

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra měření

**Framework pro ovládání technologií budov
s lokální nabídkou služeb**



leden 2017

Diplomant:

Bc. Martin Vystrčil

Vedoucí práce:

doc. Ing. Jiří Novák, Ph.D.

Poděkování

Rád bych poděkoval především vedoucímu diplomové práce doc. Ing. Jiřímu Novákovi, Ph.D. za pravidelné konzultace a věcné připomínky v rámci nich. V neposlední řadě velice děkuji své rodině a přátelům za jejich podporu po celou dobu mého studia.

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací

V Praze dne

.....

Podpis autora práce



ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	Bc. Martin Vystrčil
Studijní program:	Inteligentní budovy
Název tématu česky:	Framework pro ovládání technologií budov s lokální nabídkou služeb
Název tématu anglicky:	Building Technology Control Framework with Local Service Offer

Pokyny pro vypracování:

Seznamte se s funkčními principy existujících systémů pro monitoring a řízení technologií budov. Navrhněte koncepci a strukturu systému zaměřeného na poskytování místně specifických nabídek služeb pro uživatele pohybujícího se budovou tak, aby bylo možné jednoduché ovládání technologií budovy dostupných v místě uživatele. Pozornost věnujte také problematice bezpečnosti a úrovni oprávnění uživatelů. Navrženou koncepci realizujte a na vzorové aplikaci demonstруйте její funkčnost.

Seznam odborné literatury:

- [1] Kocourek, P. Novák, J.: Přenos informace, skriptum ČVUT FEL, 2003
- [2] Merz, H., Hansemann, T., Hübner, Ch.: Automatizované systémy budov, Grada 2007
- [3] IEEE standard 802.15.1 - Bluetooth

Vedoucí diplomové práce: doc. Ing. Jiří Novák, Ph.D.

Datum zadání diplomové práce: 18. ledna 2016

Platnost zadání do¹: 30. září 2017

Doc. Ing. Jan Holub, Ph.D.
vedoucí katedry



Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 18. 1. 2016

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Anotace

Diplomová práce se zabývá návrhem a vývojem frameworku pro ovládání technologií budov s lokální nabídkou služeb v závislosti na aktuální poloze uživatele. V práci je provedena základní rešerše systémů, které jsou v době psaní práce běžně dostupné. V práci je proveden detailní rozbor použitých hardwarových součástí. Detailní popis vyvinutého software a popis principu určování polohy uživatele. V práci je rovněž diskutováno chování systému v reálném testovacím zapojení.

Klíčová slova

lokální nabídka služeb, poloha, ovládání technologií, framework

Summary

The object of this diploma thesis is building technology control framework with local service offer based on the location of user. Basic research of commonly used system was done at the beginning of this thesis. Whole concept of the system is based on this research. There is detailed description of used hardware and software used for such system. In the text there is also description of algorithm used for locating the user. Discussion about qualities of this system tested in real environment is done.

Keywords

local service offer, location, technology control, framework

Obsah

1 Úvod	1
2 Rešerše	3
2.1 Rozdělení systémů	3
2.2 Standardy	5
2.2.1 KNX / EIB	5
2.2.1.1 Komunikační média	6
2.2.1.2 Adresace v síti	6
2.2.1.3 Individuální a skupinové adresy	6
2.2.1.4 Výrobci implementující technologii KNX	8
2.2.2 LonWorks	8
2.2.2.1 Neuron chip	8
2.2.2.2 Komunikační média	8
2.2.2.3 Vysílače	9
2.2.2.4 Adresace v síti	9
2.2.2.5 Výrobci implementující technologii LonWorks	10
2.2.3 CIB	10
2.2.3.1 Adresace v síti	10
2.2.3.2 Výrobci implementující technologii CIB	11
2.3 Shrnutí	11
3 Základní koncepce systému	11
4 Systémový návrh	13
4.1 Lokalizace uživatele	13
4.2 Infrastruktura	14
5 Implementace	15
5.1 HW návrh řešení	15
5.1.1 Řídící počítač	15
5.1.2 Vývojové desky	16
5.1.3 Stabilizátor napětí	17
5.1.4 OneWire sběrnice	17
5.1.5 Sériové rozhraní	18

5.1.6	Bluetooth rozhraní	18
5.1.6.1	Bluetooth modul	19
5.1.7	Vstupně - výstupní rozšiřující modul	19
5.2	SW návrh řešení	20
5.2.1	Embedded linux	20
5.2.2	Qt Framework	21
5.2.2.1	Signály	22
5.2.2.2	Sloty	23
5.2.3	Object Handler	23
5.2.3.1	Přenos objektů	23
5.2.4	Communication Handler	24
5.2.4.1	Bluetooth klient	25
5.2.4.2	TCP/IP klient	27
5.2.4.3	Vytvoření spojení	28
5.2.5	Database Handler	29
5.2.5.1	Inicializace databáze v serverové aplikaci	30
5.2.5.2	Inicializace databáze v GUI aplikaci	31
5.2.6	Serverová aplikace	31
5.2.6.1	Perzistentní úložiště	31
5.2.6.2	Bluetooth server	31
5.2.6.3	TCP/IP server	32
5.2.6.4	Navázání spojení	33
5.2.6.5	Ověření uživatele - LDAP	33
5.2.6.6	Ověření uživatele - lokální	35
5.2.6.7	Práce s identitou uživatele	36
5.2.6.8	Řízení technologie	37
5.2.7	Konfigurační aplikace	37
5.2.7.1	Horní informační panel	37
5.2.7.2	Levý navigační panel	38
5.2.7.3	Pravý panel obsahu	39
5.2.7.4	Panel nastavení vytápění	39
5.2.8	Uživatelská aplikace	41
5.2.8.1	Rozdíly na přihlašovací obrazovce	41
5.2.8.2	Navigační obrazovka	42

5.2.8.3	Obrazovka ovládání termostatu	42
6	Testovací zapojení	43
6.1	Statické měření síly signálu	43
6.2	Dynamické chování systému	44
7	Závěr	46
A	UML diagram ObjectHandler	50
B	UML diagram ObjectHandler pro objekt typu User	51
C	Testované zapojení systému	52
D	Naměřené hodnoty testovaného zapojení systému	53

Seznam obrázků

1	Systém přímého číslicového řízení	3
2	Sběrníkový systém	4
3	Schématické znázornění KNX instalace	5
4	Topologie KNX sběrnice [2]	7
5	Topologie zapojení sítě LON	10
6	Diagram celého systému	12
7	Řídicí deska Cubieboard2	16
8	Bluetooth modul HM-10	19
9	Vnitřní uspořádání KAQV214 [15]	20
10	Vnitřní uspořádání MOC3022 [16]	20
11	Grafické znázornění signál - slot [22]	22
12	Minimální kostra přenášených dat	24
13	Stavový automat pro Bluetooth komunikaci	26
14	Stavový automat pro WiFi komunikaci	27
15	Princip navázání spojení	29
16	Příklad dotazu a odpovědi při navazování komunikace	30
17	Stavový automat TCP/IP serveru	34
18	Schéma LDAP databáze pro testovací účely	35
19	Obsah souboru mapování skupin na povolené operace	36
20	Rozložení přihlašovací obrazovky konfigurační aplikace	38
21	Rozložení obrazovky řízení teploty	40
22	Navigační a obsahová obrazovka uživatelské aplikace	42
23	ObjectHandler abstraktní třída	50
24	UML diagram třídy User	51
25	Testované zapojení systému	52
26	Měřené body v testovaném zapojení	53

Seznam tabulek

1	Základní typy podporovaných přenosových médií	9
2	Výpis možných akcí vykonaných na objektech	24
3	Výchozí hodnoty nastavené v průběhu testování systému	39
4	Omezení pro nastavení teplotních úrovní	40
5	Staticky naměřené hodnoty v bodech dle obrázku 26.	44

1 Úvod

Řídicí systémy jsou v dnešní době běžně dostupné a v řadě velkých budov i implementované. Nicméně v průběhu implementace inteligentního řídicího systému do budovy se velmi často stane, že je systém přetechnizovaný. Taková instalace pak sice nabízí nespočetné možnosti, ale stane se velmi obtížně ovladatelnou. Ovladatelnost systému je ztížena především vysokým počtem možných nabídek a ovládacích prvků, ze kterých je nutné vybrat ten správný.

Tato skutečnost byla podnětem k vypracování diplomové práce s cílem vylepšení této situace tím, že bude docházet k nabídce pouze lokálních prvků instalace. Díky lokální nabídce dojde k razantnímu snížení počtu nabízených prvků a tím k vyšší přehlednosti uvnitř aplikace. Základním stavebním prvkem systému s podporou lokální nabídky je lokalizace uživatele uvnitř budovy.

Základní úvahou je možné dojít k tomu, že rozdělení systému na celky menší než jednotlivé místnosti nemá smysl. V případě velkých kanceláří, takzvaných *open office*, bude rozdělení na místnosti pouze virtuální podle umístění jednotlivých pracovišť. Požadavek na přesnost určení polohy uživatele uvnitř budovy tak není příliš vysoký.

Další důležitou roli při návrhu části systému určeného pro lokalizaci uživatele je výběr technologie, pomocí které bude k lokalizaci docházet. V dnešní době chytrých telefonů je tlak na to, aby bylo možné co nejvíce technologií budovy ovládat právě pomocí mobilních telefonů. Prakticky se již neseťkáme s tím, že by některý z výrobců těchto systémů vyvíjel vlastní zařízení, které by bylo určeno k přístupu k těmto technologiím. Aby byl systém konkurenceschopný, je nutné využít některou z již existujících technologií, kterými běžné chytré telefony disponují. Další nespornou výhodou chytrých telefonů je fakt, že lidé si již zvykli mít telefon neustále při sobě a tak nedochází ke snížení uživatelského komfortu.

Již při návrhu systémů, které umožňují řízení budov je nutné počítat alespoň s dvěma základními body bezpečnosti. Prvním z nich je uvážení toho, zda jsou si všichni uživatelé rovni. Obvyklá situace vypadá tak, že existuje hierarchická struktura, na jejímž vrcholu je správce systému a ve spodních patrech jsou jednotliví uživatelé. Každému patru takové struktury je možné přidělit jistá oprávnění. Uživatelům s nižšími právy pak není umožněno přidávat případně odebírat nové objekty v systému, ale pouze pracovat s již připravenými objekty.

Druhým bodem je zabezpečení vlastního přenosu příkazů a dat. Přenos dat ve většině případů probíhá prostřednictvím bezdrátového spojení pomocí technologie WiFi. Obsah komunikace (například přihlašovací údaje) je možné odposlechnout a případně využít pro potřeby útoku na systém. Je proto nutné veškerý přenos dat šifrovat pomocí asymetrických šifer. Tento způsob šifrování je velmi rozšířený a je považován za nejvíce bezpečný.

V kapitole 2 je provedena základní rešerše dostupných technologií využívaných k řízení inteligentních budov. Rešerše je zaměřena především na získání základního povědomí o standardních technologiích, které jsou běžně využívány. Nejsou zde diskutována jednotlivá řešení, která výrobci nabízejí, neboť výrobců je mnoho a technologie se neustále vyvíjejí. Nicméně u každé z uvedených technologií je uveden alespoň jeden představitel, který danou technologii implementuje. V této kapitole je rovněž zmíněna důležitost standardizace komunikačních protokolů.

V kapitole 3 jsou představeny základní konstrukční prvky systému, které jsou běžně využívány. Navíc je zde diskutována základní koncepce navrhovaného řešení, včetně zavedení základních pojmů.

V kapitole 4 je koncepce systému dále rozvedena do větších detailů včetně představení principu lokalizace uživatele a popisu použité technologie. Dále je v této kapitole popsáno základní zapojení systému, na kterém byla otestována jeho funkčnost.

V kapitole 5 je popsána implementace vzorového systému, který poskytuje nabídku lokálních služeb. Je zde podrobně rozebrán návrh hardwarové platformy, na které je realizována případová studie. Ve druhé části kapitoly je podrobně popsána softwarová část projektu. Popis softwarové části projektu je rozdělen na jednotlivé stavební bloky tak, jak byl systém navržen. Systém je realizován tak, aby potvrdil, že lze pomocí běžně dostupných komponent dosáhnout lokalizace uživatele a poskytnutí lokální nabídky.

V kapitole 6 je popsána testovací instalace a jsou zde shrnuty výsledky měření. Spolu se shrnutím dosažených výsledků jsou zde rovněž diskutována možná vylepšení, která by vedla ke zvýšení přesnosti při určování polohy.

V závěrečné kapitole jsou shrnuty základní kroky, které vedly k úspěšné realizaci frameworku pro ovládání technologií budov s lokální nabídkou. Závěrečná kapitola je navíc doplněna o další budoucí kroky, které by vedly k rozšíření systému.

2 Rešerše

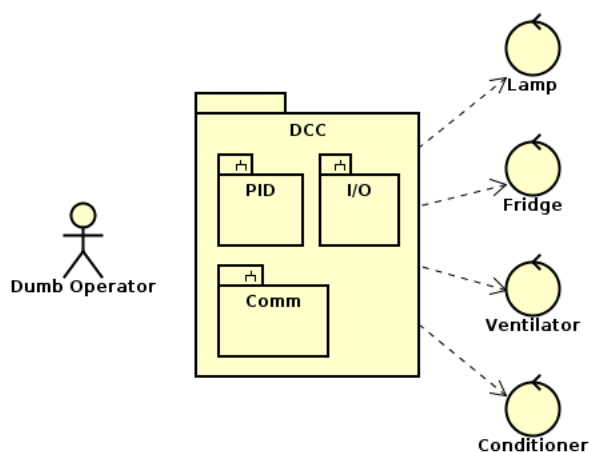
V následujících kapitolách je popsáno několik systémů sloužících pro automatizaci infrastruktury budov. V posledních letech je téma automatizace infrastruktury velmi diskutované. Velký rozmach nastal především díky velkému rozšíření sítě internet a k tomu navazujících služeb. Kvůli vysokému zájmu z řad výrobců, ale i z řad zákazníků dochází k neustálému rozšiřování stávajících a vzniku nových automatizačních systémů.

Na trhu lze nalézt řadu výrobců, kteří se pokouší prodat svůj produkt, který není žádným způsobem standardizován. V případě, že je takový produkt nainstalován, dojde dříve nebo později k potřebě jeho rozšíření, případně údržby. Po několika letech už nemusí jeho mateřská firma existovat a kvůli chybějícím standardům nebude možné systém dále spravovat. Takový systém se pak stává nepoužitelným a nutně dojde k jeho kompletní náhradě.

2.1 Rozdělení systémů

Hlavním kritériem, podle kterého se řídicí systémy dělí, je jejich fyzické připojení k prvkům infrastruktury.

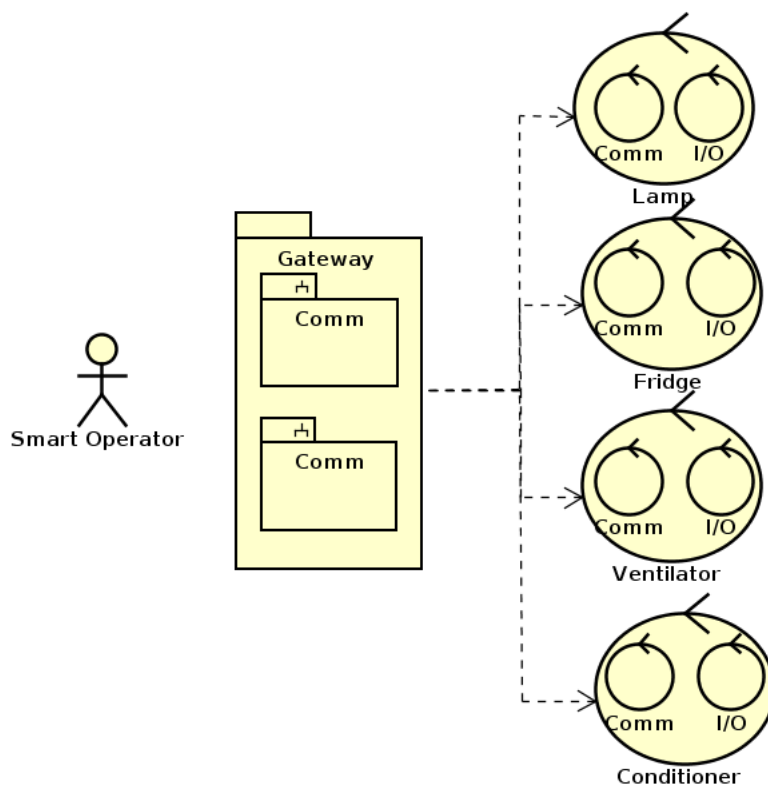
Prvním (historicky jednodušším) typem je přímé číslicové řízení (Direct Digital Control) [1]. Zprvu se jednalo o jednoduchá zařízení, která nedisponovala velkým výkonem, ale byla určena k ovládání určitého (typicky jednoho) procesu. Parametry ovládaných procesů byly předem známe a po uvedení systému do provozu už se příliš neměnily. Vzhledem k absenci dalších komunikačních rozhraní nebylo možné vytvářet složitější sítě obsahující například stovky spojených zařízení. Rovněž rozšiřitelnost takových systémů nebyla bez větších zásahů možná.



Obrázek 1: Systém přímého číslicového řízení

Jak je patrné z obrázku 1, jednoduchý systém vyžaduje vedení mnohonásobně vyššího počtu spojovacích prvků. Rovněž přidání libovolného dalšího rozšíření se neobejde bez přidání dalšího spoje. Toto je nevyhovující především s ohledem na pracnost instalace, kdy kabeláž bývá uschována pod omítkou, případně v podhledu. Instalace každého dalšího spoje s sebou tak nese nejen náklad v podobě ceny spoje. Rovněž je nutné zvážit i nepříjemnosti v podobě omezení uživatelského komfortu a to po celé délce trasy nového spoje.

Druhým, pokročilejším typem jsou systémy sběrnice. Jak již název napovídá, jedná se o systémy, které jsou vzájemně propojeny sběrnici, pomocí které mezi sebou komunikují. Nespornou výhodou tohoto typu systémů je vysoká míra modularity, neboť je možné sběrnici téměř libovolně rozšiřovat. U každého typu sběrnice ale existují maximální limity připojených zařízení, které je nutné dodržet. Rozšiřitelnost je vykoupena použitím výkonnějších součástí v koncových prvcích (mikroprocesorů), což má za následek vyšší ceny sběrnicevých zařízení.



Obrázek 2: Sběrnicevý systém

Z obrázku 2 je patrné, že komunikační médium je sdílené mezi všemi řízenými procesy. Instalace dalšího procesu tedy nebude vyžadovat přidání spoje až k centrálnímu prvku, ale pouze k nejbližšímu již instalovanému. Díky tomuto faktu je možné uspořit nejen materiál a čas strávený instalací, ale také minimalizovat snížení uživatelského komfortu na minimum.

Z výše popsaného rozdělení jasně vyplývají výhody sběrníkových systémů nad těmi přímo řízenými a proto jsou téměř všechny nové instalace navrhovány jako sběrníkové [1].

2.2 Standardy

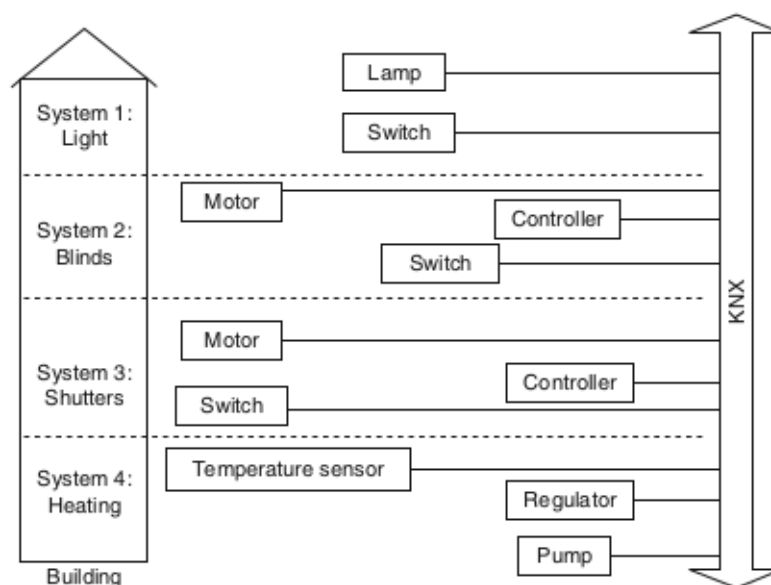
Důležitost standardizace již byla popsána v úvodní kapitole. V následujících kapitolách budou detailně popsány tři běžně používané standardy. Jedná se o standard *KNX*, standard *LonWorks* a standard *CIB*. Kromě těchto dvou základních standardů se můžeme setkat s řadou menších standardů, ty však často nepokrývají celou škálu automatizace. Patří mezi ně například standard *Dali*, který je zaměřen na řízení osvětlení.

U každého ze standardů je uvedena norma, v níž je možné nalézt jeho kompletní specifikaci. V tomto textu jsou uvedeny pouze zásadní informace, díky kterým je možné spatřit hlavní rozdíly mezi nimi.

2.2.1 KNX / EIB

Je průmyslový komunikační systém, který se v systémové technice budov používá pro síťové informatické spojení zařízení. Implementace je přizpůsobena elektrotechnické instalaci, čímž jsou zajištěny funkce a automatizované procesy v budově [1].

Zapojení KNX systému v budově je schematicky znázorněno na obrázku 3. Z obrázku je patrná sběrníková struktura s centrálním rozvodem.



Obrázek 3: Schématické znázornění KNX instalace

2.2.1.1 Komunikační média

Komunikační protokol je popsán v mezinárodním standardu ISO 14543. Standardem je podporována celá řada přenosových médií. Patří mezi ně kroucený pár (označovaný jako KNX.TP) nebo například rádiový přenos označovaný jako KNX.RF. Díky široké škále podporovaných přenosových cest je možné systém KNX napojit i na již existující rozvody, ať už se jedná o kroucené páry nebo přenos po silové části vedení. Zatímco přenos po médiu, které je vyhrazeno čistě pro potřeby systému KNX (kroucený pár) probíhá téměř bez chyb, tak v případě sdíleného média (rádio, silnoproudá instalace) může docházet (a dochází) k silnému rušení.

Napájení všech účastnických stanic na sběrnici je vyřešeno pomocí zdroje připojeného přímo na sběrnici. Každé zařízení tedy stačí ke sběrnici připojit pouze jedním párem vodičů, odpadá potřeba externího napájení. Výše popsaný způsob napájení je možný pouze v případě, že je využita některá z variant využívající fyzického přenosového kanálu. V případě využití rádiového přenosu je stále nutností zajistit napájení jednotlivých prvků sběrnice.

2.2.1.2 Adresace v síti

KNX síť je rozdělena do tří úrovní. Do nejvyšší (páteřní) úrovně se připojují oblastní spojky, které slouží k propojování jednotlivých oblastí. V případě využití více než jedné oblasti v instalaci je vhodné zavést logické dělení, například jedna oblast může odpovídat jednomu patru budovy. Maximální počet adresovatelných oblastí je 15.

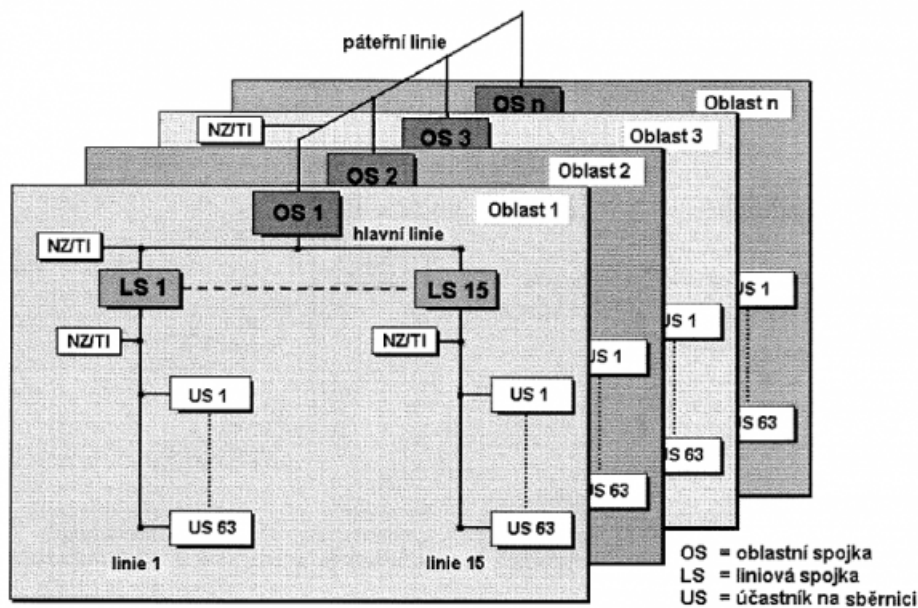
Druhým, nižším patrem hierarchie jsou linie, ke kterým jsou již připojovány jednotlivé komponenty systému. Stejně jako je možné představit si oblast jako jedno patro budovy, je možné si linii představit jako automatizační prvky stejného typu (zásuvky / žaluzie / vypínače). Maximální počet adresovatelných linií je 15, přičemž linie jsou mezi sebou propojovány pomocí liniových spojek.

Každá z linií může být dále rozdělena na 4 segmenty. Segmenty jsou k linii připojeny pomocí liniových opakovačů. Maximální počet prvků připojených do jednoho segmentu je 64. Maximální možný počet prvků zapojených do systému je pak $2^{16} = 65536$ [2].

Hierarchické rozdělení systému je schematicky znázorněno na obrázku 4.

2.2.1.3 Individuální a skupinové adresy

Každý prvek připojený do sítě KNX musí mít přiřazenu unikátní (individuální) adresu. Celá adresa jednoho prvku připojeného ve sběrnici se skládá ze tří čísel oddělených tečkou - oblast.linie.prvek (například 1.1.2) [1].



Obrázek 4: Topologie KNX sběrnice [2]

Pokud jsou v individuální adrese prvku některá pole rovna nule, pak se jedná o speciální adresy. Například adresa 1.12.0 je přiřazena liniové spojce v oblasti 1, linii 12. Podobně je tomu u adresy 2.0.0, která označuje oblastní spojku umístěnou ve druhé oblasti [1].

Pro zjednodušení programování rozsáhlých systémů byl ve standardu zaveden speciální typ adres *skupinové adresy*. Skupinové adresy lze poznat podle způsobu jejich zápisu, při němž se využívá lomítek namísto teček, například 1/2, případně 1/2/3. Při zavádění skupinových adres je vhodné logické rozdělení systému do skupin, například skupina 1/ bude použita pro řízení osvětlení, zatímco skupina 2/ bude použita pro řízení žaluzií. Za oddělovacím lomítkem pak může následovat kupříkladu označení místnosti. Skupinová adresa 1/1 by pak řídila osvětlení v místnosti obývací pokoj, zatímco skupinová adresa 1/2 by řídila osvětlení v místnosti kuchyně [1].

Ve standardu je zahrnuta podpora pro dvouúrovňové, případně tříúrovňové adresování. Pokud se ale projektant rozhodne pro jeden ze způsobů, zvolený způsob musí být použit v celém systému. Při použití dvouúrovňové adresace (častější využití) jsou 4 bity využity pro adresaci hlavní skupiny a 11 bitů je použito pro adresaci podskupiny. Číselně je tedy použitelných 16 hlavních skupin a až 2048 podskupin. Podle doporučení z [1] je vhodné skupinu 0 používat pro signalizaci alarmů.

2.2.1.4 Výrobci implementující technologii KNX

Standard KNX je jedním z nejršíreji používaných standardů. S tímto souvisí také to, že výrobce řídicích systémů s podporou KNX je možné najít v nejrůznějších zemích. Jako příklady lze uvést španělskou firmu *Futurasmus*, anglickou firmu *Livinghometech*, případně čínskou firmu *HDL*, která má pobočku i v České Republice.

2.2.2 LonWorks

Označení LonWorks v sobě zahrnuje celou řadu podružných zařízení. Kořenová část názvu této technologie - Lon znamená Local operating network [3]. Volně přeloženo jako lokální síťové služby.

Samotný komunikační protokol označovaný jako LonTalk byl vyvinut společností Echelon v roce 1989 [5] a je popsán ve standardu ISO/IEC 14908-1. Jedná se o protokol specificky navržený pro řídicí účely, který slouží pro komunikaci mezi dvěma koncovými body (peer to peer).

2.2.2.1 Neuron chip

Hlavním stavebním prvkem technologie LonWorks jsou mikrokontroléry nazývané *Neuron Chip*. Každý mikrokontrolér je složen ze třech osmibitových procesorových jednotek, fyzické paměti a referenční implementace LonTalk protokolu. Každý vyrobený čip navíc obsahuje světově unikátní identifikační číslo - *Neuron ID* [3].

Dvě procesorové jednotky se starají o komunikaci prostřednictvím LonTalk protokolu. Třetí procesorová jednotka je dostupná pro potřeby vlastní aplikace. Kompatibilita zapojení čipů od různých výrobců je zajištěna díky tomu, že každý Neuron Chip v sobě již implementuje kompletní rozhraní pro komunikaci prostřednictvím LonTalk sítě [3].

2.2.2.2 Komunikační média

Podobně jako systém KNX i LonWorks lze provozovat na řadě fyzických médií. Tabulka 1 je převzatá z [3] a uvádí základní typy podporovaných fyzických médií.

Jak je z tabulky patrné, přenosové rychlosti jsou nižší (řádově stovky kbps), stále ale naprosto vyhovující pro přenos krátkých zpráv. V případě využití typu TP/FT-10 lze fyzické zapojení realizovat bez jakýchkoliv omezení v topologii. Zapojení lze provést do hvězdy, lineární sběrnice, případně i do kruhu [3]

Označení vysílače	Médium	Přenosová rychlost	Standard
IP-852	IP tunelování	N/A	ISO/IEC 14908-4
PL-20 (A/C/N)	Silové rozvody	3.6 - 5 kbps	ISO/IEC 14908-3
TP/FT-10	Kroucená dvojlinka	78.6 kbps	ISO/IEC 14908-2
TP/XF-1250	Kroucená dvojlinka	1.25 Mbps	ISO/IEC 14908-2

Tabulka 1: Základní typy podporovaných přenosových médií

2.2.2.3 Vysílače

Každý neuron čip obsahuje referenční implementaci protokolu LonTalk. Aby byla zajištěna modulárnost systému, došlo k oddělení vlastního zpracování dat (uvnitř neuron čipu) a vlastní komunikaci se sběrnici (prostřednictvím vysílače).

Díky tomuto rozdělení je možné v jedné síti využívat čipy a vysílače různých výrobců. Důležité je pouze zachování stejného typu vysílače / přijímače na obou stranách.

2.2.2.4 Adresace v síti

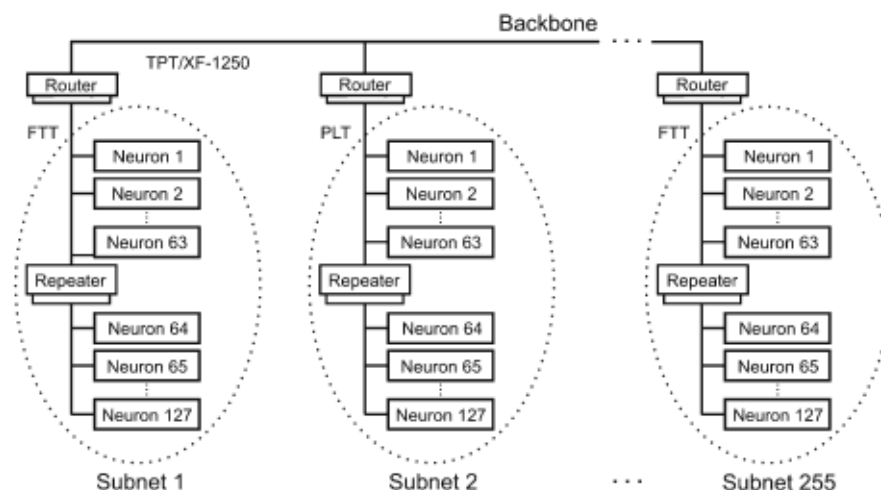
Podobně jako v případě instalace KNX, i při vytváření LON sítě je možné její dělení. Při rozdělení instalace se používá pojmu jako podsít', síť, případně doména. Hierarchicky je síť znázorněna na obrázku 5.

Nejjednodušší instalace, při nichž je zapojeno maximálně 128 účastníků, je možné provozovat jako samostatné síť. V případě využití vysílačů napájených ze sběrnice je počet omezen na 64. Rozšíření takovéto sítě na 128 prvků je možné s využitím opakovačů. Pokud jsou pro komunikaci využity vysílače s vlastním zdrojem napájení, je maximální počet účastníků zvýšen až na 128.

V případě, že je nutné zapojit více účastníků, je již zapotřebí rozdělit síť na více podsítí. Jako oddělovacích prvků se využívá zařízení nazývaných *routery*. Směrování komunikace probíhá tak, že pokud nebyl telegram doručen do podsítě daného routeru, je přeposlán do dalšího segmentu spojené sítě. Routery navíc podporují spojení více podsítí postavených na různých technologiích do jedné sítě.

Spojením více sítí vznikají největší adresovatelné celky *domény*. V jedné doméně je možné spojit až 255 podsítí (routerů). V případě, že by ani tento počet nebyl dostačující, je možné spojit další dvě domény pomocí výkonného routeru.

Nejvyšší počet připojených uzlů v jedné doméně je $255 * 127 = 32385$. Maximální velikost sítě pak může vzniknout spojením dvou domén, čímž se počet uzlů zdvojnásobí.



Obrázek 5: Topologie zapojení sítě LON

2.2.2.5 Výrobci implementující technologii LonWorks

Jedná se o druhého z největších výrobců, který je rozšířený především v zámoří. Mezi hlavní výrobce patří například americká společnost *Allen-Bradley*, německá společnost *Wilo*, případně francouzská společnost *Schneider*. Dvě poslední jmenované společnosti mají navíc pobočky v České Republice.

2.2.3 CIB

Technologie, potažmo standard CIB (Common Installation Bus) je vlastnictvím české firmy ELKO ep. Jedná se podobně jako u dvou výše popsaných standardů o sběrniceovou technologii. Zapojení sítě je v podstatě libovolné, vyjma zapojení do kruhu. Komunikace probíhá buď bezdrátově, případně prostřednictvím krouceného páru.

2.2.3.1 Adresace v síti

Na sběrnici je vždy nutné připojit hlavní řídicí jednotku, která jednak řídí komunikaci na sběrnici a druhá se stará o její napájení. Maximální počet připojených uzlů na jedné větvi je omezen na 32. Jednotky jsou proto vybaveny alespoň dvěma oddělenými větvemi. Další větve sběrnice je možné připojit pomocí rozšiřujících modulů [7].

Jako příklad lze uvést centrální jednotku CU3-01M, která disponuje dvěma svorkami pro připojení k CIB sběrnici. Jednotka navíc umožňuje rozšíření až o dalších 8 externích komunikačních jednotek, z nichž každá externí jednotka disponuje rovněž dvěma svorkami pro připojení ke sběrnici. Celkově lze tedy zapojit až $2 * 32 + 8 * 2 * 32 = 576$ uzlů [8].

2.2.3.2 Výrobci implementující technologii CIB

Standard CIB je v současné době implementována pouze v systému iNels, jež je vyvíjen stejnou společností jako vlastní technologie. Jiní výrobci implementující CIB nebyli nalezeni.

2.3 Shrnutí

Všechny výše popsané systémy poskytují dostatečné rozhraní pro ovládání inteligentní elektroinstalace a kvalitní uživatelské rozhraní potřebné k tomu, aby je bylo možné efektivně využívat. Díky standardům, které jsou propracovány do nejmenších detailů a již řadu let využívány je možné ve vývoji pokročit kupředu a více se zaměřit na uživatelský komfort.

Všechny systémy je možné škálovat do obrovských celků. V takto rozsáhlých systémech pak není z uživatelského hlediska snadné najít potřebné objekty. Může tak nastat situace, kdy celá budova bude mít implementovanou podporu pro automatizaci, ta však nebude využívána z důvodu uživatelsky nepohodlného rozhraní.

Nabízí se zde řešení v podobě nabídky pouze té části automatizace, kterou je možné v daném místě (místnosti) využít. Uživatel tak nebude nucen vyhledávat konkrétní objekt mezi stovkami dalších, ale seznam objektů se omezí na desítky. Během řešení se nepodařilo nalézt implementaci takovéto funkce u již nabízených systémů. Skutečnost, že taková funkce není zatím implementována může být způsobena například i tím, že by nebylo možné takovéto řešení realizovat.

3 Základní koncepce systému

Systémy pro automatizaci domácí infrastruktury se skládají ze tří základních stavebních prvků. Prvním, nejdůležitějším prvkem je vždy počítač s řídicím softwarem, který se stará o vlastní automatizaci systému. Aby bylo uživateli umožněno jednoduchým způsobem vstoupit do systému a provádět v něm změny, je nutné systém vybavit o další podpůrný software. V případě této realizace se jedná o dvě aplikace, které je možné spustit na mobilním telefonu, případně na standardním počítači.

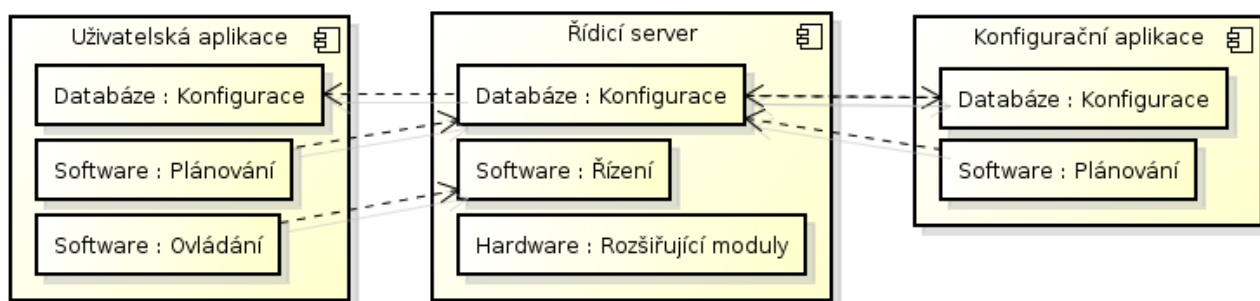
Přehledové schéma celého systému je zobrazeno na obrázku 6. Centrálním prvkem automatizace je *řídicí server*, který kromě celkové konfigurace nabízí funkční řízení všech automatizovaných celků. Kromě software pro automatizaci, je na řídicím serveru spuštěn i software určený pro komunikaci s uživatelskou nebo konfigurační aplikací.

Řídicí server je centrálním prvkem automatizace, který se stará nejen o dohled nad celým systémem, ale také o komunikaci s uživatelem. V případě rozsáhlých budov je velmi vhodné nasadit takovýchto prvků více. Nebude pak docházet k zahlcení jednoho centrálního prvku, ale

dojde k rozložení zátěže mezi jednotlivé body. U takového zapojení vznikají částečně samostatné jednotky. V případě výpadku nedojde k omezení dostupnosti veškeré technologie, ale pouze k jejímu lokálnímu omezení.

Uživatelská aplikace je zamýšlena pro běh na mobilních terminálech a uživateli nabízí jednoduché rozhraní pro vzdálené ovládání a dohledování dostupných částí systému. Typickými akcemi, jež je možné vykonávat, je ovládání automatizačních celků (osvětlení, topení, ventilace). Navíc umožňuje přidávání plánovaných akcí jako je nastavení požadované teploty v daný čas, případně automatické stažení stínící techniky.

Konfigurační aplikace je určena pro celkovou správu systému, která zahrnuje výše popsané vzdálené ovládání a nastavení časových událostí. Navíc disponuje možnostmi nastavení nových objektů do řízeného systému.



Obrázek 6: Diagram celého systému

Na komerčním trhu je dostupná celá řada systémů od různých dodavatelů. Každý z nich řeší potřebu konfigurační aplikace po svém. Řada dostupných aplikací určených pro konfiguraci domácí automatizace je založena na principech grafického programování. V případě grafického programování se umísťují grafické elementy na plochu představující daný objekt, tyto elementy se poté graficky spojují čarami představujícími logické spoje. V případě rozumného dělení systému a stálého udržování pořádku na ploše jsou tyto systémy výhodné. Pokud ale operátor umísťuje objekty chaoticky, dojde v krátkém období ke ztrátě přehledu nad projektem. Z toho důvodu byla v konfigurační aplikaci vybrána metoda textového zadávání konfigurace.

Díky textové konfiguraci je pořádek v aplikaci udržován automaticky a je vždy stejný. Systém si neklade za cíl být obecný pro všechna použití, ale zprostředkovat jednoduché rozhraní pro všechny uživatele a být dostatečně obecný na to, aby dovolil nastavení všech běžně používaných sekvencí.

Hardwarová platforma je založena na Linuxovém jednodeskovém počítači, který má dostatečný počet vstupních a výstupních pinů a zároveň s nimi nabízí i různá komunikační rozhraní. Základní deska systému není připojena ke sběrnicevým systémům, nicméně budoucí rozšíření o

připojení ke sběrnici je návrhem hardware podporováno.

Navržený komunikační protokol je založen na přenosu zpráv ve formátu XML. Tento formát byl vybrán především s ohledem na možnosti jeho budoucího využití. Veškerá data, která jsou formátována pomocí XML, je možné velmi snadno využít i mimo poskytovanou softwarovou architekturu. Nabízí se například možnost sběru aktuálních teplot z jednotlivých místností a jejich následné vyhodnocení pomocí externích aplikací.

Protokol je navržen s ohledem na bezpečnost svých uživatelů. Veškerá komunikace je proto šifrována pomocí protokolu TLS1.2. Následné ověření přístupových údajů je porovnáno proti nastavenému LDAP serveru. Komunikace mezi řídicím serverem a LDAP serverem je šifrována.

4 Systémový návrh

Ovládání menší budovy (například rodinný dům) nemá smysl optimalizovat z hlediska usnadnění navigace v aplikaci. Rodinný dům má typicky do 10 místností, nalezení té správné je tedy snadné a lehce proveditelné. Naproti tomu v případě rozsáhlé instalace (například administrativní budova / škola), kde mohou být stovky místností, se navigace v aplikaci stane obtížnou až otravnou velmi snadno.

Stávající systémy pracují tak, že se uživatel připojí na předem známou adresu (server), kde běží příslušná aplikace. Z této aplikace se stáhnou potřebná data tak, aby bylo možné z uživatelova zařízení ovládat příslušný automatizační celek.

Vylepšení současného způsobu připojení spočívá v implementaci systému, který místo uživatele rozhodne, k jaké adrese se připojit a jaká data případně stáhnout. Aby bylo možné implementovat vylepšení stávajících systémů, je nutné znát alespoň přibližnou polohu uživatele.

4.1 Lokalizace uživatele

Lokalizovat uživatele jako živou osobu uvnitř budovy není se současnou technologií proveditelné. Lokalizaci je nutné omezit na elektronický přístroj, který bude mít uživatel neustále u sebe. V dnešní době však má většina lidí alespoň jeden chytrý mobilní telefon, který u sebe neustále nosí. Z nich pak mohou být tyto systémy řízeny, a tak navrhované řešení nezavádí žádné dodatečné komplikace.

Systém se tedy omezí na lokalizaci chytrého mobilního telefonu uvnitř budovy. Vzhledem k tomu, že mobilní telefon je přenosné zařízení, je nutné lokalizaci provést pomocí některé z bezdrátových technologií. K těmto technologiím, které se v současné době u chytrých telefonů nabízejí zcela samozřejmě, patří WiFi a Bluetooth. U některých dražších modelů se pak nabízí také NFC rozhraní.

Rozhraní NFC (*Near Field Communication*) je určeno pro pomalý přenos na velmi malé vzdálenosti. Řádově se jedná o přenosy na vzdálenost 10 - 15 cm [9], přesné specifikace vycházejí z normy ISO/IEC 18092. Takto krátká komunikační vzdálenost je pro potřeby navrhovaného systému nedostatečná, není proto možné tohoto rozhraní využít.

Rozhraní WiFi (*Wireless Fidelity*) je určeno pro rychlý přenos dat na vzdálenost řádově desítky až stovky metrů. Technologie je popsána ve standardu IEEE 802.11. WiFi technologie je běžně využívána pro vysokorychlostní připojení k internetu, a proto je velmi pravděpodobné, že bude tato síť uvnitř budov již implementována. Jako řešení lokalizace by se tedy nabízelo využití již existující WiFi infrastruktury. Jakkoliv se toto řešení může zdát výhodné, zejména s ohledem na již existující infrastrukturu, bylo by nutné vytvořit přesný model budovy spolu s rozmístěním vysílačů uvnitř. Poté by se dalo za pomoci triangulace mezi viditelnými WiFi vysílači určit přibližnou polohu. Nicméně sestavení takového modelu a následné otestování celého systému by vyžadovalo nezanedbatelně vysoké úsilí.

Při navrhování WiFi sítě jsou navíc kladeny přesně opačné požadavky, než které jsou vhodné pro lokalizaci uživatele. Hlavním požadavkem při vytváření bezdrátové sítě je použití co nejmenšího počtu přístupových bodů k dostatečnému pokrytí co největší plochy [10].

Rozhraní Bluetooth bylo vyvíjeno s cílem umožnit přenosy na kratší vzdálenost (nižší desítky metrů), bez potřeby složitého vytváření infrastruktury [11]. Standard tedy už ve své základní podobě podporuje vytváření malých sítí o několika málo účastnících (piconet / scatternet).

V roce 2010 se standard navíc rozšířil o část *Bluetooth Low Energy*. Technologie BLE není zpětně kompatibilní s technologií klasického Bluetooth. Každý z výrobců Bluetooth čipů se proto musí již při výrobě rozhodnout, zda podporovat oba standardy, nebo pouze jeden z nich [12]. Mezi hlavní výhody BLE rozšíření patří především snížení spotřeby modulů a následně zkrácení odezvy při navazování komunikace. Nevýhodou je snížení přenosové rychlosti z řádově 2Mbit/s na 0.2Mbit/s [12].

Díky tomu, že se v současné době technologie Bluetooth nevyužívá jako hlavní informační kanál, je možné její využití k jiným účelům. Lze si tedy představit takové zapojení, kdy dojde k optimálnímu pokrytí míst, ve kterých se uživatelé pohybují bez snížení komfortu.

Pro lokalizaci uživatele uvnitř budov bude z výše popsaných důvodů použito bezdrátové rozhraní Bluetooth s podporou Low Energy standardu.

4.2 Infrastruktura

Pro účely testování byl systém zapojen v bytě o velikosti 2+kk. Systém byl instalován podle předpokládané instalace uvnitř administrativních budov. Nainstalovány byly dva servery tak,

aby bylo možné správně testovat poskytování lokální nabídky systémem.

Schéma instalovaného bytu spolu se znázorněným rozmístěním řídicích serverů je umístěno v příloze na obrázku 25. Vzhledem k tomu, že v době testování byly dostupné pouze dva servery, bylo možné rozlišit pouze dvě skupiny uživatelů. Předpokládané výsledky jsou takové, že do první skupiny budou zařazeni uživatelé, kteří se nacházejí buď v prostoru kuchyně, případně obývacího pokoje. Do druhé skupiny budou zařazeni uživatelé, kteří se nacházejí v prostoru pokoje.

Ve schématu zapojení můžeme vidět, že místo kolem vstupu do bytu, případně na chodbě vedoucí do obou pokojů, není přímo pokryto žádným ze serverů. K tomuto došlo v důsledku zapojení pouhých dvou serverů, v případě reálné instalace by se v tomto místě nacházel další, třetí server.

5 Implementace

V následující části textu budou popsány kroky vedoucí k implementaci inteligentního systému automatizace. Hardwarové komponenty systému byly vybrány tak, aby umožnily rychlou a bezproblémovou kompletaci. Softwarová část je postavená tak, aby s sebou nesla co nejmenší závislost na zvoleném hardwarovém řešení. Díky této nezávislosti je do budoucna možné kdykoliv vyměnit hardwarovou platformu téměř bez zásahů do softwaru.

5.1 HW návrh řešení

Hardware řídicího serveru je rozdělen do dvou desek. První z nich je vybavena konektory pro připojení řídicího počítače, připojení externího zdroje napětí a dvěma konektory pro budoucí rozšíření. Druhá z nich je vybavena vstupně / výstupními svorkami, za pomoci kterých je možné řídit standardní prvky elektroinstalace.

Rozdělení do dvou desek je výhodné nejen z hlediska místa, které deska zabírá, ale zároveň slouží i k logickému rozdělení funkcí. V cílové instalaci tak může být použita například pouze hlavní deska, která nese řídicí počítač a rozšiřující konektory. V případě využití pouze jedné poloviny systému tak dojde ke snížení potřeby místa pro instalaci při zachování všech funkcí, vyjma přímého ovládání vstupně / výstupních portů. V případě potřeby je možné systém rozšířit o druhou desku, která nese právě tyto vstupně / výstupní svorky.

5.1.1 Řídicí počítač

Základním stavebním prvkem každého systému je procesorová jednotka. Již při návrhu je nutné stanovit priority systému tak, aby dokázal splnit všechny základní požadavky. Zároveň

by se ale nemělo jednat o centrální jednotku, která by svou cenou degradovala celý systém z finančního hlediska.

Centrální jednotka může být buď vybudována na základně libovolného mikrokontroléru, kterých jsou v nabídce stovky od různých výrobců, nebo je možné využít některý z vývojových kitů, v nichž je mikrokontroler včetně vybraných periférií připojen na základní desku. U dostatečně výkonných mikrokontrolérů je často připojen síťový konektor, případně USB sběrnice.

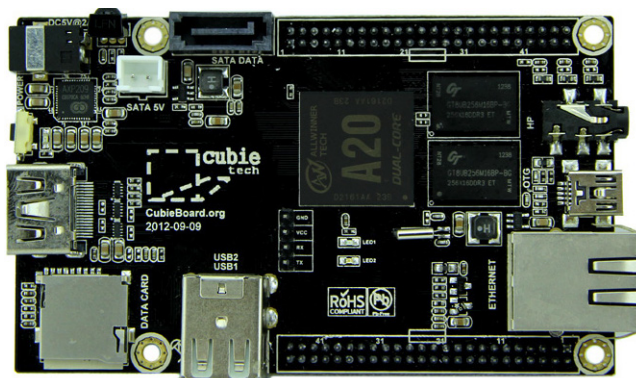
Takováto deska pak obsahuje také vstupně - výstupní piny, pomocí kterých lze tuto desku zapojit do vyvíjeného systému. Díky takové přípravě již od výrobce je možné urychlit čas nutný k vývoji základního HW vyvíjené aplikace.

5.1.2 Vývojové desky

Vývojové desky jsou nabízeny v celém výkonovém spektru. Zatímco nejméně výkonné základní desky nemají podporu pro přístup k síťovému rozhraní, ty výkonnější již nabízejí nejenom přístup k síťovému rozhraní, ale navíc i podporu pro běh operačního systému.

Pokud by došlo k výběru základní desky, která nezahrnuje podporu pro běh operačního systému, bylo by nutné vyvinout, případně získat základní knihovní funkce. Pod pojmem základních funkcí se rozumí především přístup k síťovým funkcím, případně ukládání konfiguračních dat do perzistentní paměti. Veškeré knihovní funkce by pak byly vyvinuty na míru danému mikrokontroléru a tudíž bez úprav nepoužitelné na jiné platformě.

V případě výběru základní desky s podporou běhu operačního systému není nutné vyvíjet knihovní funkce zajišťující základní komunikaci. Vyvíjený software bude navíc psán za pomoci obecných knihovních funkcí, které daný operační systém zahrnuje. Výsledný software bude tedy možné využít na libovolné vývojové desce s podporou stejného operačního systému. Podpora pro běh operačního systému je natolik zásadní, že výběr probíhal pouze mezi základními deskami s jeho podporou.



Obrázek 7: Řídicí deska Cubieboard2

Výběr hlavní řídicí desky je jedním z klíčových rozhodnutí, jež je nutné provést. Hlavním požadavkem byl počet vstupně/výstupních pinů a dalších komunikačních rozhraní na desce dostupných. Vybrána byla deska vyvinutá firmou *Cubietech* s názvem Cubieboard2. Deska disponuje 96 vyvedenými piny, z nichž lze převážnou většinu využít jako binární vstupy a výstupy, případně jim přiřadit speciální funkce. Speciální funkcí se kupříkladu rozumí nastavení dvou pinů tak, aby se tvářily jako I2C rozhraní. Celkově deska disponuje 4 x rozhraním *UART* a 1 x rozhraní *I2C*. Z dalších rozhraní, které jsou na desce připojeny lze jmenovat například ethernet, USB, případně grafické HDMI.

V realizovaném případě je zapojeno 10 pinů jako vstupy a 10 pinů jako výstupy. Rozhraní I2C je připojeno prostřednictvím převodníku na OneWire sběrnici (popsáno v kapitole 5.1.4). Dvě sériová rozhraní jsou vyvedena na externí svorkovnici pro budoucí využití (popsáno v kapitole 5.1.5), jedno sériové rozhraní je vyhrazeno pro komunikaci s bluetooth modulem (popsáno v kapitole 5.1.6). Rozhraní ethernet slouží pro připojení k síti, ostatní rozhraní zůstaly v systému nevyužity.

Na desce je osazen dvoujádrový procesor založený na architektuře ARM s maximální frekvencí 1.2GHz [14]. Jedná se o architekturu procesorů speciálně vyvinutou pro využití v zařízeních s nízkou spotřebou energie. Procesor je doplněn o 1024MB sdílené paměti RAM. Deska je navíc osazena 4GB paměti NAND, jíž je možné využít pro uložení operačního systému.

5.1.3 Stabilizátor napětí

Pro běh základní desky je nutné přivést napětí 5V, odebíraný proud se pohybuje mezi 500 až 2000 mA [14]. Při zapojení do infrastruktury automatizace se ale předpokládá napájení některým ze standardních napájecích napětí a to 12 nebo 24V.

Kvůli požadavku na snížení napětí z 24V na 5V byl jako stabilizátor vybrán LM 2575 [13]. Jedná se o standardní *step-down* spínaný regulátor. Jeho zapojení na prototypové desce je zcela dle doporučení výrobce [13].

5.1.4 OneWire sběrnice

Aby mohl vyvíjený systém zjišťovat aktuální teplotu, je nutné jeho vybavení o příslušné rozhraní. Pro měření teploty byly vybrány teplotní čidla DS18B20 od firmy Maxim [17]. Jedná se o velmi rozšířený modul, který poskytuje dostatečnou přesnost při velmi příznivé ceně.

Tato teplotní čidla komunikují prostřednictvím OneWire sběrnice. Jedná se o sběrnici, které postačují dva vodiče. Jeden vodič je určený pro nulový potenciál (GND), druhým vodičem jsou jednotlivá zařízení napájena a zároveň jím komunikují. Takovéto zapojení je doporučeno podle

dokumentace výrobce pouze pro méně rozsáhlé instalace. Při zapojení vyššího počtu teplotních čidel je doporučeno přistoupit ke tří drátovému zapojení. V případě třídrátového zapojení je třetí drát využít jako trvalý zdroj napětí [17].

Protože hlavní řídicí počítač nedisponuje rozhraním pro OneWire sběrnici, bylo jeho zapojení doplněno o převodník DS2482. Tento čip provádí převod I2C rozhraní na OneWire sběrnici. Převodník má implementovanou podporu přímo v jádru Linuxu, jeho použití tedy bylo velmi jednoduché.

5.1.5 Sériové rozhraní

Aby si systém zachoval jistou možnost dalšího rozšíření, byla na hlavní desce vyvedena další dvě sériová rozhraní, která rozšíření o nové komunikační kanály umožňují.

V případě potřeby připojení ke sběrnici KNX by se nabízelo využít například modulu *TP-UART2 Board BTM2-PCB*. Tento modul představuje převodník mezi KNX sběrnici (KNX.TP) a sériovou linkou. Samotné připojení modulu k systému by samozřejmě nebylo dostatečné, připojení by vyžadovalo úpravu softwaru. Na internetu je dostupná řada open source knihoven, díky kterým je možné přistupovat pomocí tohoto modulu na KNX sběrnici. Bylo by tedy bezpodmínečně nutné portovat některou z těchto knihoven a naimplementovat její obsluhu do systému.

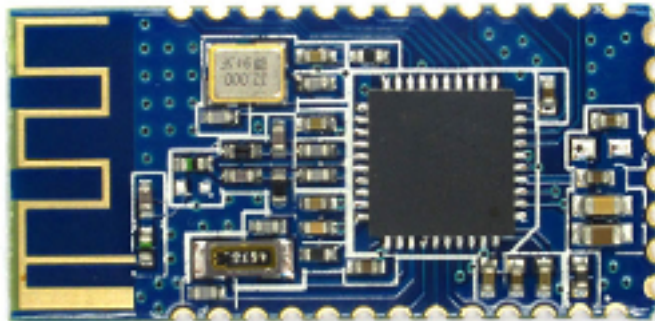
5.1.6 Bluetooth rozhraní

Jak již bylo popsáno v předchozích kapitolách, Bluetooth komunikace je v systému použita pro účely lokalizace uživatele. Na trhu je dostupná řada řídicích desek, které již mají kombinovaný WiFi / Bluetooth modul s podporou BLE přítomný na desce. Použití modulu s osazeným modulem by se mohlo zdát výhodnější, odpadá potřeba další součásti, u které by mohlo dojít k poškození, případně k nekompatibilitě. Nicméně toto řešení nebylo vybráno a to z důvodu, že pokud je Bluetooth modul externí, je možné optimalizovat jeho umístění. Je možné nainstalovat řídicí desku do rozvaděče a následně vyvést Bluetooth modul na optimální místo. Nicméně je nutné respektovat fyzická omezení sériového rozhraní.

Externí Bluetooth moduly se standardně připojují přes rozhraní *UART*. Bluetooth modul se pak může nacházet ve dvou stavech. V prvním stavu přijímá takzvané *AT příkazy*, pomocí kterých je možné modul nastavit. Do druhého stavu se modul dostane ve chvíli, kdy dojde ke spojení s jiným zařízením, poté se veškerá data poslaná přes *UART* rozhraní přenesou na vzdálenou stranu spojení. Stejně tak se veškerá data odeslaná vzdálenou stranou přenesou na sériové rozhraní *UART*.

5.1.6.1 Bluetooth modul

Jedním z prvních široce nabízených Bluetooth modulů s podporou standardu *Low Energy* byl modul založený na čipu CC2540 od firmy Texas Instruments [19]. Modul se nabízí pod označením HM-10 a je vyráběn čínskou firmou Huamao technology.



Obrázek 8: Bluetooth modul HM-10

Bluetooth modul je nutné správně nastavit tak, aby se choval jako server. V tomto módu nejen že bude možné se k němu připojit z mobilního terminálu, ale modul bude vysílat rámce, pomocí kterých bude možné jeho nalezení. Tyto rámce jsou v normě označovány jako *iBeacon* rámce. Jedná se o speciálně formátované zprávy, které jsou periodicky vysílány. Jejich obsahem je především jednoznačný identifikátor označující typ modulu a popis dostupných služeb.

5.1.7 Vstupně - výstupní rozšiřující modul

Jedná se o modul, který není pevně spojen s hlavní částí prototypu. Tento modul nabízí rozšíření o 10 vstupních a 10 výstupních portů.

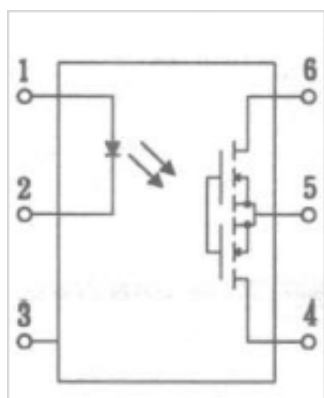
Jako vstupní optické oddělovače jsou použity integrované obvody od firmy Cosmo, PC817B. Optočleny mají izolační pevnost do 5kV.

Na vstupu optoizolátorů je LED dioda, jejíž maximální proud je 50mA, špičkově až 1A. Za normálního provozu by ale neměl přesahovat 20mA. Pro splnění těchto podmínek je nutné správně spočítat odpor zapojený s diodou do série. Výpočet vychází z hodnoty maximálního proudu z datasheetu a z maximálního vstupního napětí 24V. Výpočet je uveden v rovnici 1.

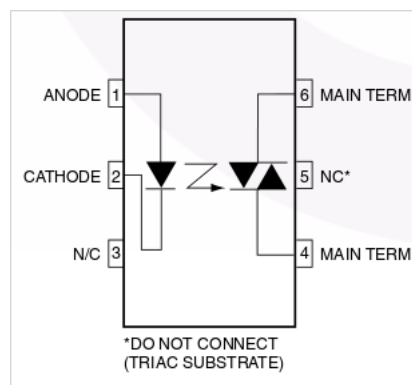
$$R = \frac{U_{max} - U_{LED}}{I_{max}} = \frac{24.0 - 1.5}{20 \cdot 10^{-3}} = 1.125k\Omega \quad (1)$$

Při výběru výstupních optočlenů je možné vybírat ze dvou typů, s triakovým výstupem nebo tranzistorovým výstupem. Použití výstupních optočlenů s triaky je omezeno pouze na spínání střídavých napětí, zatímco při správném zapojení optočlenu s tranzistorovým výstupem

je možné spínat jak střídavé tak i stejnosměrné obvody. Do zapojení byly vybrány oba dva druhy výstupních prvků.



Obrázek 9: Vnitřní uspořádání KAQV214 [15]



Obrázek 10: Vnitřní uspořádání MOC3022 [16]

Na obrázku 9 je vnitřní zapojení optočlenu s tranzistorovým výstupem. Aby bylo možné pomocí tranzistorového výstupu ovládat stejnosměrné i střídavé obvody musí se výstupní obvod zapojit mezi piny 4 a 6. Z obrázků vnitřních zapojení 9 a 10 je zřejmé, že jsou mezi sebou zaměnitelné. Vstupní signály na fotodiodu jsou rovněž stejné - 1.2V při proudu 10mA.

5.2 SW návrh řešení

Pro implementaci projektu byl vybrán framework Qt. Hlavním důvodem k výběru právě tohoto frameworku je jeho široké pole působnosti. Je možné v něm vytvářet jak konzolové aplikace bez uživatelského rozhraní, tak i aplikace s grafickým rozhraním. Největší výhodou Qt je podpora více platforem než pouze x86 PC. Tato podpora dovoluje spouštět stejnou aplikaci na běžném počítači i na mobilních zařízeních s operačními systémy Android / iOS. Kompletní seznam podporovaných platforem je k nalezení na <http://doc.qt.io/qt-5/supported-platforms.html>.

Celý projekt je rozdělen do šesti vzájemně závislých částí. Jednou z nich je aplikace běžící na serveru - *Serverová aplikace* a dvě aplikace s grafickým rozhraním - *Konfigurační aplikace*, *Uživatelská aplikace*. Zbývající tři části projektu jsou zamýšleny jako sdílené knihovny poskytující funkce a objekty, které se starají o výměnu a ukládání dat. Sdílené knihovny jsou *CommunicationHandler*, *DatabaseHandler* a *ObjectHandler*.

5.2.1 Embedded linux

Hlavní řídicí deska podporuje běh operačního systému Linux. Instalace linuxového prostředí na běžném počítači je velmi snadná. V případě vestavěných systémů je nutné dodržet několik

základních kroků, jak postupovat při sestavování celého systému.

Vstupním prvkem každého embedded linuxu je *bootloader*, česky zavaděč. Úkolem zavaděče je základní nastavení periférií a přístupu k paměti. Navíc se zavaděč stará o nahrání linuxové jádra z připojené MicroSD karty při startu systému. Při výběru bootloderu je nutné dbát na to, zda obsahuje podporu pro daný typ procesoru. Nejrozšířenějším a nejvíce podporovaným zavaděčem je *Das U-Boot*. Jedná se o univerzální open source zavaděč, který zahrnuje podporu nejen procesorů architektury ARM, ale také PowerPC, MIPS a dalších.

Linuxové jádro je nutné umístit na speciální (předem zadanou) adresu na MicroSD kartě. Je proto nutné jádro takzvaně vypálit, nestačí pouze zkopírovat na souborový systém. Zdrojové kódy linuxového jádra jsou volně dostupné. Pro úspěšné fungování jádra je nutné použití jádra s rozšířením o podporu procesorů Allwinner. Takové jádro je volně ke stažení v repozitářích udržovaných sdružením sunxi [20]. Jádro udržované sdružením sunxi je vždy založeno na některé z oficiálních verzí linuxového jádra. Do jádra jsou následně aplikovány změny, které umožní podporu všech periférií daného procesoru.

Shodnost jádra distribuovaného komunitou sunxi s oficiálním jádrem je důležitá zejména ve chvíli, kdy vznikne požadavek na aplikaci dalších změn. V tomto případě byly aplikovány změny na podporu *real-time* běhu.

Poslední součástí každého linuxového systému je základní adresářová struktura spolu se základními programy, takzvaný *filesystem*. Celý základní systém je možné sestavit podobně jako linuxové jádro ze zdrojových kódů. Sestavení celé struktury by ale zabralo velké množství času, a proto bylo využito předem připraveného základního filesystemu, dostupného online.

Sestavení jádra linuxu, včetně korektní instalace na MicroSD kartu bylo zautomatizováno pomocí skriptů napsaných v jazyce BASH.

5.2.2 Qt Framework

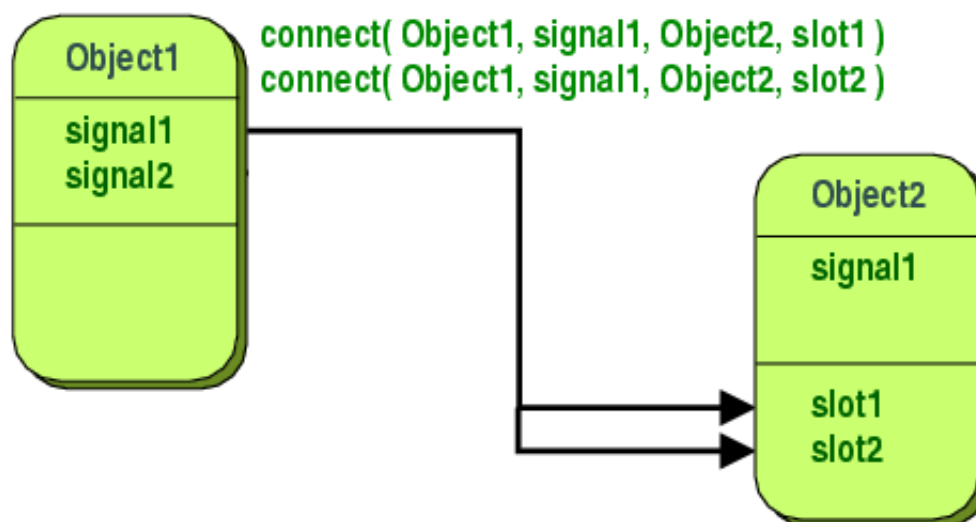
Qt Framework je založen na programovacím jazyce C++ a je vyvíjen pod open source licencemi. Jeho hlavním smyslem je poskytnutí kompaktního ekosystému knihoven a dalších nástrojů pro vývoj *cross-platform* aplikací.

Největším rozšířením jazyka C++ v Qt Frameworku je zavedení takzvaných *signálů* a *slotů* [22]. Využití vazby typu signal - slot má výhody především v grafických aplikacích. Tyto aplikace jsou charakteristické tím, že velkou část svého běhu nevykonávají žádnou akci, avšak jakmile uživatel stiskne tlačítko, je nutné okamžité vykonání obsluhy.

Většina ostatních frameworků tuto situaci řeší tak, že při kliknutí je zavolán tzv. *callback*. Jedná se o ukazatel na funkci, která má za úkol obsloužit danou akci. Nevýhodou tohoto přístupu

je nemožnost typové kontroly, zda volaná funkce vyžaduje určité parametry, a zda jsou tyto parametry zadány. Pokud není tato kontrola provedena, může dojít k neočekávaným pádům aplikace z důvodu přístupu k neinicializované paměti a podobně [22].

Navázání signálu na slot je schematicky znázorněno na obrázku 11. Jak je ze schématu zřetelné, první objekt definuje signál, jenž bude vyslán. Druhý objekt který na danou akci může reagovat definuje naopak slot. Vytvoření vazby - *connect* je možné zavolat z libovolného jiného objektu, který obsahuje reference na oba objekty. V průběhu kompilace projektu je kontrolováno, zda zdrojová funkce objektu 1 (signál) obsahuje stejný počet parametrů stejného typu jako cílová funkce objektu 2 (slot).



Obrázek 11: Grafické znázornění signál - slot [22]

Existují ale i jistá omezení, která je nutné znát a respektovat. V následujících odstavcích budou tato omezení popsána odděleně pro signály i sloty. Veškerá uvedená upozornění jsou čerpána z [22].

5.2.2.1 Signály

Signály mohou být vyslány z jakéhokoli místa v kódu, avšak je doporučeno vysílat signály pouze z objektu, který daný signál definuje. Obsah bloku slotu je vykonán okamžitě po vyslání signálu ve stejném vlákne, z něhož byl signál vyslán. Jedná se tedy o rovnocennou akci, jako zavolání běžné funkce.

Existuje zde však jedna výjimka, pokud se při vytváření spojení (*connection*) zadá jako poslední parametr typ spojení *Qt::QueuedConnection*, vykoná se blok slotu až po skončení obsluhy vyslání signálu.

Návratová hodnota signálu je vždy void čili bez návratové hodnoty.

Signály jsou ve zdrojovém kódu pouze definovány, nejsou implementovány. O jejich implementaci se stará v průběhu kompilace takzvaný *Meta Object Compiler* [23].

5.2.2.2 Sloty

Slot je vždy zavolán, pokud došlo k vyslání signálu, na který je daný slot navázán. Na rozdíl od signálu je funkce implementující slot naprosto shodná s běžnou funkcí, a lze k ní tak i přistupovat.

Zavolání slotu, který je navázán s některým signálem, není samozřejmě zdarma, je zde jisté zdržení způsobené vyhledáním správné vazby a jejím následným zpracováním. Autoři Qt uvádějí zhruba desetinásobné zpomalení v porovnání s přímým zavoláním funkce.

5.2.3 Object Handler

Jedná se o knihovnu, jež popisuje všechny objekty, jež jsou vyměňovány mezi klientskou a serverovou částí. V této knihovně jsou zavedeny dvě abstraktní třídy, které předepisují metody, jež musí každý objekt implementovat. Tyto abstraktní třídy jsou `ObjectHandler` a `DataObject`.

První třída `ObjectHandler` zavádí metody, jež jsou určeny pro serializaci a deserializaci objektů, které mají být odeslány nebo přijaty. Druhá třída `DataObject` předepisuje metody, které jsou využity pro ukládání objektů do perzistentního úložiště (databáze).

V příloze na obrázku 23 je zobrazen UML diagram vztahu (dědičnosti) mezi rozhraním `IObjectHandler` a dvěma objekty, které jej implementují. Na obrázku 24, rovněž v příloze, je zobrazen celkový UML diagram pro třídu `User`.

5.2.3.1 Přenos objektů

Spojení je vždy založeno na principu spolehlivého kanálu realizovaného prostřednictvím TCP/IP. Data, která jsou tímto spojením přenášena, není bezpodmínečně nutné chránit proti poškození při přenosu. Tato ochrana je již implementována nižší vrstvou TCP/IP v podobě výpočtu CRC.

Přenášená data jsou ve formátu XML. Jedná se o univerzální rozšiřitelný značkovací jazyk, jenž klade velký důraz na správné formátování a uzavírání tagů. Přenos ve formátu XML byl vybrán především s ohledem na možnost dalšího využití dat i mimo vyvíjené aplikace. V případě potřeby je možné jakákoliv data kdykoliv exportovat a pomocí externích nástrojů dále využívat.

Pro přenos všech objektů je zvolena jednotná kostra zprávy 12.

Každá zpráva tak obsahuje hlavičku označující, že se jedná o xml zprávu. Kořenovým atributem každé zprávy je typ operace, která má být pro daný objekt vykonána. Možné varianty

```

(1)<?xml version="1.0"?>
(2)<operation type="">
(3)  <object name=""/>
(2)</operation>

```

Obrázek 12: Minimální kostra přenášených dat

jsou detailně popsány v tabulce 2.

Posledním vnořeným atributem je jméno objektu, pro který bude daná akce vykonána. V případě, že se jedná o volání operace typu GET, je možné, aby byl obsah objektu prázdný. V takovém případě jsou navraceny všechny vyhovující hodnoty. Pokud se jedná o volání jiné operace, je nutné, aby bylo vyplněno rovněž i tělo objektu.

ADD	vytvoření nového objektu daného typu
GET	vyčtení dat z aktuálního objektu
SET	nastavení dat do aktuálního objektu
DEL	vymazání aktuálního objektu
INVOKE	vykonání příkazu na aktuálním objektu
RESP	příjem odpovědi na předchozí dotaz / příkaz

Tabulka 2: Výpis možných akcí vykonaných na objektech

Zpracování komunikace je na straně klienta omezeno pouze na odpovědi od serveru. Zpracovávají jsou pouze zprávy typu RESP. Při zpracování přijatých dat je nejdříve provedena kontrola, zda se jedná o validní (správně formátovaný) xml dokument. V kladném případě je načteno jméno objektu, kterého se zpráva týká, a operace, která má být provedena. Posledním krokem je provedení požadavku a aktualizace dat.

Zpracování komunikace je na straně serveru doplněno o krok ověření, který je vložen před vlastní zpracování operace. Cílem je zabránit neoprávněné změně objektu. Vlastní proces ověření oprávnění je popsán v příslušné části práce věnované softwaru na straně serveru.

5.2.4 Communication Handler

Jedná se o knihovnu, kterou využívá uživatelská a konfigurační aplikace. Hlavním účelem této knihovny je vyhledávání dostupných serverů a navazování komunikace s nimi. Po navázání komunikace automaticky dojde k autentizaci uživatele. Vyšší vrstvy využívající tuto knihovnu obdrží spojení, které je již připraveno k odeslání a příjmu dat.

5.2.4.1 Bluetooth klient

Komunikace pomocí Bluetooth s podporou standardu Low Energy je v Qt Frameworku podporována od verze 5.4. V této verzi probíhal vývoj velmi bouřlivě, a tak je doporučeno použití verze 5.5 a vyšší. V této verzi je již API pro přístup stabilizováno. I přesto, že většina práce s ohledem na podporu této technologie je označena za hotovou, byla v průběhu práce nalezena řada problémů.

Vzhledem k tomu, že Qt technologie je založena na principu posílání signálů do připravených slotů, je i BLE subsystém napsán tímto způsobem. Vyhledávání okolních Bluetooth zařízení probíhá tak, že objekt *QBluetoothDeviceDiscoveryAgent* zaregistruje signály *deviceDiscovered* a *finished*. Příslušné signály jsou pak obslouženy tak, že *deviceDiscovered* je vykonán vždy, když dojde k nalezení nového zařízení, signál *finished* je vykonán ve chvíli, kdy již došlo k nalezení všech okolních Bluetooth zařízení.

V průběhu testování vyhledávání dostupných zařízení se na platformě Android vyskytl problém v tom, že signál o ukončení vyhledávání byl vyslán až v rozmezí 60 - 90 s od počátku vyhledávání. Jednotlivá zařízení sice byla ve většině případů nalezena již v průběhu prvních 10 - 15 s, nicméně i zde byl pozorován velký časový rozptyl, což by nebylo pro tento systém (lokalizaci uživatele) vhodné. Uživatel by musel čekat relativně dlouhou dobu, než by bylo možné spolehlivě určit, ke kterému z dostupných zařízení se připojit. V průběhu řešení tohoto problému byl nalezen záznam o stejném chování na platformě Android i v *bugzille* Qt projektu, popis problému je možné nalézt na adrese <https://bugreports.qt.io/browse/QTBUG-46253>. Oprava je plánována ve verzi Qt Framework 5.8.

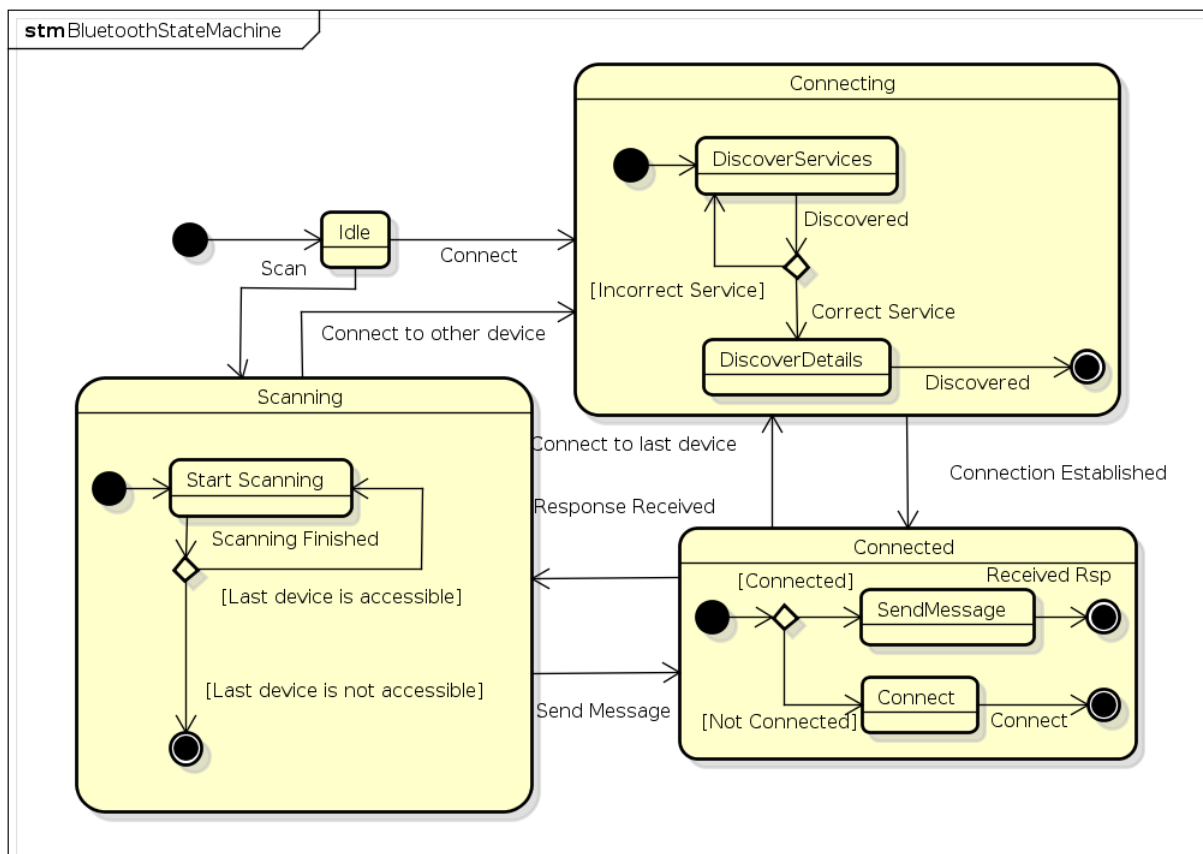
Vzhledem k nemožnosti použití interních mechanismů Qt Frameworku bylo nutné nastalou situaci vyřešit. Prvním krokem bylo ověření, zda je platforma Android schopná vyhledávat BLE zařízení lépe, než jak je tato funkce implementována uvnitř Qt. Byla tedy vyvinuta jednoduchá aplikace, jejímž jedinným úkolem bylo vyhledávat dostupná BLE zařízení.

Podobně jako v případě implementace BLE podpory uvnitř Qt Frameworku, byl i v případě platformy Android vývoj velmi bouřlivý. Podpora pro BLE byla zavedena ve verzi 18 [24], nicméně pro verze Android menší než 21 je nutné používat jiné API, než které je doporučeno používat pro verze Android vyšší než 21 včetně [25]. Vyvíjená aplikace pro vyhledávání tyto změny v API reflektuje, a proto je možné její použití jak na starších systémech Android 4.3, tak i na novějších systémech verze 5.0.

Aplikace vyvinutá přímo pro platformu Android se ukázala jako funkční, vyhledání všech dostupných zařízení je dokončeno ve zlomcích vteřin. Posledním krokem tedy bylo zavolání aplikace určené pro platformu Android napsané v jazyce Java z aplikace napsané ve frameworku

Qt v jazyce C++. Framework Qt je na tuto možnost velmi dobře připraven tím, že poskytuje rozhraní *QAndroidJniObject*, které umožňuje přímé volání zdrojového kódu Java z C++[26].

Stavový automat, který řídí přechody mezi jednotlivými stavy, je zobrazen na obrázku 13. Po spuštění příslušné aplikace započne vyhledávání aktivních Bluetooth zařízení v okolí. Vyhledávání je omezeno pouze na zařízení s podporou Low Energy. Toto vyhledávání je na platformě Android provedeno pomocí výše popsaného principu volání Java aplikace z Qt Frameworku. Změna volání je provedena ve stavu **Scanning**. Ostatní stavy automatu nejsou touto změnou dotčeny.



Obrázek 13: Stavový automat pro Bluetooth komunikaci

Jakmile je vyhledávání dostupných zařízení dokončeno, je vybrán nejbližší Bluetooth server, ke kterému se aplikace pokusí připojit. Po úspěšném navázání komunikace jsou odeslána data a čeká se na odpověď. Po přijetí odpovědi se spojení ukončí a systém opět přejde do stavu vyhledávání dostupných zařízení.

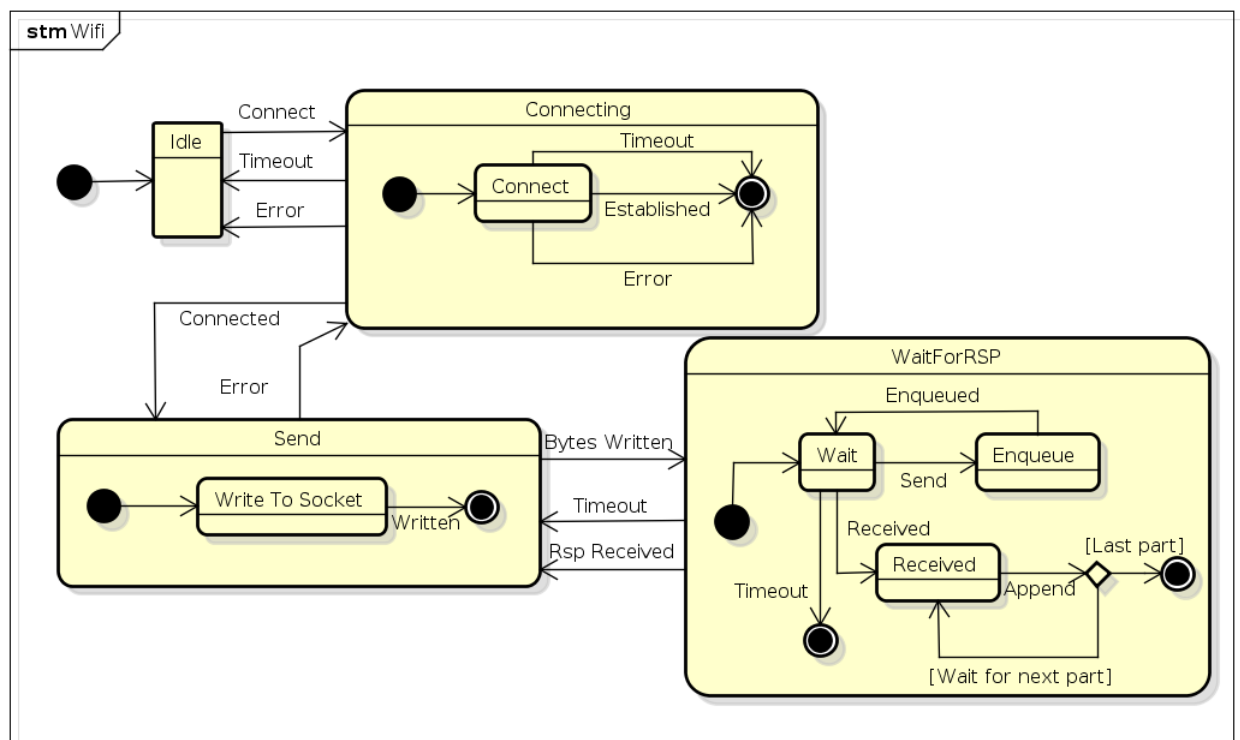
Pokud je poslední zařízení, ke kterému byla aplikace připojena stále dostupné a je v blízkosti, nedochází k novému navázování spojení. K navázání spojení k jinému serveru dojde pouze v případě, že:

1. Poslední server již není dostupný.
2. V okolí se objeví server, který je mnohem blíže.

Pro měření vzdálenosti mezi serverem a klientským terminálem je využito měření síly signálu, který klientský terminál přijímá. Síla signálu je v literatuře označována jako *RSSI* z anglického názvu Received Signal Strength Indicator.

5.2.4.2 TCP/IP klient

Jakmile jsou načteny základní údaje o serveru, je navázáno IP spojení mezi klientem a serverem. Vytvoření spojení je popsáno stavovým diagramem na obrázku 14, spojení je šifrováno pomocí SSL. Stavový automat navazování spojení je vyobrazen na obrázku 14.



Obrázek 14: Stavový automat pro WiFi komunikaci

Během navazování spojení může dojít k několika na sobě nezávislých chybovým stavům. Jednou z chyb, ke které v testovacím prostředí nedochází, nicméně v případě reálného nasazení může dojít, je nedostupnost serveru. K nedostupnosti může dojít buď v důsledku síťového problému, kdy může být spojení špatně směrováno, nebo v důsledku restartu / zaseknutí služby na serveru. Navazování spojení se v takovémto případě může prodloužit (v závislosti na platformě) ze stovek milisekund až na desítky sekund.

Pokud by tento stav nebyl nijak ošetřen, došlo by k velmi dlouhému "zamrznutí" aplikace. Tento problém je proto vyřešen tak, že těsně před navazováním spojení je spuštěn časovač. Pokud není spojení vytvořeno před expirací tohoto kontrolního časovače, je zahlášen problém vyšším vrstvám a současně s tím i uživateli. Uživatel je tak informován o nedostupnosti služby a je mu umožněno sjednat nápravu, například nahlášení tohoto incidentu správě budovy.

Přenos dat mezi klientem a serverem je ve formátu xml. Do textu tohoto xml není přidána žádná další hodnota, která by určovala velikost přenášených dat. Konec aktuálně přenášeného objektu aplikace pozná tak, že je *stream* dat ukončen pomocí dvojice symbolů `\r \n`. Stejně ukončení je použito v serverové i obou uživatelských aplikacích.

Podobně jako v případě navazování spojení, i při přenosu dat mohou nastat případy, kdy dojde k přerušení spojení a tím kompletní ztrátě dat. Pro každý přenos od klienta na server je tedy spuštěn časovač, který čeká na příchozí odpověď. Pokud není odpověď doručena do předem daného času, je aktuální spojení ukončeno. Po ukončení spojení je proveden pokus o vytvoření nového spojení na poslední známý server. Lze očekávat, že toto spojení nebude možné úspěšně vytvořit. Následně tedy dojde k opakovanému vyhledání dostupných serverů a připojení k nejbližšímu z nich.

5.2.4.3 Vytvoření spojení

Vytvoření nového spojení mezi klientem a serverem je vždy iniciováno ze strany klienta. Sekvenční diagram navazování spojení je zobrazen na obrázku 15.

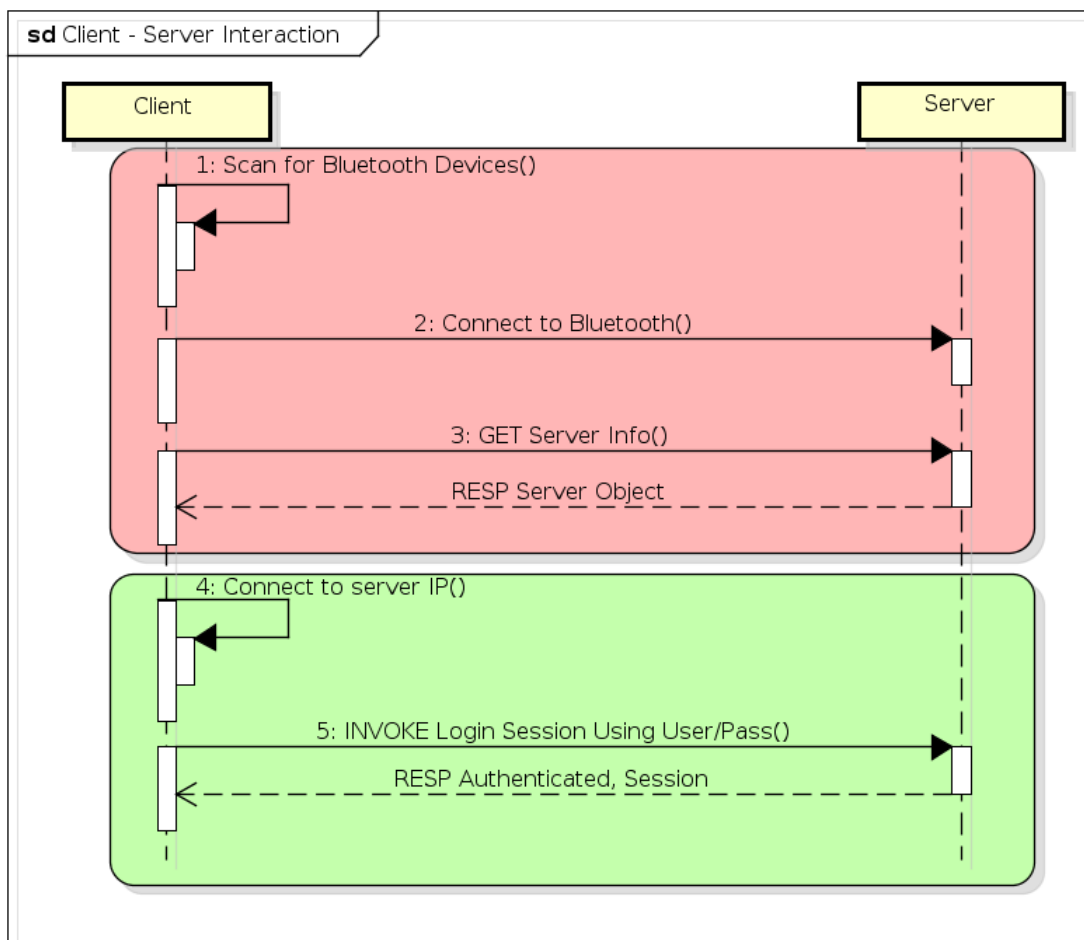
Na straně klienta dojde k vyhledání dostupného serveru pomocí Bluetooth. Následně je navázáno spojení a je vyslán požadavek na informace nutné pro připojení k serveru. Jedná se o veřejně přístupné informace, není proto vyžadováno přihlášení uživatele. Požadavek, který je vyslán z klientské aplikace je základním dotazem na objekt jménem **server**.

V odpovědi na tento dotaz (přenášené přes Bluetooth spojení) jsou obsaženy veškeré informace nutné k úspěšnému vytvoření spojení. Jedná se především o informace o IP adrese serveru a portu, na kterém naslouchá serverová aplikace.

Po přijetí těchto základních informací je spojení přes Bluetooth uzavřeno a současně s tím je navázáno spojení nové, pomocí TCP/IP protokolu přes síť internet. Prvním krokem po navázání spojení je přihlášení uživatele. Průběh přihlašování se řídí aktuálním nastavením serveru a z pohledu připojeného klienta je vždy proveden stejným způsobem.

Pokud přihlášení proběhne úspěšně, je právě vytvořené spojení označeno za kompletní. Následuje série dotazů na získání všech potřebných informací. V opačném případě je spojení uzavřeno a je opětovně spuštěna sekvence vyhledání dostupného serveru a pokus o navázání

dalšího spojení s ním.



Obrázek 15: Princip navázání spojení

V levém sloupci je ukázka dotazu na objekt typu `server`, v pravém sloupci je ukázka odpovědi 16. Díky formátování zpráv pomocí XML, je možné bez větších obtíží určit význam jednotlivých položek.

5.2.5 Database Handler

Jedná se o knihovnu sdílenou mezi všemi vyvinutými aplikacemi. Centrální funkcí této knihovny je zprostředkování přístupu k databázi. Serverová aplikace ukládá do databáze veškerá nastavení, která uživatel odešle prostřednictvím konfigurační, případně uživatelské aplikace. Současně s tím serverová aplikace periodicky ukládá další data, například naměřené teploty apod.

V ostatních aplikacích je databáze využita pro ukládání historicky zadaných údajů. Díky tomu je možné zobrazit v GUI předchozí zadaná uživatelská jména nebo například nastavení připojení k LDAP serveru.

```

<?xml version="1.0"?>
<operation type="get">
  <object name="server"/>
</operation>

<?xml version="1.0"?>
<operation type="resp">
  <object name="server">
    <name>SmonAdvanced</name>
    <address>192.168.0.235</address>
    <address_type>Statická</address_type>
    <dns>8.8.8.8</dns>
    <gateway>192.168.0.1</gateway>
    <mask>255.255.255.0</mask>
    <port>32001</port>
    <ntp>89.221.208.11</ntp>
  </object>
</operation>

```

Obrázek 16: Příklad dotazu a odpovědi při navazování komunikace

Framework Qt podporuje celou řadu běžně používaných databázových backendů (například MySQL, PostgreSQL, SQLite, Sybase a další) [27]. Ze všech podporovaných byla vybrána databáze SQLite, protože je ve výchozím stavu podporována na mobilní platformě Android. Díky tomu, že na všech platformách bude použit stejný typ databáze, je možné používat stejné SQL dotazy. Pokud by byla například v serverové aplikaci použita jiná databáze, bylo by nutné upravit syntaxi SQL dotazů tak, aby odpovídala danému typu databáze.

5.2.5.1 Inicializace databáze v serverové aplikaci

Inicializace databáze v běžném linuxovém prostředí se skládá pouze ze zavolání inicializační funkce. Tato funkce přijímá jako parametr umístění, kde má být databáze vytvořena. Pokud se jedná o databázi typu SQLite, bude celý obsah databáze uložen právě v tomto jediném souboru. Ve vyvíjené serverové aplikaci je databáze uložena na adrese `/opt/smon/database/database.db3`.

Zálohu nastavení je vhodné realizovat prostřednictvím externí utility `sqlite3`. Ta umožňuje provedení zálohy i v případě, že je databáze využívána. Nemělo by tedy dojít k porušení integrity dat vlivem právě prováděných transakcí [28].

5.2.5.2 Inicializace databáze v GUI aplikaci

Qt framework vytvoří pro každou spuštěnou aplikaci složku, do níž ukládá sdílená data. Do sdílené složky se ukládá taktéž vytvořená databáze. Spolu s vlastní databází je zde vytvořen i konfigurační soubor obsahující jméno databáze, cestu k databázi a typ databáze. Při otevírání databáze je proto nutné tento soubor načíst a otevřít databázi z daného umístění. Díky tomu, že umístění konfiguračního souboru je definováno uvnitř Qt Frameworku, je navržené řešení multiplatformní.

Vzhledem k tomu, že databáze v ovládacích aplikacích neobsahují příliš důležitá data, není nutné (ani obvyklé) tuto databázi zálohovat.

5.2.6 Serverová aplikace

Serverovou aplikaci lze rozdělit na tři hlavní bloky. Prvním blokem je TCP/IP server, který naslouchá na definovaném portu, ke kterému se jednotliví klienti připojují. Druhým blokem je Bluetooth server, který stejně jako TCP/IP vyčkává na nově přichozí spojení. Třetím blokem je řízení technologie, které běží v nekonečné smyčce nezávisle na ostatních dvou blocích.

Čtvrtým blokem, který je před zraky uživatelů skryt, je perzistentní úložiště nastavení. Toto je nutné k tomu, aby veškerá uživatelská nastavení vydržela restart systému, například v případě výpadku napájení.

5.2.6.1 Perzistentní úložiště

Perzistentní data, která je nutné ukládat mezi restarty systému, je možné rozdělit do dvou skupin. První skupinou jsou konfigurační data, tedy taková data, která uživatel nastaví do systému pomocí konfigurační aplikace. Do druhé skupiny spadají data, která jsou v systému uložena od ovládaných technologií. Převážně se jedná o historická data (typicky průběhy teploty).

Jako perzistentní úložiště byla vybrána MicroSD karta, na které je zároveň uložena celá aplikace včetně operačního systému. Zatímco jádro linuxu spolu se serverovou aplikací je uloženo na *read-only* části karty, úložiště je uloženo na *read-write* části karty. Přístup k datům je zprostředkován SQLite databází.

5.2.6.2 Bluetooth server

Bluetooth periferie není v systému využita pouze pro účely lokalizace uživatele, ale také pro přenos základních identifikačních údajů o serveru. Část dat, která jsou přenášena lze považovat za volně dostupné (jedná se o data určená k vytvoření spojení). Ostatní data, která je možné pomocí

tohoto komunikačního kanálu získat, již veřejně přístupná být nemohou. Základní myšlenkou je tedy umožnit přenos některých objektů bez vyžadování ověření, případně všech dalších objektů s příslušným ověřením.

Typické použití vypadá tak, že je ze strany klienta vytvořeno nové spojení. Do tohoto nového spojení je odeslán požadavek na objekt typu `server`. Jelikož se jedná o veřejně dostupná data, je ze strany serveru navrácen obsah tohoto objektu. Princip navázání komunikace byl již popsán v kapitole 5.2.4.3.

Případ, ve kterém dochází i k přenosu ostatních objektů, nastává především při uvádění zařízení do provozu. V zařízení nejsou nastavena ani základní komunikační rozhraní, opět se jedná především o adresu zařízení. V takovém případě je ze strany klienta nejdříve odeslán požadavek na přihlášení pomocí poskytnutých údajů. Po úspěšném přihlášení je poté odeslán požadavek na operaci `set`, která nastaví správné údaje. V dalším kroku je pak provedeno spojení prostřednictvím sítě internet.

Zásadním prvkem, díky kterému je možné považovat toto řešení za bezpečné, je možnost rozlišení mezi jednotlivými spojeními. Je důležité zaručit, aby nebylo možné jedním spojením projít sekvencí přihlášení a druhým spojením dokončit přenos dat. Vybraný Bluetooth modul zaručuje, že aktivní bude vždy pouze jedno spojení. Jakmile dojde k vytvoření spojení, stane se modul pro ostatní účastníky nedostupným. Modul navíc oznámí počátek spojení odesláním sekvence `OK+CONN` na sériovou linku. Aplikace tak dostane jasný impuls toho, že bylo vytvořeno nové spojení a může díky tomu inicializovat vnitřní proměnné. Stejně tak, po skončení přenosu dat dojde k ukončení spojení a odeslání sekvence `OK+LOST`.

V případě dlouhodobé nedostupnosti spojení přes síť internet by bylo možné celý systém provozovat pomocí komunikace pouze prostřednictvím Bluetooth. Nicméně toto řešení nelze doporučit, neboť v případě přenosů větších objektů by došlo k dlouhodobé nedostupnosti serveru pro ostatní účastníky. Ostatní účastníci by se pak pokusili přepojit na jiný, dostupný server. Tento další server by ale již nebyl v danou chvíli tím nejbližším.

5.2.6.3 TCP/IP server

Již při návrhu základního softwaru pro obsluhu spojení byly vytyčeny dva základní body. Prvním z nich byla ochrana soukromí uživatelů systému. Druhým bodem byla podpora více spojení současně.

Ochrana soukromí uživatelů je velmi podstatným prvkem systému. Žádný z uživatelů si určitě nepřeje, aby byla jeho data zneužívána třetí osobou. Současně s tím si žádný z provozovatelů systému nepřeje, aby jeho systém byl jednoduše napadnutelný. Z těchto důvodů je veškerá

komunikace mezi serverem a klientem šifrována pomocí SSL protokolu.

Obsluhu více spojení současně je možné implementovat dvěma způsoby. Pokročilejší a při vyšší zátěži efektivnější způsob využívá jednoho vlákna pro obsluhu příchozích spojení. Další vlákno (případně až N vláken) využívá pro obsluhu aktivních spojení. Těchto spojení bývá typicky více než obsluhujících vláken.

Jednodušší a přímočařejší způsob spočívá ve využití jednoho vlákna pro obsluhu příchozích spojení stejně jako v prvním případě. Pro každé příchozí spojení ale spustí nové vlákno, které obsluhuje vždy právě jedno spojení. Slabinou tohoto řešení jsou situace, kdy je nutné obsluhovat velký počet aktivních spojení, nebo když je nutné rychlé vytváření nových spojení. Výhodou tohoto přístupu je nezávislost daného spojení na ostatních. Zaseknutí jednoho spojení / vlákna žádným způsobem neovlivní chování spojení jiného.

Vzhledem k předpokladu, že k jednomu serveru bude současně připojeno řádově 50 uživatelů, byla vybrána druhá možnost. Každé aktivní spojení bude tedy obsluhováno ve vlastním vlákne. Stavový automat pro obsluhu spojení je na obrázku 17.

5.2.6.4 Navázání spojení

Spojení je vždy iniciováno směrem od uživatelské aplikace k serverové aplikaci. Jakmile je spuštěn SSL server, vyčkává na nové spojení. Ihned po přijetí je spuštěno nové vlákno, které od této chvíle obsluhuje právě vytvořené spojení. Každé nové spojení se nachází ve stavu, ve kterém přijímá pouze požadavky na přihlášení uživatele. Až po úspěšně provedeném přihlášení je stav spojení nastaven na příjem a zpracování požadavků.

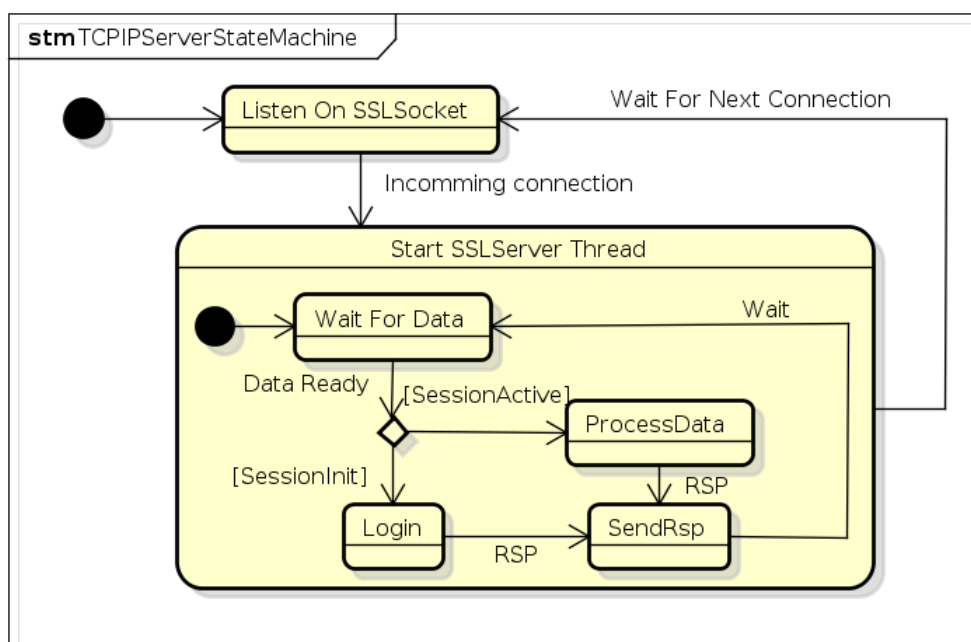
Stavový diagram výše popsaného principu je na obrázku 17.

5.2.6.5 Ověření uživatele - LDAP

Veškerá komunikace mezi klientem a serverem je šifrována pomocí SSL protokolu. Pomocí šifrování lze zabránit pouze tomu, že data nebudou viditelná pro případné útočníky, není ale možné ověřit identitu uživatele na klientské straně.

Pro centrální správu uživatelských profilů je v korporátním prostředí nejčastěji využíváno řešení postavené na technologii LDAP. LDAP je zkratkou pro *Lightweight Directory Access Protocol*, volně přeloženo jako odlehčený protokol pro přístup k adresářům [21], protokol vychází ze standardu X.500.

Jak již z rozšířeného názvu této technologie vyplývá, ukládání dat v LDAP databázi probíhá ve formátu adresářové (stromové) struktury. Každý objekt uložený v databázi je jednoznačně identifikován pomocí celé cesty.



Obrázek 17: Stavový automat TCP/IP serveru

Schéma LDAP databáze, která byla vytvořena pro testovací účely při vývoji systému, je zobrazeno na obrázku 18.

Pro vyhledávání v databázi je nutné specifikovat jedinečné jméno objektu (*distinguished name*), jehož obsah má být návratovou hodnotou. Jedinečné jméno uživatele **admin** lze rekonstruovat jako `uid=admin,ou=people,dc=ldap,dc=smon,dc=xyz`

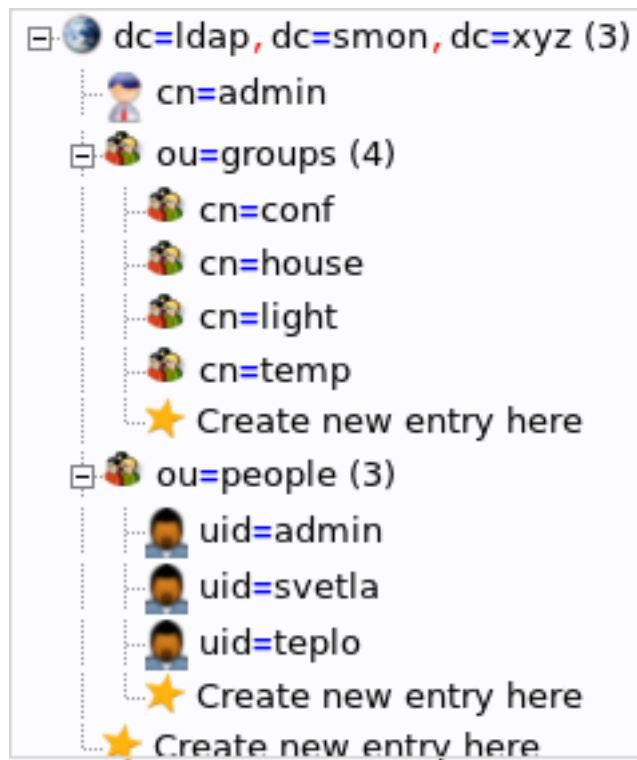
Prvním krokem je ověření uživatelského jména a hesla. Ověření si lze představit jako dotaz bez filtru na LDAP server s přihlašovacími údaji. Pokud je dotaz neúspěšný, nastala během zpracování chyba a do logů serverové aplikace je vypsán chybový kód. V případě úspěšně provedeného dotazu je navrácen chybový objekt - `No such object`.

Příklad jednoduchého dotazu bez filtru:

```
ldapsearch -x -D "uid=svetla,ou=people,dc=ldap,dc=smon,dc=xyz" -W \
-h ldap.smon.xyz
```

Dotaz začíná pomocí utility `ldapsearch`, která je využívána pro vytvoření šifrovaného spojení mezi řídicím serverem a LDAP serverem. Utilita přijímá celou řadu parametrů, které jsou podrobně vysvětleny v její manuálové stránce, popsány budou pouze některé z nich. Parametr `-D` specifikuje, pomocí kterého uživatele se dotaz provede. Pomocí parametru `-h` se nastaví adresa LDAP serveru.

Jedním ze složitějších dotazů (který je ale pro aplikaci stěžejní) je zjištění, zda je zadané uživatelské jméno součástí dané skupiny. Dotaz je možné formulovat pomocí následující kon-



Obrázek 18: Schéma LDAP databáze pro testovací účely

strukce.

```
ldapsearch -x -D "uid=admin,ou=people,dc=ldap,dc=smon,dc=xyz" -W \
  -h ldap.smon.xyz -b "uid=admin,ou=people,dc=ldap,dc=smon,dc=xyz" \
  '(memberof=cn=temp,ou=groups,dc=ldap,dc=smon,dc=xyz)'
```

Oproti předchozímu dotazu přibyl parametr `-b`, který nastaví bázi, v níž prohledávání začne. Následně se pomocí identifikátoru mezi uvozovkami nastaví filtr dotazu. Filtr dotazu je jednoznačným identifikátorem testované skupiny, v ukázkovém dotazu se jedná o skupinu `temp`.

Stejná konstrukce, jako v uvedeném příkladu (viz výše), je použita i v serverové aplikaci. Z odpovědi přijaté ze serveru je pak možné vyčíst, zda je daný uživatel v testované skupině, či nikoliv.

5.2.6.6 Ověření uživatele - lokální

Preferovaný způsob ověření přihlašovacích údajů je proveden pomocí LDAP serveru. Nastavení přístupu k serveru je nicméně nutné provést prostřednictvím konfigurační aplikace. Pro povolení takového scénáře bylo nutné přidat do systému předem definovaného uživatele. Uživatel byl označen jako `root` a je systémem považován za správce, tudíž je tomuto uživateli povoleno provádění všech operací na všech objektech.

Nastavení uživatele je uloženo v konfiguračním xml souboru v cestě `/opt/smon/conf/smon-auth.xml`. V současné implementaci existuje podpora pouze pro jednoho uživatele / administrátora.

5.2.6.7 Práce s identitou uživatele

Každé spojení, které je úspěšně vytvořeno a autorizováno, uchovává seznam skupin, v nichž je uživatel zaveden. V průběhu zpracování dotazu (popsáno v kapitole *Communication Handler*) je provedena kontrola validity. Cílem kontroly je zjistit, zda je možné provést požadovanou operaci na požadovaném objektu. V kladném případě je operace provedena a zpět je odeslána odpověď obsahující požadovaná data. V opačném případě je zpět odesláno hlášení o nepovoleném požadavku.

Seznam objektů spolu s přiřazením povolených operací na skupiny je uveden v konfiguračním xml souboru v cestě `/opt/smon/conf/smon-groups.xml`. Část souboru je zobrazena na obrázku 19.

```
<?xml version="1.0"?>
<objects>
  <permission-all>
    <group>conf</group>
  </permission-all>
  <object>
    <name>thermostat</name>
    <get>
      <group>temp</group>
    </get>
    <set>
      <group>temp</group>
      <group>conf</group>
    </set>
  </object>
```

Obrázek 19: Obsah souboru mapování skupin na povolené operace

Kořenový element `objects` zahrnuje seznam všech objektů, které je možné přenášet.

Element `permission-all` označuje takzvanou super skupinu, každý uživatel přiřazený do této skupiny může provést libovolnou operaci.

Následuje popis jednoho z přenášených objektů. Obsah elementu `name` jednoznačně identifikuje objekt, ke kterému se přiřazení vztahuje, a musí souhlasit se jménem, pod kterým je objekt v systému registrován. Poté jsou vyjmenovány jednotlivé operace, u nichž má dojít ke změně přístupu.

5.2.6.8 Řízení technologie

Část popisující řízení technologie není v této práci příliš rozvinuta. Předpokládané využití navrhovaného systému je takové, že bude napojen do již stávající technologie. Řízení technologie je tedy omezeno na jednoduché příkazy, které by byly předány již existujícím systémům řízení.

Nejjednodušším způsobem předání příkazu je sepnutí kontaktu. Pomocí tohoto způsobu je možná dvoustavová signalizace vypnuto / zapnuto. Celý princip je možné ukázat na subsystému řízení vytápění, který je v práci kompletně implementován.

V průběhu počáteční sekvence serverové aplikace je spuštěno vlákno, které periodicky načítá aktuální nastavení termostatu. V případě, že je kompletně nastaveno řízení teploty v libovolné místnosti, je provedeno porovnání aktuálně měřené teploty s teplotou požadovanou. Pokud je požadovaná teplota vyšší než nastavená, je vyslán příslušný impuls do subsystému řízení výstupů. Hodnota požadované teploty je upravena tak, aby v systému nedocházelo ke kmitání vlivem kolísání teploty v okolí nastavené hranice. Zvoleny jsou dvě hranice, vyšší z nich vznikne přidáním $0.2^{\circ}K$, nižší z nich odebráním $0.2^{\circ}K$ od požadované teploty. Vznikne tak pásmo teploty, ve kterém je vyslání požadavku - topení - blokováno i přes to, že je aktuální teplota nižší než nastavená.

5.2.7 Konfigurační aplikace

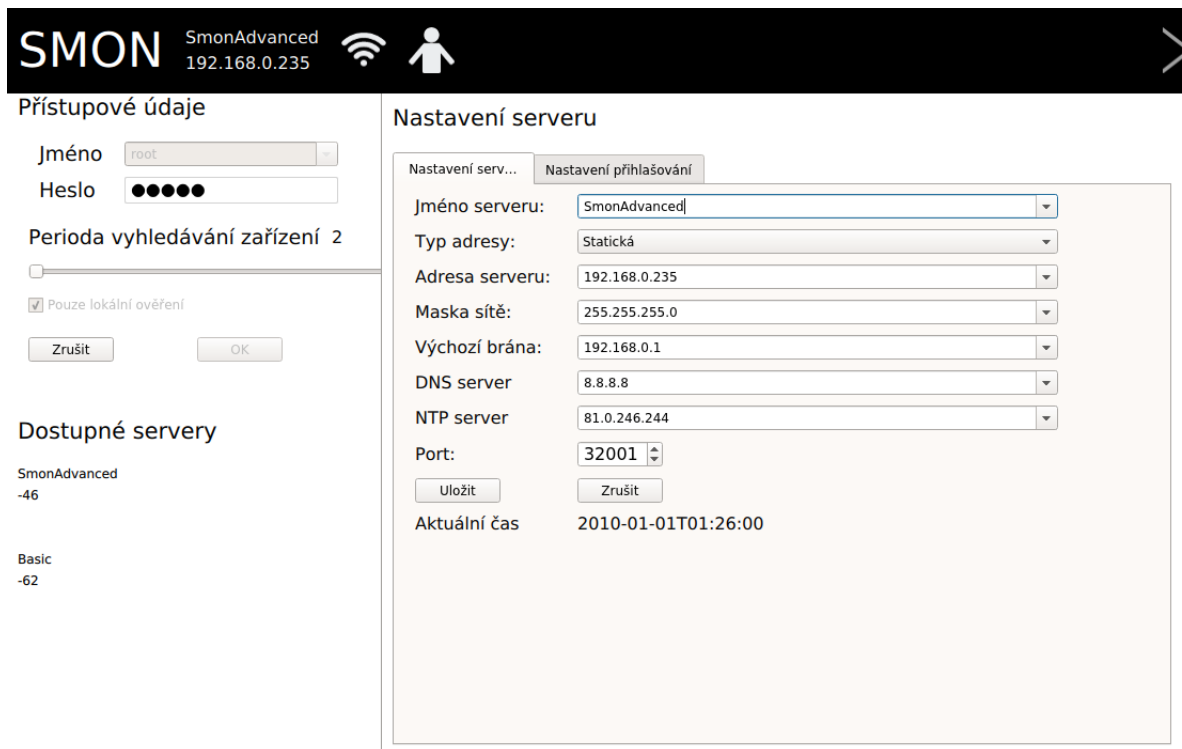
Konfigurační aplikace je základní aplikací poskytující grafické rozhraní, která slouží ke kompletnímu nastavení systému. Aplikace se skládá ze dvou hlavních obrazovek. První z nich je přihlašovací obrazovka 20. Po úspěšném spojení a přihlášení k řídicímu serveru dojde automaticky k přechodu na druhou obrazovku, která již umožňuje vlastní nastavení systému.

Obě obrazovky jsou navíc vertikálně rozděleny na levý navigační panel, který je užší, a pravý panel, v němž je zobrazen vlastní obsah.

5.2.7.1 Horní informační panel

V horním panelu je zobrazeno jméno serveru, ke kterému je aplikace v danou chvíli připojena. Identifikace je navíc doplněna o adresu daného serveru. Dále jsou zde zobrazeny ikony informující o aktuálním stavu spojení.

Ve chvíli připojování k řídicímu serveru, kdy se přenáší základní data, je zobrazena ikona představující probíhající přenos dat prostřednictvím Bluetooth. Tento stav ale trvá vždy pouze pár vteřin v průběhu připojování, a není proto na obrázku zachycen. Další ikonou je WiFi ikona, označující stav, kdy je aplikace spojena s řídicím serverem prostřednictvím sítě internet. Protože je vytvořené spojení udržováno aktivní po celou dobu běhu aplikace, je i ikona zobrazena po



Obrázek 20: Rozložení přihlašovací obrazovky konfigurační aplikace

celou dobu. Poslední ikona ve tvaru lidského těla označuje stav, kdy je daný uživatel úspěšně přihlášen. Tato ikona je rovněž zobrazena po celou dobu běhu aplikace.

V levém a pravém rohu se dále zobrazují navigační šipky v závislosti na pozici v aplikaci.

5.2.7.2 Levý navigační panel

Levý panel by měl sloužit jako navigační, nicméně na přihlašovací obrazovce je jeho funkcionality využito jiným způsobem. V panelu je zobrazeno několik polí pro vstup textu, jmenovitě uživatelského jména a hesla. Dále je možné posuvníkem zvolit periodu vyhledávání zařízení, jak je popsáno v kapitole *Communication Handler*. Dalším prvkem určeným k přihlašování je zaškrtačací tlačítko, pomocí kterého je možné ovlivnit přihlašovací proces tak, aby došlo pouze k lokálnímu ověření. Dále je zobrazen seznam dostupných serverů spolu s naměřenou silou signálu. Připojení probíhá vždy k nejbližšímu z nich.

Po úspěšně vytvořeném spojení se serverem přejde aplikace na druhou hlavní obrazovku, kde je již v levém panelu zobrazen seznam nastavených řídicích prvků. Tento seznam je zobrazen na obrázku 21. Nové prvky lze přidávat pomocí dialogu pro přidání. Odebírání prvků je realizováno kliknutím na šedý čtverec v pravé části prvku. Aktuálně vybraný prvek je zvýrazněn žlutým pozadím.

5.2.7.3 Pravý panel obsahu

Pravý panel obsahu je na přihlašovací obrazovce využit pro obecné nastavení řídicího serveru. Obecným nastavením se rozumí nastavení adresy (IP), případně adres dalších služeb, například synchronizace času (NTP) nebo služby překladu doménových jmen (DNS). Ve druhé záložce nastavení jsou pak umístěna vstupní pole pro nastavení připojení k LDAP serveru. Příklad výchozích nastavení v průběhu testování je shrnut v tabulce 3.

Jméno serveru	SmonAdvanced
Typ adresy	Statická
Adresa serveru	192.168.0.235
Maska sítě	255.255.255.0
Výchozí brána	192.168.0.1
DNS server	8.8.8.8
NTP server	81.0.246.244
Port	32001

Typ ověřování	LDAP
Adresa	104.167.98.146
Identita	uid=svetla,ou=people, dc=ldap,dc=smon,dc=xyz
Skupina	cn=temp,ou=groups, dc=ldap,dc=smon,dc=xyz

Tabulka 3: Výchozí hodnoty nastavené v průběhu testování systému

Po připojení a přechodu na druhou obrazovku je následně zobrazen panel s obsahem právě vybraného objektu. Tento stav je rovněž zachycen na obrázku 21.

5.2.7.4 Panel nastavení vytápění

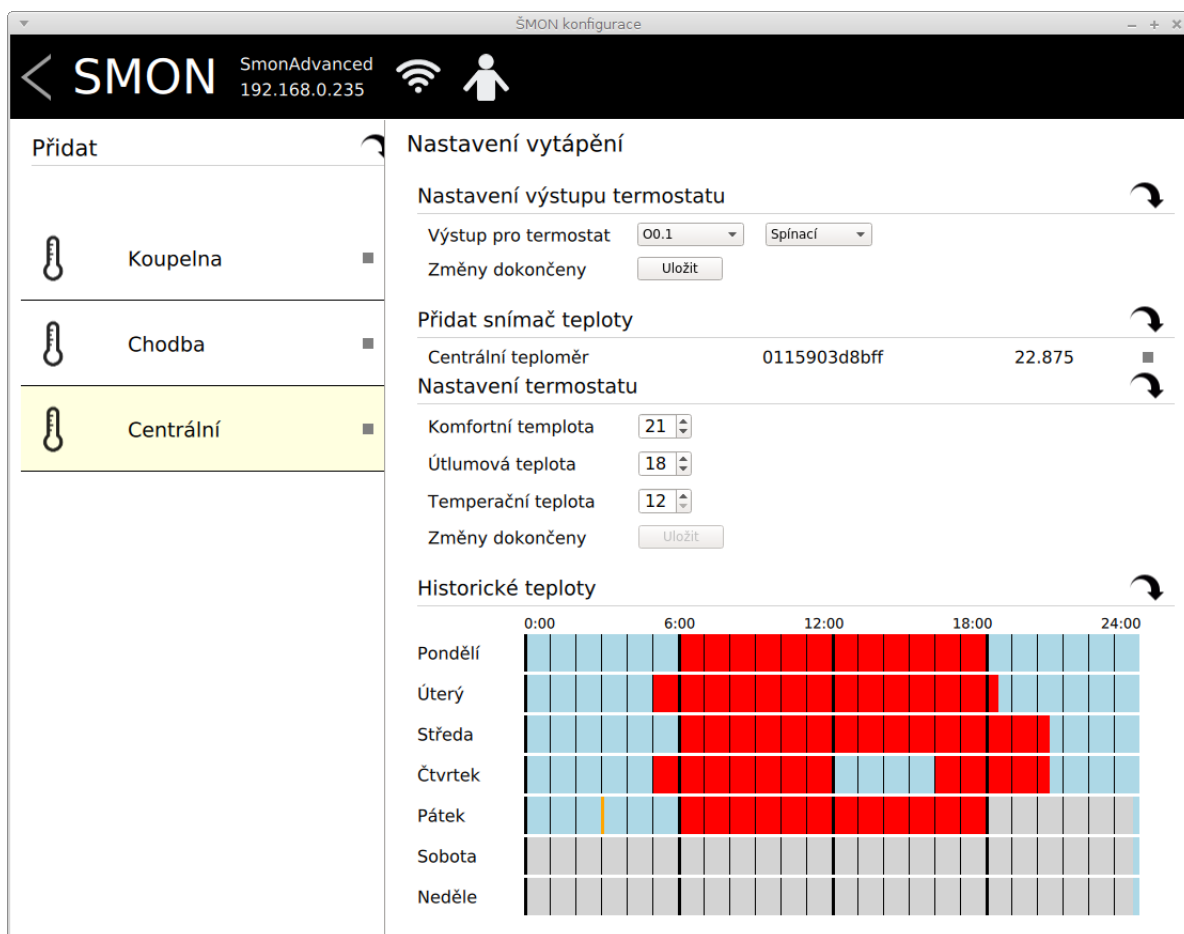
V systému je kompletně realizován systém vytápění, který je navržen na modelu jednoduchého programovatelného termostatu. Panel nastavení je popsán podle obrazovky z obrázku 21.

Nejprve je nutné nastavit výstup, který bude použit pro signalizaci požadavku topení. Tento výstup je možné nastavit buď jako spínací nebo rozpínací. V případě spínacího výstupu bude výstup aktivní, pokud vznikne požadavek na vytápění. Opačná situace bude platit, bude-li bude nastaven jako rozpínací.

Dalším krokem je registrace snímače teploty pro danou zónu. Podporované snímače teploty jsou v současné implementaci pouze ty, které jsou připojeny na *OneWire sběrnici*. V dialogovém okně registrace snímače teploty je nutné vybrat jeden z dostupných a přiřadit mu jednoznačné jméno. V hlavním panelu je pak u snímače uvedeno jméno, jeho adresa a aktuální měřená teplota.

Předposledním krokem je nastavení tří úrovní teplot. Nejvyšší úroveň je *komfortní teplota*, vyjadřující požadovanou teplotu v době, kdy je daná místnost využívána. Další dvě úrovně

- *útlumová teplota a temperační teplota* - vyjadřují požadované teploty v případě krátkodobé nepřítomnosti (přes noc), případně dlouhodobou nepřítomnost (například týdenní dovolená) [29]. Díky možnosti nastavení tří teplot je možné optimalizovat spotřebu tepla v době, kdy není nikdo přítomen na pracovišti, případně plánovaně snížit teplotu o víkendy.



Obrázek 21: Rozložení obrazovky řízení teploty

Nastavení teplot termostatu je při zadávání omezeno tak, aby nebylo možné zadat vyšší teplotu pro nižší stupeň vytápění. V případě pokusu o zadání takovéto teploty je automaticky zvýšena nastavená teplota nižší úrovně. Limity pro nastavení teploty jsou shrnuty v tabulce 4.

Úroveň	min [°C]	max [°C]
Komfortní	18	26
Útlumová	16	21
Temperační	12	17

Tabulka 4: Omezení pro nastavení teplotních úrovní

Posledním krokem nastavení je přiřazení jednotlivých úrovní do časového rozvrhu. V příkladovém

obrázku 21 je možné vidět červeně nastavenou komfortní úroveň každý pracovní den mezi 5:00 až 20:30. Začátek i konec vytápění je každý den jiný, tak jak by bylo požadováno. Vzhledem k tomu, že pátek by byl v nastaveném případě posledním pracovním dnem, bylo by možné ihned po skončení pracovní doby snížit požadovanou teplotu na úroveň temperační teploty. Tato teplota by byla udržována přes celý víkend až do nedělní půlnoci, kdy by došlo k přepnutí na vyšší útlumovou úroveň a tím k postupnému vytápění prostoru.

V průběhu tvorby rozvrhu vytápění je nutné pamatovat na to, že po každém snížení teploty místnosti, například na temperační úroveň, dojde k vychlazení nejen vzduchu uvnitř místnosti, ale současně s tím i všech předmětů v místnosti umístěných. Opětovné vytopení celé místnosti na komfortní úroveň pak bude trvat déle a zároveň dojde ke zvýšení nákladů na vytápění. Ze stejného důvodu není doporučeno, aby rozdíl teplot mezi komfortní a útlumovou teplotou nebyl větší než $2 - 3^{\circ}K$ [29].

5.2.8 Uživatelská aplikace

Uživatelská aplikace je druhou aplikací s grafickým rozhraním. Na rozdíl od aplikace konfigurační nenabízí uživatelská aplikace nastavení údajů řídicího serveru jako například jeho jméno, adresu, případně styl ověření uživatele. Naproti tomu uživatelská aplikace nabízí zjednodušené rozhraní vhodné pro běžné ovládání systému. Rozložení aplikace je velmi podobné jako v konfigurační aplikaci. V horní části je informační panel, který informuje o stavu aplikace. Horní panel byl podrobně popsán v části věnované konfigurační aplikaci, a proto zde není popisován znovu.

Hlavní část aplikace už není vertikálně rozdělena na navigační a obsahový panel, ale obsahuje pouze jeden hlavní panel. Tento hlavní panel je nutné využít jednak pro navigaci v aplikaci, stejně tak i k zobrazení vlastního obsahu. Uživatelská aplikace je rozdělena na tři hlavní obrazovky, na kterých se uživatel pohybuje. První obrazovkou je stejně jako v případě konfigurační aplikace přihlašovací obrazovka. Po úspěšném přihlášení přejde aplikace na druhou obrazovku, na níž se nachází seznam nastavených objektů. Při výběru libovolného z těchto objektů pak aplikace přejde na třetí obrazovku, kde zobrazí informace o zvoleném objektu.

Na obrázku 22a a 22b jsou zobrazeny dvě obrazovky aplikace, mezi nimiž se přechází.

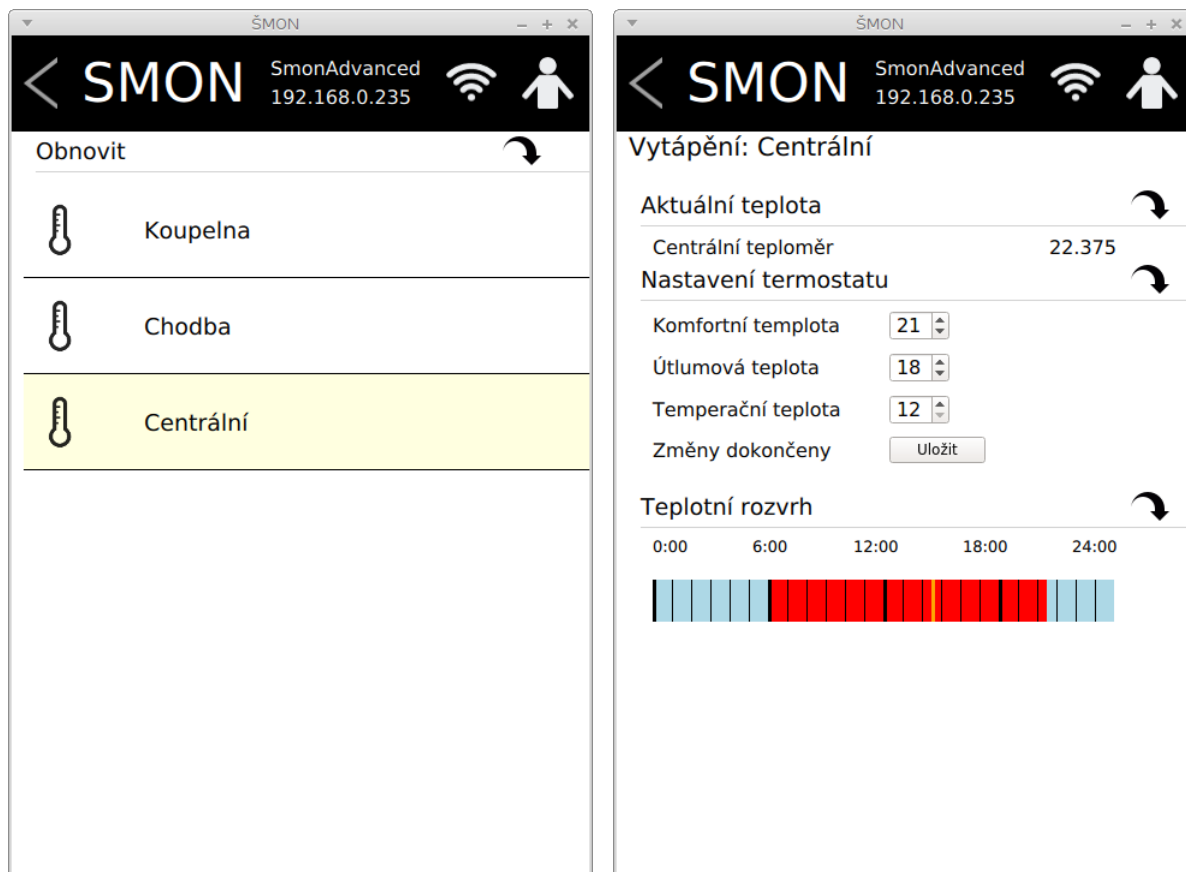
5.2.8.1 Rozdíly na přihlašovací obrazovce

Na přihlašovací obrazovce chybí tlačítko na výběr typu ověření. Toto tlačítko zde není přítomno proto, aby uživatel nemohl ovlivnit způsob přihlášení, ale musel využít takového stylu přihlášení, jaký zadal správce systému v průběhu jeho nastavení.

Kromě tohoto menšího rozdílu jsou přihlašovací obrazovky v obou aplikacích identické.

5.2.8.2 Navigační obrazovka

Na navigační obrazovce 22a je zobrazen pouze seznam nastavených objektů. Pomocí této aplikace není možné přidávat nové objekty, případně staré objekty mazat. Poslední použitý objekt je opět zvýrazněný pomocí žlutého pozadí.



(a) Rozložení obrazovky řízení teploty

(b) Rozložení obrazovky řízení teploty

Obrázek 22: Navigační a obsahová obrazovka uživatelské aplikace

5.2.8.3 Obrazovka ovládání termostatu

Podobně, jako je omezena funkcionalita na obrazovce přihlašování, je omezena i funkcionalita na hlavní obrazovce nastavení termostatu. Obrazovka je na obrázku 22b.

Na obrazovce se již nezobrazuje adresa teplotního čidla, ale pouze jeho pojmenování spolu s aktuální teplotou. Informace o přiděleném jménu teploměru spolu se jménem místnosti by měla být dostatečná v případě hlášení problému s měřením teploty.

Blok nastavení tří úrovní termostatu zůstal stejný jako v konfigurační aplikaci. Nastavení teplot jednotlivých úrovní je možné pomocí této aplikace změnit.

Část věnovaná teplotnímu rozvrhu je také značně omezena. V uživatelské aplikaci je zobrazen

pouze aktuální rozvrh, podle kterého se systém řídí. Aktuální čas je vyznačen pomocí oranžové čáry. Změna teplotního rozvrhu není z této aplikace podporována.

6 Testovací zapojení

Důležitým aspektem popsaného systému je funkce automatického připojení k lokálnímu řídicímu serveru, který poskytuje služby pro ovládání dané části budovy. Po účely testování bylo použito výše zmíněné zapojení schematicky znázorněné na obrázku 25.

6.1 Statické měření síly signálu

Síla signálu, který je vysílán Bluetooth moduly, je závislá na nastavení těchto modulů. Toto nastavení je možné změnit pomocí takzvaných AT příkazů. Dle datasheetu modulu [19] je možné nastavit sílu signálu od -23 dBm do 6 dBm. Výchozí hodnotou je pak síla signálu 0 dBm, s níž bylo měření provedeno. Řídicí servery byly vždy umístěny co nejbližší středu dané zóny, tak jak je vyznačeno na obrázku 25.

Pro měření síly signálu bylo v prvním kole využito externí aplikace pro operační systém Android, pojmenované **BLE Scanner** vyvinuté společností **Bluepixel Technologies LLP**. Ve druhém kole byla již k měření využita konfigurační aplikace vyvinutá k tomuto účelu. Obě aplikace dosahovaly srovnatelných výsledků měření.

Měření probíhalo na 18 vyznačených bodech, tak aby byl rovnoměrně pokryt celý řízený prostor. Na obrázku 26 jsou tečkami vyznačena místa, ve kterých měření probíhalo. Měření bylo prováděno vždy se stejným přístrojem. Po změně polohy se vždy vyčkalo 5 - 10 s, než došla první vyhledávací fáze. Po této kratší prodlevě byly vždy zapisovány údaje o síle signálu z obou řídicích serverů současně. Výsledky měření jsou shrnuty v tabulce 5.

Z výsledků měření je možné usoudit, že lze uskutečnit lokalizaci pomocí této technologie. V zónách, které jsou blízké některému ze serverů, lze pozorovat zásadní rozdíly v síle signálu mezi bližším a vzdálenějším serverem. Tento jev je velmi dobře pozorovatelný v bodech měření 5,6,7, případně 14,15,16. Jakmile se vzdálenost od vysílacího bodu začne zvyšovat, dochází ke zmenšení rozdílu síly signálu. Tento jev lze pozorovat v bodech měření 8, 9, 10, 11, případně 12. V těchto bodech měření hraje velkou roli především spolu s množstvím překážek, které jsou postaveny mezi vysílač a přijímač, také síla a propustnost stěn. V těch bodech měření, mezi nimiž byla vzdálenost serverů srovnatelná, a kde navíc byly oba servery skryty za zdmi ostatních pokojů, docházelo k vyrovnání síly signálu od obou serverů. Tento jev je dobře pozorovatelný v bodech měření 3, 4, 9, 18, případně 17. Nicméně tyto body měření byly umístěny na chodbě, kde by se měl nacházet další řídicí server.

#	Úroveň signálu [dBm]	Průměr [dBm]	#	Úroveň signálu [dBm]	Průměr [dBm]
1	A: -72, -70, -81, -73	-74	10	A: -78, -80, -78, -83	-80
	B: -91, -93, -90, -92	-91		B: -79, -74, -73, -72	-74
2	A: -88, -75, -79, -83	-81	11	A: -76, -78, -70, -73	-74
	B: -82, -83, -82, -78	-81		B: -74, -70, -74, -71	-72
3	A: -82, -77, -78, -77	-78	12	A: -82, -80, -80, -84	-81
	B: -71, -75, -71, -72	-72		B: -81, -77, -81, -74	-78
4	A: -69, -67, -78, -79	-73	13	A: -86, -79, -90, -78	-83
	B: -81, -78, -70, -79	-77		B: -70, -64, -65, -67	-66
5	A: -63, -62, -64, -63	-63	14	A: -76, -80, -89, -77	-80
	B: -84, -83, -87, -87	-85		B: -61, -64, -60, -61	-61
6	A: -51, -53, -55, -54	-53	15	A: -89, -84, -82, -86	-85
	B: -74, -79, -85, -81	-80		B: -65, -64, -66, -65	-65
7	A: -62, -62, -64, -64	-63	16	A: -91, -82, -82, -83	-85
	B: -83, -81, -82, -80	-81		B: -62, -72, -64, -73	-67
8	A: -70, -72, -71, -71	-71	17	A: -70, -79, -74, -82	-76
	B: -75, -69, -69, -70	-70		B: -73, -70, -81, -67	-72
9	A: -78, -79, -81, -82	-80	18	A: -85, -83, -81, -84	-83
	B: -85, -76, -77, -79	-80		B: -81, -85, -83, -83	-83

Tabulka 5: Staticky naměřené hodnoty v bodech dle obrázku 26.

Při simulovaném přesunu jednoho ze serverů z původního stanoviště na chodbu došlo k výraznému zlepšení situace. Tento simulovaný stav měření ale nelze brát jako směrodatný, neboť vlivem přesunu jednoho ze serverů nedocházelo ke vzájemnému ovlivňování tří měřených stanovišť.

Z výsledků měření jednoznačně vyplývá skutečnost, že pokrytí pouze dvěma servery bude v daném zapojení nedostatečné.

6.2 Dynamické chování systému

Pojmem dynamického chování systému se rozumí jeho chování v reálném zapojení. Z naměřených hodnot je možné vysledovat špatná a dobrá místa uvnitř budovy. Mezi horší místa budou patřit ta, na nichž není rozdíl síly signálu mezi dvěma servery dostatečně velký a stabilní. V případě, že rozdíl signálů bude kolísat, může docházet k častému přepojování mezi dvěma servery. Apli-

kace by se v takovém případě stala nepoužitelnou, neboť by nebylo možné nabízené technologie ovládat.

Cílem tohoto pozorování je určit taková místa, na nichž se aplikace bude chovat nepředvídatelně. Pokud to bude možné, tak i navrhnout takové řešení, které by zabránilo nepředvídatelným přepnutím.

Již z naměřených hodnot jasně vyplývá, že problematická místa budou v okolí vstupu do bytu, kde by bylo vhodné umístit další - třetí server. Úkolem tohoto třetího serveru by byla obsluha akcí spojená s odchodem, případně příchodem uživatelů. Typická akce při odchodu by mohla být uzamčení a zastřežení bytu, případně vyslání příkazu ostatním serverům o opuštění bytu a přechodu do úsporného režimu. V instalaci s pouze dvěma servery není úplně jasné, který z nich by se o tyto akce měl starat, neboť oba servery jsou umístěny v jiných místnostech.

Statickým pohledem na naměřené výsledky je možné odhalit i místa, která nejsou tak zřejmá, avšak mohl by v nich nastat problém. Jedná se především o místa v nejvzdálenějších rozích místností. Úrovně signálů jsou v těchto místech téměř stejné, a mohlo by zde docházet k přepojování mezi servery. Provozem a pozorováním se nicméně tyto obavy nepotvrdily, neboť systém byl nastaven tak, že k přepojení na jiný server dochází až ve chvíli, kdy je rozdíl signálů větší než $10dBm$. Při příchodu do místnosti došlo vždy k připojení ke správnému serveru a po usazení do rohu místnosti již nedocházelo k dalším přepojováním.

V případě, že by se systém nechoval správně, by bylo možné algoritmus přepojení rozšířit o další doplňující podmínku. Touto podmínkou by bylo dlouhodobé sledování síly signálu. Z další podrobné analýzy by bylo možné určit, zda není v okolí bližší server. Při správném nastavení by pak bylo možné vybrat lépe i v případě, že jsou signály z obou serverů téměř stejné.

7 Závěr

Cílem práce byl návrh frameworku pro ovládání technologií budov s lokální nabídkou služeb, jeho realizace a otestování. Na počátku práce byla provedena základní rešerše běžně dostupných a používaných řešení. Cílem rešerše bylo zjistit, zda je podobná technologie již vyvinuta a používána, či nikoliv. Výsledek rešerše byl negativní, v nabízených systémech nebyl nalezen takový, který by takovou funkci podporoval.

Dále byla provedena rešerše s cílem nalézt dostupné technologie určené k přibližné lokalizaci uživatelů uvnitř budov. Rešerší byla shledána technologie Bluetooth jako nejvíce nadějnou. Vzhledem k nižším přenosovým rychlostem, kterých je možné s pomocí Bluetooth rozhraní dosáhnout, bylo již od počátku zamýšleno, že komunikace bude podpořena prostřednictvím běžného internetového připojení. Rešerše dále odhalila, že většina dostupných chytrých telefonů již disponuje kombinovaným čipem s podporou WiFi a Bluetooth technologie s podporou Bluetooth Low Energy standardu. Díky této podpoře je tento způsob vylepšení dostupný všem uživatelům běžných chytrých telefonů.

Vzhledem k uzavřenosti aktuálně dostupných systémů domácí automatizace bylo nutné navrhnout vlastní koncepci systému tak, aby bylo možné otestovat možnosti lokalizace. Systém je postaven na základě jednodeskového počítače s podporou operačního systému. Operační systém je doplněn o sadu vylepšení podporujících zpracování úloh v reálném čase. Díky těmto vylepšením by v systému nemělo docházet k negativním zpomalením odezvy. Byly vyvinuty dvě rozšiřující desky, přičemž do hlavní desky je možné zapojit jednodeskový počítač. Hlavní rozšiřující deska je vybavena sadou komunikačních rozhraní a konektorem pro zapojení druhé rozšiřující desky. Na druhé desce jsou osazeny opticky oddělené vstupy a výstupy.

Pro účely testování lokalizace byly vyvinuty dvě aplikace určené pro ovládání prostřednictvím mobilních terminálů. Pomocí první aplikace je možné celý systém nastavit, přičemž je určena pro správce systému. Druhá aplikace obsahuje jednodušší rozhraní a je určena pro běžné uživatele, rovněž však podporuje úpravu základních hodnot v nastavení systému. Obě aplikace jsou vyvinuty pomocí frameworku Qt. Jejich funkčnost byla otestována na platformách mobilních telefonů Android a desktopové platformě Linux. Funkcionalita na ostatních platformách nebyla vyzkoušena, nicméně měla by být zaručena.

Na straně serveru byla vyvinuta aplikace, která umožňuje základní komunikaci prostřednictvím sítě Bluetooth a prostřednictvím běžného drátového spojení. Komunikace je zabezpečena pomocí šifrování protokolem SSL (TLS 1.2) a identita uživatele je ověřována prostřednictvím centrální databáze uživatelů na serveru LDAP. V aplikaci byl plně realizován subsystém zónového vytápění

a to včetně možnosti tvorby rozvrhu vytápění. Aplikace rovněž disponuje perzistentní pamětí pro nastavení systému.

Měření prokázalo, že lze pomocí takto navrženého systému určit polohu uživatele s dostatečnou přesností, aby mu byly poskytnuty pouze technologie v místě dostupné.

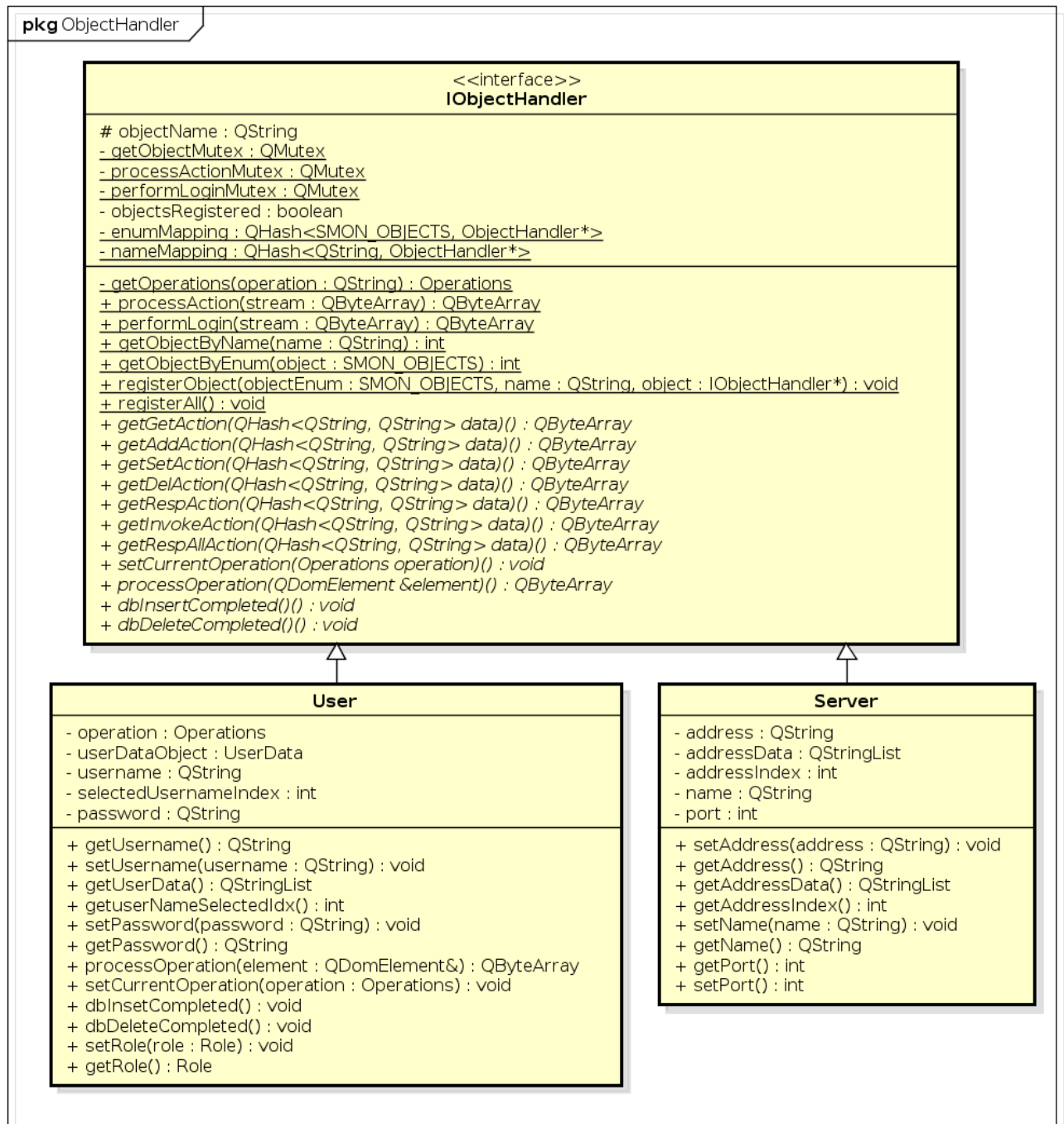
Budoucí rozšíření systému je možné především o prvky podporující komunikaci s již existujícími systémy tak, aby tento navržený systém fungoval pouze jako přístupový bod sítě. Toto rozšíření by umožnilo nasazení tohoto konceptu systému do již existujících instalací.

Reference

- [1] Merz, H.; Hansemann, T.; Hübner, Ch. *Automatizované systémy budov*; Grada, 2007.
- [2] Michalec, L. Topologie KNX, 2013. Profesionální elektronika. <http://vyvoj.hw.cz/energetika-a-smart-grids/topologie-knx.html> (accessed Nov 22, 2016).
- [3] *Introduction to the LonWorks Platform*, 2nd ed.; Echelon, 2009.
- [4] *LONWORKS TECHNOLOGY OVERVIEW*, 1st ed.; Echelon, 2009.
- [5] Sběrnice LonWorks - 2.část - LonTalk protokol. <http://automatizace.hw.cz/clanek/2005041101> (accessed Dec 10, 2016).
- [6] Klaban, J. Inels a sběrnice CIB – moderní systém inteligentní elektroinstalace. *Automa* 2008, 12, 70–71.
- [7] *CIB-tech Installation Manual* [online]. Intelligent Building Solution, 2011 [cited 13 Dec 2016].
- [8] Central unit CU3-01M, CU3-02M. http://www.elkoep.com/fileadmin/produkty/Elko/INELS-III/CU3-02M/EN/Datasheet_CU3-0xM.pdf (accessed Dec 13, 2016).
- [9] NAVEED, A. Ch. NFC - Vulnerabilities and Defense. *Conference on Information Assurance and Cyber Security (CIACS)*, 2014,
- [10] D. Plets et al., *Calculation tool for optimal wireless design and minimal installation cost of indoor wireless LANs*, 2014 IEEE Antennas and Propagation Society International Symposium (APSURSI), Memphis, TN, 2014, pp. 1165-1166.
- [11] IEEE 802.15.1. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*. IEEE, 2002. 1169 p.
- [12] Bluetooth Low Energy. https://en.wikipedia.org/wiki/Bluetooth_low_energy (accessed Nov 23, 2016).
- [13] ON Semiconductor LM2575, NCV2575 1.0 A, Adjustable Output Voltage, Step-Down Switching Regulator. 2009, p. 1-28.
- [14] Cubietech Cubieboard. <http://linux-sunxi.org/Cubieboard> (accessed Dec 13, 2016).
- [15] Cosmo Photo MOS Relay KAQV214. 2000, p. 1-7.
- [16] Fairchild Semiconductor 6-Pin DIP Random-Phase Optoisolators Triac Driver Output (250/400 Volt Peak). 2009, p. 1 - 12.

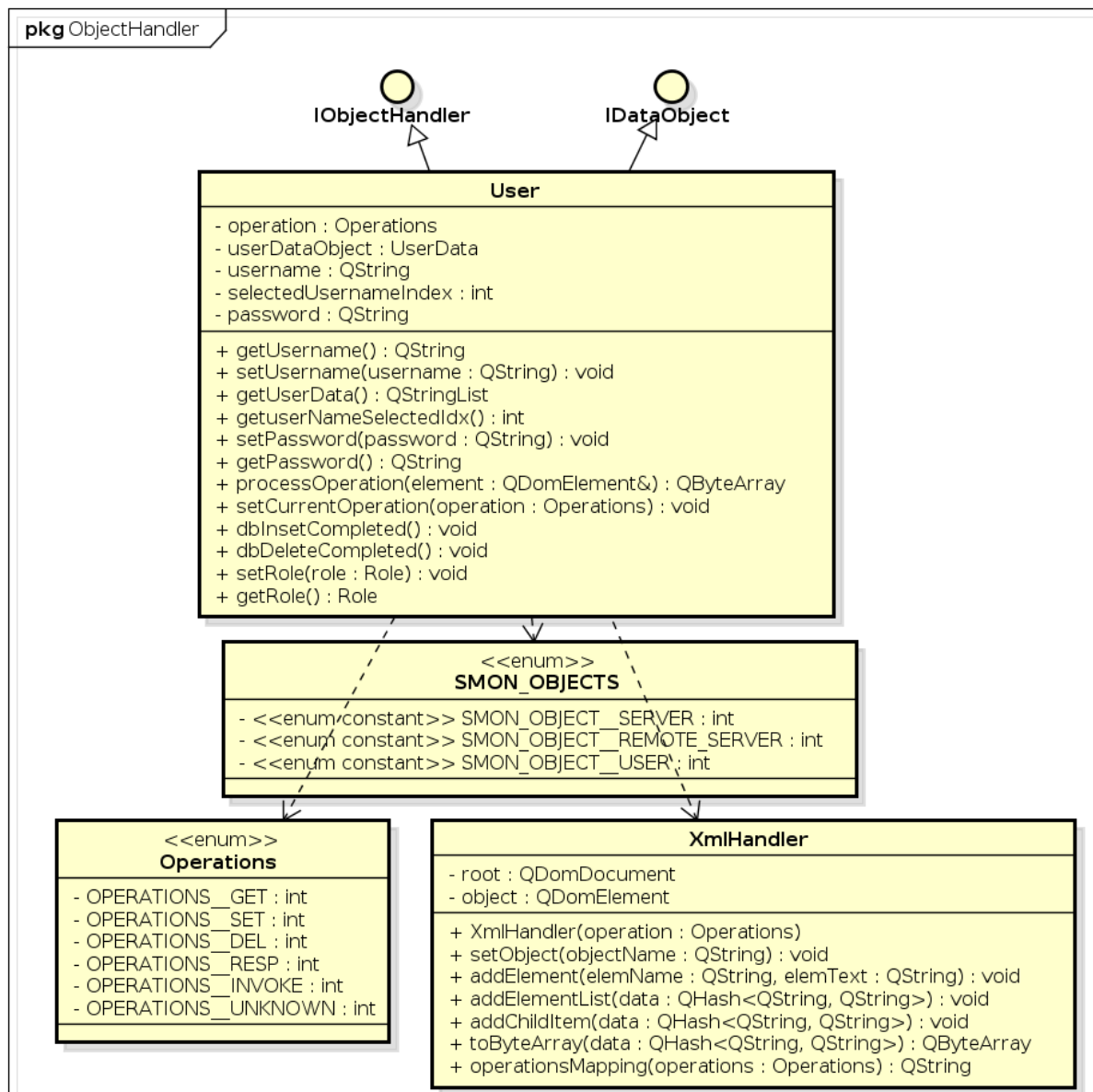
- [17] MaximIntegrated Programmable Resolution 1-Wire Digital Thermometer. 2015, p. 1-20.
- [18] MaximIntegrated DS2482-100 Single-Channel 1-Wire Master. 2015, p. 1-24.
- [19] *2.4-GHz Bluetooth low energy and Proprietary System-on-Chip*. 2013, p. 1-29.
- [20] SUNXI. http://linux-sunxi.org/Main_Page (accessed Dec 08, 2016).
- [21] LDAP Specs. <https://www.ldap.com/ldap-specs> (accessed Oct 15, 2016).
- [22] Signals & Slots. <http://doc.qt.io/qt-5/signalsandslots.html> (accessed Oct 15, 2016).
- [23] Using the Meta-Object Compiler (moc). <http://doc.qt.io/qt-5/moc.html> (accessed Oct 15, 2016).
- [24] Bluetooth Low Energy. <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html> (accessed Dec 10, 2016).
- [25] Android 5.0 APIs. <https://developer.android.com/about/versions/android-5.0.html> (accessed Dec 10, 2016).
- [26] QAndroidJniObject Class. doc.qt.io/qt-5/qandroidjniobject.html (accessed Dec 10, 2016).
- [27] QSqlDriver. <http://doc.qt.io/qt-5/qsqldriver.html> (accessed Dec 05, 2016).
- [28] Online Backup of a Running Database. <https://www.sqlite.org/backup.html> (accessed Dec 05, 2016).
- [29] Bašta, J.; Hemzal, K. *REGULACE V TECHNICE PROSTŘEDÍ STAVEB*, 1st ed.; Praha, 2009.

A UML diagram ObjectHandler



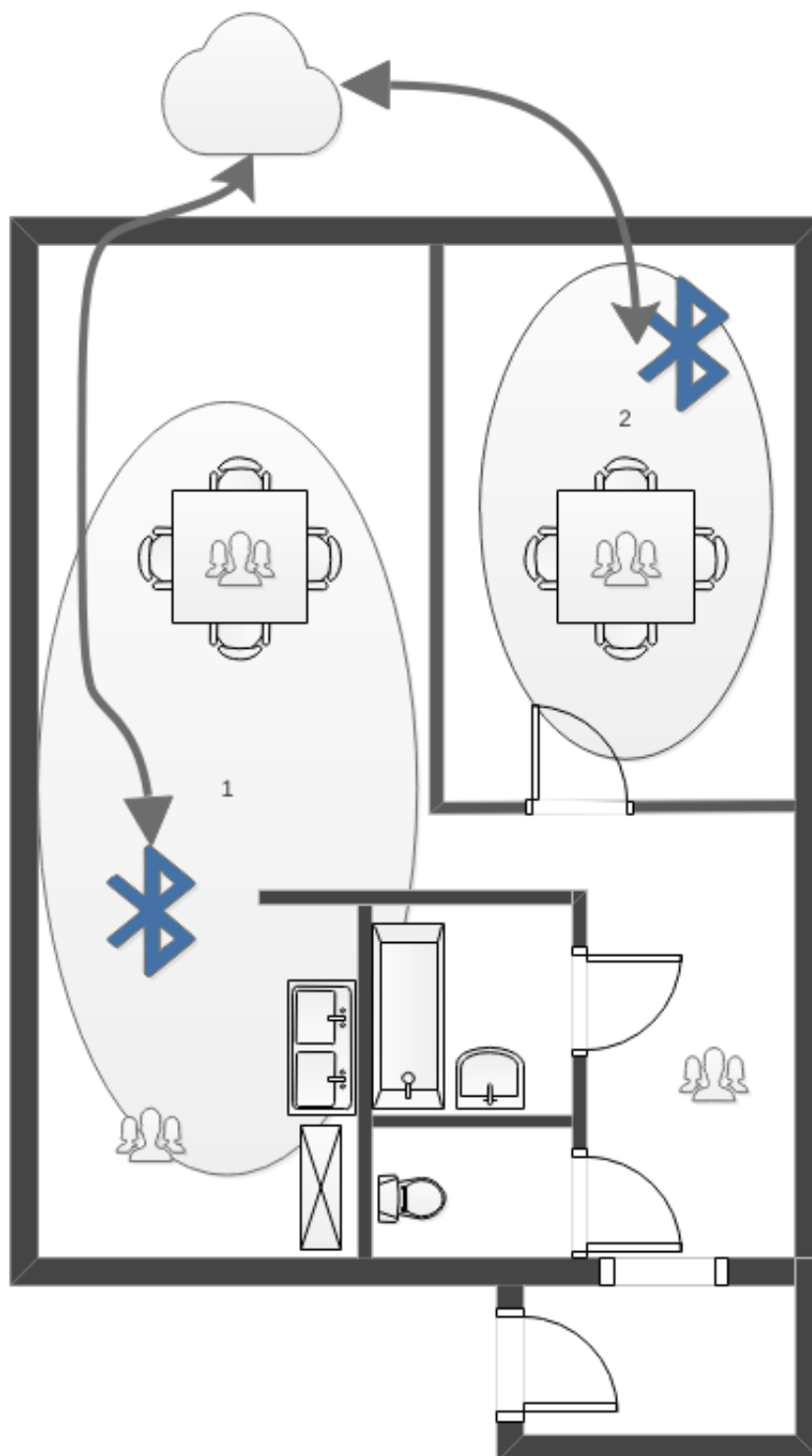
Obrázek 23: ObjectHandler abstraktní třída

B UML diagram ObjectHandler pro objekt typu User



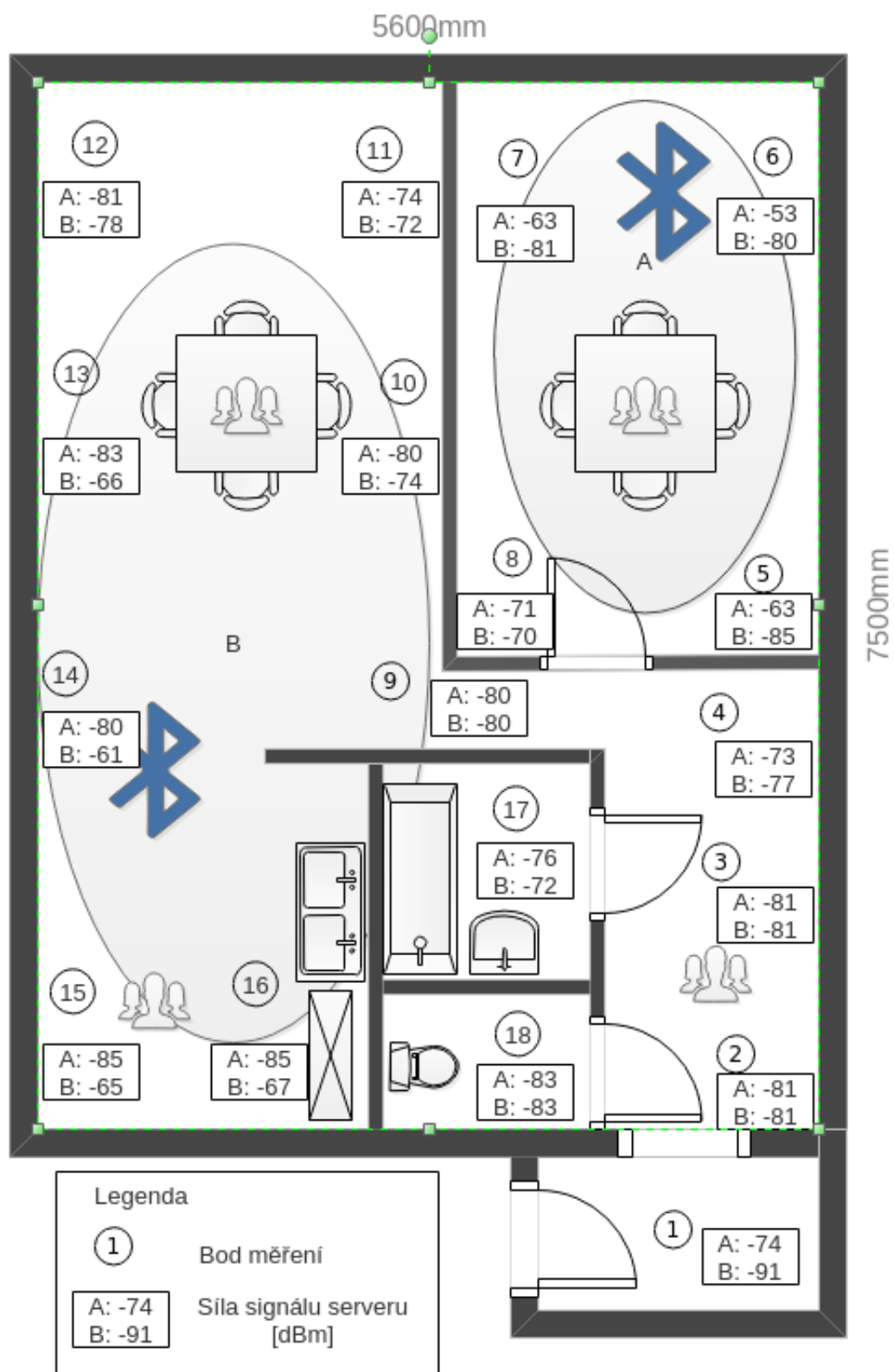
Obrázek 24: UML diagram třídy User

C Testované zapojení systému



Obrázek 25: Testované zapojení systému

D Naměřené hodnoty testovaného zapojení systému



Obrázek 26: Měřené body v testovaném zapojení