

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Lukáš Kőszegy



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ROZŠÍŘENÍ SYSTÉMU INTELIGENTNÍ DOMÁCNOSTI O OVLÁDÁNÍ TERMOREGULACE

EXTENSION OF INTELLIGENT HOME BY THERMOREGULATION CONTROL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ KÖSZEGY

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN VIKTORIN

BRNO 2016

Abstrakt

Bakalářská práce se zaměřuje na integraci termoregulátoru VPT od společnosti Thermona do systému inteligentní domácnosti s názvem BeeeOn. V teoretické části jsou popsány jednotlivé komponenty termoregulátoru VPT a systému BeeeOn. Nejdůležitější část práce tvoří návrh a implementace, které popisují získávání hodnot z termoregulátoru VPT pomocí protokolu HTTP s využitím datového souboru ve formátu JSON. V těchto kapitolách jsou také popsány k tomu potřebné součásti, které je třeba integrovat do systému BeeeOn a termoregulátoru VPT.

Abstract

This bachelor thesis is focused on the integration of the heat regulator VPT made by the company Thermona into the system for intelligent home control called BeeeOn. In the theoretical part, the most specific components of BeeeOn system and heat regulator VPT are described. The most important part of this thesis consists of the design and implementation. It focuses on gather values measured by the VPT regulator by means of the HTTP protocol. The measured values are provided in a separate JSON data file. In these parts there are also described the necessary elements, which have to be integrated into the BeeeOn system and VPT regulator.

Klíčová slova

Inteligentní domácnost, Termoregulace, VPT, BeeeOn, Adaapp, HTTP, JSON, HTML, Komunikace

Keywords

Smarthome, Thermoregulation, VPT, BeeeOn, Adaapp, HTTP, JSON, HTML, Communication

Citace

Lukáš Köszegy: Rozšíření systému inteligentní domácnosti o ovládání termoregulace, bakalářská práce, Brno, FIT VUT v Brně, 2016

Rozšíření systému inteligentní domácnosti o ovlá- dání termoregulace

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Viktorina

.....
Lukáš Kőszegy
15. května 2016

Poděkování

Rád by som sa poďakoval vedúcemu bakalárskej práce Ing. Janovi Viktorinovi za poskytnuté odborné konzultácie, pomoc a hlavne konštruktívne pripomienky pri písaní tejto práce. Tiež by som chcel poďakovať pánovi Ing. Tomášovi Novotnému za uvedenie do projektu BeeeOn ako aj ostatným členom vývojového tímu za poskytnutie odborných konzultácií na projekte.

© Lukáš Kőszegy, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	BeeeOn	4
2.1	Architektúra	4
2.2	Tabuľka zariadení	5
2.3	Koncové ovládacie stanice	6
2.4	Server	6
2.4.1	Architektúra	7
2.5	Adaptér	8
2.5.1	AdaApp	9
2.6	Senzory/Riadiace prvky	12
3	Termoregulátor VPT	14
3.1	Popis regulácie	15
3.2	Ovládanie	16
3.2.1	Displej	16
3.2.2	GSM	16
3.2.3	Ethernet a Wifi	17
4	Návrh implementácie	20
4.1	Návrhy získania hodnôt z termoregulátora VPT	20
4.1.1	Existujúce webové stránky	21
4.1.2	Rozšírenie webového rozhrania	22
4.2	Návrh integrácie do BeeeOn systému	23
4.2.1	AdaApp	23
4.3	Zhrnutie	24
5	Implementácia	25
5.1	VPT	25
5.1.1	Implementácie JSON súboru pre termoregulátor VPT	25
5.1.2	Integrácia do termoregulátora VPT	27
5.2	BeeeOn	29
5.2.1	Tabuľka zariadení	29
5.2.2	AdaApp – JSON zariadenia	31
5.2.3	AdaApp – HTTP klient	36
5.2.4	AdaApp – VPT modul	36

6 Testovanie	40
6.1 Krátkodobé testovanie	40
6.2 Dlhodobé testovanie	40
7 Plánované rozšírenie projektu	42
8 Záver	44
A Obsah CD	47

Kapitola 1

Úvod

S rozvojom moderných technológií sa stal život človeka jednoduchší a efektívnejší. Každý deň sa pracuje na nových technológiach, ktoré pomáhajú v bežnom živote. Jednou z vyvíjaných technológií je takzvaná inteligentná domácnosť. Jedna sa o skupinu technológií (zariadenia, programy a technické postupy) pre riadenie domácnosti pomocou počítača. Hlavným cieľom je zvýšiť komfort užívateľa, efektívnosť, bezpečnosť a zníženie nákladov na prevádzku domácnosti.

Inteligentná domácnosť má slúžiť hlavne ako centrálny systém k ovládaniu bežných systémov v domácnosti, ako je napríklad termoregulácia, osvetlenie, zabezpečenie domácnosti, atď. Takýto systém sa vyvíja aj na Fakulte informačných technológií VUT v Brne ako súčasť projektu Internetu veci pod názvom BeeeOn.

V rámci projektu BeeeOn už bol vytvorený celkový základ pre inteligentnú domácnosť, od zariadení (senzory) až po programové vybavenie (užívateľské rozhranie, klientska a serverová aplikácia). Napriek dlhodobému vývoju sa projekt stále nachádza v rannom štádiu vývoja a prevláda v ňom viac podpora pre zariadenia, ktorých funkcionalita umožňuje len poskytovanie dát. Stále mu chýba určitá pridaná hodnota s možnosťou vzdialeného ovládania zariadení v domácnosti.

Jedným z dôležitých aspektov, ktoré by užívatelia privítali, by bola možnosť ovládať svoju domácnosť na diaľku. Pomocou inteligentnej domácnosti vznikla by možnosť ovládať osvetlenie, zásuvky a v zimných mesiacoch termoreguláciu. Nakoľko existuje veľké množstvo termoregulátorov od rôznych výrobcov, ktoré sú vyvíjane komerčne a ich súčastí a dokumentácia nemusia byť voľne prístupné, budeme pri tejto práci používať zariadenie VPT. Dané zariadenie tiež patrí medzi komerčné projekty a je vyvíjane spoločnosťou Thermona. Nakoľko samotná spoločnosť má záujem o implementáciu ich zariadenia do systému BeeeOn, boli vývojovému tímu zapožičané zariadenia VPT a tiež spoločnosť poskytla všetky potrebné materiály a programové vybavenie pre vývoj.

Táto práca sa zameriava na návrh integrácie zariadenia VPT do systému BeeeOn, ktorá sa bude realizovať tak, aby implementovala základnú funkcionalitu zariadenia VPT s ohľadom na jednoduchosť celej integrácie a požiadavky spoločnosti Thermona.

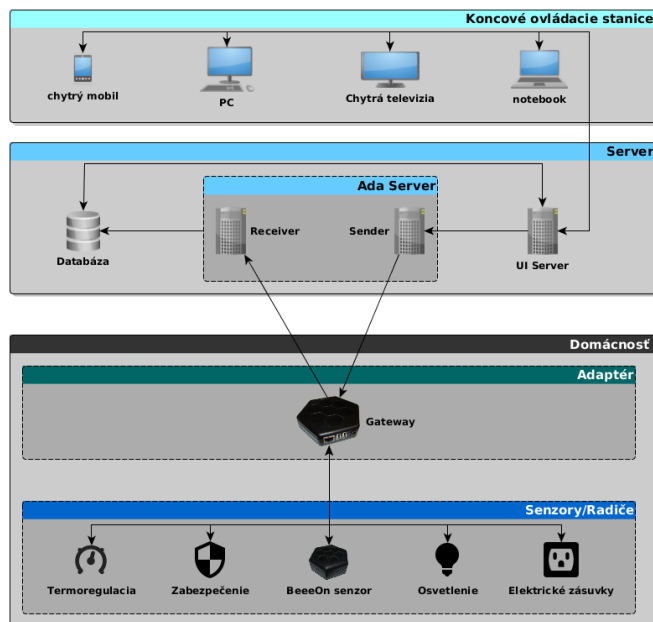
Kapitola 2

BeeeOn

Projekt BeeeOn sa vyvíja na Fakulte informačných technológií Vysokého učenia technického v Brne ako súčasť projektu Internetu vecí. Samotný vývoj započal už v roku 2014 ako projekt s komerčným zameraním a uzavretými zdrojovými kódmi. Koncom roku 2015 došlo k uvoľneniu zdrojových kódov a špecifikácii, k návrhom senzorov a zariadení s cieľom vytvoriť univerzálne riešenie pre inteligentnú domácnosť, kde je kladený dôraz na nízke zriaďovacie náklady. K dnešnému dňu sú zverejnené všetky zdrojové kódy na servery github.com [4]. Následujúce informácie o systéme BeeeOn boli čerpané z interných wiki stránok projektu [3][5][2].

2.1 Architektúra

Celý systém sa skladá z niekoľkých malých samostatných komponentov, ktoré sa následne zoskupujú do 4 hlavných skupín: **Koncové ovládacie stanice**, **Server**, **Adaptér** a **Senzory/Riadiace prvky**.



Obrázok 2.1: Architektúra BeeeOn systému.

Tieto hlavné skupiny tvoria základnú architektúru celého systému. Jednotlivé skupiny by sa mali správať v istých situáciách nezávisle od ostatných častí architektúry, čím vzniká možnosť prevádzkovať systém BeeeOn aj na neúplnej architektúre a tak je dosiahnutá určitá modularita samotnej architektúry. V aktuálnom stave je systém BeeeOn z pohľadu ukladania a získavania dát hlavne pre koncové ovládacie stanice veľmi závislý na server. V projekte BeeeOn sa ale táto závislosť plánuje znížiť, aby bolo možné získať pre užívateľské koncové zariadenia dáta priamo z Adaptéra popísaného na strane 8.

2.2 Tabuľka zariadení

Tabuľka zariadení je XML dokument v projekte BeeeOn, ktorý slúži pre centrálnu správu všetkých podporovaných zariadení. Samotná tabuľka obsahuje špecifikácie zariadení a ich vlastností, ktoré sú v projekte BeeeOn podporované. Tieto špecifikácie sú potom medzi jednotlivými aplikáciami systému BeeeOn dodržiavané.

Každé zariadenie v tabuľke má jedinečný číselný identifikátor, na základe ktorého sa rozlišuje v jednotlivých aplikáciách. Jednotlivým zariadeniam sa potom definujú moduly typu senzor, riadiaci prvok, refresh time, atď. Každý modul typu senzor a riadiaci prvok v rámci jedného zariadenia musí mať svoj vlastný jedinečný číselný identifikátor a pomenovanie. Každému modulu aj zariadeniu môžeme definovať nejaké rozširujúce vlastnosti, ako sú napríklad viditeľnosť modulu, minimálna a maximálna hodnota, meraná veličina, atď. Približná hierarchia tabuľky potom vyzerá takto:

- Zariadenia
 - Vlastnosti
 - Moduly
 - * Sensory
 - Vlastnosti
 - * Riadiace prvky
 - Vlastnosti
 - * Špeciálne moduly
 - Vlastnosti

```
1 <device id="10" name="Jablotron TP-82N Wireless thermostat" >
2   <name>T:DEV_TP_82N_THERMOSTAT</name>
3   <manufacturer>T:MANUFACTURER_JABLOTRON</manufacturer>
4   <modules>
5     <sensor id="0" type="0x02" >
6       <name>T:MOD_CURRENT_ROOM_TEMPERATURE</name>
7     </sensor>
8     <sensor id="1" type="0x02" >
9       <name>T:MOD_REQUESTED_ROOM_TEMPERATURE</name>
10    </sensor>
11    <battery id="2" type="0x08" />
12  </modules>
13 </device>
```

Zdrojový kód 2.1: Príklad tabuľky zariadení.

Pre ľahšiu integráciu do jednotlivých aplikácií sa tabuľka zariadení v XML formáte priamo nepoužíva. Na jej základe je vygenerovaný vždy výstup pre konkrétnu aplikáciu len s potrebnými vlastnosťami a hodnotami. Napríklad pre AdaApp je to hlavičkový súbor, ktorý obsahuje vo forme vektorov špecifikácie zariadení.

2.3 Koncové ovládacie stanice

Pod označením koncové ovládacie stanice si môžeme predstaviť akékoľvek zariadenie, ktoré umožňuje komunikáciu so serverom systému BeeeOn a pomocou ktorého je možné ovládať inteligentnú domácnosť. V určitých prípadoch, ale môžeme považovať za koncové stanice aj zariadenia, ktoré umožňujú len zobrazovať údaje z inteligentnej domácnosti bez možnosti jej ovládania. Medzi koncové zariadenia môžeme zaradiť osobné počítače, notebooky, chytré televízory a hlavne chytré mobilné telefóny.

Hlavným reprezentantom pre koncové ovládacie stanice v systéme BeeeOn je aplikácia s rovnakým pomenovaním, ktorá je primárne vyvíjaná pre chytré mobilné telefóny využívajúce operačný systém Android. Okrem tejto aplikácie existujú aplikácie pre chytré mobilné telefóny s operačným systémom Windows phone a je tiež možnosť priameho grafického výstupu aj zo samotného adaptéra pomocou HDMI výstupu priamo na televízore alebo monitore v domácnosti. Medzi ďalšie možnosti výstupu, ktoré sú ešte vo vývoji, patrí webové rozhranie a podpora pre chytré televízory.

Aplikácia BeeeOn používa predpísaný konzistentný dizajn a pridávanie nových zariadení je možné len pomocou Tabuľky zariadení. Takéto riešenie umožňuje ľahké pridávanie nových zariadení do aplikácie, aj bez nutnosti znalostí o programovaní pre platformu Android. Nevýhodou je ale veľká limitácia v prípade pokročilejších požiadaviek na zobrazené a ovládacie prvky.

2.4 Server

Server je v aktuálnom stave systému BeeeOn ústredným prvkom celého systému, kde je jeho úlohou poskytovať tri hlavné funkcie:

- **Proxy** – Hlavná funkcia ako proxy servera je vytvoriť medzivrstvu medzi routrom a koncovými ovládacími stanicami. Výhoda proxy servera spočíva hlavne v tom, že nie je nutné, aby malo koncové zariadenie (adaptér) pridelenú verejnú IP adresu, poprípade mohlo sa nachádzať za NAT-om¹ bez potreby ďalších nastavení, ako je port-forwarding a zároveň rieši problém s dynamickým pridelovaním IP adries od poskytovateľa internetových služieb. Táto funkcionality tiež uľahčuje aj správu zariadení, nakoľko je možné všetky adaptéry spravovať centrálnne na jednom mieste.
- **Úložisko** – Primárna časť úložiska je reprezentovaná databázou, kde sa ukladajú namerané hodnoty zo senzorov a dodatočné informácie pre fungovanie celého systému BeeeOn (napríklad: zoznam zaregistrovaných adaptérov). Server, zároveň bude slúžiť ako vyrovnávací pamäť v prípade, že by nebol adaptér napríklad z dôvodu výpadku pripojenia do siete Internet dostupný. V tomto prípade dôjde k ukladaniu riadiacich príkazov, ktoré sa prepošlú na adaptér po obnovení jeho spojenia so serverom.

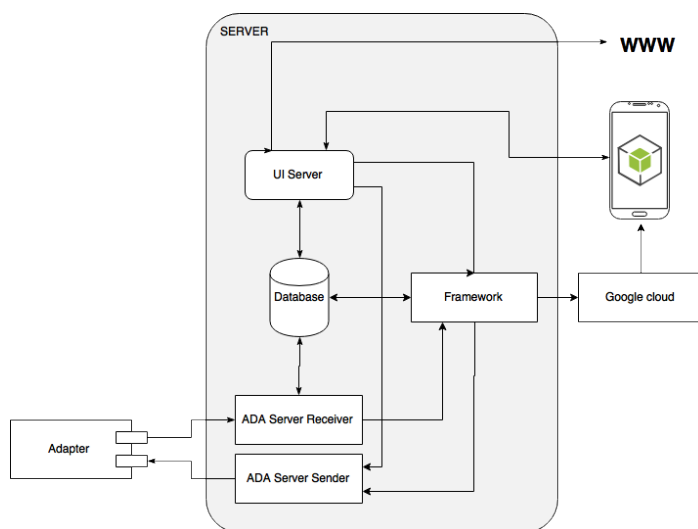
¹Network Address Translation

Pomocou funkcie úložiska môže užívateľ aj v prípade problémov získať približné (neaktuálne) informácie o stave domácnosti (napríklad: posledná zaznamenaná teplota, stav dverí – zamknuté/odomyknuté).

- **Riadiaca logika** – Server okrem centrálnej správy obsahuje aj hlavnú časť riadiacej logiky systému. Toto riešenie teda pripomína cloud. Výhoda spočíva hlavne v znížení nárokov na koncové zariadenia, ako sú adaptér, senzory a riadiace prvky. Ďalšou výhodou spoločne s úložiskom je možnosť analýzy dát, napríklad pre predikciu do budúcnosti (počasie), alebo prispôbenie sa chovaniu užívateľa na základe histórie. Nevýhoda tohto riešenia sa môže prejaviť, ak dôjde k dlhodobej poruche komunikácie medzi serverom a adaptérom. Nakoľko môže inteligentná domácnosť riadiť aj nejaké dôležité systémy, ako je termoregulácia, ohrev vody, atď., je dôležité zabezpečiť ochranu pred zlyhaním. Preto je vhodné, časť riadiacej logiky preniesť na adaptér.

2.4.1 Architektúra

Jednotlivé funkcie servera sú rozložené na nezávislé komponenty, takéto rozloženie umožňuje škálovateľnosť architektúry a rozloženie záťaže na viac serverov. Táto vlastnosť servera vo výsledku umožňuje vybudovať si cloudové riešenie.



Obrázok 2.2: Schéma servera [5].

Komponenty servera:

- **UI server** – Primárna funkcia UI servera je vytvoriť jednotné komunikačné rozhranie pre koncové ovládacie stanice a tým vytvoriť abstraktnú vrstvu nad celým systémom.
- **Ada server** – Serverová aplikácia Ada Server zabezpečuje všetkú komunikáciu s adaptéromi a ostatnými časťami servera na základe vlastného aplikačného protokolu, ktorý je postavený na XML správach. Samotná aplikácia je rozdelená na 2 časti, podľa smeru komunikácie s adaptérom:
 - **Receiver** – Hlavnou funkciou tejto časti je čítanie prichádzajúcich dát z adaptérov (získané hodnoty zo senzorov) a ich ukladanie do databázy.

- **Sender** – Umožňuje zasielanie systémových alebo užívateľských príkazov na adaptér.
- **Databáza** – Jedná sa o relačnú SQL databázu, kde sa ukladajú dáta z celého BeeOn systému. V aktuálnej vývojovej prevádzke sa pracuje s databázou PostgreSQL.

2.5 Adaptér

Adaptér reprezentuje v súčasnom stave vývoja počítač A10-OLinuXino-LIME od spoločnosti Olimex, ktorý je založený na jadre ARMv7 a disponuje operačnou pamäťou o veľkosti 512 MB. Medzi dôležitú konektorovú výbavu patrí ešte port RJ45 a UART. Viac o samotnom počítači si môžete prečítať v špecifikácii [15].

Operačný systém je linuxová distribúcia vytvorená pomocou nástroja OpenEmbedded [12] a využíva balíčkový systém opkg s vlastnými balíčkami, čím bola dosiahnutá lepšia kontrola nad konzistenciami jednotlivých adaptérov z hľadiska verzií softwaru.

Ďôležitou súčasťou každého adaptéru je navyše rozširujúci hardwarový modul, ktorý obsahuje tlakový senzor a rádio pre komunikáciu so senzormi pomocou bezdrôtovej technológie.



Obrázok 2.3: Adaptér.

V rámci systému je každý adaptér definovaný jedinečným identifikačným číslom, ktoré je generované pri prvom spustení vo výrobe. Pomocou tohto čísla sa adaptér reprezentuje v celom systéme BeeOn.

Hlavnou úlohou adaptéru je poskytnúť hardwarové a softwarové vybavenie pre jednotnú obsluhu rôznych senzorov a riadiacich prvkov v domácnosti. Teda môžeme povedať, že adaptér vytvára abstraktnú vrstvu, ktorá dovoľuje získavať a odosielať dáta na senzory a riadiace prvky jednotným spôsobom z pohľadu servera. Túto abstraktnú vrstvu vytvára hlavná aplikácia adaptéra, ktorá sa volá AdaApp a bola vytvorená v rámci diplomovej práce M. Čechmánka s názvom Brána inteligentní domácnosti v školskom roku 2014/2015 [22].

2.5.1 AdaApp

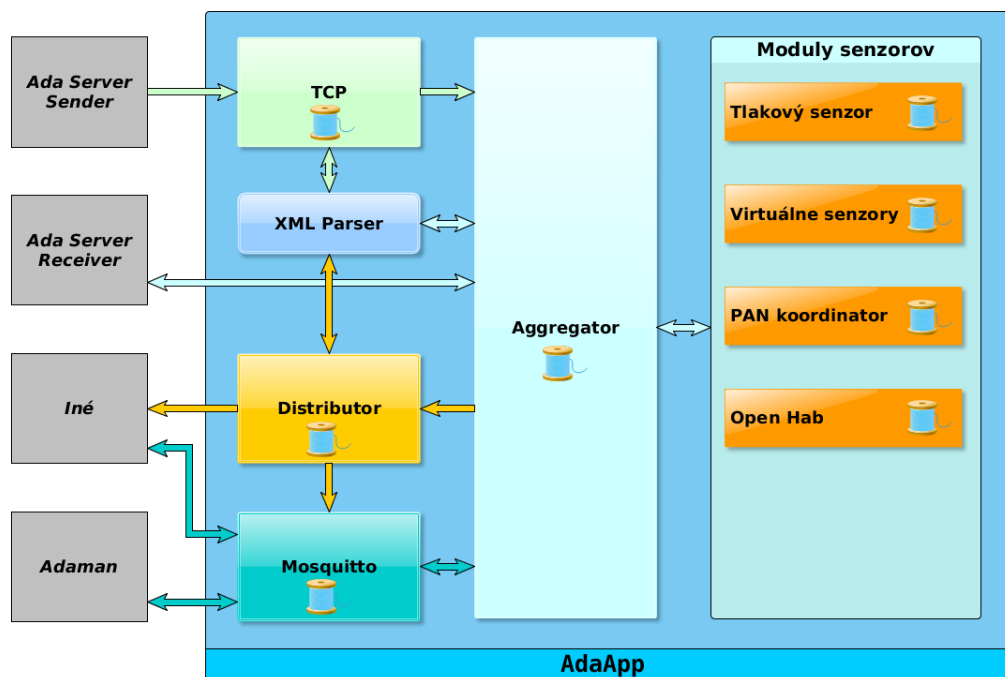
Aplikácia AdaApp je hlavná aplikácia Adaptéru, ktorá vytvára centralizované komunikačné spojenie s okolím inteligentnej domácnosti. Vo výsledku hlavnou úlohou aplikácie je vytvoriť jednotné komunikačné prostredie medzi senzormi, riadiacimi prvkami a ostatným súčastami systému BeeOn. Ďalšou funkciou z hľadiska komunikácie je vytvoriť bezpečné spojenie, ktoré by malo chrániť pred možným odposluchom dát medzi adaptérom a serverom, alebo narušením integrity komunikácie útočníkom. K tomuto účelu sú všetky TCP spojenia na server šifrované pomocou bezpečnostných protokolov SSL/TLS.

Celá aplikácia sa vyvíja v programovacom jazyku C++ pod štandardom C++11. Pri vývoji sa kladie dôraz na používanie objektovo orientovaného prístupu k programovaniu a taktiež sa pre ľahší a rýchlejší vývoj využíva framework Poco.

Architektúra

Architektúra celej aplikácie je navrhnutá tak, aby bola čo najviac modulárna, kde väčšina modulov pracuje ako samostatné vlákna. Tým vzniká jednoduchý popis funkcie main(), ktorá po spustení inicializuje základné konfigurácie pre AdaApp a overí povolené moduly. Následne sa povolené moduly spustia v samostatných vláknach a zaobstarávajú celú funkcionálnosť aplikácie. Táto architektúra umožňuje ľahko implementovať novú funkcionálnosť a hlavne podporu pre rôzne senzory a riadiace prvky.

Okrem voliteľných modulov existujú aj povinné moduly, ako sú Agregator a TCP, ktorých hlavnou úlohou je zaobstaráť komunikáciu s okolím alebo interne medzi jednotlivými modulmi.



Obrázok 2.4: AdaApp architektúra.

Aggregator

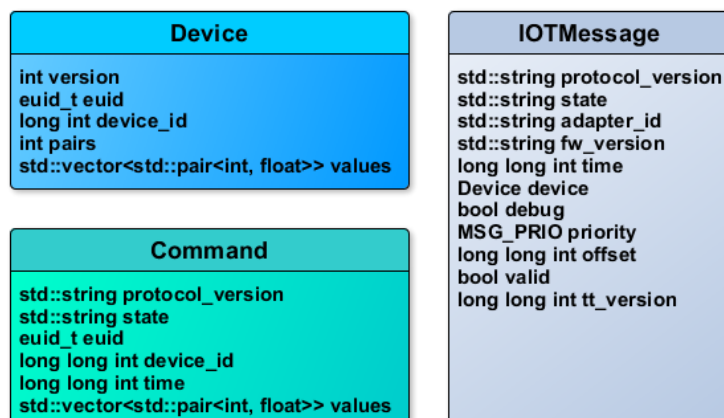
Tento modul tvorí hlavnú časť funkcionality AdaApp a tvorí centralizované miesto. Týmto miestom prechádzajú všetky správy a to smerom na Ada server a taktiež aj opačným smerom. Primárne sa ale stará o všetku odchádzajúcu komunikáciu z AdaApp na Ada server, teda zasielanie hodnôt zo senzorov na server. Pre tento účel vytvára vždy nové spojenia typu správa-potvrdenie. Okrem komunikácie so serverom zabezpečuje aj komunikáciu medzi senzorovými modulmi s ostatnými modulmi aplikácie. Medzi takéto najčastejšie prípady patria správy pre nastavenie riadiaceho prvku, ktoré pochádzajú od užívateľa. Z tohto vyplýva, že Aggregator musí mať pre správne fungovanie informácie o existencii všetkých spustených senzorových moduloch.

TCP

Modul TCP inicializuje a udržiava jedno peramentné spojenie s Ada Serverom. Toto spojenie slúži len na prijímanie správ z Ada servera, ktoré sa následne konvertujú z XML podoby na interne štruktúry a sú preposlané na Aggregator, ktorý ich prepošle ďalej na konkrétny modul, ktorému boli adresované. V aktuálnom stave toto spojenie slúži na prijímanie požiadaviek od užívateľa (napríklad: párovacia požiadavka pre spustenie *párovacieho režimu* na adaptéri alebo **zmena stavu riadiaceho prvku**).

Senzorové moduly

Tieto moduly zabezpečujú všetkú komunikáciu so senzormi alebo riadiacimi prvkami a Aggregatorom. Jeden modul väčšinou dokáže obsluhovať viacej rôznych senzorov/riadiacich prvkov, ktoré využívajú rovnaký komunikačný protokol. Samozrejme je možné vytvoriť aj niekoľko rôznych modulov pre rôzne senzory/riadiace prvky, ktoré budú využívať rovnaký komunikačný protokol. Túto možnosť je vhodné uplatniť hlavne pri veľkých rozdieloch medzi jednotlivými senzormi/riadiacimi prvkami. Každý modul implementuje rôzne spôsoby komunikácie medzi senzormi/riadiacimi prvkami, ale medzi Aggregatorom je striktné definovaný spôsob komunikácie pomocou interných štruktúr. Práve v tomto mieste vzniká prvá abstraktná vrstva v komunikácii, ktorá umožňuje jednotný spôsob komunikácie s rôznymi zariadeniami. Medzi takéto interné štruktúry patrí aj **Device**, ktorý interne reprezentuje aktuálny stav fyzického senzora/riadiaceho prvku.



Obrázok 2.5: AdaApp štruktúry pre internú komunikáciu medzi modulmi.

Každé zariadenie musí mať jedinečný číselný identifikátor (**eid**), na základe ktorého je reprezentované v systéme BeeOn a tiež musí mať definovaný identifikátor (**device_id**) pre špecifikáciu v Tabuľke zariadení. Pri prečítaní hodnôt zo senzora sa vykoná ich konvertovanie na páry, kde prvé číslo definuje identifikátor modulu v špecifikácii zariadenia podľa Tabuľky zariadení a druhé číslo obsahuje aktuálne získanu hodnotu zo zariadenia.

Hlavnou úlohou pri získavaní hodnôt zo sensorov/radiaciach prvkov je teda vytvoriť ich aktuálny stav vo forme internej štruktúry Device. Pred odoslaním do Agregatoru sa Device štruktúra prevádza do internej štruktúry IOTMessage, kde sa následne v Agregatori prevedie do správy vo formáte XML a pridajú sa základné informácie o adaptéry. Takto vytvorená správa s hodnotami sensorov sa následne odošle na server. Tento kolobeh získavania hodnôt zo sensorov a radiaciach prvkoch sa potom opakuje v pravidelných intervaloch, ktoré je možné meniť cez užívateľské rozhranie.

V prípade užívateľských požiadaviek sa pre internú komunikáciu používa štruktúra **Command**, ktorá okrem vyššie popísaných vlastností štruktúry **Device** (**eid**, **device_id** a **páry hodnôt**) má navyše tri atribúty:

- **protocol_version** (verzia protokolu) – Určuje verziu komunikačného protokolu medzi adaptérom a serverom.
- **state** (stav) – Typ správy od servera.
 - **update** (aktualizovať) – Správa pre nastavenie časového intervalu pre získanie hodnôt sensorov zo zariadenia.
 - **listen** (počúvať) – Aktivácia párovacieho režimu vo všetkých sensorových moduloch.
 - **set** (nastaviť) – Nastavenie riadiaceho prvku na požadovanú hodnotu.
- **time** – Využíva sa pre typ správy **update** a obsahuje nový časový interval pre získanie hodnôt sensorov zo zariadenia.

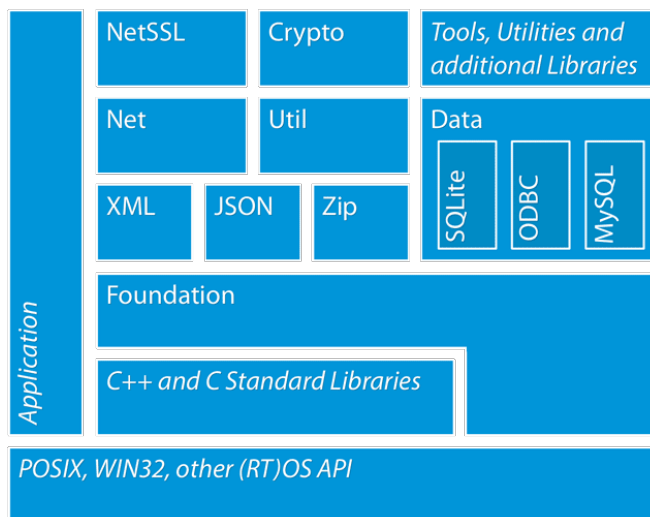
POCO

Poco je sada knižníc a zároveň framework pre programovanie v jazyku C++ [8][10]. Framework podporuje rozsiahlu škálu operačných systémov a okrem klasických počítačov je možné ho využiť vo Vstavaných systémoch a Mobilných platformách.

Poco je postavené na modulárnej architektúre, čo umožňuje využiť len potrebné súčasti frameworku. Táto architektúra znižuje nároky na výkon, aj čas potrebný na preklad. Jednotlivé moduly sú v rámci architektúry rozdelené do niekoľkých skupín:

- **Foundation** - Hlavná knižnica frameworku, ktorá obsahuje najpoužívanejšie funkcie, konštanty, podporu vlákien, atď.
- **XML** – Zastrešuje nástroje pre prácu s XML formátom – spracovanie, generovanie, atď.
- **JSON** [9] – Umožňuje ľahko pracovať s JSON formátom – spracovanie, generovanie, atď.
- **Zip** – Podpora pre komprimovaný formát Zip – vytváranie, pridávanie súborov, atď.
- **Net** – Umožňuje ľahké vytváranie sieťovej komunikácie – tvorba serverov, podpora sieťových protokolov, atď.

- **NetSSL** – Rozširuje klasickú sieťovú komunikáciu o podporu šifrovania pomocou protokolov SSL alebo TLS – podpora šifrovaných sieťových protokolov, atď.
- **Util** – Framework pre ľahké vytváranie konzolových a serverových aplikácie – spracovanie parametrov príkazového riadku a podobne.
- **Crypto** – Integruje kryptografické a súvisiace funkcie – SHA1 [7], RC4, X509, atď.
- **Data** – Umožňuje pracovať jednotným spôsobom s rozdielnymi SQL databázami (**SQLite**, **MySQL**, **ODBC**)



Obrázok 2.6: Poco architektúra [6].

2.6 Senzory/Riadiace prvky

Senzory/Riadiace prvky sa nachádzajú na spodnej vrstve celej architektúry a reprezentujú vlastnosti zariadení v domácnosti. Vo všeobecnosti sa môže jednať o akékoľvek zariadenia v domácnosti, ktoré dokážu komunikovať s okolím a umožňujú mu reportovať, alebo meniť svoj aktuálny stav. Na základe týchto vlastností rozlišujeme dva typy modulov v zariadení:

- **Senzory** – Snímajú nejakú veličinu a namerané hodnoty poskytujú adaptéru pre ďalšie spracovanie. Medzi merané veličiny môže patriť vlhkosť, teplota, intenzita svetla, elektrická spotreba, atď.
- **Riadiace prvky (Actuators)** – Môžu ovplyvňovať stav zariadenia (napríklad: ako spínač kontaktu vykurovacieho kotla). Primárne by mali slúžiť len pre nastavovanie hodnôt, ktoré sa menia skokovo a prevažne len na podnet užívateľa, ale v základe ide len o senzory s funkciou zmeny stavu na žiadosť užívateľa. Z toho vyplýva, že môžu byť použité rovnako, ako senzory pre snímanie nejakých nameraných hodnôt veličín.

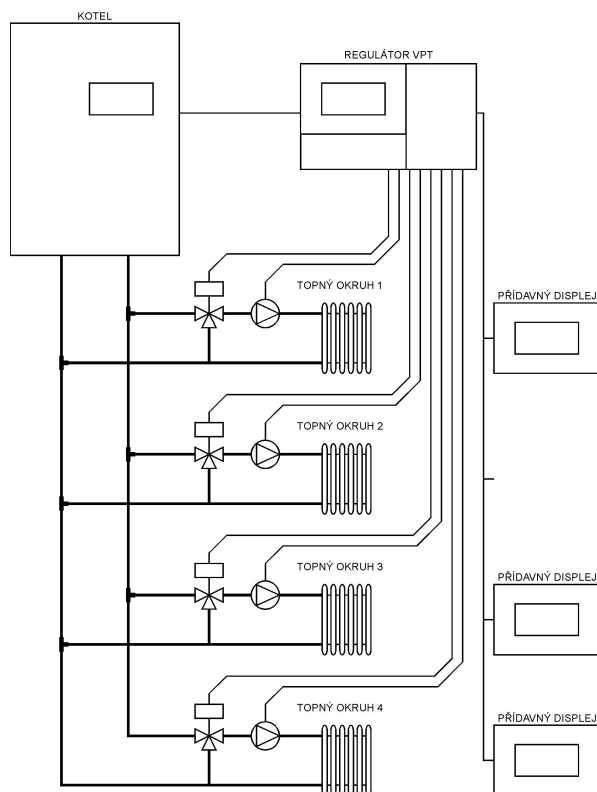
Jednotlivé zariadenia boli v rannej fáze vývoja jednoduché, najčastejšie jedno-účelové a hlavne za nízku zriaďovaciu a prevádzkovú cenu. Medzi prvé podporované zariadenia patria senzory vlastnej výroby, ktoré slúžili len pre zber údajov o teplote a vlhkosti. Pre

komunikáciu s adaptérom využívali bezdrôtovú technológiu a na vyššej vrstve FIT protokol. S postupným vývojom je ale snaha do systému BeeeOn integrovať zariadenia tretích strán, ktoré sú oproti našim vlastným sensorom zložitejšie a nevyužívajú rovnakú bezdrôtovú komunikáciu a tiež pozostávajú z väčšieho množstva sensorov a riadiacich prvkov. Medzi takéto zariadenia patrí aj termoregulátor VPT od spoločnosti Thermona.

Kapitola 3

Termoregulátor VPT

VPT termoregulátor od spoločnosti Thermona je určený pre centrálnu reguláciu vykurovania domácnosti, bytov, kancelárii, atď. Je navrhnutý primárne pre vykurovacie sústavy s jedným zdrojom tepla, ktorým môže byť kotol alebo výmenník. Termoregulátor dokáže obsluhovať nezávisle štyri zóny (okruhy vykurovania), čo je vhodné pri vykurovaní oddelených priestorov, akými môžu byť byty, dvojgeneračné domácnosti alebo kancelárie. Okrem priestorového rozdelenia vykurovacích zón, môžeme jednotlivé zóny rozdeliť aj podľa oddlišných spôsobov vykurovania, ako napríklad pomocou radiátorov, alebo podlahového vykurovania. Okrem vykurovania priestorov je možné s termoregulátorom VPT ovládať aj kotol s funkciou ohrievania teplej úžitkovej vody.



Obrázok 3.1: Zapojenie termoregulátora VPT v komplexe budov [17].

3.1 Popis regulácie

Termoregulátor VPT umožňuje pre každú zónu nastaviť päť typov regulácie. Následujúce informácie o popise regulácie boli čerpané z dokumentu popisujúci termoregulátor VPT [17].

Typy regulácie:

- **Izbový termostat** – Pri tomto type regulácie je teplota vykurovacej vody regulovaná podľa aktuálnej teploty v miestnosti, v ktorej sa nachádza senzor pre meranie teploty (napríklad: doplnkové displeje obsahujú tento senzor).
- **Ekvitermická regulácia** – V tomto prípade je teplota vykurovacej vody regulovaná na základe vonkajšej teploty bez zohľadňovania aktuálnej teploty v miestnosti. Reguláciu vykurovacej vody je možné ovplyvniť nastavením posunu ekvitermickej krivky od základnej polohy.
- **Kombinácia izbového termostatu a ekvitermickej regulácie** – Tento typ regulácie vykurovacej vody kombinuje predchádzajúce dva typy regulácie, čím umožňuje komfortnejšiu reguláciu. Teplota vykurovacej vody je teda závislá od aktuálnej teploty v miestnosti a vonkajšej teploty. V tomto type regulácie je vyžadovaná možnosť kĺzavej regulácie teploty vykurovacej vody.
- **Regulácia na konštantnú teplotu vykurovacej vody** – Pri tejto regulácii vykurovacej vody sa nevyužíva žiaden senzor. Pri regulácii sa využíva len ručne zadaná teplota vykurovacej vody.
- **Ohrev teplej úžitkovej vody** – Okrem riadenia vykurovania je možné využiť termoregulátor VPT pre riadenie ohrevu teplej úžitkovej vody. Pri tomto type regulácie termoregulátor VPT ovláda obehové čerpadlo (poprípade ventil), ktorý reguluje prietok ohrievanej vody do zásobníka vody.

Tieto typy je možné kombinovať/doplniť s časovými programami. Časové programy umožňujú konfiguráciu pre každý deň v týždni. Pri konfigurácii časových programov je možné nastaviť na každý deň 6 časových úsekov s požadovanou teplotou. Pri tejto konfigurácii sa čas zadáva vo formáte 24 hodín spoločne s minútami.

Okrem časových programov je možné pre typy regulácie nastaviť aj tri režimy prevádzky.

Režimy prevádzky:

- **Automatický** – V tomto režime sa požadovaná hodnota teploty nastavuje podľa zadaného časového programu.
- **Ručný** – Požadovaná teplota sa nastavuje ručne.
- **Dovolenka** – Podobne ako automatický režim, ale využíva sa špeciálny časový program, ktorý je platný pre každý deň v týždni, keď je aktívny tento režim.

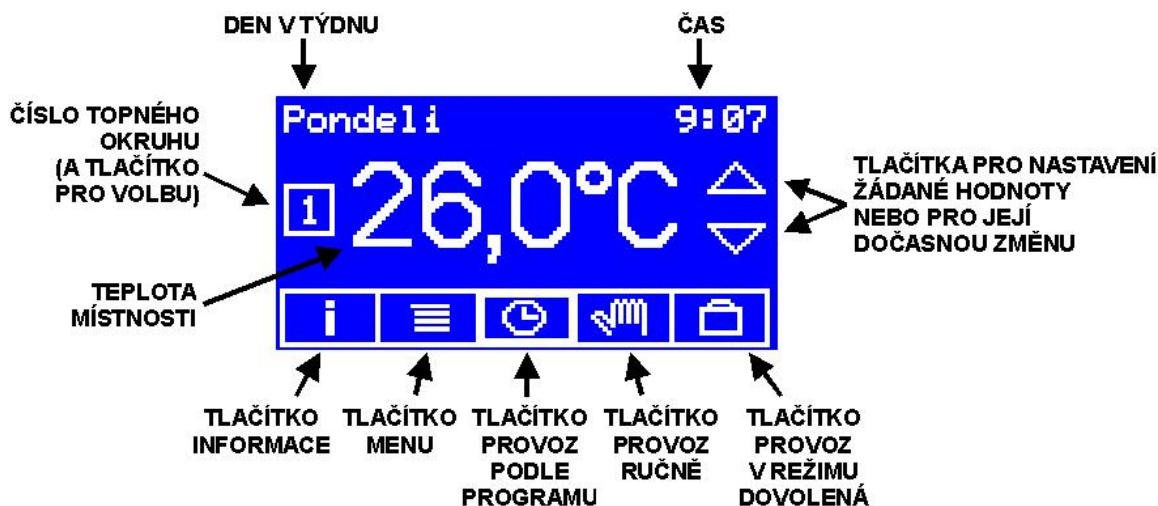
3.2 Ovládanie

Spoločnosť Thermona umožňuje ovládať termoregulátor VPT so štandardnými dotykovými displejmi, ktoré si vyžadujú fyzickú prítomnosť užívateľa. Okrem tejto možnosti spoločnosť Thermona poskytuje prídavné moduly pre vzdialené ovládanie termoregulátoru VPT s využitím technológií **Ethernet**, **Wi-fi** a **GSM** [17].

Vzdialené ovládanie umožňuje získavať aktuálny stav a nastavovať vykurovacie parametre. Vzdialené ovládanie je oproti štandardným displejom obmedzené pre pokročilé nastavenia samostatných komunikačných modulov a niektorých globálnych nastavení termoregulátora VPT. Takéto bezpečnostné obmedzenie nastavení je možné na dobu 20 minút deaktivovať cez dotykový displej aktiváciou špeciálneho aktualizáčného režimu, ktorý sa používa primárne v prípade aktualizácie webových stránok na novú verziu pre komunikačné moduly využívajúce technológiu **Ethernet** a **Wi-fi**. Práve tieto dva moduly boli poskytnuté spoločnosťou Thermona s požiadavkou o integráciu do systému BeeeOn.

3.2.1 Displej

Termoregulátor VPT sa v základe dodáva s dotykovým displejom, ktorý umožňuje jeho konfiguráciu [20]. K VPT je možné pridať aj prídavné displeje, ktoré sú vhodné pre viac zónové vykurovanie, kde každý displej môže obsluhovať svoju zónu. Nakoľko je ale rozmer štandardného a prídavného displeja malý, je ovládacie rozhranie veľmi obmedzené. Väčšina konfigurácii je rozdelená do logických skupín, kde v rámci každej skupiny sa sekvenčne prepínajú jednotlivé možnosti konfigurácie. Takéto riešenie je veľmi nevhodné pre pokročilé konfigurácie, ktoré vyžadujú manipuláciu s veľkým množstvom funkcií, ktoré termoregulátor VPT poskytuje.



Obrázok 3.2: Ovládacia obrazovka termoregulátora VPT [21].

3.2.2 GSM

Pre komunikáciu pomocou GSM je možné dokúpiť k termoregulátoru VPT prídavný modul VPTGSM. S využitím tohto modulu je možné ovládať termoregulátor VPT pomocou príkazov v SMS správach. Modul umožňuje získavať aktuálny stav všetkých zón ako aj na-

stavovať ich parametre. Pre funkčnosť tohto modulu je nutné vlastniť aktívnu SIM kartu u nejakého operátora.

3.2.3 Ethernet a Wifi

Moduly VPTLAN [18] a VPTWIFI [19] umožňujú vzdialené ovládanie termoregulátora VPT v rámci lokálnej siete alebo siete Internet. Tieto moduly poskytujú rovnaké užívateľské rozhranie založené na jednoduchej webovej stránke. Jediný rozdiel, ako vyplýva z názvu jednotlivých modulov je v použitej technológii pre pripojenie do lokálnej siete. Rovnako ako GSM modul je možné na webovej stránke získať aktuálny stav pre všetky zóny spoločne s kotlom a nastavovať ich parametre. Takéto riešenie oproti štandardným displejom umožňuje ľahšiu konfiguráciu pokročilých funkcií, aj lepší prehľad aktuálneho stavu celého systému, nie len jednej zóny.

Samotné moduly sú postavené na zariadeniach od spoločnosti Connect One pre sieťovú komunikáciu. V prípade Ethernetovej a Wi-fi verzie je použitý čip CO2144 [16]. Tento čip komunikuje pomocou sériovej linky s termoregulátorom VPT a zároveň zabezpečuje webový server pre komunikáciu s užívateľom. Okrem webového servera s využitím protokolu HTTP disponuje podporou aplikačných protokolov FTP, POP3, SMTP, Telnet. S využitím protokolov SSL3/TLS1 vie poskytnúť šifrované aplikačné protokoly HTTPS a FTPS.

Webové stránky

CO2144 podľa využitia obsahuje 2 rozdielne webové stránky:

The screenshot shows the 'Pokojevý regulátor VPT' web interface. It features a sidebar on the left with navigation options: VPT, Stav, Parametry zón, Ruční režim, Společné parametry, Časové programy, Poruchy a historie, and Nastavení. The main content area is titled 'Parametry topných zón' and contains a table with columns for 'parametr', 'zóna 1', 'zóna 2', 'zóna 3', and 'zóna 4'. The table lists various parameters such as 'Režim provozu', 'Pásmo regulace pokojového termostatu', 'Diference vypnutí pokojového termostatu', 'Použití ekvitem v pokojovém termostatu', 'Sklon ekvitemní křivky', 'Základní posun ekvitemní křivky', 'Práh venkovní teploty', 'Minimální teplota topné vody', 'Maximální teplota topné vody', 'Minimální teplota teplé užitkové vody', 'Maximální teplota teplé užitkové vody', 'Protimrazová ochrana', 'Práh protimrazové ochrany', 'Teplota topné vody při protimrazové ochraně', 'Měření teploty místnosti', 'Pohon směšovacího ventilu', 'Teplota topné vody pro ohřev TUV', 'Diference vypnutí při ohřevu TUV', and 'Nastavený typ provozu topení'. Each parameter has a corresponding input field or dropdown menu. A 'Odešli změny' button is located at the bottom right of the table. The footer of the page reads 'BASPELIN, s. r. o. 2015'.

parametr	zóna 1	zóna 2	zóna 3	zóna 4
Režim provozu	Ruční	Čas. program	Čas. program	Čas. program
Pásmo regulace pokojového termostatu (3 - 10)	6 °C	6 °C	6 °C	6 °C
Diference vypnutí pokojového termostatu	0,3 °C	0,3 °C	0,3 °C	0,3 °C
Použití ekvitem v pokojovém termostatu	Ano	Ano	Ano	Ano
Sklon ekvitemní křivky (0,1 - 6,0)	1,6	1,6	1,6	1,6
Základní posun ekvitemní křivky (-20 - +40)	+0 °C	+0 °C	+0 °C	+0 °C
Práh venkovní teploty (0 - 30)	12 °C	12 °C	12 °C	12 °C
Minimální teplota topné vody (20 - 50)	25 °C	25 °C	25 °C	25 °C
Maximální teplota topné vody (30 - 90)	80 °C	80 °C	80 °C	80 °C
Minimální teplota teplé užitkové vody (20 - 50)	30 °C	30 °C	30 °C	30 °C
Maximální teplota teplé užitkové vody (40 - 90)	60 °C	60 °C	60 °C	60 °C
Protimrazová ochrana	Ne	Ne	Ne	Ne
Práh protimrazové ochrany (0 - 15)	5 °C	5 °C	5 °C	5 °C
Teplota topné vody při protimrazové ochraně (25 - 50)	40 °C	40 °C	40 °C	40 °C
Měření teploty místnosti	Displej	Čidlo	Čidlo	Čidlo
Pohon směšovacího ventilu	Žádný	0-10 V	0-10 V	0-10 V
Teplota topné vody pro ohřev TUV (50 - 90)	80 °C	80 °C	80 °C	80 °C
Diference vypnutí při ohřevu TUV (1 - 10)	2 °C	2 °C	2 °C	2 °C
Nastavený typ provozu topení	Ohřev TUV	Vypnuto	Vypnuto	Vypnuto

Obrázok 3.3: Webové stránky termoregulátoru VPT pre moduly VPTLAN a VPTWIFI.

- **Webové stránky termoregulátoru VPT** – Slúžia ako webové rozhranie pre termoregulátor VPT. Tieto webové stránky sú svojím charakterom statické webové stránky, ale obsahujú špeciálne premenné, ktoré sa mapujú na registre termoregulátora VPT. Pri načítaní potom čip CO2144 nahrádza tieto špeciálne premenné hodnotami v príslušných registroch. V stránkach sa využíva aj javascript primárne, len pre preloženie niektorých hodnôt z registrov do čitateľnejšej formy pre užívateľa. Ide prevažne o slovné hodnoty, ktoré neobsahujú diakritiku a majú predpísanú požadovanú veľkosť a preto sú doplnené o potrebný počet znakov podtržítom (_).
- **Konfiguračné webové stránky** – Tieto stránky sú dostupné na URL /ichip a ide o interné stránky, ktoré slúžia na konfiguráciu samotného čipu CO2144. Na týchto stránkach je možné meniť nastavenie siete, ako aj aktualizovať firmware čipu alebo webové stránky termoregulátoru VPT na novú verziu.

Nastavenie riadiacich prvkov

Nastavenie riadiacich prvkov termoregulátora VPT je možné pomocou zmeny hodnôt jeho interných registrov, ktoré sú číslované 4 až 5 ciferným číslom. Takto označený register sa pri komunikácii pomocou čipu CO2144 mapuje na špeciálne premenné, ktoré môžeme rozdeliť na tri typy:

- **Registre určené na čítanie** – Registre, ktoré majú format **Pxxx** (x je číslo 0 až 9) sú určené len k čítaniu a poskytujú prevažne senzorové hodnoty. Medzi takéto hodnoty patrí napríklad aktuálna teplota v miestnosti.
- **Registre určené pre čítanie a zápis** – Formát týchto registrov je **PExxx** (x je číslo 0 až 9) a okrem čítania umožňujú zápis hodnôt. Práve tieto registre sa používajú pre nastavenie riadiacich prvkov termoregulátora VPT. Príkladom pre tento register je režim vykurovania pre jednotlivé zóny.
- **Špeciálny register BEEEE0** – Register BEEEE0 je špeciálny register o veľkosti 64 znakov. Bol pridaný pre systém BeeeOn pre odlaďovanie možných problémov pri komunikácii s adaptérom. Tento register slúži pre ukladanie špeciálnych časových razítok, ktoré sa skladajú z niekoľkých častí a pri správnej komunikácii môže BEEEE0 register obsahovať reťazec BeeeOn 192.168.0.121/1091640985557517:22.03.2016+00 22:42:45R.

Popis jednotlivých častí časového razítka:

- **Začiatok** – Obsahuje len reťazec **BeeeOn** ako identifikátor projektu. Veľkosť 6 znakov.
- medzera (\x20) – Veľkosť 1 znak.
- **IP adresa adaptéra** – Obsahuje IP adresu adaptéra z ktorej prišla požiadavka na termoregulátor VPT. Veľkosť 7 až 15 znakov.
- lomítko (/) – Veľkosť 1 znak.
- **Identifikátor adaptéra** – Jedinečný identifikátor adaptéra v rámci systému BeeeOn. Jeho primárne využitie je v prípade použitia viac adaptérov. Veľkosť 16 znakov.
- dvojbodka (:) – Veľkosť 1 znak.
- **Deň** – Deň odoslania požiadavky. Veľkosť 2 znaky.

- bodka (.) – Veľkosť 1 znak.
- **Mesiac** – Mesiac odoslania požiadavky. Veľkosť 2 znaky.
- bodka (.) – Veľkosť 1 znak.
- **Rok** – Rok odoslania požiadavky. Veľkosť 4 znaky.
- plus (+) – Veľkosť 1 znak.
- **Časové pásmo** – Časové pásmo nastavené na adaptéry. Veľkosť 1 až 2 znaky.
- medzera (\x20) – Veľkosť 1 znak.
- **Hodina** – Hodina odoslania požiadavky. Veľkosť 2 znaky.
- dvojbodka (:) – Veľkosť 1 znak.
- **Minuta** – Minúta odoslania požiadavky. Veľkosť 2 znaky.
- dvojbodka (:) – Veľkosť 1 znak.
- **Sekunda** – Sekunda odoslania požiadavky. Veľkosť 2 znaky.
- **Akcia** – Typ akcie, ktorú vykonal adaptér na termoregulátori VPT.
 - * **P** – Napáranie termoregulátora VPT.
 - * **R** – Prečítanie hodnôt z termoregulátoru VPT.
 - * **S** – Nastavenie registra na termoregulátore VPT.

Nastavenie registrov VPT prebieha teda cez komunikačné moduly VPTLAN a VPTWIFI pomocou špeciálnych premenných. Tieto špeciálne premenné sa nastavujú pomocou HTTP metódy typu GET na niektorú z webových stránok termoregulátora VPT. Pri metóde GET je špeciálna premenná zadaná ako kľúč, ktorému je priradená požadovaná hodnota. Pri každom nastavení termoregulátor VPT robí kontrolu hodnoty na určité kritéria, ako je napríklad minimálna alebo maximálna teplota a v prípade zlých hodnôt nedôjde k zmene interných registrov termoregulátora VPT. Pre vyššiu bezpečnosť je možné nastaviť prístupové heslo pre zmenu špeciálnych premenných. V takomto prípade pri každej metóde GET, ktorá bude chcieť zmeniť hodnotu špeciálneho registra dôjde k presmerovaniu na stránku s formulárom pre zadanie hesla. Heslo je generované ako SHA1 hash z konkatenácie zadaného hesla s náhodne vygenerovaným číslom, ktoré je možné nájsť v javascriptovej funkcii na stránke s formulárom pre zadanie hesla.

Kapitola 4

Návrh implementácie

Spoločnosť Thermona požadovala v rámci integrácie termoregulátora VPT do systému BeeeOn podporu zobrazovať a nastavovať niektoré základné hodnoty termoregulátora VPT. Tiež bola požadovaná podpora komunikácie s termoregulátorom VPT pomocou modulov **VPTLAN** a **VPTWIFI**, ktoré boli spoločnosťou Thermona zapožičané. Pre tieto moduly bola požadovaná integrácia podpory hesla pre bezpečné nastavenie hodnôt.

Názov hodnoty	Režim podpory
Typ vykurovania	čítať/nastaviť
Režim prevádzky	čítať/nastaviť
Aktuálna teplota v miestnosti	čítať
Požadovaná teplota v miestnosti	čítať/nastaviť
Aktuálna teplota vykurovacej vody	čítať
Požadovaná teplota vykurovacej vody	čítať/nastaviť
Protimrazová ochrana	čítať/nastaviť
Stav kotla	čítať
Režim kotla	čítať
Aktuálna teplota vykurovacej vody	čítať
Aktuálna vonkajšia teplota	čítať
Aktuálny výkon kotla	čítať
Aktuálny tlak kotla	čítať
Aktuálna chyba na kotly	čítať
BeeeOn register	čítať/nastaviť

Tabuľka 4.1: Požadované hodnoty pre integráciu od spoločnosti Thermona.

Z pohľadu systému BeeeOn boli požiadavky zamerané na jednoduchú implementáciu bez nutnosti narušenia architektúry niektorého z komponentov systému BeeeOn a splnenia požiadavok pre integráciu hodnôt od spoločnosti Thermona.

4.1 Návrhy získania hodnôt z termoregulátora VPT

V rámci návrhov pre získanie hodnôt z termoregulátora VPT s využitím modulov **VPTLAN** a **VPTWIFI**, ktoré poskytuje webový server sa zameriame na návrhy, ktoré dokážu daný webový server využiť. Po analýze možností čítania hodnôt z registrov termoregulátora VPT boli navrhnuté dva prístupy integrácie termoregulátora VPT do systému BeeeOn:

- využitie existujúcich webových stránok k priamemu extrahovaniu hodnôt
- vloženie vlastnej webovej stránky s vhodne definovanou dátovou štruktúrou

4.1.1 Existujúce webové stránky

Tento návrh sa spolieha na použitie existujúcich webových stránok termoregulátora VPT a priamym extrahovaním hodnôt z HTML kódu webových stránok. Jeho výhoda spočíva v tom, že nie je nutný žiaden zásah do termoregulátora VPT. Pri podrobnejšej analýze, ale boli zistené komplikácie, ktoré zvyšovali obtiažnosť implementácie do systému BeeeOn. V nasledujúcom texte budú uvedené jednotlivé komplikácie s možnosťou ich riešenia.

Zistené komplikácie:

- **Neštandardné stránky** – Webové stránky termoregulátora VPT nespĺňali žiaden štandard jazyka HTML. Okrem chýbajúcej definície **DOCTYPE**, ktorú by mala obsahovať každá webová stránka, boli zistené nadbytočné uzatváracie elementy pre bunky v tabuľkách. S využitím XML Parsera z frameworku Poco vznikal problém pri parsovaní webových stránok kvôli nadbytočným elementom. Tento problém spôsoboval vygenerovanie výnimky a nebolo možné webové stránky termoregulátora VPT jednoduchým spôsobom parsovať. Okrem XML parsera v knižnici Poco boli vyskúšané ešte knižnice **pugixml**, **libxml++** a **HTML tidy**. Z týchto knižníc, len **HTML tidy** dokázala s využitím možnosti pre automatickú opravu rozparsovať webové stránky termoregulátora VPT do interných štruktúr.
- **Hodnoty na rôznych stránkach** – V prípade existujúcich webových stránok sú požadované hodnoty zobrazované na rôznych webových stránkach. Túto komplikáciu je možné riešiť vytvorením mapovania jednotlivých hodnôt na korektnú webovú stránku.
- **Rôzne cesty k hodnotám** – V rámci webových stránok boli jednotlivé hodnoty uložené v tabuľkách. Tieto tabuľky ale nemajú rovnaké rozmery a jednotlivé bunky neobsahujú ani jedinečný identifikátor. Preto je potrebné pre rôzne hodnoty definovať cestu v rámci webovej stránky na korektný element. V tomto prípade, by bolo možné pre zjednodušenie implementácie využiť XPath.
- **Rôzny prístup k hodnotám** – Okrem rôznych ciest k hodnotám sú použité na zobrazenie hodnôt dve rôzne metódy.
 - **Priamy prístup** – Je použitý pre číselné hodnoty, kde je daná hodnota uložená priamo medzi dvoma elementami HTML kódu (najčastejšie bunky tabuľky).
 - **Nepriamy prístup** – Používa sa pre slovné hodnoty. Hodnoty sú v tomto prípade tiež uložené podobne ako v priamom prístupe s tým rozdielom, že sú navyše uložené v javascriptovej funkcii, ako argumenty funkcie. Na stránkach sa používajú 2 typy funkcií, ktoré sa rozlišujú len na základe počtu argumentov funkcie. Tieto funkcie majú za úlohu nahradiť hodnotu z registra termoregulátora VPT do spisovnej podoby pre užívateľa. Nakoľko XML parser nedokáže rozparsovať javascriptový kód na jednotlivé časti je uložené volanie funkcie vo forme reťazca. Pre získanie aktuálnej nastavenej hodnoty z argumentov funkcie je teda nutné vytvoriť regulárny výraz pre každý typ funkcie.

- **Externé spravovanie webových stránok** – Nakoľko sú webové stránky termoregulátora VPT vytvorené a spravované spoločnosťou Thermona, môže dochádzať k nekompatibilitě pri vydaní novej verzie webových stránok.

4.1.2 Rozšírenie webového rozhrania

Dalším riešením integrácie termoregulátora VPT do systému BeeeOn je rozšírenie aktuálnych webových stránok o vlastný súbor, ktorý by prezentoval dáta v podobe vhodnej pre strojové spracovanie. V praxi sa k tomuto účelu používajú hlavne dva jazyky: **XML** a **JSON**.

XML je značkovací jazyk definovaný konzorciom W3C v špecifikácii Extensible Markup Language [13]. V poslednej dobe sa však značne rozširujú iné jazyky, hlavné JSON (JavaScript Object Notation). Dôvodom je väčšia úspornosť pásma pri zachovaní čitateľnosti ako z pohľadu človeka, tak i počítača. Jazyk JSON se rozšíril i vďaka tomu, že se jedná o podmnožinu veľmi rozšíreného jazyka JavaScript, teda je jeho implementácia ve webových prehliadačoch jednoducho implementovateľná.

Dáta, ktoré termoregulátor VPT poskytuje môžeme rozdeliť podľa rôznych parametrov:

1. Podľa jednotky v ktorej je údaj uvedený:

- bar
- stupneň Celsia (°C)
- bezrozmerné číslo
- sklon (ekvitermnej) krivky
- výkon (%)
- bitové pole

2. Rozdelenie dát podľa fyzického rozloženia vykurovacieho systému.

- Termoregulátor VPT rozlišuje až štyri nezávislé vykurovacie okruhy - zóny.
- Okrem vykurovacích okruhov, termoregulátor VPT umožňuje reportovať stav obsluhovaného kotla.

3. Rozdelenie podľa režimov, v ktorých termoregulátor VPT pracuje.

Vzťahy medzi dátovými položkami sú riešené na iných úrovniach systému BeeeOn. Cieľom rozšírenia webového rozhrania je len možnosť jednoduchého získavania dát z termoregulátora VPT a to v štandardnej podobe.

Z predchádzajúceho textu vyplýva, že nie je na tejto úrovni nutné popisovať zložité štruktúry a ani vyhodnocovať zložité vzťahy medzi dátovými položkami. Tiež množstvo dát, ktoré sa takto z termoregulátora VPT prenáša do systému BeeeOn je relatívne malé, (menšie ako 1 kB). Pragmaticky je pre reprezentáciu dát použitý formát JSON. Okrem toho spracovanie dát vo formáte JSON má jednoduchšie aplikačné rozhranie a nižšiu pamäťovú náročnosť na spracovanie. V opačnom prípade by mohol byť XML formát vhodnejší.

Táto rozširujúca stránka ma tiež odstrániť čo možno najviac komplikácií z pôvodných webových stránok. Má teda ísť o jednu samostatnú stránku, ktorá bude nezávislá od ostatných webových stránok. Takéto riešenie by umožnilo, aby bola stránka vytvorená a spravovaná vývojovým tímom projektu BeeeOn.

4.2 Návrh integrácie do BeeeOn systému

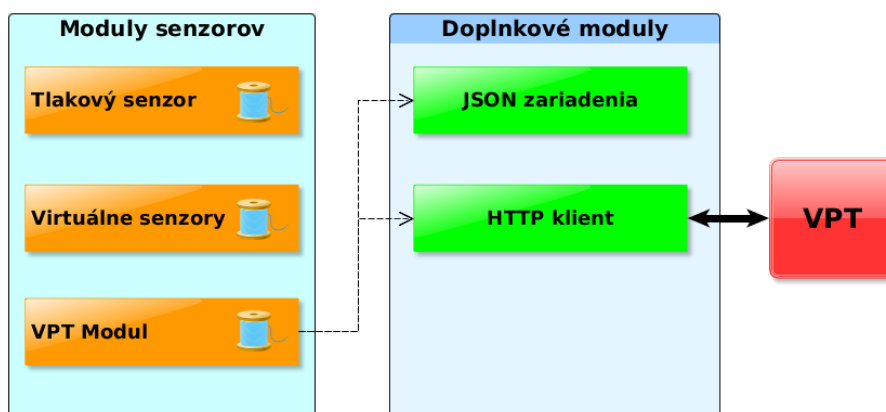
V systéme BeeeOn bolo nutné integrovať podporu termoregulátora VPT na dvoch miestach. Prvým miestom bola integrácia podporovaných vlastností (senzorov a riadiacich prvkov) do Tabuľky zariadení. Integrácia termoregulátora VPT do Tabuľky zariadení je popísaná v kapitole 5 na strane 29.

4.2.1 AdaApp

Druhým miestom pre integráciu termoregulátora VPT bola hlavná aplikácia adaptéra **AdaApp**. V tejto aplikácii bolo potrebné pre termoregulátor VPT zabezpečiť podporu komunikácie prostredníctvom protokolu HTTP, extrahovanie hodnôt z JSON súboru a konvertovanie nevhodných hodnôt na hodnoty systému BeeeOn.

Návrh

Aby sa vyhovelo požiadavkám zo strany systému BeeeOn a nedošlo k narušeniu základnej architektúry aplikácie, bude všetkú komunikáciu s termoregulátorom VPT zabezpečovať samostatný **modul VPT**, ktorý bude bežať v samostatnom vlákne, podobne ako iné moduly pre komunikáciu so senzormi alebo riadiacimi prvkami. Pre lepšiu modularitu a udržiavanie kódu budú zavedené ešte dva doplnkové moduly: **JSON zariadenia** a **HTTP klient**.



Obrázok 4.1: Návrh integrácie modulov pre VPT do aplikácie AdaApp.

JSON zariadenia

Modul JSON zariadenia má zaobstarať podporu JSON formátu. Jeho primárnou úlohou je extrahovanie hodnôt z JSON súboru a konvertovanie reťazcových hodnôt na číselné možnosti pre systém BeeeOn.

Okrem týchto úloh pre vyššiu univerzálnosť celého riešenia, musí JSON modul vedieť spravovať špecifikácie podporovaných zariadení. V tomto prípade nejde o špecifikácie zariadení z Tabuľky zariadení, ale o špecifikácie JSON formátov na stránkach zariadení a popríklad doplnkové vlastnosti pre zariadenie. Medzi doplnkovými vlastnosťami si môžeme predstaviť zoznam riadiacich prvkov spolu s URL adresami a kľúčmi pre ich nastavenie pomocou HTTP metódy GET. Okrem toho, tieto špecifikácie obsahujú informácie o podporovaných skupinách v rámci hlavnej skupiny **sensors** a názvy senzorov v rámci jednotlivých

skupín. Tieto špecifikácie tiež obsahujú mapovanie názvu hodnôt na identifikátory senzorov a riadiacich prvkov v Tabulke zariadení, ako aj potrebné údaje pre konvertovacia funkciu reťazcových hodnôt na číselné možnosti do systému BeeeOn. Presný popis špecifikácií podporovaných zariadení je v sekcii 5.2.2 Špecifikácia zariadení na strane 32.

HTTP klient

Doplnkový modul HTTP klient pridáva do aplikácie AdaApp podporu komunikácie prostredníctvom protokolu HTTP. Teda jeho hlavnou úlohou je zabezpečiť komunikáciu s termoregulátorom VPT. Okrem komunikácie musí HTTP klient tiež vedieť zaobstarať prehľadávanie siete zamerané na detekciu termoregulátorov VPT pripojených v sieti.

VPT modul

VPT modul v tomto prípade vystupuje ako centrálny prvok, ktorý zaobstaráva beh celého senzorového modulu v samostatnom vlákne. Jeho hlavnou úlohou je obsluhovať zvyšné dva doplnkové moduly **JSON zariadenia** a **HTTP klient**, medzi ktorými sprostredkúva komunikáciu. Medzi ďalšie funkcie, ktoré musí VPT modul zaobstarať je komunikácia s Aggregátorom a centrálna správa napárovaných termoregulátorov VPT. Okrem toho bude možné VPT modul využiť v prípade rozširujúcich sa požiadaviek na úpravu získaných hodnôt z termoregulátora VPT, čo umožní ponechať moduly **JSON zariadenia** a **HTTP klient** univerzálnymi.

4.3 Zhrnutie

Ako bolo naznačené v sekcii 4.2.1 **AdaApp** po konzultácii so spoločnosťou Thermona, bolo umožnené integrovať vlastný dátový súbor vo formáte JSON do termoregulátora VPT. Spoločnosť Thermona poskytla všetky zdrojové kódy webových stránok, ako aj potrebnú dokumentáciu pre ich vytvorenie a uloženie do modulov VPTLAN a VPTWIFI.

Tiež bolo umožnené, aby projekt BeeeOn spravoval tento dátový súbor vo formáte JSON. Týmto rozhodnutím sme získali aj kontrolu nad jeho ďalším vývojom a udržiavaním. Vďaka získaným privilegiam, môžeme v budúcnosti predísť problémom s nekompatibilnými verziami dátového súboru a ľahšie riešiť vzniknuté problémy.

V rámci systému BeeeOn bude prebiehať výsledná implementácia na dvoch miestach:

- **Tabulka zariadení**
- **AdaApp**

Tento návrh implementácia termoregulátora VPT rozšíri funkcionality AdaApp o podporu JSON formátu a komunikačný protokol HTTP. Túto funkcionality zaobstarajú dva nové doplnkové moduly, ktoré boli navrhnuté pre potreby termoregulátora VPT. Pri ich návrhu sa brala do úvahy možnosť ich ďalšieho použitia aj pre iné zariadenia. Výsledné rozšírenie aplikácie AdaApp pridá až tri moduly, z toho dva sú univerzálne (**JSON zariadenia** a **HTTP klient**). Tretí **VPT modul** je priamo určený pre termoregulátor VPT.

Kapitola 5

Implementácia

Integrácia termoregulátora VPT do systému BeeeOn okrem rozšírenia AdaApp a Tabuľky zariadení vyžaduje rozšírenie na samotnom termoregulátore VPT. Cela implementácia obnáša vytvorenie dvoch nových formátov pre súbory vo formáte JSON, naprogramovanie troch nových modulov v jazyku C++ a pridanie špecifikácie termoregulátora do Tabuľky zariadení.

5.1 VPT

Pri implementácii vlastného dátového súboru vo formáte JSON, budú bližšie rozpísané jednotlivé časti JSON formátu a ich funkcionality. V rámci implementácie tiež bude opísaný postup pridania nového dátového súboru do termoregulátora VPT.

5.1.1 Implementácie JSON súboru pre termoregulátor VPT

Na základe návrhu sa pri integrácii termoregulátora VPT využije možnosť rozšírenia pôvodných webových stránok o dátový súbor vo formáte JSON. Finálna implementácia súboru je napísaná v anglickom jazyku. Dôvodom pre výber anglického jazyka bolo jeho využitie ako základného jazyka pre systém BeeeOn. Nakoľko je súbor vo formáte JSON určený pre spracovanie pomocou počítača, nemal by výber jazyka predstavovať žiadne problémy, aj keď je termoregulátor VPT určený pre český trh. Výsledná implementácia pre termoregulátor VPT s bližším popisom jednotlivých častí a príkladmi potom vyzerá nasledovne:

Úroveň 1 – Všeobecné informácie o zariadení

```
1 {  
2   "vendor": "Thermona",  
3   "device": "Regulator VPT LAN v1.0",  
4   "version": 2016021100,  
5   "id": "~MACA~",  
6   "sensors": {  
7     ...
```

Zdrojový kód 5.1: 1. úroveň dátového súboru vo formáte JSON (skrátaná verzia).

- **vendor** (výrobca): Thermona – Určuje výrobcu daného zariadenia. V aktuálnej implementácii ide o doplnkový parameter, ktorý sa nepoužíva. Jeho využitie je plánované

v budúcnosti, aby eliminoval problémy s duplicitnými názvami zariadení od rôznych výrobcov.

- **device** (zariadenie): Regulátor VPT LAN v1.0 – Určuje konkrétne zariadenie od daného výrobcu. Tento parameter sa používa pri výbere korektnej špecifikácie dátového súboru vo formáte JSON pre modul **JSON zariadenia**.
- **version** (verzia): 2016021100 – Verzia dátového súboru vo formáte JSON slúži ako doplnkový parameter pre modul **JSON zariadenia** pri výbere konkrétnej špecifikácie. Zavedenie verzie ma umožniť ľahšie zavádzať nové verzie dátových súborov a pomôcť udržiavať spätnú kompatibilitu so staršími verziami. Dôvodom zavedenia tohto parametru sú hlavne predané termoregulátory VPT koncovým užívateľom, kde nie je možné na diaľku upgradovať verzie dátových súborov. Formát verzie je potom založený na dátume, kedy boli vykonané zmeny. Presný formát je v poradí rok, mesiac, deň a revízia. Prvé tri údaje, ktoré určujú dátum sa menia v prípade, keď sú vykonané vo formáte väčšie zmeny. Posledné dvojčiferné číslo určuje revíziu od poslednej veľkej zmeny. Toto číslo je určené len pre malé zmeny vo formáte. V tomto prípade je dôležité si uvedomiť, že pod zmenou formátu sa myslí zmena senzorových skupín, alebo nejaké zmeny v senzoroch alebo riadiacich prvkoch. Samotné úrovne JSON formátu dátového súboru musia byť podľa tohto popisu dodržané.
- **id** (MAC adresa): 0003941BAF5D – MAC adresa termoregulátora VPT zariadenia bude v systéme BeeeOn použitá ako jedinečný číselný identifikátor zariadenia. Tento identifikátor je tiež v projekte označený ako **euaid** a jeho veľkosť je 32 bitov. Prvý bajt **euaid** je ale rezervovaný ako identifikátor určitého typu skupiny zariadení. V prípade termoregulátora VPT je hodnota tohto bajtu 0xA4. Nakoľko je veľkosť MAC adresy 48 bitov a v **euaid** je možné použiť len 24 bitov, budú pre identifikáciu termoregulátora VPT v systéme BeeeOn použité len posledné 3 bajty MAC adresy. Tento spôsob získania **euaid** môže viesť k vytvoreniu rovnakého **euaid** pre viac zariadení. V prípade termoregulátora VPT, by ale tento jav nemal nastať, nakoľko čipy, ktoré zabezpečujú komunikáciu pomocou protokolu HTTP sú od spoločnosti Connect One a táto spoločnosť ma zaregistrovaný len OUI 00-03-94 [1].
- **sensors** (senzory) – Zapuzdruje do seba všetky senzory, ktoré sú následne rozdelené do skupín.

Úroveň 2 – Skupiny senzorov

```
1 "sensors": {  
2     "ZONE_1": {...},  
3     "ZONE_2": {...},  
4     "ZONE_3": {...},  
5     "ZONE_4": {...},  
6     "BOILER": {...}  
7 }
```

Zdrojový kód 5.2: 2. úroveň dátového súboru vo formáte JSON (skrátaná verzia).

Skupiny senzorov v JSON formáte v dátovom súbore sú reprezentované poliami pomenovaných štruktúr. Týchto štruktúr je v implementácii termoregulátora VPT celkovo päť. Prvé štyri z nich reprezentujú jednotlivé zóny a posledná skupina zapuzdruje senzory určené

pre kotol a samotný termoregulátor VPT. Samotné skupiny zón majú identické pomenovania senzorov, ktorých je v každej skupine 21. Skupina určená pre kotol, potom obsahuje šesť senzorov pre kotol a dva nevyužitú senzory pre termoregulátor VPT (Aktuálna IP adresa, BeeOn register).

Úroveň 3 – Sensory

```
1 "ZONE.1": {
2     "MOD_BOILER_OPERATION_TYPE": "~P019~",
3     "MOD_BOILER_OPERATION_MODE": "~PE044~",
4     "MOD_REQUESTED_ROOM_TEMPERATURE": "~P023~",
5     "MOD_CURRENT_ROOM_TEMPERATURE": "~P032~",
6     ...
7 },
8 ...
```

Zdrojový kód 5.3: 3. úroveň dátového súboru vo formáte JSON (skrátaná verzia).

Na poslednej úrovni sú samotné senzory. Ide o páry názvov senzorov a ich hodnôt. Názvy senzorov boli odvodené z Tabuľky zariadení a hodnota v implementácii reprezentuje názov špeciálnej premennej. Pri načítaní webovej stránky potom komunikačný modul termoregulátora VPT nahradí názov špeciálnej premennej za hodnotu v danom registri.

Pre zobrazenie všetkých hodnôt bol vybraný dátový typ reťazec. Dôvodom k tomuto kroku aj pre číselné hodnoty bolo používanie čiarky pri číslach s plávajúcou desatinnou čiarkou. Podľa štandardu ECMA-404 [11] môžu byť čísla s plávajúcou desatinnou čiarkou oddelené len pomocou bodky (.). Ďalším dôvodom bolo aj zobrazovanie znakov (napríklad: --) pri určitých hodnotách v prípade poruchy, alebo nezapojeného senzoru.

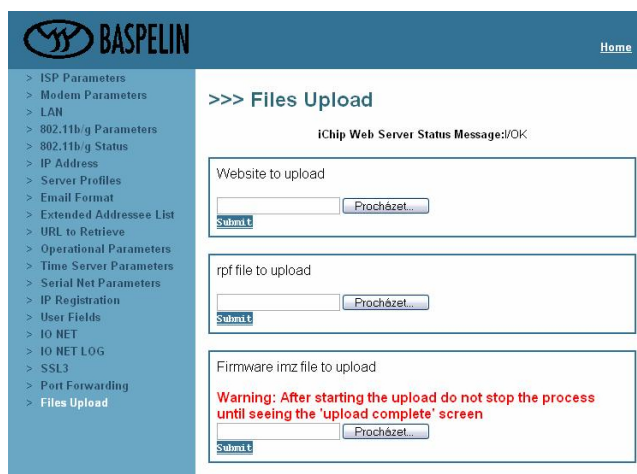
5.1.2 Integrácia do termoregulátora VPT

K integrácii rozširujúceho dátového súboru vo formáte JSON je potrebné zdrojové kódy webových stránok termoregulátora VPT. Nahratie rozšírených webových stránok je potom možné pomocou konfiguračných webových stránok komunikačného modulu termoregulátora VPT.

Pre nahratie sa používa formát IMG, ktorý je potrebné vytvoriť pomocou aplikácie od spoločnosti Connect One s názvom iChipConfig Utility. Pre vytvorenie požadovaného obrazu IMG sa budeme riadiť podľa **sekcie 3.5** v užívateľskom manuály [14]. Pre správne vygenerovanie obrazu použijeme nasledujúci postup:

1. **Zdrojové kódy** – Zdrojové kódy musia byť umiestnené v jednom spoločnom adresári, ktorý si otvoríme/označíme v aplikácii iChipConfig Utility. Ku zdrojovým kódom webových stránok je potrebné ešte pred otvorením vložiť vytvorený dátový súbor vo formáte JSON pod názvom **values.json**.
2. **Úvodná stránka** – V okne nižšie s obsahom adresára kliknutím na súbor index.html vykonáme jeho označenie za úvodnú stránku. Skontrolujeme výber dobrého súboru v okne **Default Page** (Úvodná stránka). Ak súbor nepôjde vybrať kliknutím, môžeme jeho názov priamo napísať do poľa pre **Default Page**.
3. **Platforma** – V sekcii platforma vyberieme cieľový čip, pre ktorý budú webové stránky určené. Pre termoregulátor VPT použijeme možnosť **CO2144**.

4. **Parametre** – Pre správnu funkciu je potrebné nahráť zoznam špeciálnych premenných, ktoré sa mapujú na registri termoregulátora VPT. Tento zoznam je súčasťou zdrojových kódov webových stránok pre termoregulátor VPT. Kliknutím na **Load file** (Nahráť súbor) vyvoláme okno pre nahranie súboru, kde vyberieme požadovaný súbor s názvom **ParLen4.txt**.
5. **Vytvorenie obrazu** – Výsledný obraz vo formáte IMG vytvoríme kliknutím na tlačidlo **Pack** (Zabalíť).
6. **Zapnutie aktualizáčného režimu** – Pred aktualizáciou webových stránok je potrebné pomocou displeja termoregulátora VPT aktivovať v nastaveniach aktualizáčny režim.
7. **Nahratie nového obrazu** – Vo webovom prehliadači si otvoríme konfiguračnú stránku termoregulátora VPT na adrese **http://xxx.xxx.xxx.xxx/ichip** (xxx.xxx.xxx.xxx – IP adresa termoregulátora VPT). Na konfiguračnej stránke si v ľavej časti klikneme na sekciu **Files Upload** (Nahranie súborov). V danej sekcii budeme pracovať s prvým rámčekom s názvom **Website to upload** (Nahranie webových stránok). Kliknutím na tlačidlo **Browse** (Prehľadávať) si vyberieme vytvorený obraz webových stránok vo formáte IMG. Následné nahranie nových webových stránok potvrdíme kliknutím na tlačidlo **Submit** (Nahráť).
8. **Testovanie nových stránok** – Ak prebehlo nahranie obrazu webových stránok v poriadku, môžeme otestovať ich funkčnosť otvorením implementovaného dátového súboru vo formáte JSON. V prehliadači implementovaný dátový súbor otvoríme na adresa **http://xxx.xxx.xxx.xxx/values.json** (xxx.xxx.xxx.xxx – IP adresa termoregulátora VPT). Pri správnom nahraní, mala by sa nám zobrazíť stránka a špeciálne premenné by mali byť nahradené obsahom príslušných registrov. V niektorých prípadoch, hneď po nahraní novej verzie webových stránok je potrebné chvíľu počkať, pokiaľ sa obsahy registrov budú zobrazovať. V každom prípade, by ale nemalo byť možné vidieť názvy špeciálnych premenných.



Obrázok 5.1: Konfiguračná webová stránka termoregulátora VPT [19].

5.2 BeeeOn

Implementácia do systému BeeeOn je rozsiahlejšia oproti rozšíreniu webových stránok termoregulátora VPT. Väčšia časť implementácie sa bude týkať rozšíreniu aplikácie AdaApp, kde jednotlivé rozšírenia budú rozobrané v rámci každého pridaného modulu podľa návrhu.

5.2.1 Tabuľka zariadení

Implementácia podpory termoregulátora VPT je aktuálne zariadenie s najväčším počtom senzorov a riadiacich prvkov v Tabuľke zariadení. Ich celkový počet je 64. Jednotlivé senzory, alebo riadiace prvky sú buď číselného typu, alebo na základe zoznamu možností. Konkrétnu implementáciu si ukážeme na jednom senzore a jednom riadiacom prvku.

Zoznam možností

Zoznam možností je použitý pre senzory alebo riadiace prvky: typy regulácie, režimi prevádzky, režim kotla, stav kotla, protimrazová ochrana. Ako ukážkový popis implementácie si uvedieme riadiaci prvok protimrazová ochrana pre zónu 1.

```
1 <actuator id="11" type="0x01" >
2     <order>13</order>
3     <group>T:ZONE_1</group>
4     <name>T:FROST_PROTECTION</name>
5     <values name="T:VAL_FROST_PROTECTION" >
6         <value id="0">T:OFF</value>
7         <value id="1">T:ON</value>
8     </values>
9     <rules>
10         <if value="0" >
11             <hide-module id="12" />
12             <hide-module id="13" />
13         </if>
14     </rules>
15 </actuator>
```

Zdrojový kód 5.4: Protimrazová ochrana pre zónu 1 v Tabuľke zariadení.

- **actuator** (riadiaci prvok) – Obsahuje dva atribúty, ktoré obsahujú jedinečný identifikátor (**id**) senzoru alebo riadiaceho prvku v zariadení. Atribút **type** určuje typ hodnoty s ktorou daný senzor, alebo riadiaci prvok pracuje. Podporované typy sú definované v súbore **types.xml**.
- **order** (poradie) – Určuje poradie pri zobrazení v skupine na koncových ovládacích stanicach (napríklad: android aplikácie BeeeOn).
- **group** (skupina) – Zoskupuje senzory alebo riadiace prvky do logickej skupiny. V prípade termoregulátora ide o zóny vykurovania (ZONE_1, ZONE_2, ZONE_3, ZONE_4) a kotol (BOILER).
- **name** (názov) – Slovné pomenovanie, ktoré sa používa pre koncové ovládacie stanice. Tento názov sa ešte pomocou jazykových súborov mapuje do konečnej podoby, podľa cieľového jazyka.

- **values** (hodnoty) – Hodnoty sa používajú len pre typ hodnôt 0x01, ktorý označuje zoznam možností. Atribút **name** (názov) je následne tiež použitý pre koncové ovládacie stanice pre mapovanie jednotlivých možností na korektnú podobu v cieľovom jazyku.
- **value** (hodnota) – Hodnota popisuje jednu možnosť zo zoznamu možností, ktoré môže daný senzor alebo riadiaci prvok nadobudnúť. Atribút **id** v tomto prípade označuje číselnú reprezentáciu danej hodnoty v systéme BeeeOn, ktorá sa používa v celom systéme BeeeOn pre popis aktuálneho stavu senzoru, alebo riadiaceho prvku. Všetky hodnoty teda musia byť konvertované do tejto podoby, aby vedeli s nimi komponenty systému BeeeOn pracovať. Následná hodnota elementu sa potom tiež používa pre koncové ovládacie stanice pre mapovanie na korektný preklad.
- **rules** (pravidlá) – Pravidlá sa používajú pre pridanie doplnkovej logiky prevažne pre koncové ovládacie stanice. V prípade termoregulátora VPT sa pracuje len s elementom **hide-module**.
- **if** (podmienka) – Podmienka obsahuje atribút **value** (hodnota), kde je uvedená hodnota pre daný senzor alebo riadiaci prvok, pri ktorej sa majú aplikovať jednotlivé pravidlá.
- **hide-module** (skryť-modul) – Je pravidlo, ktoré určuje pre koncové ovládacie stanice, aby nezobrazilo senzor alebo riadiaci prvok s identifikačným číslom v rovnakom zariadení, ktorý je uvedený v atribúte **id**.

Číselný typ

Číselný typ senzorov a riadiacich prvkov prevláda v implementácii termoregulátora VPT nad senzormi a riadiacimi prvkami, ktoré pracujú so zoznamom možností. Okrem vyššie vymenovaných senzorov a riadiacich prvkov v sekcii **Zoznam možností**, sú všetky senzory alebo riadiace prvky práve číselného typu. Termoregulátor VPT využíva číselné typy reprezentujúce: teplotu (°C), tlak (hPa) a výkon (%). Ako bolo spomenuté v sekcii **2.6** Sensory/Riadiace prvky, riadiace prvky sú vlastne senzory s rošírenou funkciou pre zmenu svojho stavu na podnet užívateľa. Preto ich implementácia v tabuľke zariadení je veľmi podobná a väčšina elementov bola popísaná v sekcii **Zoznam možností** na strane **29**. Ako príklad si v tom prípade uvedieme senzor pre aktuálnu teplotu vykurovacej vody pre zónu 1. Nakoľko bola väčšina elementov už popísaná zameriame sa len na nepopísané elementy **constraints**, **min**, **max**, **step**.

```

1 <sensor id="5" type="0x02" >
2   <order>4</order>
3   <group>T:ZONE_1</group>
4   <name>T:MOD_CURRENT_WATER_TEMPERATURE</name>
5   <constraints>
6     <min>20.0</min>
7     <max>120.0</max>
8     <step>1</step>
9   </constraints>
10 </sensor>

```

Zdrojový kód 5.5: Aktuálna teplota vykurovacej vody pre zónu 1 v Tabuľke zariadení.

- **constraints** (obmedzenia) – Ako vyplýva z názvu samotného elementu, ide o obmedzenie rozsahu hodnôt, ktoré môže senzor nadobúdať. Ich využitie sa používa hlavne v prípade riadiacich prvkov na koncových ovládacích staniciach, pre obmedzenie zmeny stavu riadiaceho prvku na neplatnú hodnotu.
- **min** (minimálna hodnota) – Minimálna hodnota určuje spodnú hranicu rozsahu hodnôt.
- **max** (maximálna hodnota) – Maximálna hodnota určuje hornú hranicu rozsahu hodnôt.
- **step** (krok) – Krok určuje rozsah medzi dvoma najbližšími hodnotami.

Jazykova podpora

Okrem implementácie podpory termoregulátora VPT do Tabuľky zariadení bolo potrebné v rovnakom repozitári pre koncové ovládacie stanice implementovať názvy jednotlivých senzorov a riadiacich prvkov pre podporované jazyky. V projekte BeeeOn je pre koncové ovládacie stanice podpora troch jazykov: anglicky, česky a slovensky. Implementácia je pre každý jazyk rovnaká, jediný rozdiel je v hodnote elementov, ktorá sa mení podľa príslušného jazyka. Ako ukážku použijeme protimrazovú ochranu, nakoľko obsahuje aj implementáciu hodnôt zoznamu možností.

```

1 <string name="frost_protection" >Protimrazová ochrana</string>
2 <values name="val_frost_protection" >
3     <string name="off" >Vypnuta</string>
4     <string name="on" >Zapnuta</string>
5 </values>
```

Zdrojový kód 5.6: Jazyková podpora pre protimrazovú ochranu v slovenskom jazyku.

- **string** (reťazec) – Atribútu **name** (názov) je pridelená hodnota elementu **name** pre senzor alebo riadiaci prvok z Tabuľky zariadení, pre ktorý chceme definovať preklad v cieľovom jazyku. Potom hodnota elementu **string** definuje preklad v cieľovom jazyku.
- **values** (hodnoty) – Element hodnoty zoskupuje všetky možnosti zo zoznamu možností pre hodnoty definované v atribúte **name** (názov), ktorý odkazuje na hodnotu atribútu **name** v elemente **values** v Tabuľke zariadení pre nejaký senzor, alebo riadiaci prvok.

5.2.2 AdaApp – JSON zariadenia

Pri implementácii doplnkového modulu JSON zariadenia bolo nutné zabezpečiť funkcionálnu podporu JSON formátu a spracovanie špecifikácií zariadení. Pre podporu JSON formátu bol využitý v implementácii Poco JSON Parser s využitím interných štruktúr DynamicStruct. Najviac práce si vyžiadala implementácia špecifikácií zariadení.

Mapovanie na špecifikácie zariadení

Implementácia špecifikácií zariadení nevyžadovala len implementáciu kódu pre spracovanie, ale aj zmenu adresovej štruktúry konfiguračných súborov aplikácie **AdaApp**. Táto zmena spočívala hlavne v pridaní konfiguračného súboru **json.ini** a adresára **json_devices**,

ktorý bude obsluhovať špecifikácie zariadení vo formáte JSON. Samotný konfiguračný súbor **json.ini** je pomocný súbor, ktorý sa používa pre mapovanie na korektnú špecifikáciu zariadenia. Takéto riešenie bolo vybrané pre zvýšenie bezpečnosti, aby nebolo možné podvrhnutím neplatného dátového súboru vo formáte JSON termoregulátora VPT otvoriť nejaký systémový súbor.

```
1 [Regulator VPT LAN v1.0]
2 2016021100=regulator-vpt-lan-v1.0.json
```

Zdrojový kód 5.7: Konfiguračný súbor pre mapovanie špecifikácií.

Samotný obsah konfiguračného súboru je jednoduchý. Súbor je rozdelený na sekcie, kde každá sekcia má rovnaké pomenovanie, ako názov zariadenia, ktorý je hodnotou kľúča **device** v dátovom súbore vo formáte JSON v termoregulátore VPT. Rovnakým spôsob sú riešené názvy kľúčov v sekciách, ktoré majú zhodný názov s hodnotou v kľúči **version**. Ako hodnoty kľúčov sú potom uvedené názvy súborov obsahujúce špecifikáciu zariadenia pre konkrétnu verziu dátového súboru vo formáte JSON, ktoré sú uložené v adresári **json_devices**. Týmto spôsobom je zabezpečená podpora rôznych dátových súborov vo formáte JSON pre termoregulátor VPT.

Špecifikácia zariadení

Špecifikácia zariadenia bola vytvorená za účelom podpory rôznych zariadení a dátových súborov vo formáte JSON. Samotná špecifikácia zariadenia je tiež implementovaná vo formáte JSON, čím nebolo nutné do modulu JSON zariadenia implementovať podporu ďalších formátov.

```
1 {
2     "vendor": "Thermona",
3     "device": "Regulator VPT LAN v1.0",
4     "device_id": 6,
5     "functions": [ "sensors", "actuators", "converter" ],
6     "sensors": { ... },
7     "actuators": [ ... ],
8     "converter": [ ... ]
9 }
```

Zdrojový kód 5.8: Špecifikácia termoregulátora VPT – úroveň 1 (skrátaná verzia).

- **vendor** (výrobca) – Určuje výrobcu konkrétného zariadenia.
- **device** (zariadenie) – Názov zariadenia, pre ktorý je špecifikácia určená. Táto hodnota by mala byť zhodná s hodnotou rovnakého kľúča v dátovom súbore vo formáte JSON.
- **device_id** (identifikátor zariadenia) – Definuje identifikátor zariadenia v Tabuľke zariadení, ktorý sa používa v interných štruktúrach pre komunikáciu medzi modulmi.
- **functions** (funkcie) – Ide o kolekciu podporovaných funkcií pre zariadenie. Aktuálne sú podporované tri funkcie: senzory, riadiace prvky a konvertor. Tieto funkcie ďalej musia byť definované v zdrojovom kóde a špecifikácia obsahuje potrebné hodnoty pre ich funkčnosť. Výhodou využitia tejto kolekcie je možnosť, aby bola špecifikácia univerzálna.

Sensors (Senzory)

Funkcia senzory je určená pre čítanie hodnôt z dátového súboru vo formáte JSON. Táto funkcia obsahuje všetky podporované skupiny senzorov na webovej stránke aj zoznam senzorov.

```
1 "sensors": {
2     "groups": ["ZONE_1", "ZONE_2", "ZONE_3", "ZONE_4", "BOILER"],
3     "ZONE_1": [{
4         "name": "MOD_BOILER_OPERATION_TYPE",
5         "id": 0,
6         "action": "convert"
7     }],
8     ...
9     {
10        "name": "MOD_REQUESTED_ROOM_TEMPERATURE",
11        "id": 2,
12        "action": false
13    },
14    ...
15    ],
16    "ZONE_2": [...],
17    "ZONE_3": [...],
18    "ZONE_4": [...],
19    "BOILER": [...]
20 }
```

Zdrojový kód 5.9: Špecifikácia termoregulátora VPT – senzory (skrátaná verzia).

- **groups** (skupiny) – Táto kolekcia obsahuje názvy skupín, ktoré zoskupujú senzory v dátovom súbore vo formáte JSON. Tieto názvy sú tiež použité pre názvy skupín senzorov v špecifikáciách.
- **Skupiny senzorov** – Skupiny senzorov, zoskupujú senzory rovnako ako v dátovom súbore vo formáte JSON a tiež s nimi zdieľajú rovnaký názov.
- **name** (názov) – Obsahuje názov podporovaného senzoru.
- **id** (identifikátor) – Určuje identifikátor senzoru/riadiaceho prvku pre zariadenie v Tabuľke zariadení.
- **action** (akcia) – Doplnkový parameter, ktorý určuje ďalší postup po získaní hodnoty z dátového súboru vo formáte JSON. Akcia primárne určuje či je potrebné hodnotu pred použitím konvertovať (hodnota: **convert**) do formátu pre systém BeeeOn. Tento parameter tiež môže byť využitý pre definovanie iných funkcií po načítaní hodnôt, ktoré môžu byť implementované v zdrojovom kóde, čím sa rozširujú možnosti využitia modulu JSON zariadenia aj pre iné zariadenia.

Riadiace prvky

Kolekcia reprezentujúca funkciu riadiace prvky je zhodná so skupinami senzorov z funkcie senzory. V tomto prípade každá štruktúra reprezentuje riadiaci prvok v rovnakom formáte ako senzory, ale rozširuje ho o dva nové parametre a jednému mení význam.

```
1 "actuators": [{
2     "id": 0,
3     "variable": "PE040",
4     "url": "values.json",
5     "action": "convert",
6     "name": "MOD_BOILER_OPERATION_TYPE"
7 },
8 ...
9 {
10     "id": 6,
11     "variable": "PE020",
12     "url": "values.json",
13     "action": false
14 },
15 ...
16 ],
```

Zdrojový kód 5.10: Špecifikácia termoregulátora VPT – riadiace prvky (skrátaná verzia).

- **url** – Definuje webovú stránku v URL adrese, na ktorú sa robí HTTP požiadavka s využitím metódy GET za účelom zmeniť stav riadiaceho prvku.
- **variable** (premenná) – Názov kľúča pre HTTP metódu GET pomocou ktorého sa nastavuje príslušný riadiaci prvok.
- **name** (názov) – Názov v tomto prípade sa používa v kombinácii s akciou **convert**. Táto hodnota sa ďalej použije vo funkcií **converter** pre vyhľadanie správneho zoznamu kombinácií pri prevode hodnoty pre potreby termoregulátora VPT.

Konvertor

Funkcia **converter** (Konvertor) slúži pre prevod hodnôt medzi systémom BeeeOn a termoregulátorom VPT. V prípade termoregulátora VPT sa využíva len pre konvertovanie reťazcov na číselné možnosti zo zoznamu možností v Tabuľke zariadení a naopak.

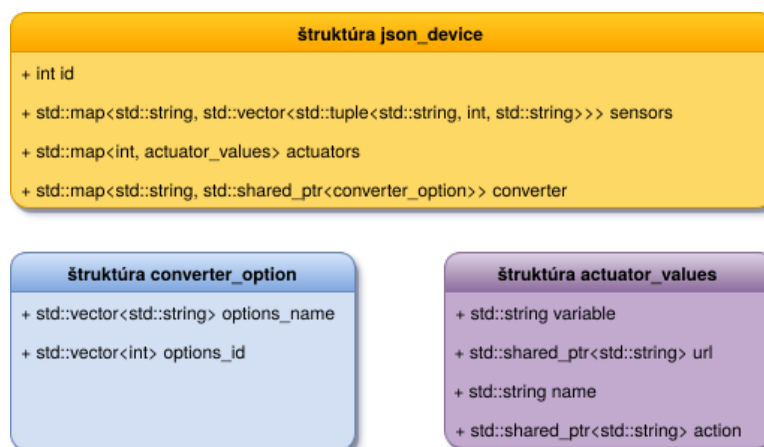
```
1 "converter": [{
2     "support": ["MOD_BOILER_OPERATION_TYPE"],
3     "options": [
4         ["Vypnuto.", 0 ],
5         ["Pok.ter.", 1 ],
6         ["Ekviterm", 2 ],
7         ["Tep.vody", 3 ],
8         ["OhrevTUV", 4 ]
9     ]
10 },
11 ...
12 ]
```

Zdrojový kód 5.11: Špecifikácia termoregulátora VPT – konvertor (skrátaná verzia).

- **support** (podpora) – Definuje názvy podporovaných senzorov alebo riadiacich prvkov, ktoré využívajú nižšie uvedené kombinácie možností.
- **options** (možností) – Obsahuje kolekcie kombinácií možností pre prevod, kde prvá hodnota z kombinácie v kolekcii reprezentuje hodnotu určenú pre termoregulátor VPT a druhá jej reprezentáciu pre systém BeeeOn.

Načítanie špecifikácie

Pre rýchlejšie spracovanie špecifikácie boli vytvorené interné štruktúry pre modul JSON zariadenia, do ktorých sa požadovaná špecifikácia načíta.



Obrázok 5.2: Interné štruktúry pre špecifikáciu zariadení.

Štruktúra **json_device** reprezentuje špecifikáciu celého zariadenia, kde identifikátor **id** odpovedá hodnote **device.id** a jednotlivé mapy reprezentujú funkcie zo špecifikácie. V rámci implementácie kódu je pre každú premennú v štruktúre vytvorená samostatná funkcia z dôvodu odlišnosti formátov jednotlivých funkcií v špecifikácií. Implementácia veľmi pracuje s mapami, tento kontajner bol vybraný hlavne pre integrovanú metódu **find** (vyhľadávať).

Pre jednotlivé možnosti konvertora a riadiace prvky boli vytvorené samostatné štruktúry **converter_option** a **actuator_values**. Tieto štruktúry sa pri načítaní ukladajú do hlavnej štruktúry **json_device** ako súčasť kontajnera **std::map** a umožňujú efektívne vyhľadávanie na základe kľúčov. V prípade konvertora sa využíva ako kľúč názov senzora alebo riadiaceho prvku z kolekcie **support** v špecifikácií. Pri riadiacom prvku kľúč reprezentuje jeho identifikátor podľa Tabuľky zariadení a to z dôvodu, že sa táto štruktúra využíva explicitne pre požiadavky na zmenu stavu riadiaceho prvku termoregulátora VPT. Tieto požiadavky samozrejme pochádzajú od koncového užívateľa a teda správa, ktorá dorazí, využíva internú reprezentáciu zariadení v systéme BeeeOn.

Senzory pre internú reprezentáciu využívajú tri zanorené kontajnery. Prvý kontajner umožňuje ľahké vyhľadávanie skupín senzorov, na základe vyhľadavacieho kľúča, ktorý reprezentuje názov skupiny senzorov zo špecifikácie. Druhý kontajner reprezentuje jednotlivé senzory v skupine senzorov. Nakoľko sa hodnoty získavajú na základe tejto špecifikácie v postupnom poradí je pre sekvenčný prechod využitý iterátor cez **std::vector**. Posledný

kontajner je typu **tuple** a obsahuje špecifikácie senzora. Tieto špecifikácie sú zoradené podľa vzoru: názov senzora, identifikátor v Tabulke zariadení a akcia.

5.2.3 AdaApp – HTTP klient

Implementácia HTTP klienta je veľmi jednoduchá, nakoľko protokol HTTP zabezpečuje trieda `Poco::Net::HTTPClientSession`. V implementácii HTTP klienta je teda dôležitejšie rozšírenie pre detekciu termoregulátora VPT v sieti.

Detekcia termoregulátora VPT v sieti

Princíp detekcie je veľmi jednoduchý. Pri detekcii sa prehľadáva rozsah IP adries na sieťach v ktorých je pripojený adaptér. Tento rozsah je definovaný medzi IP adresou siete a IP adresou broadcastu. Z dôvodu vysokej časovej náročnosti na prehľadanie veľkých sietí sa rozsah siete obmedzuje na sieť s prefixom 24 pre menšie prefixy ako 24.

S rozsahom IP adries siete je možné začať prehľadávať sieť na prítomnosť termoregulátorov VPT. Prehľadávanie siete prebieha v dvoch fázach. V prvej fáze sa testuje otvorenie portu na ktorom majú termoregulátory VPT komunikovať. Každé pripojenie má definovaný časový limit **10 milisekúnd**, čo by nemalo pri pripojení pomocou Ethernetu alebo Wi-fi predstavovať v lokálnej sieti a bežných podmienkach žiadne problémy. Pri úspešnom pripojení sa spojenie uzavrie a IP adresa je zaznamenaná ako potenciálny termoregulátor VPT. Keď sa prehľadávanie v prvej fáze ukončí, je vrátený modulu VPT zoznam potenciálnych IP adries, ktoré môžu predstavovať termoregulátory VPT.

Druhá fáza už prebieha v module VPT, kde prebieha už párovanie termoregulátorov VPT. Pri párovaní už sa v module HTTP klient využíva podpora HTTP protokolu, kde sa na potenciálnu IP adresu vytvorila požiadavka na prevzatie webovej stránky na URL ceste `/values.json`. Tieto HTTP požiadavky pracujú s väčším časovým limitom, ktorý je nastavený až na 3 sekundy. Väčší časový limit počas druhej fázy bol zvolený primárne s prihliadnutím, že by sa na sieti nemal nachádzať veľký počet webových serverov.

Tieto časové obmedzenia spoločne s obmedzením rozsahu siete hlavne umožňujú prehľadať celú sieť do 20 sekúnd. Interval 20 sekúnd je zvolený na základe párovacieho režimu na koncových ovládacích stanicách, ktorý trvá presne 30 sekúnd. Rozdiel 10 sekúnd je kvôli časovej rezerve, nakoľko rýchlosť prehľadávania siete závisí od počtu pripojených zariadení a hlavne počtu webových serverov na sieti.

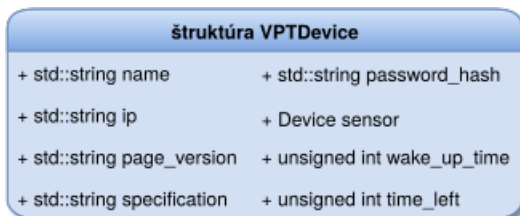
Jedným z prvých problém, ktorý vznikol pri implementácii spôsoboval framework Poco. V prípade, že nebola explicitne definovaná broadcastová IP adresa na rozhraní, framework Poco vráti pre sieťové rozhranie IP adresu pridelenú na rozhraní ako broadcastovú IP adresu. Pre vyriešenie problému bol zdroj IP adresy broadcastu použitý výpočet z IP adresy na sieťovom rozhraní v kombinácii s IP adresou masky siete.

5.2.4 AdaApp – VPT modul

Implementácia VPT modulu je prispôbená iným sensorovým modulom aplikácie AdaApp. Medzi takéto prispôsobenia patrí hlavne implementácia vlákien, ktorú zabezpečuje trieda `Poco::Runnable` z ktorej VPT modul dedí a pridanie konfiguračného súboru `vpt_sensor.ini` medzi ostatné konfiguračné súbory aplikácie AdaApp. Hlavnú časť implementácie modulu VPT tvorí potom spravovanie napárovaných termoregulátorov VPT, podpora hesla pre nastavovanie riadiacich prvkov pre termoregulátor VPT a riadenie komunikácie medzi termoregulátorom VPT s ostatnými časťami aplikácie BeeOn.

Spravovanie napárovaných termoregulátorov VPT

Spravovanie termoregulátorov VPT je zabezpečené pomocou štruktúry **VPTDevice**. Táto štruktúra reprezentuje základné údaje potrebné pre komunikáciu, vyhľadávanie špecifikácie zariadenia a podpory časovača pre získavanie hodnôt z termoregulátora VPT.



Obrázok 5.3: Interná štruktúra reprezentujúca napárovaný termoregulátor VPT.

Popis štruktúry VPTDevice:

- **name** (názov) – Obsahuje názov zariadenia, ktoré sa počas párovania prevezme z dátového súboru vo formáte JSON z termoregulátora VPT.
- **ip** (ip adresa) – Reprezentuje IP adresu termoregulátora VPT.
- **page_version** (verzia webových stránok) – Rovnako ako názov je z webových stránok prebratá ich verzia.
- **specification** (špecifikácia) – Tento parameter slúži pre zníženie počtu operácií tým, že spája názov a verziu webových stránok.
- **password_hash** (hash hesla) – Ukladá vygenerovaný hash potrebný pre prihlásenie do termoregulátora VPT.
- **sensor** (senzor) – Interná štruktúra reprezentujúca zariadenie v aplikácií AdaApp.
- **wake_up_time** (čas prebudenia) – Uchováva časový interval medzi dvomi aktualizáciami hodnôt senzorov z termoregulátora VPT.
- **time_left** (zostávajúci čas) – Slúži pre odpočet času do ďalšej aktualizácie hodnôt senzorov. Po skončení každej aktualizácii je znovu nastavený na čas prebudenia. Minimálna hranica pre termoregulátor VPT je nastavená na 15 sekúnd.

Podpora hesla

V aktuálnej implementácii sa pracuje len s podporou jedného hesla pre všetky termoregulátory VPT obsluhované jedným adaptérom. Heslo pre termoregulátory VPT je uložené v konfiguračnom súbore **vpt_sensor.ini** pod kľúčom **password** (heslo). Nakoľko je heslo vyžadované len pri nastavení riadiacich prvkov, jeho potrebu nie je možné detekovať pri klasickom párovaní, ktoré sa spolieha len na načítanie hodnôt. Preto detekcia potreby hesla bola zavedená v kóde, ktorý má na starosti preposielanie požiadavky do modulu HTTP klient. V tomto prípade môže byť požiadavka odoslaná až 3-krát, kde prvá požiadavka pre nastavenie riadiaceho prvku je odoslaná bez hesla. Vrátený obsah načítanej webovej stránky sa následne kontroluje na formát. V prípade, že ide o dátový súbor vo formáte JSON prebehlo nastavenie riadiaceho senzoru v poriadku. Ak sa vráti webová stránka vo

formáte HTML, ide o stránku s požiadavkou pre zadanie hesla. Z tejto stránky sa spojením nasledujúcich dvoch reťazcov `'var randnum = '` a `([0-9]+)` vytvorí regulárny výraz pre vyhľadanie náhodného vygenerovaného čísla potrebného pre hashovaciu funkciu. Následne sa z konkatenácie tohto náhodného čísla a hesla s konfigurácie vygeneruje hash, ktorý sa uloží do `password_hash`. Takto vygenerovaný hash sa následne vždy pripojí do HTTP metódy GET ako hodnota kľúča `_HOSTPWD`.

Riadenie komunikácie

Ako bolo naznačené v návrhu VPT modul zaobstaráva všetku komunikáciu medzi jednotlivými modulmi aplikácie AdaApp. Z hľadiska komunikácie si bližšie popíšeme implementáciu komunikácie medzi jednotlivými modulmi pri načítaní hodnôt zo senzorov a nastavení riadiaceho prvku.

Načítanie hodnôt:

1. (**VPT Modul**) – Získanie hodnôt napárovaných termoregulátorov VPT iniciuje VPT modul, kde uplynul zostávajúci čas na hodnotu **0** v štruktúre **VPTDevice**.
2. (**VPT modul** → **HTTP klient**) – VPT modul následne volá modul HTTP klienta s požiadavkou pre získanie obsahu dátového súboru vo formáte JSON z termoregulátora VPT s IP adresou uloženou v štruktúre **VPTDevice**.
3. (**HTTP klient** ↔ **Termoregulátor VPT**) – HTTP klient komunikuje prostredníctvom HTTP protokolu s termoregulátorom VPT za účelom získania webovej stránky `values.json`.
4. (**HTTP klient** → **VPT modul**) – Načítaný dátový súbor vo formáte JSON je vrátený do VPT modulu ako `std::string`.
5. (**VPT modul** → **JSON zariadenia**) – Následne VPT modul volá JSON zariadenia, kde predáva obsah súboru spolu s názvom špecifikácie pre dané zariadenie.
6. (**JSON zariadenia**) – JSON zariadenie vyhľadá potrebnú špecifikáciu a na jej základe získa požadované hodnoty z `Poco::DynamicStruct`, kde sa predtým uloží naparsovaný obsah webovej stránky.
7. (**JSON zariadenia** → **VPT modul**) – Hodnoty senzorov sú vrátené do VPT modulu ako vektor, ktorý sa následne uloží do atribútu `sensor` v štruktúre **VPTDevice**.
8. (**VPT modul** → **Aggregator**) – V predposlednom kroku je atribút `sensor` zo štruktúry **VPTDevice** uložený do štruktúry **IOTMessage** ako atribút `device` a následne sa táto štruktúra odošle aggregatoru pre odoslanie na server.

Nastavenie riadiaceho prvku:

1. (**Aggregator** → **VPT modul**) – Aggregator ako prvé osloví jednotlivé senzorové moduly, aby sa na základe **euid** alebo **device_id** určil správny modul, ktorému bude predaná štruktúra **Command** popísaná na strane 11.
2. (**VPT modul**) – Na základe **euid** sa vyhledá štruktúra **VPTDevice**, ktorá reprezentuje cieľový termoregulátor VPT.
3. (**VPT modul** → **JSON zariadenia**) – VPT modul volá JSON zariadenia s identifikátorom riadiaceho prvku, hodnotou a názvom špecifikácie, za účelom vygenerovania URL požiadavky.
4. (**JSON zariadenia** → **VPT modul**) – URL pre požiadavku je zostavená z cesty k stránke, kľúča a hodnoty. Kľúč je špeciálna premenná spolu s adresou stránky získaná zo špecifikácie a špeciálnej premenej je priradená hodnota z argumentu. Výsledná cesta je vrátená do **VPT modulu** vo formáte **std::string**.
5. (**VPT modul** → **HTTP klient**) – Predaná cesta z JSON zariadenia je spoločne s IP adresou predaná do HTTP klienta.
6. (**HTTP klient** → **Termoregulátor VPT**) – HTTP klient kontaktuje prostredníctvom HTTP protokolu termoregulátor VPT na nastavenej ceste.
7. (**HTTP klient** → **VPT modul**) – Zobrazená webová stránka z danej cesty je vrátená do VPT modulu vo formáte **std::string**.
8. (**VPT modul** ↔ **JSON zariadenia**) – Obsah dátového súboru je predaný JSON zariadeniu pre určenie, že obsah dátového súboru je vo formáte JSON. Návratová hodnota z JSON zariadenia je typu **bool**.
9. (**VPT modul**) – Ak modul JSON zariadenia potvrdí, že dátový súbor je vo formáte JSON, nastavenie hodnoty prebehlo bez problémov a proces nastavenia riadiaceho prvku je ukončený. V prípade, že nejde o JSON formát, je potrebné pri nastavení riadiaceho prvku použiť heslo. Postup komunikácie medzi modulmi sa potom ešte môže 2-krát opakovať od piateho bodu, kde je do URL ešte pridaný kľúč **_HOSTPWD** pre heslo spolu s hashom hesla.

Kapitola 6

Testovanie

Testovanie integrácie termoregulátora VPT do systému BeeeOn prebiehalo kontinuálne počas celej doby implementácie. Počas týchto testov sa nová funkcionálna v prvých fázach testovala bez integrácie do systému BeeeOn. Tento prípad testov platí iba pre pridané moduly do aplikácie AdaApp, ktoré sa testovali buď samostatne pridaním vlastnej funkcie `main()`, ktorá aktivovala vlákno modulu VPT, alebo neprepojením VPT modulu s modulom Aggregator. Výstup v takomto prípade bol kontrolovaný pomocou zaznamenaných logov.

V konečnej fáze, keď integrácia obsahovala všetky potrebné vlastnosti pre systém BeeeOn a funkcie pre komunikáciu s termoregulátorom VPT, modulom Aggregator a medzi integrovanými modulmi sa spôsob testovania zmenil. Testovanie v tejto fáze sa už nezameriava len na nové funkcie, ale kontroluje sa funkčnosť ako celok. Na základe doby trvania a spôsobu ich vykonávania môžeme testy rozdeliť na krátkodobé a dlhodobé.

6.1 Krátkodobé testovanie

Tieto testy sú veľmi podobné testom v prvej fáze integrácie termoregulátora VPT. Testy nemajú žiaden konkrétny scénar a slúžia hlavne pre testovanie nových funkcií, alebo opráv chýb. Vykonávateľom testov je priamo programátor. Rozdiel oproti testom v pravej fáze integrácia spočíva v tom, že jednotlivé súčasti sú integrované do systému BeeeOn. Preto sa nekontroluje už len výpis z logov, ale aj zobrazenie na koncových ovládacích stanicach spoločne napríklad s databázou a výpismi logov z iných modulov. Tento spôsob testovania prebieha v takzvanej vývojovej prevádzke systému BeeeOn, kde sa testuje nová funkcionálna komponentov a kde môže dôjsť aj k poškodeniu uložených dát.

6.2 Dlhodobé testovanie

Dlhodobé testovanie prebieha v dlhšom časovom intervale a vykonáva sa dvoma spôsobmi:

- **Automatizované testy** – Ku kontrole základnej funkcionality sa používajú automatizované testy. K vykonávaniu automatizovaných testov je použitý systém Jenkins. Jednotlivé testy sú krátkodobého charakteru, ale nakoľko majú predpísaný scénar a vykonávajú sa pravidelne, sú zaradené medzi dlhodobé spôsoby testovania. Tento typ testov patrí medzi najkratšie používané a preto ich implementácia nepokrýva ešte plnú funkcionálnu integráciu termoregulátora VPT do systému BeeeOn. V aktuálnej implementácii sa testujú tieto aspekty:

- Správne ukončenie modulu.
- Kontrola prehľadanych sieťových rozhraní na prítomnosť termoregulátorov VPT.
- Detekované termoregulátory VPT na sieti.
- Korektné prečítanie hodnôt z termoregulátora VPT uložených v databáze.

Pri týchto testoch sa používa okrem zapožičaného termoregulátora VPT aj jednoduchý program, ktorý simuluje správanie sa a funkcionálnosť termoregulátora VPT.

- **Testery** – Pre dlhodobé testy sa využívajú aj externé testery. Testery pracujú už s produkčnou verziou systému BeeOn, ktorá prešla krátkodobými a automatizovanými testami. Tiež je produkčná verzia väčšinou o 1 až 2 mesiace staršia, ako vývojová verzia. Medzi testery zamerané na integráciu termoregulátora VPT patrí hlavne spoločnosť Thermona, ktorá využíva na testovania až dva adaptéry. Okrem spoločnosti Thermona integráciu termoregulátora VPT v produkčnej verzii testujeme aj na škole s využitím dvoch termoregulátorov VPT, kde k jednému je pripojený simulátor kotla.

Kapitola 7

Plánované rozšírenie projektu

Integrácia termoregulátora VPT priniesla nové požiadavky na systém BeeeOn, ktoré nebolo možné v tejto práci implementovať, nakoľko tieto zmeny zasahujú do celého systému BeeeOn.

Plánované rozšírenia a požiadavky:

1. Rozšírenie komunikačného protokolu

- Aktualný komunikačný protokol vo verzií 1.0 medzi Ada serverom a AdaApp nedokáže pri párovaní termoregulátora VPT zaslať z koncovej ovládacej stanice prístupové heslo pre nastavenie riadiacích prvkov. Z tohto dôvodu bolo implementované aj hlavné heslo v konfiguračných súboroch. V pripravovanej verzií protokolu 1.1 je pridaná podpora zasielania hesla.
- Jedným z nedostatkov aktuálnej implementácie je chýbajúca podpora pre zapamätanie si napárovaných termoregulátorov VPT. Tento problém spôsobuje, že VPT modul si musí napárovať každý termoregulátor VPT, ktorý nájde na sieti, aj keď sa jeho hodnoty užívateľovi nebudú zobrazovať. Tento problém rieši tiež verzia protokolu 1.1, ktorá bude zasielať zoznam napárovaných zariadení uložených v databáze.
- Termoregulátor VPT môže nastaviť jednotlivým vykurovacím zónam názvy. Tieto názvy, ale nie je možné preniesť do systému BeeeOn, nakoľko protokol umožňuje zasielať len číselné hodnoty.

2. Rozšírenie types.xml a chybové stavy

- Pri riešení chybových stavov kotla, ktoré sú reprezentované jedným číslom a hodnota konkrétneho bitu určuje nejaký typ poruchy, bol systém BeeeOn rozšírený o dátový typ vyjadrujúci bitové pole o dĺžke 32 bitov.
- Aktuálna implementácia na koncových staniciach ani Tabuľka zariadení neumožňuje popísať sémantický význam jednotlivých bitov, preto chybové stavy nie sú korektne implementované a následne zobrazované.

3. Podpora pre časové programy

- Veľmi dôležitú časť funkcionality termoregulátora VPT tvoria časové programy. Tieto časové programy nie sú momentálne podporované zo strany ovládacích

koncových staníc, nakoľko si vyžadajú špeciálne rozmiestnenie ovládacích prvkov pre efektívnu prácu a užívateľskú prívetivosť. Nakoľko je termoregulátor VPT prvé zariadenie s časovými programmi, neexistuje v systéme BeeeOn pre nich podpora.

- Pri prvých diskusiách integrácie časových programov, bol podaný návrh pre oddelenie GUI od Tabuľky zariadení tak, aby každému zariadeniu bolo možné definovať samostatné GUI podľa grafického manuálu systému BeeeOn.

4. **Spôľahlivé prepínanie riadiacich prvkov** – Termoregulátor VPT má medzi prepnutím stavu riadiaceho prvku určitú časovú odozvu. Tento stav prináša do systému BeeeOn nežiadúci efekt, keď sa po prepnutí stavu riadiaceho prvku na koncovej ovládacej stanici znovu obnoví pôvodný stav riadiaceho prvku, ktorý bol načítaný v ďalšej aktualizácii senzorov.

Preto sú niekedy potrebné až dve štandardné načítania hodnôt senzorov z termoregulátora VPT pre správne zobrazenie stavu. Tento efekt môže vytvoriť až 30 sekundový časový interval, kedy užívateľ nemá istotu, že došlo k zmene stavu riadiaceho prvku na požadovanú hodnotu.

Tento problém, by mal riešiť nový stav **zanepřázdněný** spoločne so zásuvným modulom, ktorý bude umiestnený medzi Agregátorom a Sensorovými modulmi. Jeho hlavnou úlohou bude kontrolovať hodnoty zo senzorov a v prípade potreby meniť ich stav na zaneprázdněný po určitý časový interval - podľa potrieb zariadenia, kedy bude možné bezpečne aktuálne získaný stav senzoru prehlásiť za správny.

Kapitola 8

Záver

V tejto práci bol analyzovaný termoregulátor VPT v súvislosti s jeho integráciou ako rozširujúceho prvku do systému BeeeOn. Na základe tejto analýzy bola do systému BeeeOn implementovaná podpora termoregulátora VPT tak, aby mimo iného spĺňala aj požiadavky firmy Thermona, ktorá tieto termoregulátory dodáva.

V systéme BeeeOn nedošlo k narušeniu architektúry v žiadnom z komponentov systému BeeeOn a aktuálna implementácia termoregulátora VPT je zariadenie s najväčším počtom senzorov a riadiacich prvkov. Aplikácia AdaApp bola rozšírená o podporu protokolu HTTP a JSON. Tieto rozšírenia boli implementované pomocou dvoch nových modulov (**HTTP klient** a **JSON zariadenia**). Nakoľko sú tieto moduly navrhnuté univerzálne je ich možné použiť aj v iných senzorových moduloch.

Integrácia termoregulátora VPT do systému BeeeOn zo strany termoregulátora bola problematickejšia, nakoľko pôvodne plánované získavanie hodnôt z webových stránok termoregulátora VPT nebolo možné jednoduchým spôsobom implementovať, kvôli nedodržaniu žiadneho štandardu HTML a dvom rôznym spôsobom prístupu k hodnotám. Tieto problémy, ale vyriešil vlastný dátový súbor vo formáte JSON, ktorý je centralizovaným miestom pre zobrazovanie hodnôt senzorov s jednotným spôsobom prístupu z termoregulátora VPT.

Práca na projekte bude pokračovať ďalej aj po ukončení bakalárskej práce, kde plánujem v spolupráci s vývojovým tímom systému BeeeOn riešiť všetky plánované rozšírenia projektu, ktoré sú uvedené v kapitole 7 na strane 42. Okrem plánovaných rozšírení bude potrebné aj doplniť automatizované testy pre systém Jenkins, aby pokrývali plnú funkcionálnosť integrácie termoregulátora VPT do systému BeeeOn.

Literatura

- [1] Association, I. S.: *IEEE Public Organizational Unique Identifier List [online]*. IEEE Standards Association, [Online; navštívené 9. 2. 2016].
URL <http://standards-oui.ieee.org/oui/oui.txt>
- [2] BeeeOn: *BeeeOn Adaptér [online]*. BeeeOn, [Online; navštívené 8. 10. 2015].
URL <https://ant-2.fit.vutbr.cz/projects/adapter/wiki>
- [3] BeeeOn: *BeeeOn – Architektúra [online]*. BeeeOn, [Online; navštívené 6. 10. 2015].
URL <https://ant-2.fit.vutbr.cz/projects/iot/wiki/Architektura>
- [4] BeeeOn: *BeeeOn – Repozitáre [online]*. BeeeOn, [Online; navštívené 15. 3. 2016].
URL <https://github.com/BeeeOn>
- [5] BeeeOn: *BeeeOn Server – Architektúra [online]*. BeeeOn, [Online; navštívené 27. 2. 2016].
URL <https://ant-2.fit.vutbr.cz/projects/server/wiki>
- [6] GmbH, A. I. S. E.: *POCO C++ Libraries – Features [online]*. Applied Informatics Software Engineering GmbH, [Online; navštívené 8. 3. 2016].
URL <http://pocoproject.org/features.html>
- [7] GmbH, A. I. S. E.: *POCO C++ Libraries – The Hashing Framework [online]*. Applied Informatics Software Engineering GmbH, 2006–2010, [Online; navštívené 8. 3. 2016].
URL <http://pocoproject.org/slides/145-Hashing.pdf>
- [8] GmbH, A. I. S. E.: *POCO C++ Libraries [online]*. Applied Informatics Software Engineering GmbH, 2006–2016 [cit. 2016-3-8], [Online; navštívené 8. 3. 2016].
URL <http://pocoproject.org/>
- [9] GmbH, A. I. S. E.: *POCO C++ Libraries [online]*. Applied Informatics Software Engineering GmbH and Contributors, 2016 [cit. 2016-3-8], [Online; navštívené 8. 3. 2016].
URL <http://pocoproject.org/docs/Poco.JSON.html>
- [10] GmbH, A. I. S. E.: *POCO C++ Libraries – Reference Library [online]*. Applied Informatics Software Engineering GmbH and Contributors, 2016 [cit. 2016-3-8], [Online; navštívené 8. 3. 2016].
URL <http://pocoproject.org/docs/>
- [11] International, E.: *Standard ECMA-404 – The JSON Data Interchange Format [online]*. Ecma International, 2013, [Online; navštívené 14. 12. 2015].
URL <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>

- [12] Kolektiv autorov: *OpenEmbedded [online]*. OpenEmbedded, 1997–2016 [cit. 2016-3-2], [Online; navštívené 2. 3. 2016].
URL http://www.openembedded.org/wiki/Main_Page
- [13] Kolektiv autorov: *Extensible Markup Language (XML) [online]*. W3C, 2013-2015, [Online; navštívené 8. 12. 2015].
URL <https://www.w3.org/XML/>
- [14] Ltd., C. O.: *iChip Config Utility User Manual [online]*. Connect One Ltd., [Online; navštívené 12. 11. 2015].
URL http://www.connectone.com/wp-content/uploads/2012/06/iChip_Config_Utility_Manual.pdf
- [15] Olimex: *A10-OLinuCino-LIME-4GB [online]*. Olimex, 1997–2016 [cit. 2016-3-2], [Online; navštívené 2. 3. 2016].
URL <https://www.olimex.com/Products/OLinuCino/A10/A10-OLinuCino-LIME-4GB/>
- [16] One, C.: *iChip CO2128/CO2144 [online]*. Connect One, 2012, [Online; navštívené 11. 11. 2015].
URL http://www.connectone.com/?page_id=178
- [17] Thermona: *Regulátor vytápění VPT – všeobecný popis*. Thermona, příloha A: /doc/termoregulator-vpt/Popis-regulatoru-VPT.pdf.
- [18] Thermona: *Zónový regulátor vytápění VPT – návod k instalaci a použití komunikačního modulu VPTLAN*. Thermona, příloha A: /doc/termoregulator-vpt/Navod-VPTLAN.pdf.
- [19] Thermona: *Zónový regulátor vytápění VPT – návod k instalaci a použití komunikačního modulu VPTWiFi*. Thermona, příloha A: /doc/termoregulator-vpt/Navod-VPTWiFi.pdf.
- [20] Thermona: *Zónový regulátor vytápění VPT – návod k použití*. Thermona, příloha A: /doc/termoregulator-vpt/Navod-k-pouziti-regulatoru-VPT_A5.pdf.
- [21] Thermona: *Zónový regulátor vytápění VPT – příručka pro rychlý začátek*. Thermona, příloha A: /doc/termoregulator-vpt/Rychly-zacatek-VPT_A5.pdf.
- [22] ČECHMÁNEK, B. M.: *Brána inteligentní domácnosti [online]*. MARTIN ČECHMÁNEK, [Online; navštívené 02. 01. 2016].
URL <http://www.fit.vutbr.cz/study/DP/DP.php.cs?id=17240&file=t>

Příloha A

Obsah CD

Pribalené CD obsahuje všetky zdrojové kódy v ktorých prebiehala implementácia ako aj niektorú použitú literatúru pri návrhu integrácie.

- **src/** – Zdrojové kódy súčasti systému BeeeOn v ktorých prebiehala implementácia ako aj vlastnú webovú stránku vo formáte JSON pre termoregulátor VPT.
 - **adaapp-new** – Zdrojové kódy aplikácie AdaApp.
 - **common/specification/** – Obsahuje Tabuľku zariadení a definíciu typov.
 - **common/specification/languages** – Jazyková podpora pre koncové ovládacie stanice.
 - **thermona/devices_files** – Webové stránky termoregulátora VPT.
 - **thermona/adaapp-devices_specification** – Špecifikácia termoregulátora VPT pre aplikáciu Adaapp.
 - **testing/beeeon_tests/adaapp_configs** – Automatizované testy aplikácie Adaapp.
- **doc/** – Literatúra použitá pri integrácii termoregulátora VPT.
- **pdf/** – PDF verzia bakalárskej práce.
- **tex/** – Zdrojové súbory pro LATEX k vytvoreniu PDF verzie bakalárskej práce.
- **tex/fig** – Použité obrázky v bakalárskej práci.
- **img/** – Pracovné verzie obrázkov vytvorených pre bakalársku prácu.