



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

INTELIGENTNÍ DŮM SE SENZORY TŘETÍCH STRAN

SMART HOME WITH THE THIRD PARTY SENSORS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

PETER TISOVČÍK

Ing. JAN VIKTORIN

BRNO 2017

Abstrakt

Cílem této bakalářské práce bylo seznámit se s vybranými bezdrátovými senzory využívajícími Z-Wave a OASiS protokol a vytvořit jednotné rozhraní pro BeeeOn systém, které umožní komunikaci s vybranými prvky. V teoretické části práce jsou popsány senzorové prvky a USB zařízení, která slouží ke komunikaci s těmito prvky i operačním systémem, ke kterému jsou zařízení připojena. V rámci praktické části byla rozšířena aplikace BeeeOn Gateway, která zprostředkovává komunikaci s vybranými prvky. K tomuto účelu slouží komponenta pro správu zařízení, která může být snadno rozšířena o podporu dalších komunikačních protokolů.

Abstract

The aim of this thesis was to acquaint with selected wireless sensors, which use Z-Wave and OASiS protocol and to create the uniform interface for BeeeOn system providing communication with the selected sensors. The theoretical part of the thesis describes sensors and USB devices that serve for communication with sensors or operating system to which they are connected. In the practical part, the BeeeOn application was extended by device manager which can support the other communication protocols.

Klíčové slová

Inteligentní domácnost, BeeeOn, ZeroMQ, Jablotron, Turrís Gadgets, Z-Wave, AdaApp, BeeeOn Gateway

Keywords

Smart Home, BeeeOn, ZeroMQ, Jablotron, Turrís Gadgets, Z-Wave, AdaApp, BeeeOn Gateway

Citácia

TISOVČÍK, Peter. *Inteligentní dům se senzory třetích stran*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Viktorin Jan.

Inteligentní dům se senzory třetích stran

Prehlásenie

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Viktorina. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Peter Tisovčík

18. mája 2017

Podakovanie

Rád bych poděkoval vedoucímu bakalářské práce Ing. Janu Viktorinovi za poskytnuté odborné konzultace, ať už při psaní textové části práce, návrhu nebo implementaci. Poděkování patří i Ing. Josefu Hájkovi, Ph.D. za pomoc při řešení problémů s připojováním sensorových prvků.

Obsah

1	Úvod	4
2	Projekt BeeeOn	6
2.1	Architektúra systému BeeeOn	6
2.2	BeeeOn adaptér	7
2.2.1	Popis implementácie modulov v AdaApp	9
2.3	Tabuľka zariadení	9
2.4	Pridanie nového zariadenia do systému BeeeOn	10
3	Bezdrôtové senzorové prvky	11
3.1	Turris Gadgets	11
3.1.1	USB adaptér pre pripojenie periférii	12
3.1.2	AC-88 diaľkovo ovládaná zásuvka	13
3.1.3	JA-81M bezdrôtový detektor otvorenia	14
3.1.4	JA-82SH detektor otrasu alebo náklonu	15
3.1.5	JA-83M magnetický detektor	15
3.1.6	JA-83P – PIR detektor	16
3.1.7	JA-85ST detektor dymu a teploty	17
3.1.8	RC-86K diaľkový ovládač	17
3.1.9	TP-82N Termostat	18
3.2	Z-Wave a Z-Wave Plus zariadenia	18
3.2.1	USB Dongle Aeotec ZW090 Z-Stick	20
3.2.2	Aeotec Multisensor 6 – ZW100C	20
3.2.3	D-Link DCH-Z120 – Senzor pohybu	20
3.2.4	Fibaro FGK-107 – Magnetický kontakt s binárnym vstupom a meraním teploty	20
3.2.5	Philio PST02-1C – Multisenzor	20
3.2.6	Popp 123627 – Bezdrôtovo ovládaná zásuvka	21
4	Komunikácia BeeeOn Gateway so senzormi	22
4.1	Základné ZeroMQ prenosové vzory	22
4.1.1	Dotaz-odpoveď (request-response)	23
4.1.2	Vydavateľ-odberateľ (publisher-subscriber)	24
4.1.3	Zreťazené spracovanie (pipeline)	25
4.1.4	ZeroMQ typy prenosov	25

5	Návrh rozšírenia Gateway aplikácie	27
5.1	Návrh pripájania senzorov tretích strán	27
5.2	Identifikácia požiadaviek pre komunikáciu	30
5.3	Pripojenie Jablotron zariadení do prostredia OS Linux	32
5.4	Pripojenie Z-Wave zariadení do prostredia OS Linux	32
5.5	Návrh príkazov	32
5.5.1	Registrowanie manažéra zariadení do prostredníka	32
5.5.2	Zoznam napárovaných zariadení	33
5.5.3	Posledná nameraná hodnota na zariadení	33
5.5.4	Napárovanie nového zariadenia	33
5.5.5	Odpárovanie pridaného zariadenia	34
5.5.6	Nastavenie hodnoty na zariadení	34
6	Testovanie	36
6.1	Vlastnosti komunikačných protokolov	36
6.2	Jednotkové testy	37
6.3	Funkčné testy manažérov zariadení	37
6.3.1	Test registrowania manažéra zariadení	37
6.3.2	Test získania zoznamu napárovaných zariadení	38
6.3.3	Test poslednej nameranej hodnoty na zariadení	38
6.3.4	Test napárovania nového zariadenia	39
6.3.5	Test odpárovania nového zariadenia	39
6.3.6	Test nastavenia hodnoty na zariadení	39
7	Záver	41
	Literatúra	42
	Prílohy	45
A	Fotografie zariadení	46

Zoznam obrázkov

2.1	Architektúra systému BeeeOn	6
2.2	Architektúra AdaApp aplikácie	8
2.3	Štruktúra zariadenia v tabuľke zariadení	9
4.1	Dotaz-odpoveď (request-response) – ZeroMQ prenosový vzor	23
4.2	Vydavateľ-odberateľ (publisher-subscriber) – ZeroMQ prenosový vzor	24
4.3	Zreťazené spracovanie (pipeline) – ZeroMQ prenosový vzor	25
4.4	Typy koncových bodov	25
5.1	Návrh rozšírenia BeeeOn Gateway aplikácie	28
5.2	Odosielanie dát z manažérov zariadení do prostredníka (komunikácia N-to-1)	30
5.3	Odoslanie správy vybraným manažérom zariadení (komunikácia 1-to-N)	31
5.4	Nastavenie hodnoty na senzore z BeeeOn brány (komunikácia 1-to-1)	31
5.5	Stavy po prijatí správy na nastavenie hodnoty v Gateway	34
5.6	Stavy po prijatí správy na nastavenie hodnoty v manažéroch zariadení a Gateway	35
A.1	BeeeOn adaptér s pripojenou rozširujúcou doskou	46
A.2	USB zariadenia pre pripojenie bezdrôtových senzorov – Aeotec Z-Stick USB Dongle, Jablotron Dongle	46
A.3	Jablotron zariadenia – AC-88, JA-81M, JA-83M, JA82-SH, JA-83P, TP-82N	47
A.4	Z-Wave zariadenia – Aeotec Multisenzor 6, Dlink DCH-Z120, Philio PST02-1C, Popp 123627, Fibaro FGK-107	47

Kapitola 1

Úvod

S rozvojom moderných technológií si ľudia snažia zväčšovať svoje pohodlie. Vďaka tejto ľudskej vlastnosti sa vyvíjajú stále nové a nové technológie, ktoré pomáhajú zjednodušovať zložité, ale aj jednoduché činnosti. Jedná sa o úkony, ktoré boli pred pár rokmi plne v ľudských rukách, ale nástupom moderných technológií tieto činnosti vykonávajú stroje. Príkladom takýchto technológií je napríklad práčka, umývačka riadu, či robotický vysávač a pod.

V poslednom období zachádza tento trend ešte o niečo ďalej v podobe inteligentných domácností, ktorých hlavným cieľom je znížiť náklady na chod domácnosti a zároveň dosiahnuť, čo najlepšiu bezpečnosť a komfort pri používaní. Majiteľ inteligentného domu získava alternatívu k ručnému ovládaniu spotrebičov v podobe vzdialeného ovládania a sledovania spotrebičov, alebo senzorov, a to odkiaľkoľvek s prístupom k internetu. Príkladom využitia takéhoto vzdialeného ovládania je automatické otvorenie garáže, zapnutie ohrievania vody v bazéne, alebo automatické vetranie podľa teploty a vlhkosti v byte a vonku.

Vízia inteligentnej domácnosti sa ustálila aj na Fakulte informačných technológií v Brne, kde sa vyvíja projekt BeeeOn. Tento projekt sa zameriava na zjednotenie ovládania senzorov od rôznych výrobcov.

Jedným z kritérií pri výbere systému pre riadenie domácnosti je množstvo zariadení, ktoré systém ponúka. Táto práca sa zameriava na rozšírenie podpory pripájania senzorov tretích strán. Okrem samotnej podpory nových zariadení vznikla jednoduchá možnosť pridávania podpory ďalších zariadení. Existujúci BeeeOn systém bol rozšírený o možnosť pripájať zariadenia od firmy Jablotron a zariadenia podporujúce Z-Wave technológiu.

Text práce je rozdelený do niekoľkých častí. V prvej časti práce sú informácie o fungovaní BeeeOn systému, jeho architektúre a spôsobe pridávania nových zariadení do tohto systému.

Ďalšia časť práce popisuje vybrané senzorové prvky od firmy Jablotron a zariadenia podporujúce Z-Wave technológie pre pripojenie do systému BeeeOn. Ďalej sa v tejto časti nachádza popis využívaných bezdrôtových protokolov, ktorými komunikujú vybrané senzory a popis zariadení, ktoré umožňujú komunikovať s týmito bezdrôtovými protokolmi.

Štvrtá kapitola je zameraná na popis použitého transportného protokolu (ZeroMQ) pre vytvorenie jednotného rozhrania, ktoré integruje komunikáciu s jednotlivými senzormi.

Návrh pripojenia senzorov tretích strán a jednotného komunikačného protokolu je popísaný v piatej časti. V tejto časti sa tiež nachádza analýza požiadaviek na implementáciu transportného protokolu pre pripájanie zariadení tretích strán.

Na túto časť nadväzuje posledná kapitola s testovaním, ktorá overuje funkčnosť implementovaného riešenia. Okrem testovania sa tu nachádzajú aj vlastnosti komunikačných protokolov získané pri testovaní.

Testovanie sa skladá z integračných testov, ktoré overujú funkčnosť komunikácie medzi BeeOn Gateway a senzormi. Do testov sú zahrnuté aj jednotkové testy, ktoré testujú funkčnosť vybraných komponentov.

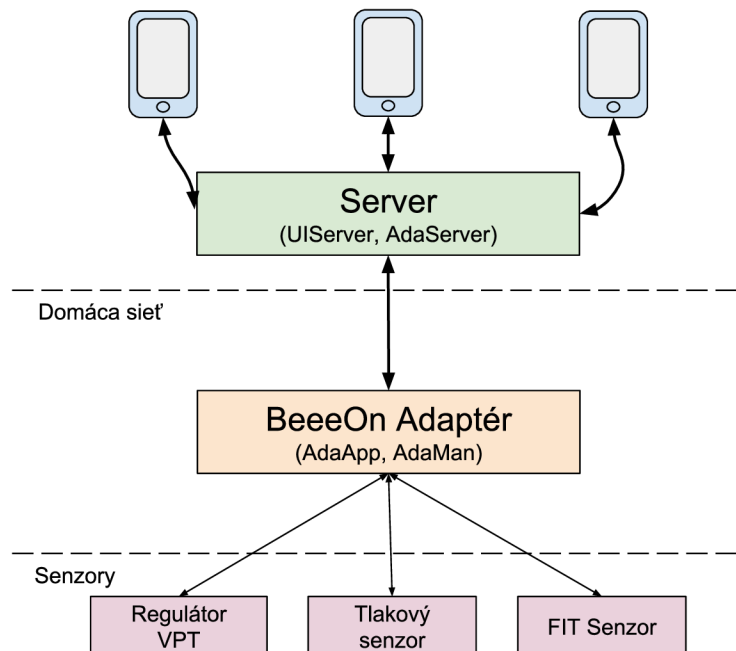
Kapitola 2

Projekt BeeeOn

BeeeOn je open source projekt vyvíjaný na Fakulte informačných technológií Vysokého učenia technického v Brne v oblasti Internetu vecí (Internet of Things). Cieľom projektu je vytvoriť ľahko ovládateľný a rozšíriteľný systém pre pripájanie zariadení tretích strán. Dôraz sa kladie na zjednotenie ovládania prvkov od rôznych výrobcov naprieč rôznymi senzorovými protokolmi.

2.1 Architektúra systému BeeeOn

Systém sa rozdeľuje do niekoľkých častí, ako je znázornené na obrázku 2.1. Na najspodnejšej časti sa nachádzajú senzorové zariadenia tretích strán, komunikujúce rôznymi protokolmi. Tieto zariadenia umožňujú zber hodnôt teploty, vlhkosti, svetelnosti, pohybu, otrasu a ďalších veličín.



Obr. 2.1: Architektúra systému BeeeOn

Príkladom zariadení pre zber hodnôt, ktoré sú implementované v BeeeOn adaptéri je regulátor VPT, ktorý slúži pre riadenie vykurovacej sústavy, FIT senzory pre meranie teploty a vlhkosti, či tlakový senzor.

Okrem zberu informácií dokážu niektoré senzory meniť svoj stav. Tieto zariadenia sa označujú ako aktory. Medzi aktory patrí napr. zásuvka, ktorú je možné zapnúť i vypnúť, alebo žiarovka, ktorej je možné nastaviť intenzitu osvetlenia. Niektoré senzorové prvky podporujú nastaviť vlastnosť refresh time, ktorý umožňuje zvoliť dobu, po ktorej moduly odosielať namerané hodnoty, prípadne iné veličiny. Tieto zariadenia najčastejšie komunikujú s bránou prostredníctvom bezdrôtovej siete. Brána môže podporovať radu bezdrôtových sietí vďaka externým USB adaptérom.

BeeeOn adaptér je zariadenie, ktoré sa stará o príjem dát zo senzorov a v požadovanom formáte preposiela prijaté dáta na server. Na adaptéri sa nachádzajú aplikácie, ktoré komunikujú s jednotlivými bezdrôtovými protokolmi pomocou pripojených USB zariadení.

Úlohou serveru je registrovať adaptére, senzory, užívateľov, prípadne ukladať prijaté dáta zo senzorov. Server pre komunikáciu s bránami a adaptérami sa nazýva AdaServer. Umožňuje kontrolu prijatých dát, ich prevod, ak je to vyžadované a následne ich uloženie do databáze. Okrem neho sa na servery nachádza UIServer, ktorý poskytuje protokol pre komunikáciu s Android aplikáciou. AdaServer a UIServer medzi sebou komunikujú za účelom sprostredkovania príkazov od užívateľa. Pomocou Android aplikácie môžu byť všetky tieto dáta prezentované užívateľovi a to formou grafov a zobrazením kľúčových informácií na jednej obrazovke.

2.2 BeeeOn adaptér

BeeeOn adaptér (viď príloha A.1) je hardwarová komponenta, ktorá je postavená na open source hardwaru. Základ brány tvorí vývojová doska A10-OLinuXino-LIME od spoločnosti Olimex. Na doske je osadený 1 GHz Cortex-A8 procesor, 512 MB pamäť, 100 Mbit Ethernet konektor a 160 GPIO pinov. Doska ponúka niekoľko rozhraní pre komunikáciu s ďalšími perifériami: SATA, HDMI a USB konektor. K tejto doske je pomocou GPIO pinov pripojená rozširujúca doska obsahujúca rádio MRF89XA od spoločnosti Microchip pre komunikáciu s bezdrôtovým FIT protokolom, obvod RTC s batériou a tlakový senzor [2]. Na vývojovej doske pre BeeeOn bránu od spoločnosti Olimex je použitá linuxová distribúcia zostavená pomocou prekladového systému OpenEmbedded¹.

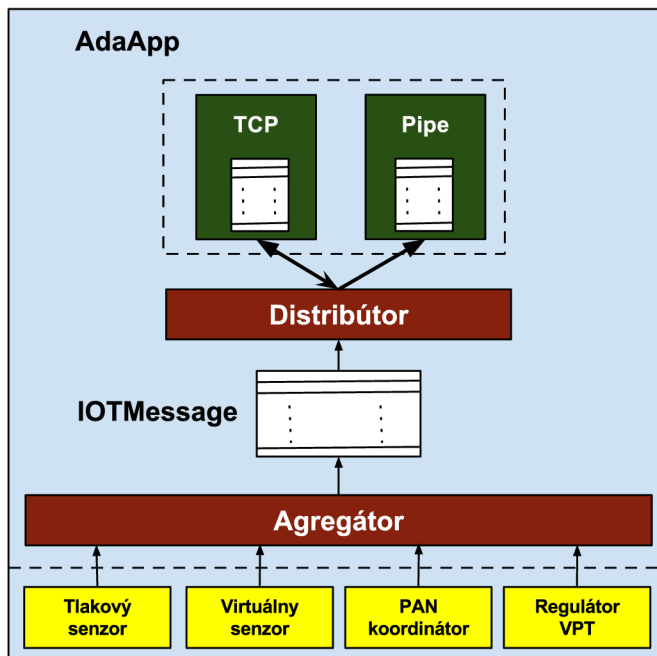
Adaptér slúži na pripájanie senzorov a aktorov tretích strán. Na jednej strane obsahuje hardwarové porty na pripojenie USB zariadení a na druhej strane obsahuje programové rozhranie pre komunikáciu so serverom. Každá brána je identifikovaná šesťnásť miestnym decimálnym identifikátorom, ktorý je jedinečný v BeeeOn systéme. Identifikačné číslo je pridelené pri výrobe adaptéra a jeho formát je nasledovný: XYYY YYYY YYYY YYYZ. Význam jednotlivých znakov v identifikátore [25]:

- X – verzia identifikátora
- Y – náhodne vygenerované číslice identifikátora v desiatkovej sústave
- Z – kontrolný súčet

Senzorové zariadenie je identifikované 64-bitovým identifikátorom. Prvých 8 bitov slúži na identifikovanie, akým typom protokolu zariadenie komunikuje (ďalej označovaný ako prefix senzoru).

¹<http://www.openembedded.org/>

Na adaptéri sú spustené dve základné aplikácie AdaApp a AdaMan. AdaMan slúži na distribúciu aktualizácií na adaptére. Je to aplikácia typu klient-server. Na adaptéroch je spustený klient, ktorý sa dotazuje servera na dostupnosť nových aktualizácií. Dostupné aktualizácie sa stiahnu a vykoná sa ich inštalácia.



Obr. 2.2: Architektúra AdaApp aplikácie

AdaApp aplikácia (ďalej označovaná ako AdaApp) je jadrom adaptéru a slúži na komunikáciu so serverom. Architektúra aplikácie je znázornená na obrázku 2.2. Po štarte sa spustia jednotlivé vlákna s povolenými modulmi (tlakový senzor, virtuálny senzor, PAN koordinátor a regulátor VPT), ktoré komunikujú s pripojenými senzormi. PAN koordinátor slúži na pripájanie FIT senzorov.

Modul je časť AdaApp, ktorá sa stará o komunikáciu so senzormi. Jednotlivé moduly prevádzajú dáta do jednotnej štruktúry *IOTMessage*. Následne sa táto správa predá do Agregátora, ktorý túto správu prepošle do Distribútora a exportuje do požadovaných výstupných modulov. Medzi výstupné moduly patrí výstup na linuxovú rúru v CSV jazyku a klient pre komunikáciu so serverom, ktorý komunikuje pomocou správ v XML jazyku.

AdaApp umožňuje získať posledný stav aktora a tým ho nastaviť. V prípade výpadku spojenia so serverom, správy nie sú zahadzované, ale ukladajú sa do vyrovnávajúcej pamäte a po obnovení spojenia sa dáta postupne odošlú. Aplikácie zabezpečujúce komunikáciu so senzormi môžu byť súčasťou AdaApp ako moduly, alebo môžu byť spúšťané osobitne. Osobitne spúšťané aplikácie komunikujú s AdaApp pomocou transportného protokolu *MQTT*² a vlastného aplikačného protokolu. Aplikačný protokol je vo forme jednoduchých textových správ. Komunikácia medzi adaptérom a serverom prebieha XML správami. Pre tento prenos dát je použitý zabezpečený kanál.

²<http://mqtt.org/>

2.2.1 Popis implementácie modulov v AdaApp

V súčasnej implementácii AdaApp sú moduly implementované niekoľkými spôsobmi. Jednotlivé implementácie vychádzali z jednotlivých potrieb modulov a vývoja AdaApp.

Zo začiatku sa jednotlivé moduly implementovali priamo do AdaApp bez pevne definovaného rozhrania. Takto bol implementovaný napr. modul pre virtuálny senzor. Virtuálny senzor je senzor, ktorý posiela testovacie hodnoty pre ladenie aplikácie bez nutnosti mať pripojený reálny senzor. Pri implementácii modulov bez rozhraní sa duplikuje kód, ktorý je spoločný pre všetky moduly. Ďalej je potrebné byť oboznámený s architektúrou adaptéru pre správne použitie jednotlivých tried a metód.

Ďalšou vývojovou fázou bolo vytvorenie modulu, ktorý by komunikoval s AdaApp pomocou vlastného aplikačného protokolu a transportného protokolu *MQTT*. Modul využívajúci vlastný komunikačný protokol je napr. PAN koordinátor, ktorý je oddelený od AdaApp. Nevýhodou tejto implementácie je špecifický aplikačný protokol pre konkrétnu implementáciu modulov. PAN koordinátor využíva ako formát aplikačného protokolu čiarkou oddelené hodnoty znakov definované v ASCII tabuľke.

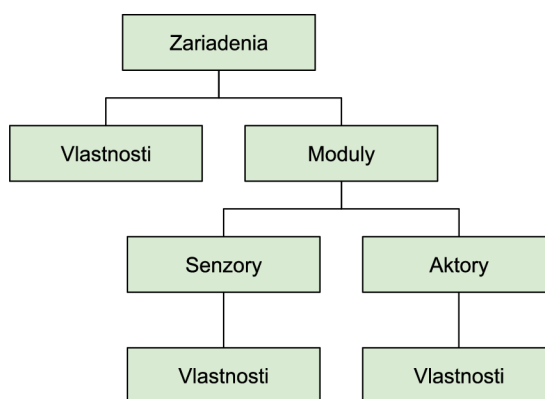
Poslednou možnosťou bolo vytvorenie abstraktnej triedy `ModuleADT`, ktorá definovala spoločné rozhranie pre nové moduly. Toto rozhranie bolo až doteraz využívané pri návrhu nových modulov.

V tejto práci sú tieto prístupy zhodnotené a zjednotené do spoločného dostatočne obecného princípu.

2.3 Tabuľka zariadení

Návrh tabuľky zariadení vznikol na základe požiadaviek pre centrálnu správu zariadení v BeeOn systéme. Táto tabuľka je používaná naprieč celým systémom a pomocou aplikácie *DeviceTransformer* sú vytvárané potrebné súbory pre začlenenie nových prvkov na strane Android aplikácie, databáze a AdaApp aplikácie.

Tabuľka zariadení definuje vlastnosti jednotlivých senzorových prvkov, ktoré sú podporované BeeOn systémom. Umožňuje definovať merané a nastavované hodnoty spolu s ich dátovými typmi. Okrem meraných veličín umožňuje vkladať popis zariadení a hodnôt v rôznych jazykoch, ktoré sa zobrazujú v Android aplikácii. Preklad do rôznych jazykov je riešený špeciálnymi názvami, ktoré sú mapované na jednotlivé lokalizácie, ako je znázornené v ukážke 2.1. Informácie o zariadeniach sú popísané v XML súbore.



Obr. 2.3: Štruktúra zariadenia v tabuľke zariadení

Každé zariadenie je v tabuľke zariadení identifikované jedinečným identifikátorom, ktorý slúži na určenie, o aký prvok sa jedná. Každý prvok je definovaný menom (*name*), výrobcom (*manufacturer*) a podporovanými modulmi (*modules*), ako je znázornené na obrázku 2.3. Modulom sa označuje každý senzor na senzore zariadení, napr. ak sa jedná o senzor teploty a vlhkosti (dva senzory), tak jednotlivé moduly sú:

- modul pre meranie teploty
- modul pre meranie vlhkosti

2.4 Pridanie nového zariadenia do systému BeeeOn

V prípade, že sa pridá podpora nového zariadenia do AdaApp bez aktualizácie tabuľky zariadení, vznikajú určité chyby v BeeeOn systéme. Pokiaľ sú cez adaptér odosielané dáta zo zariadení, ktoré adaptér nemá uvedené vo svojej internej statickej tabuľke, tak sú takéto dáta zahadzované. Ak by sa tabuľka aktualizovala len v rámci BeeeOn adaptéra, do databázy na serveri by sa nevložili žiadne údaje. V Android aplikácii by sa zase zariadenie zobrazilo ako neznáme a novo pridané moduly by zostali bez popisu.

Pre pridanie nového zariadenia do BeeeOn systému je nutné splniť určité kroky:

- Pridať podporu nového sensorového prvku do AdaApp – implementovať komunikáciu s USB adaptérom, ktorý umožňuje komunikovať s bezdrôtovým protokolom.
- Pridať popis nového zariadenia do tabuľky zariadení – definovať vlastnosti a moduly nového prvku.
- Preložiť Android aplikáciu s novou tabuľkou zariadení, aby sa užívateľom zobrazili informácie o senzore a popisy modulov, ktoré novo pridané zariadenie obsahuje.
- Upraviť trigger v databáze podľa novej tabuľky zariadení, aby sa pri napáraní nového zariadenia vytvorili v databáze potrebné moduly pridaného senzora.
- Preložiť AdaApp s novou tabuľkou zariadení, aby bolo možné identifikovať zariadenia a overiť, či senzor odosiela hodnoty z preddefinovaných modulov.

```
1 <device id="25" name="FIBARO Door/Window Sensor FGK-107">
2   <name>T:DEV_Z_WAVE_FGK_107</name>
3   <manufacturer>T:MANUFACTURER_FIBARO</manufacturer>
4   <modules>
5     <sensor id="0" type="0x01">
6       <name>T:FIBARO_FGK_107_MAGNETIC_DOOR_CONTACT</name>
7       <values name="T:FIBARO_FGK_107_MAGNETIC_DOOR_CONTACT">
8         <value id="0">T:CLOSED</value>
9         <value id="1">T:OPENED</value>
10        <value id="2">T:UNKNOWN</value>
11      </values>
12    </sensor>
13    <battery id="1" type="0x08" />
14  </modules>
15 </device>
```

Zdrojový kód 2.1: Ukážka popisu zariadenia v tabuľke zariadení s jazykovo nezávislými popismi

Kapitola 3

Bezdrôtové senzorové prvky

V súčasnosti je veľký záujem o senzorové prvky, o čom svedčí veľké množstvo výrobcov a druhov týchto zariadení na trhu. Pokles dopytu po senzorových zariadeniach si môžeme predstaviť len ťažko a môžeme skôr predpokladať zvyšujúci sa záujem o tieto zariadenia.

Senzorový prvok je zariadenie, ktoré umožňuje merať hodnoty alebo ovládať svoje výstupy. Väčšina senzorových prvkov obsahuje niekoľko senzorov, ktoré merajú rôzne veličiny. Tieto jednotlivé senzory na sensorovom prvku sú v BeeeOn systéme označované ako moduly.

Táto časť je zameraná na popis bezdrôtových senzorových prvkov, ktoré komunikujú pomocou bezdrôtových uzatvorených protokolov OASiS a Z-Wave. Bezdrôtový protokol OASiS využívajú zariadenia Turrís Gadgets. Z-Wave využívajú zariadenia od firiem Aeotec, D-Link, Fibaro, Philio a Popp.

3.1 Turrís Gadgets

Jedná sa o projekt spoločnosti Jablotron Alarms a neziskového výskumného projektu združenia CZ.NIC, ktorý sa snaží vytvoriť sadu zariadení Turrís Gadgets pre domácnosť. Túto sadu je možné pripojiť pomocou Turrís Donglu k smerovaču Turrís¹, prípadne k inému zariadeniu, ktoré má podporu tohto USB zariadenia. Sada obsahuje niekoľko typov zariadení, a to pre meranie teploty, detekovanie pohybu, dymu, otvorenia alebo otrasu. Pre komunikáciu je použitý bezdrôtový uzavretý protokol OASiS, ktorý je nástupcom Jablotron Profi [5]. Dosah zariadení je v rozmedzí 1–300 metrov v závislosti od zariadení a veľkosti prekážok medzi vysielačom a prijímačom.

Niektoré bezdrôtové senzorové prvky podporujú zasielanie správ typu BEACON. Tento typ správy je použitý pre monitorovanie funkčnosti zariadení, ktoré túto správu posielajú. Princíp monitorovania pomocou tejto správy spočíva v odosielaní BEACON správ každých 9 minút. Túto správu spracúva USB zariadenie, ktoré je pripojené k hostiteľskému systému. Na túto správu sa neodpovedá, má len informačný charakter. V prípade, že túto správu predbehne iný typ správy, tak sa jej odoslanie posunie o 9 minút [1].

Niektoré Jablotron prvky podporujú detekciu sabotáže. Sabotáž je možné rozdeliť na dva typy:

- otvorenie zariadenia
- odlepenie zariadenia

¹<https://www.turris.cz/cs/>

Prvý typ sabotáže je otvorenie zariadenia, ktoré sa umiestňuje pomocou pripravenej sady na uchytenie do steny. Druhý typ tvoria zariadenia, ktoré sa upevňujú obojstrannou páskou. Zariadenia, ktoré je možné umiestniť obojstrannou páskou, obsahujú vo vnútri tlačidlo, na ktorom je umiestnená pružina. Táto pružina presahuje výšku zariadenia a po upevnení sa pôsobením pružiny toto tlačidlo stlačí. Pri odstránení zariadenia sa táto pružina uvoľní a tlačidlo sa rozpojí, čo signalizuje sabotáž. Detektor pohybu má podobné tlačidlo a pružinu. Pružina však v tomto detektore neprečnieva zo zariadenia a slúži na detekovanie otvorenia senzoru. Stav sabotáže sa posielajú v správe typu TAMPER.

OASiS protokol neumožňuje posielanie informácie o konkrétnom stave batérie. Avšak zariadenia napájané batériou pri odosielaní dát odosielajú informáciu LB, či má batéria nízky stav alebo je stav batérie v poriadku [4].

Súhrnné informácie o jednotlivých zariadeniach sú uvedené v tabuľke 3.1.

	Podpora správy BEACON	Batériové napájanie	Podpora detekcie sabotáže	Životnosť batérie [rok]	Udávaný dosah [m]
AC-88	x	x	x	-	1
JA-81M	✓	✓	✓	3	300
JA-82SH	✓	✓	✓	2	1
JA-83	✓	✓	✓	3	300
JA-83M	✓	✓	✓	3	300
JA-85ST	✓	✓	✓	3	300
RC-86K	x	✓	x	4	30
TP-82N	x	✓	x	1	100

Tabuľka 3.1: Vlastnosti zariadení Jablotron

3.1.1 USB adaptér pre pripojenie periférii

Turris Dongle (Obr. A.2) pracuje na frekvencii 868,5 MHz. K hostiteľskému systému je pripojený pomocou FTDI-FT230X prevodníku. Na ovládanie je následne použitá sériová linka s komunikačnou rýchlosťou 57 600 baudov, ôsmimi dátovými bitmi a jedným stop bitom.

Každé zariadenie má svoj unikátny identifikátor v rozsahu 0x180000–0xCFFFFFF. Pre komunikáciu medzi bezdrôtovými prvkami a Turris Donglom je nutné, aby boli v zariadení uložené sériové čísla komunikujúcich zariadení. K jednému USB Donglu je možné pripojiť až 32 zariadení.

Správy protokolu OASiS sú textové a rozdeľujú sa na tri typy:

- príkaz (od hostiteľa k USB zariadeniu)
- odpoveď (od USB zariadenia k hostiteľovi)
- správa (od USB zariadenia k hostiteľovi)

Správy a odpovede protokolu OASiS používajú ako oddelovač symbol konca riadku (LF – 0x0A, ďalej označované ako '\n'). Ak sa jedná o príkaz pre USB zariadenie, správa začína symbolom escape (ESC – 0x1B, ďalej označené ako '\x1B') a končí znakom konca riadka. Medzi oddelovačmi správ sa nachádza text správy (ďalej označovaný ako MESSAGE). Prijatá správa a odpoveď je vo formáte \n MESSAGE \n, formát príkazu je \x1B MESSAGE \n.

V prípade, že modul prijme neplatnú správu, je odosielateľ informovaný odpoveďou `\n ERROR \n`. Pokiaľ je príkaz v požadovanom formáte, tak je hostiteľ informovaný odpoveďou `\n OK \n`. Všetky správy sú bez medzier za úvodným a pred ukončujúcim znakom [4].

Registrowanie zariadenia prebieha príkazom `\x1B SET SLOT:XX [YYYYYYYY] \n`.

- `XX` – číslo slotu v rozmedzí 00–31
- `YYYYYYYY` – sériové číslo zariadenia v dekadickej sústave
- `OK, ERROR` – odpoveď na zadaný príkaz

Odregistrovanie zariadenia prebieha príkazom `\x1B SET SLOT:XX [-----] \n`.

- `XX` – číslo slotu v rozmedzí 00–31
- `OK, ERROR` – odpoveď na zadaný príkaz

Vyčítanie zariadenia (slotu) prebieha príkazom `\x1B GET SLOT:XX \n`.

- `XX` – číslo slotu v rozmedzí 00–31
- `YYYYYYYY` – sériové číslo zariadenia v dekadickej sústave
- `\n SLOT:XX [YYYYYYYY] \n` – odpoveď na zadaný príkaz

3.1.2 AC-88 diaľkovo ovládaná zásuvka

Jedná sa o bezdrôtovú zásuvku (Obr. A.3) obsahujúcu relé, ktoré dokáže zapínať, či vypínať pripojené zariadenie. Príkladom použitia tejto zásuvky je bezdrôtové zapnutie/vypnutie pripojených elektrických zariadení (svietidlo, počítač, televízor). K jednej Jablotron sady je možné pripojiť maximálne 2 zásuvky. Zásuvka je napájaná zo siete.

Pri stlačení tlačidla na zásuvke sa odošle správa vo formáte [4]:

`\n [YYYYYYYY] AC-88 RELAY:? \n`

- `YYYYYYYY` – sériové číslo zariadenia v dekadickej sústave
- `AC-88` – typ zariadenia
- `RELAY:0` – rozpojené relé, tlačidlom na zásuvke alebo bezdrôtovo
- `RELAY:1` – zopnuté relé, tlačidlom na zásuvke alebo bezdrôtovo

Nastavenie zásuvky stavovým paketom prebieha príkazom [4]:

`\x1B TX ENROLL:? PGX:? PGY:? ALARM:? BEEP:(NONE|SLOW|FAST) \n`

- `ENROLL:0` – nevysiela sa registračný signál pre prijímač
- `ENROLL:1` – vysiela sa registračný signál pre prijímač
- `PGX:0` – výstup je rozpojený (zásuvka je vypnutá)
- `PGX:1` – výstup je spojený (zásuvka je zapnutá)
- `PGY:0` – výstup je rozpojený (zásuvka je vypnutá)

- PGY:1 – výstup je spojený (zásuvka je zapnutá)
- ALARM:0 – poplach nie je aktívny (siréna je vypnutá)
- ALARM:1 – poplach je aktívny (siréna je zapnutá)
- BEEEP:NONE – zvuk sirény nie je aktívny
- BEEEP:SLOW – pípanie sirény je pomalé
- BEEP:FAST – pípanie sirény je rýchle
- OK, ERROR – odpoveď na zadaný príkaz

Výstup PGX je mapovaný na zásuvku, ktorá je uložená na menšom čísle slotu (napr. SLOT:00). Na vyššom čísle slotu (napr. SLOT:01) je uložená zásuvka, na ktorú je mapovaný výstup PGY. Pri nastavovaní zariadení je doporučené vysielanie stavového paketu vykonať 3-krát za sebou. Časový odstup medzi odoslaním jednotlivých paketov je 200–500 milisekúnd. Pri nastavovaní by sa mala nastavovať len jedna hodnota. Ak je nutné nastaviť niekoľko hodnôt, je doporučené dodržať odstup jednu sekundu medzi stavovými paketmi [4].

3.1.3 JA-81M bezdrôtový detektor otvorenia

Magnetický detektor (Obr. A.3) je určený na detekovanie otvorenia okien alebo dverí. Tento senzor je možné doplniť senzorom, ktorý umožňuje zopnúť kontakt na výstupe. Vo vnútri zariadenia sa nachádza prepínač, ktorým je možné nastaviť zapnutie/vypnutie magnetického detektora. Druhým prepínačom vo vnútri zariadenia je možné nastaviť, či detektor leží v prístupovej ceste a poskytuje odchádzajúce a prichádzajúce oneskorenie. Senzor ďalej umožňuje nastaviť, či má reagovať len na oddialenie alebo aj na priblíženie magnetu [10]. Zariadenie podporuje detekciu sabotáže (otvorenie zariadenia), správu typu BEACON a je napájané batériou s približnou životnosťou 3 roky. Udávaný dosah je okolo 300 metrov pri priamej viditeľnosti.

Zariadenie môže odoslať nasledujúce typy správ [4]:

```
\n [YYYYYYYY] JA-81M SENSOR LB:? ACT:? \n
\n [YYYYYYYY] JA-81M TAMPER LB:? ACT:? \n
\n [YYYYYYYY] JA-81M BEACON LB:? \n
```

- YYYYYYYY – sériové číslo zariadenia v dekadickej sústave
- JA-81M – typ zariadenia
- SENSOR – udalosť na senzore (oddialenie alebo priblíženie magnetu)
- TAMPER – informácie o sabotáži (premiestnenie senzora)
- BEACON – ohlásenie zariadenia
- LB:0 – informácia o nízkom stave batérie
- LB:1 – informácia o dobrom stave batérie
- SENSOR ACT:0 – otvorenie (oddialenie magnetu od senzoru)
- SENSOR ACT:1 – zatvorenie (priblíženie magnetu k senzoru)

- TAMPER ACT:0 – sabotáž (odlepenie zariadenia)
- TAMPER ACT:1 – zabezpečenie zariadenia (upevnenie zariadenia)

3.1.4 JA-82SH detektor otrasu alebo náklonu

Detektor (Obr. A.3) pracuje v dvoch režimoch. Prvý režim slúži na detekciu otrasu, ktorý je možné použiť ako indikátor otvorenia dverí alebo okien hrubou silou. Druhý režim slúži na detekciu náklonu a je možné ho použiť pre indikáciu otvorenia okien. Medzi režimami je možné vyberať pomocou zabudovaného prepínača na zariadení. Okrem tohto prepínača sa tu nachádza potenciometer, ktorým sa určuje citlivosť snímaných hodnôt [11]. Zariadenie podporuje detekcie sabotáže, správu typu BEACON a je napájané batériou s približnou životnosťou 2 roky. Komunikačný dosah je udávaný okolo 300 metrov pri priamej viditeľnosti.

Zariadenie môže odoslať nasledujúce typy správ [4]:

```
\n [YYYYYYYY] JA-82SH SENSOR LB:? \n
\n [YYYYYYYY] JA-82SH TAMPER LB:? ACT:? \n
\n [YYYYYYYY] JA-82SH BEACON LB:? \n
```

- YYYYYYYY – sériové číslo zariadenia v dekadickej sústave
- JA-82SH – typ zariadenia
- SENSOR – udalosť na senzore (náklon, otras)
- TAMPER – informácie o sabotáži (premiestnenie senzora)
- BEACON – ohlásenie zariadenia
- LB:0 – informácia o nízkom stave batérie
- LB:1 – informácia o dobrom stave batérie
- ACT:0 – sabotáž (odlepenie zariadenia)
- ACT:1 – zabezpečenie zariadenia (upevnenie zariadenia)

3.1.5 JA-83M magnetický detektor

Detektor JA-83M (Obr. A.3) je určený na detekciu otvorenia okien alebo dverí. Obsahuje detektor a magnet. Vysielacia časť sa upevňuje na pevnú časť dverí, okien a magnet na pohyblivú časť. Po oddialení alebo priblížení magnetu sa odošle informácia o tomto stave. Vo vnútri zariadenia sa nachádza prepínač, ktorým je možné nastaviť, či detektor leží v prístupovej ceste a poskytuje odchádzajúce a prichádzajúce oneskorenie. Senzor ďalej umožňuje nastavenie, či má reagovať len na oddialenie alebo aj na priblíženie magnetu [12]. Zariadenie podporuje detekcie sabotáže (otvorenie zariadenia), správu typu BEACON a je napájané batériou s približnou životnosťou 3 roky. Udávaný dosah je okolo 300 metrov pri priamej viditeľnosti.

Zariadenie môže odoslať nasledujúce typy správ [4]:

```
\n [YYYYYYYY] JA-83M SENSOR LB:? ACT:? \n
\n [YYYYYYYY] JA-83M TAMPER LB:? ACT:? \n
\n [YYYYYYYY] JA-83M BEACON LB:? \n
```

- YYYYYYYY – sériové číslo zariadenia v dekadickej sústave
- JA-83M – typ zariadenia
- SENSOR – udalosť na senzore (oddialenie alebo priblíženie magnetu)
- TAMPER – informácie o sabotáži (premiestnenie senzora)
- BEACON – ohlásenie zariadenia
- LB:0 – informácia o nízkom stave batérie
- LB:1 – informácia o dobrom stave batérie
- SENSOR ACT:0 – otvorenie (oddialenie magnetu od senzoru)
- SENSOR ACT:1 – zatvorenie (priblíženie magnetu k senzoru)
- TAMPER ACT:0 – sabotáž (odlepenie zariadenia)
- TAMPER ACT:1 – zabezpečenie zariadenia (upevnenie zariadenia)

3.1.6 JA-83P – PIR detektor

Zariadenie (Obr. A.3) je určené na detekciu pohybu osôb v interiéri budov a obytných priestorov. Obsahuje dva prepínače, ktorými je možné nastaviť stupeň odolnosti falošných poplachov a zároveň, či má zariadenie poskytovať odchodové a príchodové oneskorenie. V dôsledku šetrenia batérie prejde senzor po 15 minútach od zatvorenia krytu do režimu spánku. V prípade zaznamenania pohybu, PIR detektor odošle správu a ďalších 5 minút nereaguje na pohyby (spánok senzora). Po uplynutí 5 minút sa zariadenie prebudí a nepretržite monitoruje priestor než je detekovaný ďalší pohyb. Čas spánku senzora je možné skrátiť na 1 minútu [13]. Zariadenie podporuje detekciu sabotáže (otvorenie zariadenia), správu typu BEACON a je napájané batériou s približnou životnosťou 3 roky. Udávaný dosah je okolo 300 metrov pri priamej viditeľnosti.

Zariadenie môže odoslať nasledujúce typy správ [4]:

```
\n [YYYYYYYY] JA-83P SENSOR LB:? \n
\n [YYYYYYYY] JA-83P TAMPER LB:? ACT:? \n
\n [YYYYYYYY] JA-83P BEACON LB:? \n
```

- YYYYYYYY – sériové číslo zariadenia v dekadickej sústave
- JA-83P – typ zariadenia
- SENSOR – udalosť na senzore (náklon, otras)
- TAMPER – informácie o sabotáži (premiestnenie senzora)
- BEACON – ohlásenie zariadenia
- LB:0 – informácia o nízkom stave batérie
- LB:1 – informácia o dobrom stave batérie
- ACT:0 – sabotáž (otvorenie krytu zariadenia)
- ACT:1 – zabezpečenie zariadenia (zatvorenie krytu zariadenia)

3.1.7 JA-85ST detektor dymu a teploty

Jedná sa o zariadenie (Obr. A.3) slúžiace k detekovaniu požiarneho nebezpečia. Zariadenie obsahuje dva samostatné detektory. Optický detektor slúži na detekovanie väčších častíc, ktoré sa nachádzajú v hustom dyme. Druhým, menej citlivým detektorom je detektor teploty, ktorý slúži na detekovanie tepla. Tento detektor sa používa pri horení kvapalín, ktoré produkujú malé množstvo dymu s väčším teplom. Zariadenie podporuje detekciu sabotáže (otvorenie zariadenia), správu typu BEACON a je napájané batériou s približnou životnosťou 3 roky. Udávaný dosah je okolo 300 metrov pri priamej viditeľnosti [3].

Zariadenie môže odoslať nasledujúce typy správ [4]:

```
\n [YYYYYYYY] JA-85ST SENSOR LB:? \n
\n [YYYYYYYY] JA-85ST BUTTON LB:? \n
\n [YYYYYYYY] JA-85ST TAMPER LB:? ACT:? \n
\n [YYYYYYYY] JA-85ST DEFECT LB:? ACT:? \n
\n [YYYYYYYY] JA-85ST BEACON LB:? \n
```

- YYYYYYYY – sériové číslo zariadenia v dekadickej sústave
- JA-85ST – typ zariadenia
- SENSOR – udalosť na senzore (požiar)
- TAMPER – informácie o sabotáži (premiestnenie senzora)
- BEACON – ohlásenie zariadenia
- DEFECT – porucha zariadenia
- LB:0 – informácia o nízkom stave batérie
- LB:1 – informácia o dobrom stave batérie
- TAMPER ACT:0 – sabotáž (odlepenie zariadenia)
- TAMPER ACT:1 – zabezpečenie zariadenia (upevnenie zariadenia)
- DEFECT ACT:0 – výskyt poruchy
- DEFECT ACT:1 – odstránenie poruchy

3.1.8 RC-86K diaľkový ovládač

Jedná sa o bezdrôtový diaľkový ovládač (Obr. A.3), ktorý sa chová ako dva samostatné ovládače. Príkladom použitia ovládača je zamknúť alebo odomknúť dvere na diaľku. Každá polovica ovládača má vlastnú adresu. Zariadenie nepodporuje detekcie sabotáže (otvorenie zariadenia) ani správu typu BEACON. Je však možné vyvolať udalosť PANIC, ktorá sa aktivuje stlačením oboch tlačidiel na danej polovici ovládača. Zariadenie je napájané batériou s približnou životnosťou 4 roky. Udávaný dosah je okolo 30 metrov pri priamej viditeľnosti [9].

Zariadenie môže odoslať nasledujúce typy správ [4]:

```
\n [YYYYYYYY] RC-86K ARM:? LB:? \n
\n [YYYYYYYY] RC-86K PANIC LB:? \n
```

- YYYYYYYY – sériové číslo zariadenia v dekadickej sústave
- RC-86K – typ zariadenia
- ARM:0 – stlačenie tlačidla pre odomknutie
- ARM:1 – stlačenie tlačidla pre zamknutie
- PANIC – stav tiesne
- LB:0 – informácia o nízkom stave batérie
- LB:1 – informácia o dobrom stave batérie

3.1.9 TP-82N Termostat

Termostat (Obr. A.3) slúži k meraniu a regulácii teploty. Požadovanú teplotu je možné nastaviť priamo na termostate. Termostat odosiela dva typy teplôt a to aktuálnu a nastavenú teplotu. Teplota sa posiela ako dvojčiferné číslo s presnosťou na jedno desatinne miesto (??.)°C. Stupeň Celzia (značka °C) sú posielané ako ASCII znaky 0xA7 a 0x43 (ďalej značené ako \xA743). Zariadenie nepodporuje detekcie sabotáže (otvorenie alebo odlepenie zariadenia) ani správu typu BEACON. Zariadenie je napájané batériou s približnou životnosťou 1 rok. Udávaný dosah je okolo 100 metrov pri priamej viditeľnosti [6].

Zariadenie môže odoslať nasledujúce typy správ [4]:

```
\n [YYYYYYYY] TP-82N SET:??.\xA743 LB:? \n
\n [YYYYYYYY] TP-82N INT:??.\xA743 C LB:? \n
```

- YYYYYYYY – sériové číslo zariadenia v dekadickej sústave
- TP-82N – typ zariadenia
- SET:??.\xA743 – nastavená teplota (v °C)
- INT:??.\xA743 – aktuálna teplota (v °C)
- LB:0 – informácia o nízkom stave batérie
- LB:1 – informácia o dobrom stave batérie

3.2 Z-Wave a Z-Wave Plus zariadenia

Z-Wave je bezdrôtový komunikačný protokol navrhnutý pre riadenie, sledovanie a zisťovanie stavu zariadení. O vývoj protokolu sa stará firma Sigma Design. Je predovšetkým využívaný v domácich podmienkach a je optimalizovaný na minimálnu spotrebu energie pri zachovaní dostatočnej latencie systému. Patrí medzi protokoly, ktoré nie sú voľne dostupné, a je medzinárodne štandardizovaný organizáciou Z-Wave ALLIANCE².

Protokol Z-Wave definuje niekoľko frekvencií v rozmedzí 865,2–926,0 MHz v závislosti na regióne. Pre Európu sú definované frekvencie 868,42 MHz a 869,85 MHz [26]. Z-Wave je určený pre prenos malých dátových paketov s nízkou latenciou. Prenosové rýchlosti môžu dosahovať 9,6 kbit/s, 40 kbit/s, prípadne 100 kbit/s.

²<http://z-wavealliance.org/>

Rýchlosť závisí na šírke kanálu a od regiónu, v ktorom sa dané zariadenie používa [20]. Dosah je udávaný na približne 30 metrov v závislosti od prekážok medzi zariadeniami. Pri tomto protokole je použitá mesh topológia siete. Každé zariadenie umožňuje okrem prijímania a odosielania dát preposielať dáta ďalej do siete. Preposielanie dát je možné maximálne cez 4 hopy (zariadenia).

Súhrnné informácie o ďalej popísovaných zariadeniach a o funkciách, ktoré podporujú, sú uvedené v tabuľke 3.2.

	AeoTec Multisensor 6 ZW100C	D-Link DCH-Z120	Fibaro FGK-Z120	Philio PST02-1C	Popp 123601
Napájanie zo siete	✓	x	x	x	✓
Batériové napájanie	✓	✓	✓	✓	x
Podpora detekcie sabotáže	x	✓	x	✓	x
Životnosť batérie [rok]	2	2	2	2	-
Podpora Z-Wave Plus	✓	✓	x	✓	x
Udávaný dosah [m]	150	10	30	30–70	300

Tabuľka 3.2: Vlastnosti zariadení Z-Wave

Konkrétnu Z-Wave sieť identifikuje 32 bitový identifikátor nazývaný *HomeID*. Zariadenia pripájané do Z-Wave siete sú adresované 8 bitovým identifikátorom *NodeID*, ktorý je v danej sieti unikátny. Niektoré identifikátory v rámci siete sú použité pre internú komunikáciu a špeciálne účely. Z tohto dôvodu je možné pripojiť k jednej Z-Wave sieti 232 prvkov. Jednotlivé siete vytvára zariadenie, ktoré sa označuje ako *primary controller* alebo *USB Dongle*.

Pri pridávaní zariadenia do siete sa odošle párovacia požiadavka pomocou USB zariadeniu a párované zariadenie sa následne ohlási. Po ohlášení sa danému zariadeniu priradí *HomeID* a *NodeID*.

Protokol Z-Wave definuje tzv. *komunikačné triedy*, ktoré zahŕňajú príkazy slúžiace pre komunikáciu zo zariadeniami. Aby bola komunikácia možná, musí komunikujúce zariadenia podporovať totožnú *komunikačnú triedu*. V súčasnej verzii protokolu existuje niekoľko desiatok definíc týchto tried [21] [22]. Príkladom môže byť trieda `Binary Switch Command Class`, ktorá slúži na riadenie zariadení so stavmi zapnuté, vypnuté a povolené, zakázané alebo `Battery Command Class` na zistenie stavu úrovne batérie zariadenia.

Nástupcom Z-Wave je Z-Wave Plus, ktorý rozširuje a vylepšuje možnosti Z-Wave. Hardwarová platforma zlepšila dosah a predĺžila životnosť batérie. Pridala sa možnosť bezdrôtovej aktualizácie (OTA³). Pre zvýšenie dosahu a spoľahlivosti prenosu boli pridané RF kanály [7].

³Over The Air upgrading

Typickým znakom Z-Wave zariadení je pridanie, či odstránenie senzového prvku tlačidlom trikrát za sebou v definovanom časovom intervale. K pridaniu alebo odstráneniu zariadenia je potrebné USB zariadenie.

3.2.1 USB Dongle Aeotec ZW090 Z-Stick

Aeotec Z-Stick Dongle (Obr. A.2) umožňuje ovládať zariadenia v Z-Wave sieti a tak dokáže kontrolovať osvetlenie, bezpečnostné senzory alebo merať teplotu. Zariadenie umožňuje pripojiť až 232 zariadení. V zariadení sa nachádza batéria, ktorá spolu s tlačidlom umožňuje bez použitia počítača pridať alebo odstrániť zariadenie zo Z-Wave siete. Stav siete a zariadenia je indikovaný pomocou LED diódy [18]. Toto USB zariadenie umožňuje pridávať a odoberať zariadenia aj bez použitia hostiteľského systému. USB zariadenie je možné vrátiť do výrobných nastavení pomocou tlačidla umiestneného na zadnej strane ktoré treba držať pokiaľ sa nezmení červená farba na modú.

3.2.2 Aeotec Multisensor 6 – ZW100C

Senzor Aeotec Multisensor 6 (Obr. A.4) predstavuje šesť samostatných sensorov, ktoré sa nachádzajú v jednom zariadení. Multisenzor umožňuje detekovať vlhkosť, teplotu, intenzitu svetla, UV žiarenie, pohyb a otras. Dosah zariadenia pri priamej viditeľnosti je okolo 150 metrov. Zariadenie je možné napájať batériou s približnou životnosťou 2 roky alebo USB napájacím adaptérom. Senzor podporuje Z-Wave Plus [19].

3.2.3 D-Link DCH-Z120 – Senzor pohybu

Senzor pohybu (Obr. A.4) umožňuje informovať o neoprávnenom vstupe do budovy alebo domu. Okrem pohybového senzoru obsahuje aj senzor teploty a svetelnosti. Dosah zariadenia je okolo 10 metrov pri priamej viditeľnosti. Zariadenie je napájané batériou s približnou životnosťou 2 roky. Senzor podporuje Z-Wave Plus [17].

3.2.4 Fibaro FGK-107 – Magnetický kontakt s binárnym vstupom a meraním teploty

Univerzálny senzor (Obr. A.4) kombinujúci detekciu otvorenia dverí alebo okien pomocou magnetického senzoru spolu s teplotným a binárnym sensorom. Binárny senzor umožňuje použitie externého senzoru (tlačidlo). Pre pripojenie teplotného čidla je nutné pripojiť externý snímač DS18B20⁴. Dosah zariadenia pri priamej viditeľnosti je približne 30 metrov. Zariadenie je napájané batériou s približnou životnosťou 2 roky. Senzor podporuje Z-Wave Plus [16].

3.2.5 Philio PST02-1C – Multisenzor

Multisenzor Philio PST02-1C (Obr. A.4) v sebe obsahuje senzor teploty, osvetlenia a magnetický kontakt, ktorý je možné použiť na detekciu otvorenia, či zavretia dverí alebo okien. Dosah zariadenia pri priamej viditeľnosti je okolo 30 metrov. Zariadenie je napájané batériou s približnou životnosťou 2 roky. Senzor podporuje Z-Wave Plus [14].

⁴DS18B20 - Digitálny teplotný senzor

3.2.6 Popp 123627 – Bezdrôtovo ovládaná zásuvka

Zásuvku Popp 123627 (Obr. A.4) je možné zapínať a vypínať bezdrôtovo alebo pomocou zabudovaného tlačidla na zapnutie alebo vypnutie. Od iných zásuviek sa odlišuje možnosťou automatického vypínania a funkciou LED nočného svetla, ktoré je možné nastaviť. Zásuvka obsahuje zabudovaný opakovač, ktorý zlepšuje dosah Z-Wave siete. Dosah zariadenia pri priamej viditeľnosti je okolo 30 metrov. Zariadenie je napájané batériou s približnou životnosťou 2 roky. Senzor nepodporuje Z-Wave Plus [15].

Kapitola 4

Komunikácia BeeeOn Gateway so senzormi

Pre komunikáciu manažérov zariadení (viď podkapitola 5.1) s BeeeOn Gateway bol zvolený transportný protokol ZeroMQ (ZMTP) pre svoje široké možnosti, ktoré sú popísané v tejto kapitole.

Jedná sa o tzv. ZeroMQ message transport protocol (ZMTP) alebo taktiež ZeroMQ. Je to protokol s otvoreným štandardom¹ a zdrojovými súbormi. Protokol je orientovaný na zasielanie asynchrónnych správ v počítačovej sieti, ktorý je distribuovaný pod LGPLv3 licenciou. Knižnica je nadstavbou nad sieťovými soketmi, ktorá sa zameriava predovšetkým na ľahké použitie a výkon. Základ knižnice ZeroMQ je napísaný v programovacom jazyku C, aby sa dosiahol, čo najlepší výkon s ohľadom na použité zdroje. Vďaka širokej komunite je možné ZeroMQ používať vo väčšine programovacích a skriptovacích jazykov (C, C++, Java, C#, PHP, Bash, Python, Go a ďalších) [23, s. 7–9].

4.1 Základné ZeroMQ prenosové vzory

ZeroMQ ponúka niekoľko typov prenosových vzorov, ktoré sa líšia svojim použitím. Prenosové vzory popisujú spôsob, akým je možné komunikovať, spracovávať komunikáciu, a zároveň aké typy soketov je možné použiť. Komunikácia a spracovanie môže prebiehať medzi dvomi alebo viacerými klientmi. Ďalšou možnosťou spracovania je zrefazovaná komunikácia.

Okrem popisovaných základných vzorov existujú aj nasledujúce, ktoré sú rozšírené o ďalšie možnosti. Príklad rozšírených vzorov sú napr. Lazy Pirate pattern, Binary Star pattern [24, s. 141], Clone pattern, Espresso pattern [24, s. 245] a ďalšie.

Jednotlivým návrhovým vzorom je potrebné priradiť správne typy soketov. Typ soketu definuje sémantiku smerovania správ, ktoré prichádzajú, odchádzajú alebo sa nachádzajú vo frontách. Jednotlivé fronty sa môžu vytvárať pri prijímaní ale aj pri odosielaní dát [24, s. 34]. Typy soketov nie je možné ľubovoľne kombinovať a sú určité prípustné a neprípustné kombinácie.

V uvedených príkladoch komunikácie kruhy znázorňujú komunikujúce body a šípky smer komunikácie.

¹<https://rfc.zeromq.org/>

4.1.1 Dotaz-odpoveď (request-response)

Prenosový vzor dotaz-odpoveď (Obr. 4.1) je príkladom komunikácie klienta a servera, kde klient posiela dáta na server, ktorý mu na prijaté dáta odpovie. Tento prenosový vzor slúži pre vzdialené volanie procedúr a distribúciu úloh [24, s. 37]. Pomocou tohto vzoru je možné implementovať synchronnú ale aj asynchrónnu komunikáciu, ktoré sa nastavujú podľa zvolených typov soкетов. Prenosový vzor umožňuje detekovať odpojenie klienta tým, že mu nepríde odpoveď na odoslanú požiadavku [24, s. 142].



Obr. 4.1: Dotaz-odpoveď (request-response) – ZeroMQ prenosový vzor

Podporované typy soкетов v tomto vzore [8]:

- **REQ** je typ soketu používaný, keď chce klient odoslať požiadavku a zároveň prijíma odpoveď zo služby, ku ktorej je pripojený. Každá požiadavka je odosielaná pomocou round-robin plánovača a každá prijatá odpoveď je spárovaná s odoslanou správou. Operácia pre odoslanie dát je blokujúca v prípade, že sa druhá strana odpojí a po pripojení sa odošlú nedoručené správy.
- **REP** tento typ soketu je používaný ako služba pre prijímanie požiadaviek z klientov, ktorým odosiela odpovede. Každá prijatá požiadavka sa vyberá pomocou fair-queue plánovača zo všetkých pripojených klientov.
- **DEALER** je typ soketu používaný pre pokročilejšie vzory, ktorý rozširuje vzor dotaz-odpoveď. Každá prijatá správa je odosielaná pomocou round-robin plánovača všetkým pripojeným klientom a každá prijatá správa je spracovaná fair-queue plánovačom. Operácia pre odoslanie dát je blokujúca v prípade, že sa druhá strana odpojí a po pripojení sa odošlú nedoručené správy.
- **ROUTER** tento typ soketu je, podobne ako **DEALER**, používaný pre pokročilejšie vzory a rozširuje vzor dotaz-odpoveď. Po prijatí správy tento typ soketu musí vybrať identifikáciu zo správy, ktorá identifikuje odosielateľa. Prijaté správy sú spracovávané pomocou fair-queue plánovača. Pri odosielaní správ sa odstráni prvá časť správy (identifikácia), ktorá sa použije na určenie, komu má byť správa odoslaná. Klientovi prídu len odoslané dáta. Odosielané správy odpojeným klientom sú zahadzované a neukladajú sa pre neskoršie odoslanie po pripojení klienta.

Povolené kombinácie soкетов [24, s. 86]:

- z **REQ** do **REP**
- z **DEALER** do **REP**
- z **REQ** do **ROUTER**
- z **DEALER** do **ROUTER**
- z **DEALER** do **DEALER**

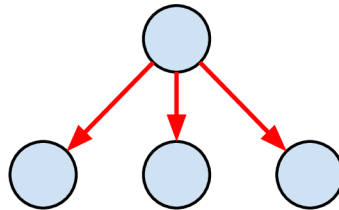
- z ROUTER do ROUTER

Zakázané kombinácie soketov [24, s. 86]:

- z REQ do REQ
- z REQ do DEALER
- z REP do REP
- z REP do ROUTER

4.1.2 Vydavateľ-odberateľ (publisher-subscriber)

Tento prenosový vzor (Obr. 4.2) slúži pre distribúciu dát jedným smerom a to z vydavateľa do odberateľov [24, s. 37]. Tento vzor neumožňuje komunikáciu opačným smerom a pre túto komunikáciu je potrebné použiť iný prenosový vzor. V popisovanom vzore je možné použiť socket typu vydavateľ (PUB) a odoberateľ (SUB). Odberateľ sa môže pripojiť k viac ako jednému vydavateľovi. Príkladom použitia, môžu byť aplikácia, ktorá slúži na komunikáciu, kde sa do nej jednotliví užívatelia pripájajú a dostávajú správy z jedného centrálného uzla, ktorý všetky nové správy prepošle pripojeným užívateľom. Tento vzor sa môže využiť aj pri paralelných výpočtoch, kedy vydavateľ vytvára úlohy a jednotlivé úlohy môžu byť paralelne spracované v jednotlivých odberateľoch [24, s. 16].



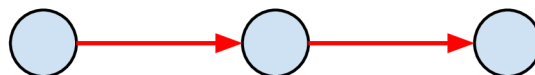
Obr. 4.2: Vydavateľ-odberateľ (publisher-subscriber) – ZeroMQ prenosový vzor

Podporované typy soketov v tomto vzore [8]:

- PUB tento typ soketu je používaný pre publikovanie dát všetkým pripojeným odberateľom a neumožňuje prijímanie správ. Správy pre odpojených odoberateľov sú zahadzované.
- SUB tento typ soketu je používaný pre prijímanie dát od vydavateľa. Pre prijímanie dát z odoberateľa je potrebné sa prihlásiť k odberu týchto dát. Tento typ soketu neumožňuje odosielať dáta.

4.1.3 Zreťazené spracovanie (pipeline)

Zreťazené spracovanie (Obr. 4.3) slúži pre ZeroMQ prenosové vzory, ktoré majú niekoľko krokov a cyklov.



Obr. 4.3: Zreťazené spracovanie (pipeline) – ZeroMQ prenosový vzor

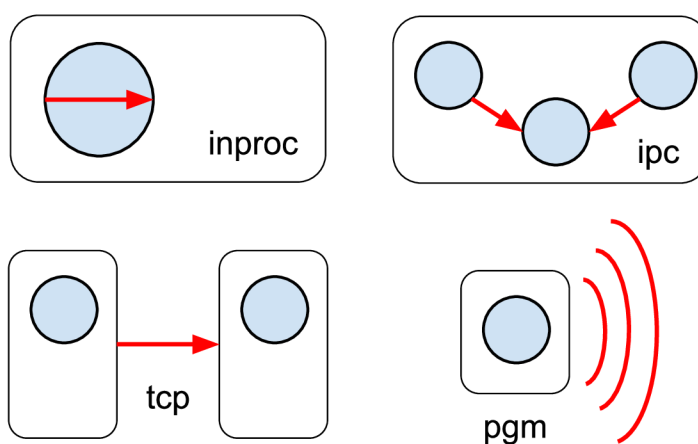
V prípade ak sa niektorý pracovník odpojí, odosielať príkazu sa o tom nedozvie. Pracovník prijíma úlohy a stará sa o ich vykonanie. Podobne ako vydavateľ-odberateľ umožňuje komunikovať len jedným smerom, a to v smere od zadávateľa úlohy k jednotlivým pracovníkom.

Podporované typy soketov v tomto vzore [8]:

- PUSH tento typ soketu je používaný pre odosielanie dát do nižšej vrstvy zreťazeného spracovania. Odosielanie dát prebieha pomocou round-robin plánovača. Prijímanie dát nie je možné. Pri odpojení pripojených klientov sú dáta ukladané a po opätovnom pripojení sa odošlú.
- PULL tento typ soketu je používaný pre zreťazené spracovanie prijatých správ z vyššie položených zreťazených spracovaní. Prijaté správy sú spracovávané pomocou fair-queue plánovača. Odosielanie dát týmto soketom nie je možné.

4.1.4 ZeroMQ typy prenosov

ZeroMQ poskytuje unikastový (inproc, ipc a tcp) a multikastový (pgm, epgm) prenos [24, s. 35]. Podporované prenosové typy sú znázornené na obrázku 4.4. ZeroMQ umožňuje zasielanie správ v podobe objektov (binary blob), ktoré sú pri prenose prevádzané na byty. Takýto typ umožňuje uchovávať dáta, kde je ich veľkosť limitovaná veľkosťou pamäte. ZeroMQ umožňuje automatické znovupripojenie na TCP sokety.



Obr. 4.4: Typy koncových bodov

Zoznam podporovaných transportných typov [24, s. 33]:

- **inproc** – komunikácia vlákien v jednom procese
- **ipc** – komunikácia medzi niekoľkými vláknami alebo procesmi
- **tcp** – komunikácia v jednej sieti
- **pgm, epgm** – komunikácia v rámci multikastových skupín

Kapitola 5

Návrh rozšírenia Gateway aplikácie

Pôvodná implementácia AdaApp aplikácie slúžila len na preposielanie a prevod hodnôt z jedného typu senzorov. Návrh AdaApp nepočítal s obecným princípom pripájania ďalších druhov senzorov, od ktorých by sa preposielali informácie na server. Preto tento návrh bolo potrebné rozšíriť o podporu pridávania nových senzorov. AdaApp počítala len so synchronnými operáciami a na asynchrónne spracovanie nebol vytvorený žiaden mechanizmus. Medzi asynchrónne operácie sa zaraďujú všetky operácie, ktorým nie je určený čas ich ukončenia. V prípade BeeeOn Gateway to môže byť nastavenie hodnoty na senzore (viď podkapitola 5.5.6), kde výsledok nastavenia môže prísť okamžite ale aj po uplynutí určitého času.

S príchodom nových senzorov bolo potrebné zasahovať do jadra aplikácie, ktoré na to nebolo stavané. Okrem podpory nových druhov senzorov mala aplikácia problémy s narastajúcim počtom nových typov správ, ktoré si vyžadovali tieto senzory. V konečnom dôsledku sa dospelo do stavu, kedy sa do aplikácie dopĺňali súčasti, ktoré riešili len veľmi malý okruh problémov a neriešili vzniknuté problémy systematicky. Príkladom nesystematického riešenia môžu byť rôzne možnosti pripájania zariadení alebo spúšťanie príkazov pre jednotlivé senzorové prvky.

Súčasný spôsob implementovania nových modulov do AdaApp nie je presne definovaný a prináša radu nevýhod. Rozhranie modulov sa mení na základe požiadaviek, ktoré nové moduly zo sebou prinášajú. Moduly sú implementované priamo v AdaApp alebo s ňou len komunikujú za použitia vlastného protokolu (viacej informácií v kapitole 2.2.1).

V návrhu pre pripájanie senzorov tretích strán (viď podkapitola 5.1) bol rozšírený existujúci základ novej BeeeOn Gateway aplikácie (ďalej označovaná ako Gateway), ktorá nahradí aktuálne používanú AdaApp. V novom návrhu sa myslelo na systematické riešenie pridávania nových senzorových protokolov a spúšťanie príkazov pre jednotlivé protokoly.

5.1 Návrh pripájania senzorov tretích strán

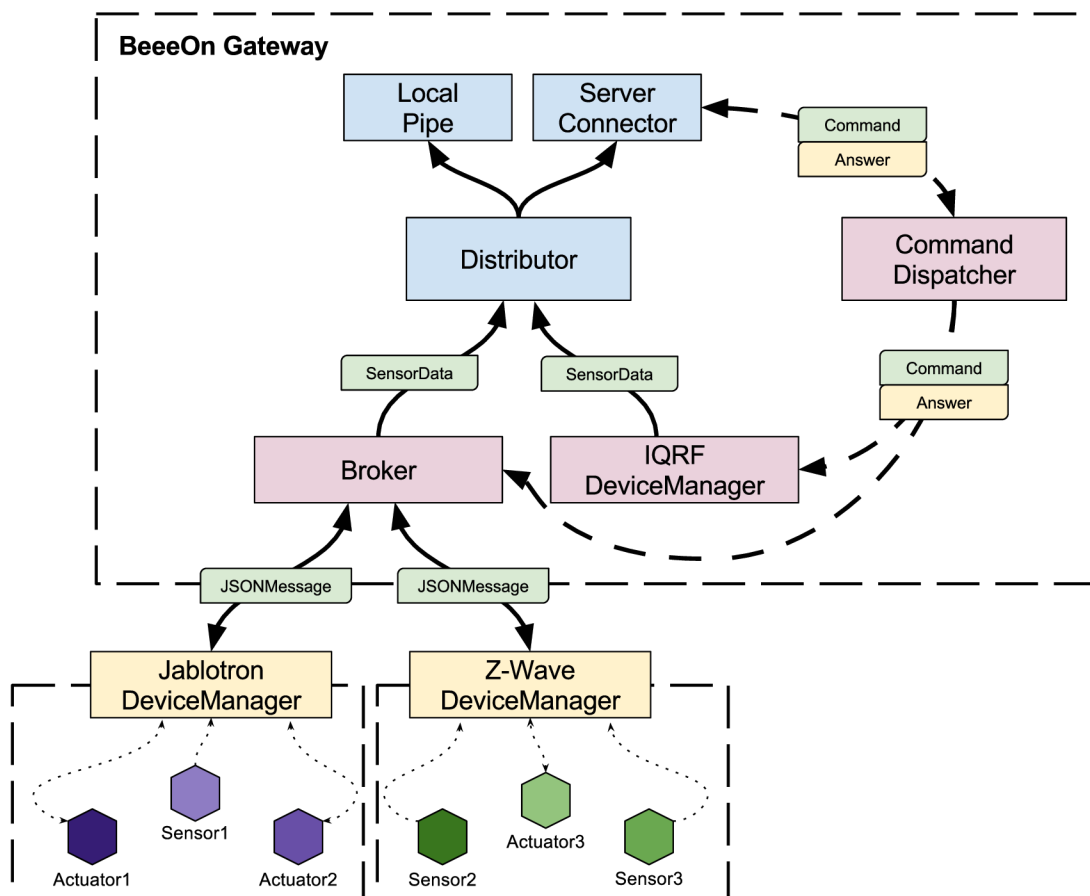
Pri implementácii návrhu sa predpokladá striktný objektovo orientovaný prístup. Výhody objektovo orientovaného prístupu sú zapúzdrenie, abstrakcia, polymorfizmus a dedičnosť. Vďaka skrytým implementačným detailom vo vnútri objektu je možné pre komunikáciu medzi objektami používať len definované rozhranie danej triedy.

Gateway je rozdelená do niekoľkých celkov (Obr. 5.1), ktoré sa môžu spúšťať samostatne.

Najdôležitejšou časťou je samostatná Gateway, ktorá na jednej strane prijíma dáta z rôznych typov zariadení a na druhej strane poskytuje rozličné spôsoby exportu týchto informácií. Ďalšími súčasťami Gateway sú manažéri zariadení (DeviceManager), ktorí sa starajú o komunikáciu s konkrétnymi sensorovými prvkami.

O príjem dát vo vnútri aplikácie sa stará distribútor (Distributor), ktorý prijaté dáta (SensorData) prepošle registrovaným exportérom. Príkladom týchto exportérov môže byť linuxová rúra (LocalPipe), alebo klient (ServerConnector), ktorý komunikuje zo serverom. Každý z týchto exportérov môže prijaté dáta formátovať podľa svojej potreby. Táto základná časť bola rozšírená o možnosť rozposielania príkazov v rámci brány a prostredníka (Broker), ktorý komunikuje s pripojenými sensorovými manažérmi.

Odosielateľ príkazov (CommandDispatcher) zastrešuje volanie a spracovanie požadovaných príkazov (Command). Príkazom sa rozumie operácia, ktorá má niečo vykonať alebo spustiť. Príkladom príkazu môže byť zistenie poslednej nameranej hodnoty ale aj spustenie párovacieho režimu. Popis príkazov je uvedený v kapitole 5.5. Ktorákoľvek komponenta v rámci brány môže posilať ale aj spracovávať príkazy, ktoré považuje pre svoju funkčnosť za dôležité. Odpoveď (Answer) môže obsahovať niekoľko výsledkov, pretože príkazy môžu byť spracované niekoľkými komponentami súčasne.



Obr. 5.1: Návrh rozšírenia BeeeOn Gateway aplikácie

Prostředník (Broker) vytvárá rozhraní pre pripájanie manažérov zariadení k bráne. Na jednej strane umožňuje komunikovať s manažérmi pomocou správ (JSONMessage) v JSON jazyku. Popis správ sa nachádza v kapitole 5.5. Na druhej strane komunikuje s distribútorom, ktorému predáva prijaté namerané hodnoty. Prostředník okrem spracovania prijatých dát zabezpečuje komunikáciu s odosielateľom príkazov, čo umožňuje preposielanie príkazov jednotlivým manažérom zariadení.

Manažéri zariadení prekladajú správy z prostředníka na operácie, ktoré sú špecifické pre dané sensorové prvky. Tento preklad prebieha aj opačným smerom. Príkladom je rôzne spracovanie dát zo sensorových prvkov Z-Wave (ZWaveDeviceManager) a Jablotron (JablotronDeviceManager) v týchto manažéroch zariadení a ich unifikovaný výstup pre prostředníka.

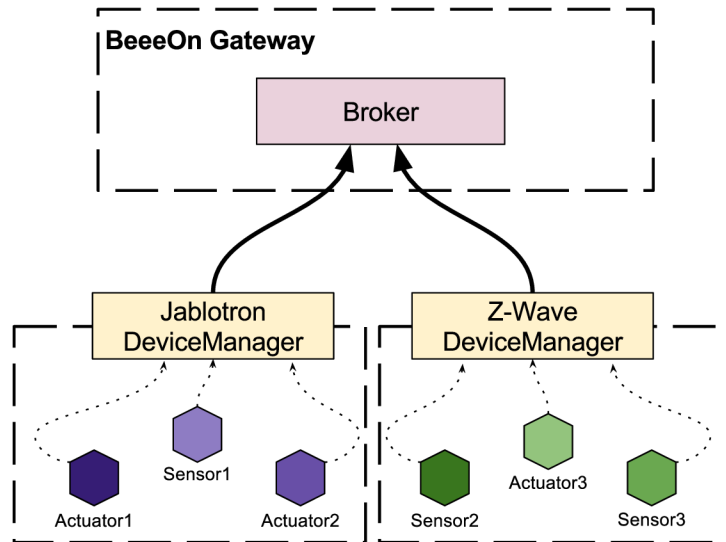
Každý manažér zariadení sa viaže k určitému protokolu alebo množine zariadení, s ktorými dokáže komunikovať. Manažérom zariadení je priradený prefix, ktorý túto skupinu zariadení identifikuje. Z-Wave manažér zariadení komunikuje so zariadeniami, ktoré používajú Z-Wave protokol. Jablotron manažér zariadení komunikuje so zariadeniami, ktoré podporujú bezdrôtový protokol OASiS od firmy Jablotron.

Sensorové prvky pripájané pomocou prostředníka využívajú existujúce knižnice, prípadne vlastnú implementáciu spracovania. Manažér zariadenia IQRF (IQRFDeviceManager) podporuje aplikáciu, ktorá umožňuje exportovať dáta zo sensorových prvkov. Príkladom môže byť IQRF GW daemon¹, ktorý komunikuje s okolím definovaným formátom. Tieto sensorové prvky sa budú pripájať priamo na bránu, kde stačí implementovať klienta pre spracovanie definovaného formátu z tejto aplikácie. Manažér zariadení IQRF nie je súčasťou implementácie.

¹<https://github.com/iqrfsdk/iqrf-daemon>

5.2 Identifikácia požiadaviek pre komunikáciu

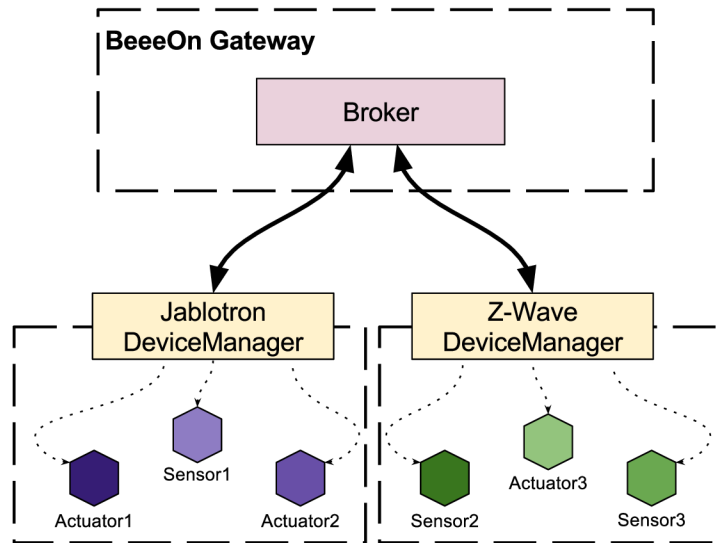
Komunikácia medzi prostredníkom a jednotlivými manažérmi zariadení vyžaduje určité požiadavky. Požiadavky vychádzajú zo spôsobu, akým sú dáta posielané do prostredníka z manažérov zariadení a opačne.



Obr. 5.2: Odosielanie dát z manažérov zariadení do prostredníka (komunikácia N-to-1)

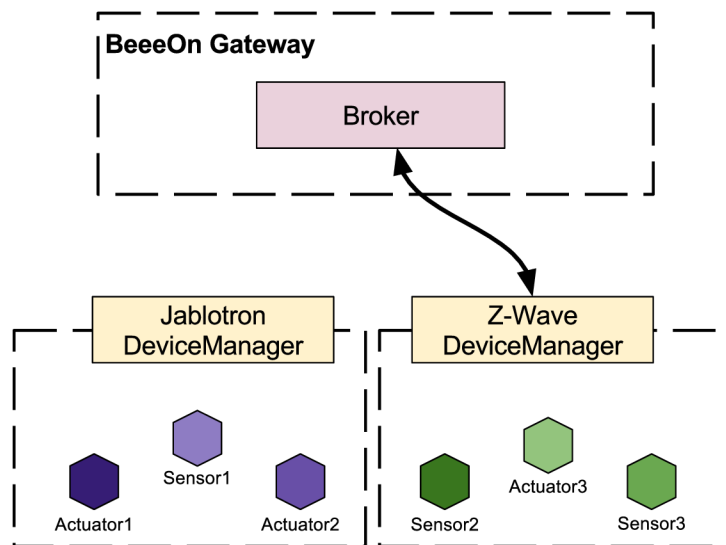
Jedná sa teda o komunikáciu N-to-1, ako je zobrazené na obrázku 5.2. Komunikáciu inicializuje manažér, ktorý odosiela dáta. Okrem samostatnej komunikácie je nutné zabezpečiť spoľahlivé doručenie nameraných hodnôt v prípade prerušenia spojenia s prostredníkom. Riešením je ukladať namerané hodnoty a pri obnovení spojenia tieto dáta postupne odoslať.

Okrem odosielania nameraných hodnôt je nutné brať do úvahy riadiace správy pre senzorové prvky v opačnom smere. Tento typ správ je možné rozdeliť na dve skupiny v závislosti na tom, či je správa určená všetkým manažérom zariadení alebo len jednému konkrétnemu.



Obr. 5.3: Odoslanie správy vybraným manažérom zariadení (komunikácia 1-to-N)

Ak sa jedná o správu pre všetky moduly jedná sa o komunikáciu, kedy brána potrebuje poslať rovnakú správu všetkým manažérom zariadení, ako je znázornené na obrázku 5.3. Druhou možnosťou je odoslať správu len vybraným manažérom zariadení. Jedná sa o komunikáciu 1-to-N. Komunikáciu inicializuje brána. Príkladom takejto komunikácie môže byť zapnutie párovacieho režimu na všetkých alebo len na vybraných manažéroch zariadení.



Obr. 5.4: Nastavenie hodnoty na senzore z BeeeOn brány (komunikácia 1-to-1)

V prípade, že sa jedná o správu, ktorá je určená len jednému konkrétnemu manažérovi zariadení, ako je znázornené na obrázku 5.4, komunikuje sa len medzi dvomi uzlami (komunikácia 1-to-1). Komunikáciu inicializuje brána. Príkladom takejto komunikácie je zistenie informácií o zvolenom manažérovi zariadení alebo nastavenie hodnoty na senzore.

5.3 Pripojenie Jablotron zariadení do prostredia OS Linux

Pre implementáciu bolo nutné vytvoriť manažéra zariadení, ktorý spracováva údaje z Turris Dongla. Manažér zariadení umožňuje zistiť, či je zariadenie pripojené k BeeeOn bráne. Po pripojení tohto zariadenia je nutné načítať prvky, s ktorými je možné komunikovať. Po načítaní zariadení je potrebné nastaviť stav zásuviek podľa hodnôt zo servera a následne manažér zariadení môže spracovávať prijaté správy zo senzorov. Na spracovanie vyčítaných dát boli použité regulárne výrazy a konečný automat pre postupné spracovávanie jednotlivých častí prijatej správy. Po spracovaní sa prijaté dáta pošlú do prostredníka na bráne, ktorý ich prepošle na server. Výsledkom návrhu je JablotronDeviceManager.

5.4 Pripojenie Z-Wave zariadení do prostredia OS Linux

Pre komunikáciu s USB zariadením je možné použiť knižnicu OpenZWave² a jej nadstavby pre rôzne programovacie jazyky. Samotná knižnica umožňuje prijať pakety zo Z-Wave siete a ponúka možnosť identifikovať tieto dáta, prípadne ich vyčítať.

Po prijatí dát sa zistí, o aký typ správy sa jedná. Ak sa jedná o namerané hodnoty, zistí sa o akého výrobcu a produkt sa jedná. Následne sa spracujú prijaté dáta a odošlú sa prostredníkom, ktorý ich prepošle na server. Výsledkom návrhu je ZWaveDeviceManager.

5.5 Návrh príkazov

Medzi najdôležitejšie časti návrhu patrí návrh príkazov. Príkazom sa rozumie operácia, ktorá má niečo vykonať alebo spustiť. Príkladom príkazu môže byť zistenie poslednej nameranej hodnoty ale aj spustenie párovacieho režimu. Jednotlivé príkazy sa dajú používať v rámci Gateway, ale aj v rámci jednotlivých manažérov zariadení bez ohľadu na to, či je daný manažér zariadení priamo súčasťou Gateway alebo s Gateway komunikuje pomocou prostredníka. Ak manažér zariadení komunikuje pomocou prostredníka (Broker), jednotlivé príkazy sú prevedené do JSON jazyka a späť.

5.5.1 Registrovanie manažéra zariadení do prostredníka

Manažér zariadení sa musí pri spustení registrovať do prostredníka. Dôvodom tejto registrácie je pridelenie vygenerovaného identifikátora, pomocou, ktorého prostredník posiela dáta jednotlivým manažérom zariadení. Manažéri zariadení odosielajú dáta s týmto identifikátorom, aby prostredník vedel, od ktorého manažéra zariadení prišli údaje. Pre vygenerovanie nového identifikátora je nutné dodať prefix manažéra zariadení, ktorému sa má vygenerovať identifikátor. Výsledný vygenerovaný identifikátor má nasledujúci hexadecimálny formát `XXYY` a skladá sa z dvoch častí:

- `XX` – prefix manažéra zariadení
- `YY` – poradové číslo v rámci zadaného prefixu

Príkaz pre registrovanie manažéra zariadení do prostredníka je volaný z každého manažéra zariadení a odpovedá na neho prostredník v Gateway.

²<http://www.openzwave.com/>

5.5.2 Zoznam napárovaných zariadení

Po vygenerovaní identifikátora manažéra zariadení je nutné načítať zoznam napárovaných zariadení zo servera, aby bola zachovaná konzistencia dát medzi údajmi na serveri a údajmi v Gateway aplikácii. Pre získanie napárovaných zariadení zo servera je potrebné uviesť prefix manažéra zariadení.

Bez tohto príkazu by mohlo dôjsť k situácii, kedy by sa posielali namerané údaje z nena-párovaných zariadení. Ďalší problém by mohol nastať pri odstránení lokálneho konfigurač-ného súboru, ktorý obsahuje zoznam napárovaných zariadení. Tento súbor by bol náhradou príkazu a obsahoval by pre každého manažéra zariadení zoznam napárovaných zariadení. Konfiguračný súbor by mohol byť odstránený nešťastnou náhodou, reinstalovaním systému na adaptéri alebo znefunkčnením adaptéra, kedy by daný manažér zariadení nepoznal žiadne napárované zariadenie.

Tento príkaz je volaný z každého manažéra zariadení a odpovedá na neho prostredník v Gateway. Odpoveďou je zoznam identifikátorov zariadení, s ktorými je možné komunikovať a stav, či je tento zoznam platný alebo nie.

5.5.3 Posledná nameraná hodnota na zariadení

Po načítaní zoznamu zariadení je nutné hodnoty na sensorových prvkoch, ktorým je možné nastavovať údaje zosynchronizovať s hodnotami, ktoré boli naposledy nastavené a uložené na serveri. Pre získanie poslednej nameranej hodnoty na sensorovom prvku je potrebné poznať identifikáciu sensorového prvku a jeho modulu. Ak by hodnoty neboli zosynchro-nizované, mohlo by dôjsť počas nedostupnosti Gateway k zmene týchto hodnôt ručným nastavením na niektorých typoch senzorov.

Typickým príkladom použitia tohto príkazu je výpadok elektrickej energie v budove, v ktorej sa nachádza bezdrôtovo ovládaná zásuvka. Pri obnovení dodávky elektrickej energie si zásuvka nepamätá posledný stav, v ktorom bola (zapnutá/vypnutá). Preto je nutné sa na tento stav dotázať servera, ktorý má uloženú túto hodnotu. Tento príkaz je volaný z každého manažéra zariadení a odpovedá na neho prostredník v Gateway. Výsledkom tohto príkazu je posledná nameraná hodnota na zadanom sensorovom prvku, prípadne posledná požadovaná hodnota u aktora. Okrem poslednej nameranej hodnoty sa v odpovedi nachádza aj stav, či je táto hodnota platná alebo nie.

5.5.4 Napárovanie nového zariadenia

Pred tým ako sa môžu prijímať namerané hodnoty od sensorových prvkov musí byť prvok registrovaný v BeeOn systéme (pridaný v databáze na serveri), inak sú jeho namerané hodnoty ignorované v manažéri zariadení a neodošlú sa na server.

Pre pridanie nového zariadenia sa musí zapnúť párovací režim, ktorý umožní po sta-novenú dobu prijímať dáta aj zo zariadení, ktoré nie sú registrované v BeeOn systéme. Prostredník po prijatí príkazu na napárovanie, prepošle tento príkaz všetkým pripojeným manažérom zariadení. Odpoveďou na tento príkaz je úspešné alebo neúspešné zapnutie páro-vacieho režimu od jednotlivých pripojených manažérov zariadení. Novému zariadeniu stačí odoslať namerané údaje a užívateľ v Android aplikácii môže potvrdiť pridanie zariadenia k svojmu adaptéru.

5.5.5 Odpárovanie pridaného zariadenia

Podobný proces ako pri pridávaní zariadenia nastáva aj pri jeho odpárovaní. Ak je požadované odstránenie napárovaného zariadenia z BeeeOn systému, je potrebné poznať jeho identifikátor. Po odpárovaní nebude možné prijímať údaje z daného sensorového prvku

Prostredník po prijatí príkazu na odpárovanie, prepošle tento príkaz príslušnému manažérovi zariadení, ktorý má pripojený daný sensorový prvok. Správny manažér zariadení sa vyberie na základe prefixu v identifikátore zariadenia, ktoré sa má odpárovať. Výsledkom tohto príkazu je stav, ktorý značí, či prebehlo odpárovanie úspešne alebo nie.

5.5.6 Nastavenie hodnoty na zariadení

Niektoré sensorové prvky umožňujú meniť svoj stav. Tento stav je možné nastavovať pomocou Android aplikácie. Pre nastavenie hodnoty je potrebné poznať identifikáciu prvku a hodnotu, ktorá sa má nastaviť na vybraný sensorový prvok.

Prostredník po prijatí príkazu na nastavenie hodnoty prepošle tento príkaz príslušnému manažérovi zariadení, ktorý má pripojený požadovaný sensorový prvok. Podobne ako pri odpárovaní sa správny manažér zariadení vyberie na základe prefixu v identifikátore zariadenia, ktoré sa má nastaviť. Výsledkom tohto príkazu je stav, v ktorom sa nachádza nastavenie hodnoty.

Odoslaním požiadavky na nastavenie hodnoty na sensorový prvok môže Gateway túto požiadavku potvrdiť a poslať na server správu o akceptovaní (`gw_accepted`), zároveň prepošle tento príkaz do správneho manažéra zariadení. Okrem úspešného potvrdenia môže byť poslaný stav (`gw_busy`), kedy brána nemôže spracovať príkaz. Jednotlivé stavy, ktoré môžu nastať po prijatí správy sú znázornené na obrázku 5.5. Dôvodom neúspešného spracovania môže byť plná tabuľka pre nastavenie hodnôt alebo nedostupnosť potrebného manažéra zariadení, ktorý sa má postarať o nastavenie hodnôt.

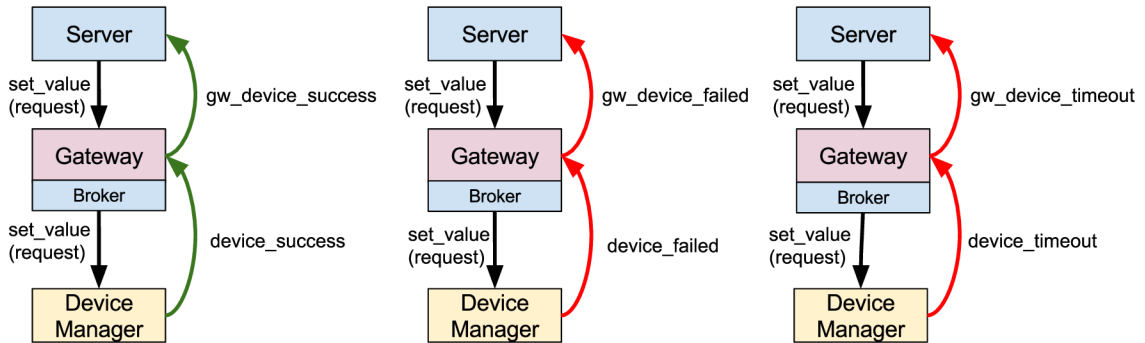


Obr. 5.5: Stavy po prijatí správy na nastavenie hodnoty v Gateway

Manažér zariadení, ktorému je určený príkaz ho spracuje (Obr 5.6) a pošle požiadavku na nastavenie hodnoty na danom zariadení. Výsledkom úspešného nastavenia hodnoty môže byť informačná správa o úspechu nastavenia hodnoty v rámci protokolu, ktorý daný manažér zariadení obsluhuje. Ak neexistuje takáto informačná správa, je možné potvrdenie poslať po prijatí nameraných dát, ktoré obsahujú aj dáta zo zmeneného modulu. Potvrdenie nastavenia (`device_success`) manažér zariadení prepošle prostredníkovi, ktorý informuje server o nastavení (`gw_device_success`).

V prípade neúspechu môžu nastať dve situácie. Prvou možnosťou je, že manažér zariadení nedokáže spracovať požiadavku, prípadne vie, že daný sensorový prvok nie je dostupný na základe informácii z daného protokolu. Informáciu o tom, že na danom prvku nie je možné nastaviť hodnotu (`device_failed`) pošle manažér zariadení prostredníkovi, ktorý informuje server, že nie je možné nastaviť hodnotu (`gw_device_failed`).

Druhou možnosťou je, že daný sensorový prvok nie je dostupný a správu vôbec neprijal ani po uplynutí stanoveného času. Stanovený čas je doba, po ktorú by malo dôjsť k nastaveniu hodnoty. Ak sa nastavenie nevykoná do stanoveného času dôjde k odoslaniu informácie (device_timeout) z manažéra zariadení do prostredníka o neúspešnom nastavení kôli vypršaní tejto doby. Prostredník po prijatí tejto informácie prepošle neúspech spôsobený vypršaním času na nastavenie na server (gw_device_timeout).



Obr. 5.6: Stav po prijatí správy na nastavenie hodnoty v manažéroch zariadení a Gateway

Dôvod prečo je rozlišovaná informácia z manažéra zariadení a prostredníka je, že na jednotlivé správy reaguje server a prostredník odlišne. Príkladom môže byť informácia o zlyhaní nastavenia (device_failed). Na túto informáciu prostredník reaguje znovuposlaním požiadavky na nastavenie hodnoty ale len v prípade, že nebol prekročený časový limit nastavovania. Pri informácii o úspešnom nastavení (device_success), prípadne pri informácii o vypršaní časového intervalu (device_timeout) sa táto informácia prepošle na server, kde sa rozhodne, čo sa má ďalej vykonať. Server môže požiadavku na nastavenie hodnoty znovu poslať do Gateway, prípadne môže len informovať užívateľa o tomto stave. Ak server prijme informáciu, že brána je zaneprázdnená (gw_busy), skúsi požiadavku o nastavenie odoslať znova. V prípade úspešného odoslania správy môže byť užívateľ informovaný o tomto stave.

Kapitola 6

Testovanie

Cieľom testovania bolo dokázať správnosť implementácie manažérov zariadení Jablotron, Z-Wave a funkčnosť jednotného komunikačného rozhrania spolu s prostredníkom (Broker). Okrem testovania funkčnosti implementácie sú v tejto kapitole zahrnuté aj jednotkové testy dôležitých komponentov.

6.1 Vlastnosti komunikačných protokolov

Počas testovania boli zistené určité spoločné, ale aj odlišné vlastnosti. Medzi spoločné vlastnosti patrí dosah zariadení, ktorý dosahuje v uzavretých priestoroch približne 25 metrov.

Čo sa týka spotreby energie, zariadenia používajúce Z-Wave protokol majú väčšiu spotrebu oproti Jablotron zariadeniam. Dôvodom môže byť náročnejšia implementácia protokolu ale aj väčšia komplexita zariadení, kedy na jednom zariadení je pripojených viacej senzorov.

Nevýhodou Jablotron zariadení, že používajú uzavretý protokol, ktorý podporuje komunikáciu len s veľmi obmedzeným počtom typov zariadení. Oproti tomu Z-Wave môžu používať rôzni výrobcovia a patrí tak medzi rozšírené komunikačné protokoly.

Protokoly Z-Wave a Jablotron sa líšia v spôsobe pripájania zariadení do siete. Pre pridanie zariadenia do Z-Wave siete je nutné zapnúť párovací režim na USB zariadení, ktoré vytvára sieť. Následne musí pripojované zariadenie odoslať žiadosť o pripojenie do siete, na ktorú USB zariadenie zareaguje priradením *HomeID* a *NodeID* (viď podkapitola 3.2).

Postup pri pridávaní zariadení do Jablotron siete je odlišný. Hlavným rozdielom je, že pripojované zariadenie nemusí byť v dosahu USB zariadenia. Pripájanie prebieha zápisom sériového čísla senzoru do USB zariadenia.

Čo sa týka odpárovania senzorov jednoduchšie riešenie ponúka protokol Jablotron. Pokiaľ je potrebné zariadenie odpárovať stačí len odoslať požiadavku na odstránenie zariadenia s požadovaným sériovým číslom.

Postup odpárovania zariadení u Z-Wave protokolu je podobný ako proces pripájania. Najprv je nutné odstrániť senzor z USB zariadenia a následne je aktivovaný režim pre odpárovanie zariadenia. Pre kompletné odpárovanie zariadenia je potrebné vykonať určitú akciu na zariadení (napr. opakované stlačenie tlačidla). Pokiaľ nie je zariadenie kompletne odpárované, nie je možné ho pripojiť do inej Z-Wave siete.

6.2 Jednotkové testy

Jednotkovými testami bola overená funkčnosť dôležitých častí implementovaného návrhu. Medzi testované časti patrí:

- odosielateľ príkazov (CommandHandler) – overenie, že dokáže odosielateľ príkazov odoslať a spustiť príkaz na registrovanom poslucháčovi,
- vytváranie a spracovanie príkazov v JSON jazyku – overenie, že všetky implementované príkazy a odpovede na príkazy sa správne prevádzajú do JSON jazyka a späť,
- odoslanie príkazu z prostredníka do klienta – otestovanie funkčnosti jednotlivých príkazov s testovacím klientom za účelom overenia, že prostredník správne implementuje spracovanie jednotlivých príkazov.

6.3 Funkčné testy manažérov zariadení

Najdôležitejšou časťou celého testovania bolo otestovanie navrhnutého riešenia. Testovanie prebiehalo medzi sensorovým prvkom a umelo vytvoreným klientom, ktorý simuluje komunikáciu so serverom (ServerConnector). Testy simulujú reálnu prácu so senzormi a požiadavky, ktoré posiela užívateľ pomocou Android aplikácie do BeeeOn systému. Simulovanie Android aplikácie a servera prebieha pomocou testovacieho klienta.

Každý test má svoj vlastný konfiguračný súbor, ktorý definuje chovanie testovacieho klienta v danom teste. Tento súbor umožňuje nastaviť, s akými zariadeniami sa má v teste pracovať a ak je potrebné, je možné vyvolať príkaz (napárovanie zariadenia, odpárovanie zariadenia, nastavenie hodnoty) s určitými parametrami. Spustenie takéhoto príkazu je možné nastaviť s časovým oneskorením, ktoré je možné meniť v konfiguračnom súbore. Okrem časového oneskorenia sa tu nachádza možnosť nastaviť dĺžku spusteného párovacieho režimu. Pri nastavovaní hodnoty je možné definovať v konfiguračnom súbore, ktorému zariadeniu a na ktorom module sa nastaví definovaná hodnota. Spomenuté príkazy umožňujú spustiť definované akcie s časovým oneskorením, ktoré sa taktiež dá nastaviť v konfiguračnom súbore.

Pre každý test boli vybrané zariadenia s požadovanými vlastnosťami. Na každý test je potrebný manažér zariadení (ZWaveDeviceManager, JablotronDeviceManager), preložené zdrojové súbory Gateway aplikácie a konfiguračné súbory. Každý test vyžaduje spustenie Gateway aplikácie a následné spustenie vybraného manažéra zariadení.

6.3.1 Test registrovania manažéra zariadení

Cieľom testu je registrovať manažéra zariadení do prostredníka. Manažérovi zariadení sa priradí identifikátor, ktorým sa identifikuje pri odosielaní údajov do prostredníka (Broker). Očakávaný výstup je registrovanie manažéra zariadení do prostredníka.

Pre spustenie testu je potrebný Jablotron manažér zariadení (JablotronDeviceManager). Manažér zariadenia po spustení odošle požiadavku na získanie identifikátora a následne prostredník (Broker) odošle vygenerovaný identifikátor.

1 `register device manager id: 900`

Zdrojový kód 6.1: Ukážka napárovaneého manažéra zariadení

Overenie priradenia identifikátora pre manažéra zariadení je možné pomocou výpisov z Gateway aplikácie (6.1).

6.3.2 Test získania zoznamu napárovaných zariadení

Cieľom testu je získať zoznam napárovaných zariadení zo servera pre konkrétny manažér zariadení.

Očakávaný výstup je zoznam registrovaných zariadení, od ktorých je možné preposlať správu prostredníkov (Broker). Zoznam napárovaných zariadení bude obsahovať len jedno zariadenie.

Pre spustenie testu je potrebný Z-Wave manažér zariadení (ZWaveDeviceManager) a USB Dongle pre ovládanie Z-Wave zariadení. Pred spustením testu musí byť USB zariadenie vo výrobnom nastavení (viď 3.2.1).

Podobne ako v predchádzajúcom teste sa po spustení manažéra zariadení odošle požiadavka na získanie identifikátora a následne prostredník (Broker) odošle vygenerovaný identifikátor. Po získaní identifikátora sa manažér zariadení dotáže prostredníka na zoznam zariadení. Po prijatí tohto zoznamu môže manažér zariadení preposielať správy do prostredníka (Broker).

```
1 {
2   "device_list" : [
3     {
4       "device_id" : "0xa80000ef1f430202"
5     }
6   ],
7   "message_type" : "device_list_result",
8   "result_status" : 1
9 }
```

Zdrojový kód 6.2: Ukážka zoznamu napárovaných zariadení

Overenie priradenia identifikátora pre manažéra zariadení je možné pomocou výpisov z Gateway aplikácie (6.2). Odpoveď na zoznam napárovaných zariadení bude zariadenie (device_id) s označením 0xa80000ef1f430202 a úspešným stavom odpovede (result_status).

6.3.3 Test poslednej nameranej hodnoty na zariadení

Cieľom testu je získať poslednú nameranú hodnotu zo servera pre konkrétny senzorový prvok pripojený k použitému manažéru zariadení.

Očakávaným výstupom je zapnutie zásuvky. V tomto teste bola zásuvka pred vypnutím Gateway a manažéra zariadení zapnutá (definované v konfiguračnom súbore).

Pre spustenie testu je potrebný Jablotron manažér zariadení (JablotronDeviceManager), diaľkovo ovládaná zásuvka Jablotron AC-88 a USB zariadenie pre pripojenie periférii. Pred spustením testu musí byť zásuvka vypnutá.

Pred požiadanim servera o poslednú nameranú hodnotu na senzore musí manažér zariadení požiadať o priradenie identifikátora a následne aj zoznamu zariadení, od ktorých môže preposielať správy do prostredníka (Broker). Následne manažér zariadení požiada server o poslednú nameranú hodnotu na danej zásuvke.

Overenie nastavenia poslednej nameranej hodnoty na senzorovom prvku sa vykoná vizuálne, a to tak, že tlačidlo je podfarbené červenou farbou (zapnuté). Ďalšou možnosťou je pripojenie na linuxovú rúru, kde Gateway preposiela prijaté dáta zo senzorových prvkov.

6.3.4 Test napárovania nového zariadenia

Cieľom testu je pridať nové zariadenie do BeeeOn systému, aby mohlo posielat namerané hodnoty na server.

Očakávaný výstup je pridanie nového zariadenia do manažéra zariadení a zároveň vyčítanie nameraných údajov zo senzora.

Pre spustenie testu je potrebný Z-Wave manažér zariadení (ZWaveDeviceManager), Aeotec Multisenzor 6 a USB Dongle pre ovládanie Z-Wave zariadení. Pred spustením testu musí byť USB zariadenie vo výrobnom nastavení 3.2.1 a multisenzor musí byť odpárovaný.

Pred požiadanim servera o poslednú nameranú hodnotu na senzore musí manažér zariadení požiadať o priradenie identifikátora a následne aj zoznamu zariadení, od ktorých môže preposielat správy do prostredníka (Broker). Následne manažér zariadení požiada server o poslednú nameranú hodnotu na danej zásuvke.

Overenie napárovania nového zariadenia do BeeeOn systému sa vykoná pripojením na linuxovú rúru, kde Gateway preposiela prijaté dáta zo senzorových prvkov.

6.3.5 Test odpárovania nového zariadenia

Cieľom testu je odstrániť zariadenie z BeeeOn systému, aby nemohlo posielat namerané dáta.

Očakávaný výstup je odstránenie pridaného zariadenia z manažéra zariadení.

Pre spustenie testu je potrebný Jablotron manažér zariadení (JablotronDeviceManager), diaľkovo ovládaná zásuvka Jablotron AC-88 a USB zariadenie pre pripojenie periférii.

Pred odpárovaním zariadenia sa musí manažér zariadení dotázať na identifikátor, ktorý mu bude následne priradený a taktiež na zoznam zariadení, od ktorých môže preposielat správy do prostredníka (Broker). Po získaní zoznamu napárovaných zariadení a získaní poslednej nameranej hodnoty sa spustí časovač, ktorý po vypršaní časového limitu spustí odpárovanie zariadenia. Manažér zariadení odstráni zariadenie zo zoznamu napárovaných zariadení.

Overenie odstránenia diaľkovo ovládanej zásuvky sa vykoná pripojením na linuxovú rúru a následným stlačením tlačidla na zásuvke. Po stlačení sa neobjaví žiadna nová hodnota, senzor nie je v zozname napárovaných zariadení.

6.3.6 Test nastavenia hodnoty na zariadení

Cieľom testu je nastaviť hodnotu pre konkrétny senzorový prvok pripojený k manažéru zariadení.

Očakávaný výstup je zmena stavu senzoru, ktorý je pripojený k manažéru zariadení, na takú hodnotu, ako bola zadaná v konfiguračnom súbore. Po vykonaní príkazu sa v tomto teste zásuvka vypne.

Pre spustenie testu je potrebný Jablotron manažér zariadení (JablotronDeviceManager), diaľkovo ovládaná zásuvka Jablotron AC-88 a USB zariadenie pre pripojenie periférii. Stav zásuvky po zapnutí aplikácii sa načíta z konfiguračného súboru. Predpokladá sa, že zásuvka bude zapnutá.

Pred nastavením hodnoty na senzore sa musí manažér zariadení dotázať na priradenie identifikátora a následne aj zoznamu zariadení, od ktorých môže preposielat správy do prostredníka (Broker). Ďalej sa manažér zariadení dotáže servera na poslednú nameranú hodnotu na danej zásuvke. Po získaní zoznamu napárovaných zariadení a získaní poslednej

nameranej hodnoty sa spustí časovač, ktorý po vypršaní časového limitu spustí odpárovanie zariadenia. Zariadenie sa odstráni zo zoznamu zariadení, od ktorých je možné prijímať dáta.

Overenie nastavenia poslednej nameranej hodnoty na senzоровom prvku sa vykoná vizuálne, a to tak, že tlačidlo je podfarbené červenou farbou (zapnuté). Ďalšou možnosťou je pripojenie na linuxovú rúru, kde Gateway preposiela prijaté dáta a sleduje stav prepnutia.

Kapitola 7

Záver

V tejto práci boli analyzované možnosti vybraných sensorových prvkov Jablotron a Z-Wave. Ďalej boli diskutované možnosti ich pripojenia do operačného systému Linux. Zariadenia od firmy Jablotron je možné pripojiť pomocou USB zariadenia Turris Dongle a senzory podporujúce bezdrôtový protokol Z-Wave za použitia Aeotec Z-Stick USB zariadenia. Následne bol navrhnutý jednotný komunikačný protokol pre pripájanie zariadení tretích strán. Ďalej bola rozšírená BeeeOn Gateway aplikácia, ktorá sprostredkováva komunikáciu s vybranými sensorovými prvkami. Komplexný systém pre pridávanie podpory nových protokolov bol navrhnutý a implementovaný. Celkom bola umožnená komunikácia zo šestnástimi zariadeniami od rôznych výrobcov.

Integrácia jednotného komunikačného protokolu do novej BeeeOn Gateway aplikácie bola problematická, pretože väčšina operácií nemá presne definovaný čas ukončenia, čo viedlo na vytvorenie asynchrónneho volania operácií. Aktuálna verzia novej BeeeOn Gateway aplikácie ešte nemá implementované pripojenie na server a neumožňuje posielat príkazy z Android zariadení. Tento problém bol vyriešený vytvorením testovacieho klienta, ktorý nahrádza klienta komunikujúceho z reálnym serverom. Takýto testovací klient v sebe zahŕňa potrebnú logiku pre testovanie a potrebné konfiguračné údaje si načítava zo súboru.

Literatúra

- [1] *FAQ - nejčastěji kladené dotazy Project: Turris / Gadgets*. [Online; navštívené 15.03.2017].
URL https://www.turris.cz/gadgets/faq#co_znamená_report_beacon
- [2] *Gateway - BeeOn*. [Online; navštívené 10.01.2017].
URL <https://beeeon.org/wiki/Gateway>
- [3] *JA-85ST Kombinovaný detektor kouře a teplot bezdrátový / Jablotron*. [Online; navštívené 14.03.2017].
URL <https://www.jablotron.com/cz/produkt/kombinovany-detektor-koure-a-teplot-bezdratovy-397/>
- [4] *Referenční manuál Project: Turris / Gadgets*. [Online; navštívené 14.03.2017].
URL <https://www.turris.cz/gadgets/manual>
- [5] *Systém OASiS 868 MHz*. [Online; navštívené 14.03.2017].
URL <https://www.jabloshop.cz/system-oasis-868-mhz>
- [6] *TP-82N Bezdrátový pokojový termostat / Jablotron*. [Online; navštívené 14.03.2017].
URL <https://www.jablotron.com/cz/produkt/bezdratovy-pokojovy-termostat-430/>
- [7] *What is Z-Wave Plus / Vesternet*. [Online; navštívené 28.01.2017].
URL <http://www.vesternet.com/what-is-z-wave-plus>
- [8] *zmq_socket(3) - 0MQ Api*. [Online; navštívené 1.05.2017].
URL <http://api.zeromq.org/4-2:zmq-socket>
- [9] *RC-86 bezdrátový ovládač. Září 2009*, [Online; navštívené 10.03.2017].
URL https://www.jabloshop.cz/index.php?route=product/product/download&download_id=163&product_id=55
- [10] *JA-81M bezdrátový detektor otevření univerzální vysílač*. Duben 2011, [Online; navštívené 14.03.2017].
URL https://www.jabloshop.cz/index.php?route=product/product/download&download_id=115&product_id=124
- [11] *JA-82SH bezdrátový detektor otřesu nebo náklonu*. Březen 2011, [Online; navštívené 14.03.2017].
URL https://www.jabloshop.cz/index.php?route=product/product/download&download_id=119&product_id=117

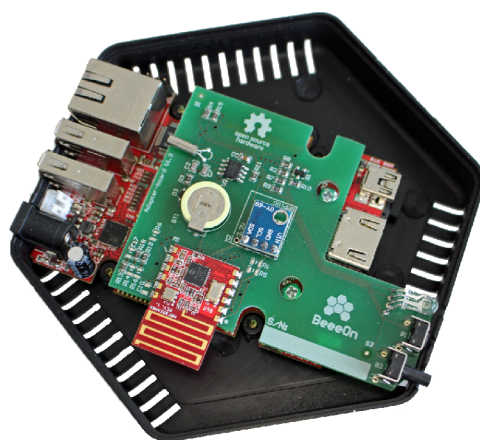
- [12] *JA-83M Bezdrátový magnetický detektor otevření*. Duben 2011, [Online; navštívené 14.03.2017].
URL https://www.jabloshop.cz/index.php?route=product/product/download&download_id=125&product_id=113
- [13] *JA-83P bezdrátový PIR detektor pohybu osob*. Duben 2011, [Online; navštívené 10.03.2017].
URL https://www.jabloshop.cz/index.php?route=product/product/download&download_id=126&product_id=941
- [14] *Slim Multisensor PST02-A/B/C*. Září 2011, [Online; navštívené 14.01.2017].
URL https://doc.eedomus.com/files/PST02_Manual-2014-09-17.pdf
- [15] *Z-Wave Wall Plug Switch*. Září 2013, [Online; navštívené 16.01.2017].
URL <https://www.koti.sk/files/popp/zasuvka/zwave-popp-bezdrotova-zasuvka.pfd.pdf>
- [16] *Door Window Sensor FGK-101-107 manual*. Leden 2014, [Online; navštívené 14.01.2017].
URL https://www.mojefibaro.cz/wp-content/uploads/2015/09/mn_magneticky_kontakt.pdf
- [17] *Battery Motion Sensor DCH Z120 manual*. Květen 2015, [Online; navštívené 14.01.2017].
URL http://www.dlink.com/cz/cs/-/media/consumer_products/dch/dch-z120/datasheet/dch_z120_datasheet_cz.pdf
- [18] *Z-Wave Plus Aeotec Z-Stick USB controller manual*. Červen 2015, [Online; navštívené 25.01.2017].
URL http://www.vesternet.com/downloads/dl/file/id/664/product/1541/z_wave_aeon_labs_z_stick_usb_controller_gen5_manual.pdf
- [19] *Z-Wave Plus Aeotec Multisensor 6 manual*. Srpen 2016, [Online; navštívené 15.01.2017].
URL http://www.vesternet.com/downloads/dl/file/id/764/product/1545/z_wave_aeon_labs_multisensor_6_gen5.pdf
- [20] Alliance, Z.-W.: *Z-Wave Transceivers – Specification of Spectrum Related Components*. 2014, [Online; navštívené 23.01.2017].
URL <http://z-wavealliance.org/wp-content/uploads/2015/02/ZAD12837-1.pdf>
- [21] Alliance, Z.-W.: *Z-Wave Command Class Specification, A-M*. Srpen 2016, [Online; navštívené 15.01.2017].
URL <http://z-wave.sigmadesigns.com/wp-content/uploads/2016/08/SDS12657-12-Z-Wave-Command-Class-Specification-A-M.pdf>
- [22] Alliance, Z.-W.: *Z-Wave Command Class Specification, N-Z*. Září 2016, [Online; navštívené 15.01.2017].
URL <http://z-wave.sigmadesigns.com/wp-content/uploads/2016/08/SDS12652-13-Z-Wave-Command-Class-Specification-N-Z.pdf>

- [23] Hintjens, P.: *Code Connected Colume 1*. iMatrix Corporation, 2013, ISBN 9781481262651.
- [24] Hintjens, P.: *ZeroMQ: Messaging for Many Applications*. O'Reilly Media, 2013, ISBN 9781449334062.
- [25] Novotný, T.: *Identifikace adaptéru*, Únor 2015, [Online; navštívené 10.01.2017] URL https://antdev.fit.vutbr.cz/redmine/projects/adapter/wiki/Identifikace_adaptéru.
- [26] Sigma Designs, I.: *Z-Wave Frequency Coverage*. Prosinec 2012, [Online; navštívené 17.01.2017].
URL http://z-wave.sigmadesigns.com/wp-content/uploads/2016/08/Z-Wave_Frequency_Coverage.pdf

Prílohy

Príloha A

Fotografie zariadení



Obr. A.1: BeeOn adaptér s pripojenou rozširujúcou doskou



Obr. A.2: USB zariadenia pre pripojenie bezdrôtových senzorov – AeoTec Z-Stick USB Dongle, Jablotron Dongle



Obr. A.3: Jablotron zariadenia – AC-88, JA-81M, JA-83M, JA82-SH, JA-83P, TP-82N



Obr. A.4: Z-Wave zariadenia – Aeotec Multisenzor 6, Dlink DCH-Z120, Philio PST02-1C, Popp 123627, Fibaro FGK-107