

**Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta**

**Řešení ovládání inteligentního domu pomocí systému
pro heterogenní zařízení**

Bakalářská práce

Stanislav Dvořák

Vedoucí práce: Mgr. Jiří Pech, Ph. D.

České Budějovice 2016

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Student: Stanislav Dvořák
(jméno, příjmení, tituly)

Obor – zaměření studia: 1801R001 / Aplikovaná informatika
.....

Školitel: Mgr. Jiří Pech
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PřF:
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant:
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma bakalářské práce: Řešení ovládání inteligentního domu pomocí systému pro heterogenní zařízení.

Cílem práce je návrh a sestavení systému, který bude ovládat inteligentní dům pomocí zařízení od různých výrobců. Práce by měla splňovat následující body:

1. Sestavení a popis množiny zařízení pro ovládání inteligentního domu a volba vhodné zastřešující platformy.
2. Konstrukce funkčního systému na základě předchozího bodu a popis řešení.

Základní doporučená literatura :

KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IP a systémem DNS. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.

PINKER, Jiří. Mikroprocesory a mikropočítače. 1. vyd. Praha: BEN - technická literatura, 2004, 159 s. ISBN 80-7300-110-1.

UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. 1. vyd. Brno: Computer Press, 2013, 232 s. ISBN 978-80-251-4116-8.

VALEŠ, Miroslav. Inteligentní dům. 1. vyd. Brno: ERA, 2006, viii, 123 s. 21. století. ISBN 80-7366-062-8.

Financování práce :

Vedoucí práce: Mgr. Jiří Pech, Ph.D.

podpis : 

U externích vedoucích fakultní garant práce..... podpis :

Garant oboru bakalářského studia

..... podpis :

Vedoucí oddělení: PhDr. Milan Novák, Ph.D.

podpis 

Případný souhlas vedoucího ústavu AV podpis :

V Českých Budějovicích dne 27.10.2015

Převzal/a dne.....

podpis : 

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou v veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 19.4.2016

Stanislav Dvořák

Poděkování

Tímto bych zde rád poděkoval vedoucímu bakalářské práce panu Mgr. Jiřímu Pechovi, Ph. D. za jeho rady, přístup, odborné vedení a za čas, který mi věnoval při řešení dané problematiky.

Dvořák, S., 2016: Řešení ovládání inteligentního domu pomocí systému pro heterogenní zařízení. [The solution of a smart house operational control via the heterogeneous devices. Bc. Thesis, in Czech.] – 42 p., Faculty of Science, University of South Bohemia in České Budějovice, Czech Republic.

Abstrakt

Tématem bakalářské práce je vytvoření inteligentní domácnosti, od návrhu konkrétního řešení po popis jeho realizace na reálném příkladu. Popsaný koncept inteligentní domácnosti vychází z řídicí jednotky Raspberry Pi, která je ovládána prostřednictvím centralizovaného systému openHAB, pomocí něhož je ovládáno osvětlení v domě a topný systém.

Klíčová slova: inteligentní dům, řídicí systém, automatizace, osvětlení, topný systém, openHAB, Raspberry Pi

Abstract

The topic of this bachelor thesis is creation of smart home, from design of specific solution to describe its implementation on a real example. Described concept of smart home is based on Raspberry Pi control unit, which is controlled via openHAB centralized system, by which house lightning and heating system are controled.

Keywords: smart home, control system, automation, lighting, heating system, openHAB, Raspberry Pi

Obsah

1	Úvod	1
1.1	Cíle práce	1
1.2	Metodika	2
2	Teoretická část	4
2.1	Inteligentní dům – charakteristika.....	4
2.2	Centralizované řídicí systémy	5
2.2.1	OpenHAB	5
2.2.2	Home Assistant	5
2.3	Řídicí jednotka – Raspberry Pi	6
2.3	Možností připojení zařízení k řídicí jednotce.....	7
2.3.1	Drátové ethernet připojení	7
2.3.2	Sériová linka RS232 a RS485.....	8
2.3.3	Bezdrátové řešení.....	9
2.4	Prvky inteligentní domácnosti.....	10
2.4.1	Osvětlení	10
2.4.2	Topné systémy	10
2.4.3	Zabezpečení	11
2.4.3	Multimédia.....	11
2.4.5	Prezentační vrstva	12
3	Praktická část	13
3.1	Návrh.....	13
3.1.1	Popis objektu.....	13
3.1.2	Výběr vhodného centralizovaného řídicího systému.....	13
3.1.3	Parametry konkrétních prvků.....	14

3.1.4	Bindings - vazby	18
3.2	Realizace navrženého projektu	21
3.2.1	Schéma zapojení domácí sítě	21
3.2.2	Prvotní konfigurace zařízení	21
3.3	Instalace a konfigurace openHAB	22
3.3.1	Příprava SD karty z Windows	22
3.3.2	Spuštění Raspbian a instalace openHAB.....	23
3.3.3	Konfigurace openHAB runtime a vazeb.....	25
4	Závěr.....	38
	Citovaná literatura	39
	Seznam obrázků.....	41
	Seznam tabulek	42

1 Úvod

S rozvojem nových zařízení v domácnostech se výrobci snaží dovést svá zařízení k dokonalosti. Aby došlo k zefektivnění a symbióze zařízení v domácnosti, musí se začít koordinovat jejich funkce. Pro takovou kooperaci zařízení vznikl pojem inteligentní dům. Inteligentní dům je samostatné zařízení. Každý takový dům se od ostatních liší svou neopakovatelnou individualitou, kombinací typů heterogenních zařízení a specifickou logikou ovládacích prvků. Díky neustálému monitorování energetické spotřeby dochází nejen k úsporám spotřeby a tím pádem i provozních nákladů, ale například i k většímu komfortu a bezpečnosti uživatelů domácnosti. Nejčastěji se řídí vnitřní klimatické podmínky, vytápění budov, ventilace, osvětlení a bezpečnost objektu.

Tato práce se zabývá problematikou ovládnání inteligentního domu pomocí systému pro heterogenní zařízení, a to od návrhu po realizaci v konkrétní domácnosti.

Tuto práci jsem si vybral z důvodu perspektivity daného oboru. Samotná myšlenka inteligentních staveb není sice nová, ale až v současné době se díky jejich výhodám a propagaci zvýšila jejich realizace a tyto stavby se staly oblíbeným a často využívaným produktem na trhu. Tyto faktory přispívají k již zmíněné perspektivitě. Dalším motivem podněcujícím volbu tohoto tématu byla možnost realizace automatizace ve skutečné domácnosti. Na trhu jsou již hotová komplexní zařízení, ale napojením heterogenních produktů lze ušetřit zákazníkům finanční zdroje.

Práce je rozdělena do čtyř hlavních kapitol. První kapitola seznamuje s problematikou inteligentní domácnosti. Druhá kapitola, teoretická část, se zabývá seznámením s dostupnými centralizovanými řídicími systémy, dále s řídicí jednotkou Raspberry Pi a možnými prvky inteligentní domácnosti. Ve třetí části, praktické, je seznámení s vybranými zařízeními včetně jejich specifikace, seznámení s konfigurací programu openHAB a následně zapojení zařízení do systému. Ve čtvrté části je zhodnocení bakalářské práce.

1.1 Cíle práce

- Návrh inteligentního řešení domácnosti v konkrétním domě
- Volba vhodné řídicí jednotky
- Volba vhodného řídicího systému
- Realizace konkrétního projektu

1.2 Metodika

Při vypracování tématu mé bakalářské práce nazvané „Řešení ovládání inteligentního domu pomocí systému pro heterogenní zařízení“, konkrétně při řešení ovládání osvětlení a topného systému, jsem využil následující hardware:

- Raspberry Pi 2 Model B
(<http://rpishop.cz/raspberry-pi-pocitace/170-raspberry-pi-2-1024-mb-ram.html>)
- 8GB microSDHC Class 10 paměťová karta
(<http://rpishop.cz/raspberry-pi-prislusenstvi/58-8gb-microsdhc-karta-kingston-cl10-adapter.html>)
- TP-Link TL-WN725N 150Mbps Nano Wifi N USB Adapter
(<https://www.alza.cz/tp-link-tl-wn725n-d341839.htm>)
- eQ-3 MAX!+ termostatické hlavice
(<http://www.conrad.cz/bezdratova-programovatelna-termostaticka-hlavice-eq-3-max.k519588>)
- eQ-3 MAX! Cube Lan Gateway konfigurátor termostatických hlavic
(<http://www.conrad.cz/konfigurator-termostatickych-hlavic-eq-3-max-cube-lan-gateway.k560896>)
- Qubino relé, model ZMNHND1
(<http://www.ebay.com/itm/QUBINO-Z-Wave-Plus-1D-Relay-Flush-Module-ZMNHND1-/121818105608>)
- Z-Wave ZW090 USB modul
(<http://www.amazon.com/Aeon-Labs-Aeotec-Z-Wave-Zstick/dp/B00X0AWA>)
- TP-Link TD-W8961NB 300Mbps Wireless N ADSL2+ (Verze firmware: 3.24.18.0_MT7550)
(<https://www.alza.cz/tp-link-td-w8961nb-d290869.htm>)
- Eaton PowerWare 5110 1000VA
(<https://www.alza.cz/eaton-ups-powerware-5110-500va-d197287.htm>)

Dále pro vlastní realizaci jsem využil tento software:

- Win32 Disk Imager – zapsání operačního systému na paměťovou kartu
- Raspbian (Kernel version 4.1) – operační systém
- Java 1.8.0.65
- openHAB – řídicí systém

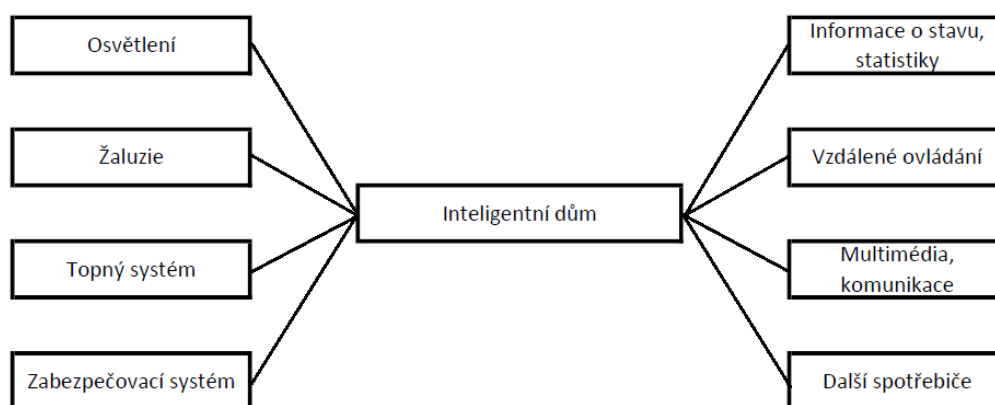
- MAX! PC software – aplikace pro prvotní konfiguraci eQ-3 MAX! Cube Lan Gateway
- Aeon Labs IMA tool – aplikace pro prvotní konfiguraci Z-wave zařízení
- openHAB Designer – aplikace pro konfiguraci konfiguračních souborů openHABu
- NetBeans IDE 8.1 – vývojové prostředí JAVA, využité pro vytvoření pravidel v openHABu
- Windows 10 Pro 64 bit
- iceweasel – webový prohlížeč
- Cisco Packet Tracer Student – schémata v práci

Díky kombinaci veškerých těchto prvků bylo dosaženo funkčního řešení v realizovaném domě.

2 Teoretická část

2.1 Inteligentní dům – charakteristika

Pojem inteligentní dům představuje domácí prostředí, které je díky elektroinstalaci v domě či bytě částečně nebo kompletně automatizované, což výrazně uživatelům ulehčuje ovládání domu, přináší více komfortu a podává veškeré informace o tom, co se aktuálně děje. Tím dochází k automatizaci domu. Díky zařízením jako jsou tablety, mobilní telefony nebo počítače máme přístup k centrální řídicí jednotce, která zprostředkovává ovládání světel, topných systémů, ventilace, klimatizace, kamerových systémů a dalších zařízení, které uživatel domácnosti potřebuje. Systém umožňuje monitorování vnitřních i vnějších teplot podle nainstalovaných čidel. Dokáže monitorovat také aktuální spotřebu elektrické energie. Uživatel si díky statistikám systému může vyhodnotit, kde dochází ke ztrátám a vyhnout se jim. Všechny úlohy systému a nastavení by měly být jednoduché a pro všechny uživatele domu srozumitelné. Na obrázku číslo 1 je schéma funkcí inteligentního domu.



Obrázek 1 - Schéma funkcí inteligentního domu

Podstatou inteligentního domu je řídicí jednotka zprostředkovávající komunikaci jednotlivých prvků mezi sebou a přístup uživatele k funkcím systému. Jednotlivá zařízení mohou být k řídicí jednotce připojena více způsoby. Využívá se drátové a bezdrátové řešení instalace. U novostaveb je potřeba, aby projektant již s tímto faktem počítal a návrh chytré elektroinstalace přizpůsobil, protože není vhodné zasahovat do zdi vysekáváním omítek pro dodatečnou instalaci kabeláže. Bezdrátová instalace je praktičtější v případech, kde nechceme zasahovat do zdí, například v panelových domech. Obě řešení jsou ale vzájemně plně kompatibilní a není problém je kombinovat.

V 60. letech 20. století byl v Japonsku prezentován „inteligentní dům“, v němž řízení veškerých funkcí řídil počítač. Tyto práce se však nesetkaly se širokým uplatněním v praxi. Počátkem 70. let 20. století energetická krize a prudký nárůst cen ropy způsobily nastartování vývoje projektů směřujících ke snížení spotřeby energií na vytápění budov, jejich osvětlování atd. Prvních úspěchů bylo dosaženo poměrně rychle. Byly prezentovány výsledky německých výrobců nejen kvalitnějších otopných systémů, ale i nově koncipovaných elektrických instalací. Za počátek zrodu jednotné koncepce inteligentní elektroinstalační techniky lze považovat rok 1987, v němž založily firmy Berker, Gira, Merten a Siemens společnost Instabus Gemeinschaft. Jejich cílem bylo vyvinout systém pro měření, řízení, regulaci a sledování provozních stavů v budovách. (Trtík, 2009)

2.2 Centralizované řídicí systémy

Na trhu se dnes nachází velké množství řešení pro řízení inteligentních domácností. Pro každou budovu se může hodit jiné řešení. Závisí to na mnoha faktorech, například na zařízení, která budeme chtít ovládat. Je potřeba si nejprve zjistit, zda jsou daná zařízení podporovaná výrobcem (existují ovladače), nebo zda budeme vybírat komerční či open-source řešení. Definice open-source je: „*Open-source software je takový software, k němuž zákazník dostane od jeho tvůrce zdrojový kód a může jej dále upravovat*“. (Anonymous, 2004)

2.2.1 OpenHAB

OpenHAB je svobodný software, který slouží k ovládání inteligentních domů. Je založen na technologii Java za pomoci frameworku OSGi (Open Services Gateway initiative), tudíž pro svůj provoz žádá přítomnost Java Runtime Environment, která zajišťuje nezávislost na platformě a architektuře procesoru. Díky OSGi frameworku je možné přidávat či odebírat moduly a přitom není potřeba zastavit službu. OpenHAB ve své podstatě neposkytuje přímo služby pro ovládání jednotlivých zařízení, nýbrž nabízí prostředí sjednocující ovládání těchto zařízení buď přes webové rozhraní, nebo aplikacemi pro mobilní zařízení, jako jsou tablety či mobilní telefony s operačními systémy Android, iOS, nebo Windows Phone (dnes už Windows 10 Mobile).

2.2.2 Home Assistant

Home Assistant je open-source program sloužící k řízení inteligentních domů založený na skriptovacím jazyce Python 3. Lze jej nainstalovat jak na počítač Raspberry Pi, tak

i na klasický počítač, na kterém je nainstalovaný Python. V současné době existují pro vzdálené ovládání aplikace pro mobilní operační systémy iOS a Android. (GitHub, 2016)

2.2.3 Domoticz

Domoticz je open-source systém automatizace domácnosti, který umožňuje sledovat a konfigurovat různá zařízení. Je napsán v programovacím jazyce C++. Lze jej spustit na Raspberry Pi, Windows, Linux, Mac OS X a embedded zařízeních. Uživatelské rozhraní, které se automaticky přizpůsobí pro stolní počítač či mobilní zařízení, je napsané v HTML5. (Domoticz, 2016)

2.3 Řídicí jednotka – Raspberry Pi

Pro řízení inteligentního domu je potřeba také zvolit vhodnou řídicí jednotku. Nabízí se výběr mezi notebookem či desktopovým počítačem, nicméně obě tato řešení mají poměrně vysokou spotřebu (řády Wattů). Pro účely tohoto projektu byla zvolena platforma Raspberry Pi 2 Model B, jelikož je malých rozměrů (85 x 56 x 21 mm), vyniká nízkou spotřebou a nízkými pořizovacími náklady. Je vybavena oproti předešlé verzi vyšší kapacitou operační paměti RAM, více USB porty a pro nás klíčovým LAN konektorem pro přístup k místní síti a internetu (vzdálené ovládání, synchronizace času). Na obrázku číslo 2 je Raspberry Pi znázorněno a v tabulce číslo 1 je uvedena hardwarová výbava Raspberry Pi. (Raspberry Pi, 2015)

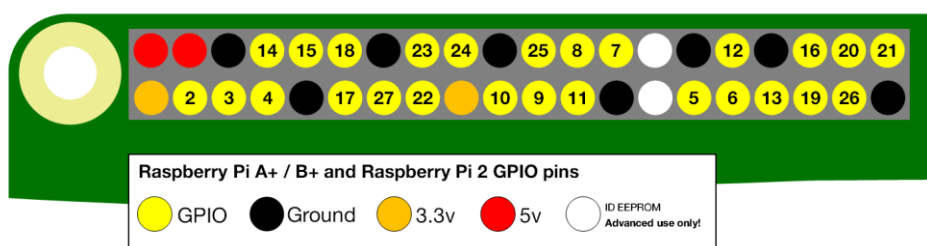
Tabulka 1 – HW výbava Raspberry Pi 2 Model B

Procesor	Broadcom BCM2836 900MHz ARM Cortex-A7 quad-core
Grafická karta	VideoCore IV 3D GPU (250 MHz, 24 GFLOPS)
Operační paměť	1GB (sdílená s grafickou kartou)
USB 2.0	4
Síťová karta LAN (RJ45)	100Mbit
Další výbava	2×20pinový GPIO, CSI/DSI rozšíření pro kameru a dotykový displej
Napájení	micro USB, 5V



Obrázek 2 - Raspberry Pi 2 Model B

Důležité pro náš projekt je zejména rozhraní GPIO, které obsahuje celkem 40 pinů. Hned čtyři z těchto pinů slouží k napájení (2 x 3,3V; 2 x 5V) 8 x zem, dva piny pro EEPROM desky. Zbylé porty jsou SPI (sériové rozhraní, vstupně/výstupní), IO a I2C. Na obrázku číslo 3 je znázorněné rozložení pinů.



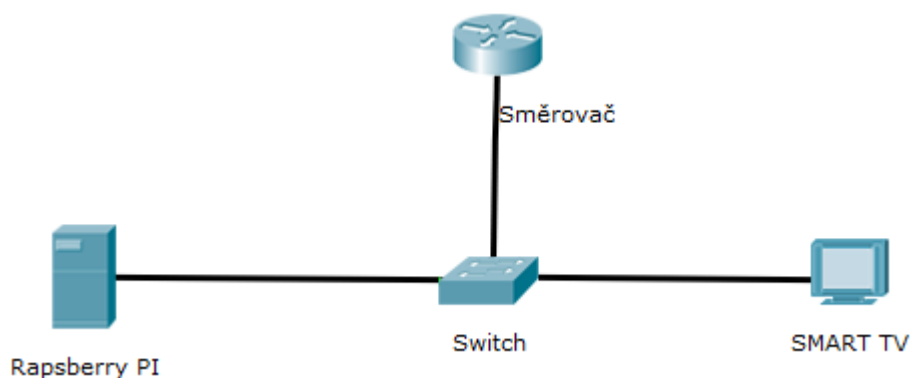
Obrázek 3 - Rozložení GPIO na Raspberry Pi 2 Model B

Zdroj: Raspberry Pi, 2015

2.3 Možností připojení zařízení k řídicí jednotce

2.3.1 Drátové ethernet připojení

Jednou z možností, jak propojit zařízení se serverem, je pomocí LAN, pokud jej zařízení podporuje. Na síti LAN spolu komunikuje několik stanic zpravidla na sdíleném médiu. Na jedné síti LAN se používá stejný linkový protokol (např. Ethernet). V současné době se pod pojmem LAN míní tzv. rozšířené LAN, které jsou tvořeny jednotlivými LAN. Termín rozšířené LAN definují Kabelová a Dostálek (2008) jako: „Rozšířené LAN vzniknou spojením jednotlivých LAN pomocí přepínačů.“ Na obrázku číslo 4 je příklad zapojení SMART TV do sítě.

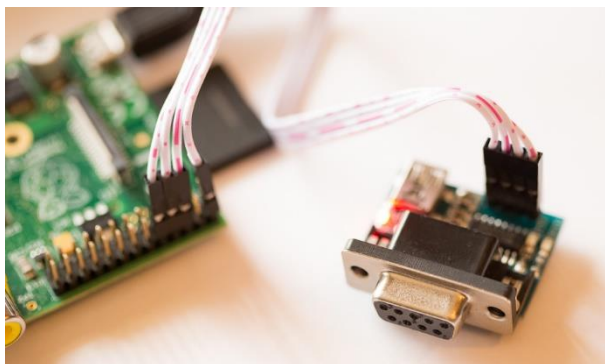


Obrázek 4 - Schéma zapojení SMART TV do sítě LAN

2.3.2 Sériová linka RS232 a RS485

Další možnost je využití sériových linek RS232 či RS485.

- RS232 je rozhraní pro přenos informací vytvořeno původně pro komunikaci dvou zařízení do vzdálenosti 20 m. Pro větší odolnost proti rušení je informace po propojovacích vodičích přenášena větším napětím než je standardních 5 V. Přenos informací probíhá asynchronně pomocí pevně nastavené přenosové rychlosti a synchronizace sestupnou hranou startovacího impulsu. (Olmr, 2005)
- Standard RS485 je navržen tak, aby umožňoval vytvoření dvou vodičového poloduplexního vícebodového sériového spoje. Má stejný základ jako standard RS232, od kterého se liší především jinou definicí napěťových úrovní, nepřítomností modemových signálů, možností vytváření sítí (též sběrnice) stávající se z až 32 zařízení a možností komunikace na vzdálenost až 1200m (proti 20 m u RS232). (Anonymous, 2015)



Obrázek 5 - Modul rozšiřující Raspberry PI o RS232

Zdroj: Hunt, 2013

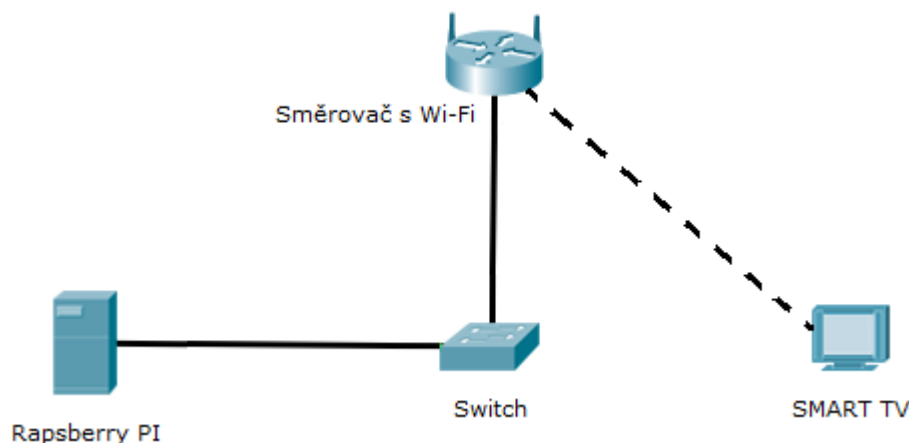
Počítač Raspberry PI není osazen přímo konektory RS232 ani RS485. Díky cenově dostupnému čipu MAX3232 na tištěném spoji s TTL měničem a konektorem RS232 lze

o tento konektor poměrně snadno rozšířit připojením ke čtyřem pinům (3,3V; Ground; 2x GPIO). Na obrázku číslo 5 je znázorněný rozšiřující modul.

2.3.3 Bezdrátové řešení

V domácnosti existuje řada zařízení, která podporují bezdrátové komunikační technologie. Bezdrátové řešení přináší výhody přenosu informací mezi jedním či více body, které nejsou fyzicky propojeny, nebo fyzické propojení ani nepodporují. Maximální přenosové vzdálenosti se mohou lišit v závislosti na prostředí (pokud jsou zdi betonové, tak je dosah velice omezen) a použité technologii pro přenos dat. Tato práce se zabývá především krátkým dosahem mezi serverem a zařízením, tím pádem se nabízí technologie Bluetooth a Wi-Fi. Na obrázku číslo 6 můžeme vidět schéma zapojení SMART TV prostřednictvím Wi-Fi do lokální sítě.

- Wi-Fi připojení je založeno na standardu 802.11a/b/g/n/ac, základní frekvence jsou 2,4 GHz a 5 GHz. Přenosové rychlosti a vzdálenost se pak liší podle standardu.
- Bluetooth připojení je založeno na standardu 802.15. Pracuje na frekvenci 2,4 GHz (jako Wi-Fi).



Obrázek 6 - Schéma zapojení SMART TV do sítě LAN prostřednictvím Wi-Fi

- Možné je využití infračerveného přenosu IR. Tuto technologii popisuje Zdařil (2009) takto: „Přenos pomocí infračerveného (IR) rozhraní se řadí mezi jednu z nejstarších metod mobilní bezdrátové komunikace. Tuto éru pravděpodobně nastartovala jedna z prvních aplikací moderního bezdrátového přenosu, dálkové ovládání televizních přijímačů. Postupem času však tato metoda začala prostupovat do nejrůznějších odvětví elektroniky. V dnešní době jsou

infračerveným rozhraním (chceme-li také infra portem) vybaveny téměř všechny mobilní telefony, nejrůznější periferní zařízení osobních i stolních počítačů, digitální fotoaparáty a kamery a v neposlední řadě samozřejmě dálkové ovladače.“

- Další možností je Li-Fi. Tuto technologii popisuje Všetěčka (2015) takto: *„Li-Fi, někdy označovaná jako Visible light communications (VLC). Data jsou přes Li-Fi přenášena pomocí mikroblikání, které by lidské oko nemělo postřehnout, a proto mu nevadí. Modemem mohou být například LED žárovky kombinované se světelným senzorem. V podstatě se tak přenos dat po optických vláknech přesunul do vzduchu se všemi výhodami a problémy, které to může přinášet. Ke komunikaci se využívá standard IEEE 802.11, se kterým pracuje i klasická Wi-Fi.“*

2.4 Prvky inteligentní domácnosti

2.4.1 Osvětlení

Pro komfort uživatele inteligentního domu je základní funkcí osvětlení ve většině částí budovy. Správná funkčnost by měla být taková, aby uživatel měl k dispozici přehled aktuálně zapnutých i vypnutých světel, mohl je jednotlivě ovládat a nastavovat jas. Na trhu jsou dostupná i relé, která nám zařídí postupné rozsvěcení a zhasínání osvětlení, což je při odchodu z budovy příjemné. V tomto přehledu by neměly chybět statistiky, ze kterých můžeme vygenerovat časový harmonogram na automatizaci osvětlení. Osvětlení budovy dále lze sofistikovaně propracovat s použitím pohybových čidel, která mohou rozsvěcet a zhasínat světla na základě přítomnosti uživatele.

2.4.2 Topné systémy

Nezbytnou součástí domácnosti v našem zeměpisném pásmu je vytápění. Pro individuální automatickou regulaci teplot v jednotlivých místnostech jsou potřeba teplotní senzory. Pokud se v budově používají elektrické přímotopy, tak se vytápění řídí napřímo. V případě, že je nainstalováno podlahové topení či radiátory, jsou nezbytné akční členy, tzv. aktuátory, které ovládají termostaty na topení. Topný systém lze zpětně díky statistikám vyhodnotit a v případě využití pohybových čidel individuálně pro každou místnost přizpůsobit tak, aby byl co nejefektivnější a energeticky úsporný. Podstatnou součástí inteligentního systému vytápění je možnost jeho dálkového ovládání pomocí tabletu, mobilního telefonu či webového prohlížeče v počítači.

2.4.3 Zabezpečení

Pod pojmem zabezpečení domácnosti si lze představit komponenty alarmu, jako jsou kontakty dveří, hlásiče požárů, pohybové senzory, snímače otisků prstů, interkomy či kamerové systémy. Zabezpečení domácnosti je bezpochyby odůvodněnou součástí inteligentní domácnosti. Při odchodu uživatelů z domu se alarm aktivuje a dokáže upozornit na neoprávněný vstup pomocí SMS, hovoru nebo emailové zprávy. Kamerový systém díky pohybovým sensorům nemusí nahrávat 24 hodin denně, ale pouze při podnětu ke spuštění. Záznam nebo živý přenos lze sledovat i mimo domácnost. Vstupní dveře či garáž mohou být zabezpečeny pomocí kódu, klíče, zařízení na snímání otisků prstů či jiných biometrických údajů. Způsobů zabezpečení je nespočet, důležité jsou požadavky a možnosti majitele.

Zabezpečení je potřeba vnímat i ze softwarového hlediska, jelikož při napadení centralizovaného systému by si útočník mohl v extrémních případech například odemknout zamčené dveře, otevřít okna atp. Z tohoto důvodu je vhodné monitorovat síťovou komunikaci firewallem. Firewall definuje společnost Kaspersky Lab (2015) takto: *„Brána firewall slouží jako ochrana počítače proti virům, červům, trojským koním a hackerským útokům hrubou silou. Tato brána může být softwarová (bezpečnostní program) nebo hardwarová (fyzický směrovač), přičemž oba typy vykonávají stejnou funkci: prohledávají příchozí provoz, zda neobsahuje data nacházející se na seznamu zakázaných položek. Brány firewall prohledávají všechny pakety s daty (menší části větších celků, které jsou rozděleny kvůli snadnějšímu přenosu), zda neobsahují škodlivý kód.“* Dále je důležité každému zařízení, které má vlastní síťové konfigurační rozhraní, nastavit silné heslo, a to minimálně osm znaků s použitím velkých a malých písmen, číslic a speciálních znaků.

2.4.3 Multimédia

Pro komfort uživatelů patří i multimédia v domácnosti. Pod tímto pojmem si lze představit poslech hudby po celém domě, kterou si v jednotlivých pokojích lze zapnout například vypínačem na zdi. V případě pokrytí Wi-Fi či kabelových ethernetových rozvodů lze využívat chytrá přenosná zařízení, na kterých lze přehrávat hudbu, sledovat filmy nebo konzumovat internet. Tímto konceptem lze využívat i rodičovskou kontrolu, která může zprostředkovat vypnutí světla či zneprístupnění určitých webových stránek. Zabudované reproduktory pro poslech hudby lze využít i jako hlasový komunikační

system, či budík. Při sofistikované implementaci lze využít i systém na roztažení/zatažení žaluzií či rolet. Všechny tyto prvky stupňují pocit komfortu.

2.4.5 Prezentační vrstva

Uživatelské rozhraní centralizovaného systému umožňuje prohlížet statistiky nebo ovládat jednotlivé prvky domu. Tento systém je přístupný z lokální sítě, lze se na něj také při správné konfiguraci směrovače připojit z internetu. Do uživatelského rozhraní se lze připojovat z počítačů přes webové rozhraní nebo pomocí aplikací pro mobilní telefony či tablety, tedy pokud má jejich operační systém naprogramovanou příslušnou přístupovou aplikaci.

3 Praktická část

3.1 Návrh

3.1.1 Popis objektu

Instalace prvků inteligentní domácnosti je realizována ve skutečné budově. Jedná se o přízemní novostavbu o výměře přibližně 165 metrů čtverečních, kde již v projektu bylo s vybranými prvky počítáno. Konkrétně se jedná o osvětlení a topný systém. Jelikož se majitel nemovitosti chce vyhnout budoucím stavebním zásahům, tak je inteligentní instalace připravena i pro více zařízení pro případ, že by se v budoucnu rozhodl investovat do dalších inteligentních prvků. Dům má zavedené pevné připojení k internetu pomocí pevné linky, ADSL technologií. Majitel nemovitosti má zaplacenou veřejnou IP adresu pro vzdálenou správu.

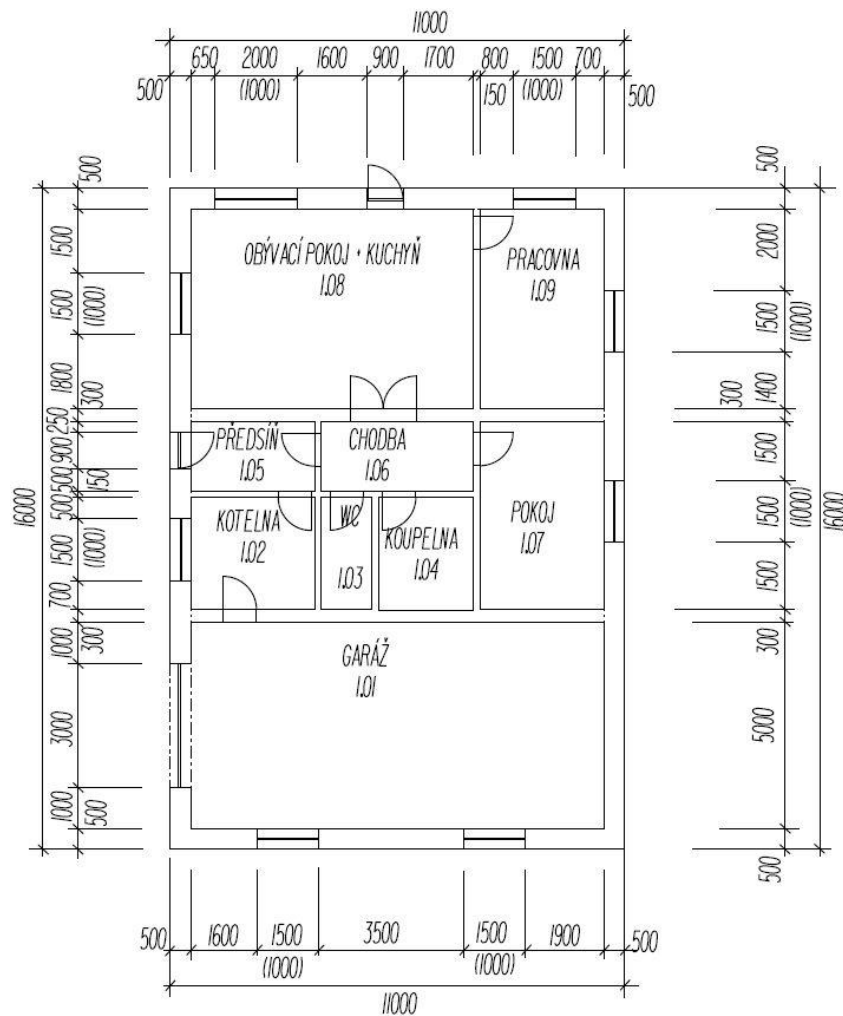
Dům je rozvržen tak, že za hlavním vchodem, který je situován na východ, se nachází předsíň, se kterou sousedí kotelna. Za předsíní následuje chodba. Z chodby lze vstoupit do obytných částí, tedy obývacího pokoje spojeného s kuchyňským koutem, koupelny, na toaletu a do pokoje. Půdorys domu s popisky je na obrázku číslo 7.

3.1.2 Výběr vhodného centralizovaného řídicího systému

Při výběru vhodného centralizovaného řídicího systému je potřeba se ohlížet k řídicí jednotce, na které bude řídicí systém nainstalovaný. V tomto případě je centralizovanou řídicí jednotkou zařízení Raspberry Pi 2 Model B, tudíž musíme vybrat systém, který lze na tomto hardwaru spustit. V této práci vybíráme mezi třemi řídicími systémy, tedy openHAB, Home Assistant a Domoticz. Všechny tyto systémy jsou open-source řešením, jinými slovy mají otevřený zdrojový kód a lze je provozovat na Raspberry Pi. Podrobnější porovnání nalezneme v tabulce číslo 2.

Tabulka 2 - porovnání řídicích systémů

	Domoticz	Home Assistant	openHAB
Kompatibilita s R. PI	Ano	Ano	Ano
Licencování	Open-source	Open-source	Open-source
Mobilní aplikace	Ano	Ano	Ano
Programovací jazyk	C++	YAML standard	JAVA
Podpora Z-wave (rozšířený protokol)	Ano	Ano	Ano

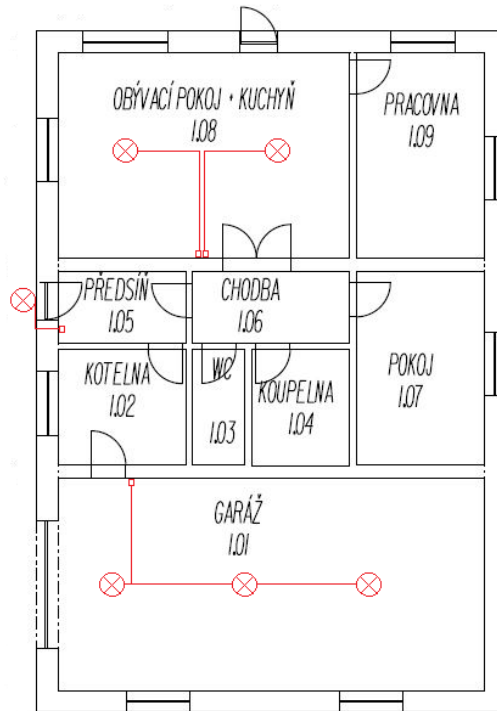


Obrázek 7 - Půdorys domu

Po porovnání vlastností systémů, které vyšlo nerozhodně, jsem musel rozhodnout subjektivně, s ohledem na mnou ovládaný programovací jazyk a budoucí rozšiřitelnost systému. OpenHAB je ze všech těchto řídicích systému nejmodulárnější, mezi další argument mé volby patří široká komunita uživatelů openHABu, díky které je vyvinuto velké množství bindů (bind je volitelný balíček, který se používá pro rozšíření funkčnosti openHABu, neboli je to „ovladač“, kterým se konfiguruje příslušné zařízení), konkrétně i pro zařízení v realizovaném domě. OpenHAB je založen na Java technologii, tudíž je nezávislý na platformě a architektuře procesoru.

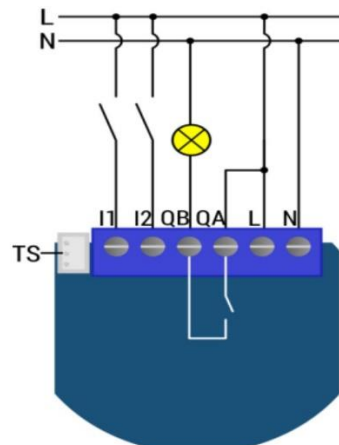
3.1.3 Parametry konkrétních prvků

V obývacím pokoji, u vchodu do domu a v garáži realizovaného domu je realizováno centralizované ovládání osvětlení tak, jak je znázorněno na obrázku číslo 8.



Obrázek 8 - Umístění osvětlení

Pro realizaci inteligentního osvětlení je využit spínací modul Z-Wave Plus – Qubino (model ZMNHND1), jehož pořizovací hodnota se pohybuje okolo 1 500 Kč včetně DPH. Tento modul je instalován do standardních spínacích krabiček pod schodišťový vypínač na zdi. Schéma zapojení tohoto relé je na obrázku číslo 9.



Obrázek 9 - Schéma zapojení relé Z-Wave Plus – Qubino ZMNHND1

Zdroj: Domotique-store, 2015

N je nula, L – fáze, QA – vstup z jiné fázové jednotky, QB – výstup na okruh (pro spotřebič), I1 a I2 – vstupy ze spínačů (nebo tlačítek), TS slouží k propojení s digitálním teplotním senzorem. Osvětlení funguje i bez ohledu na počítač.

Pro komunikaci mezi modulem a Raspberry Pi slouží USB modul Aeon Labs Aeotec Z-Wave Z-Stick (ZW090), znázorněný na obrázku číslo 10. Z-Wave je bezdrátový komunikační protokol pro domácí automatizaci. Pracuje na frekvenci 868.42 MHz, vyhýbá se tedy interferenci se standardy Wi-Fi a Bluetooth. (Z-Wave, 2016)



Obrázek 10 - USB modul ZW090

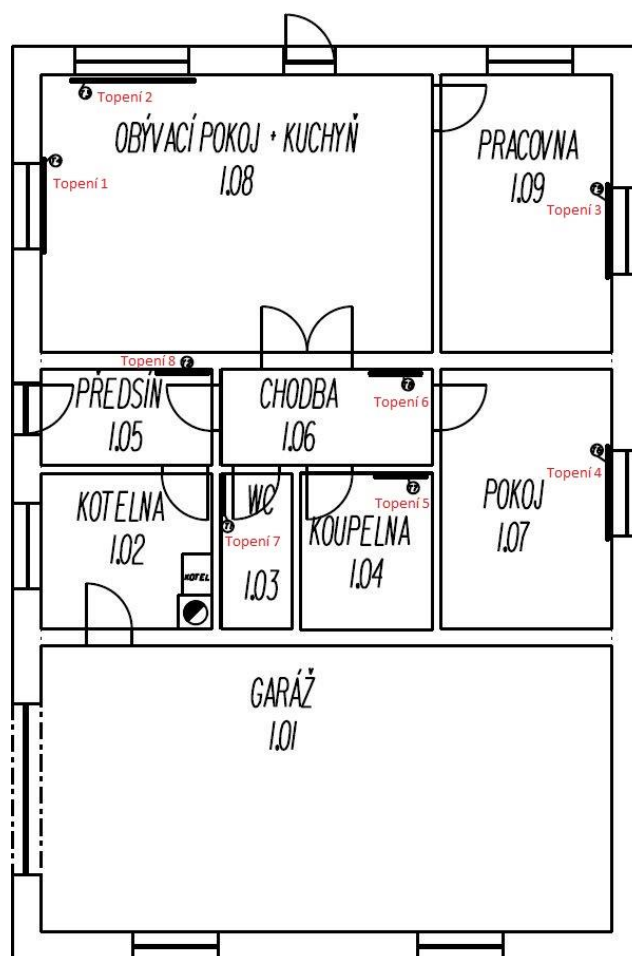
Zdroj: The smartest house, 2015

Topný systém je realizován ve všech místnostech s výjimkou kotelny a garáže, kde nejsou zavedeny radiátory (dle obrázku číslo 12). Topný systém je založen na kombinovaném (dřevo/hnědé uhlí) zplynovacím kotli Atmos C 18 S. Palivo je uskladněno ve druhé polovině garáže. Radiátory jsou od výrobce Korado, model RADIK KLASIK různých rozměrů, dle šíře oken.

Na radiátorech jsou namontované bezdrátové programovatelné termostatické hlavice eQ-3 MAX!+ (obrázek číslo 11), jejichž hodnota je okolo 1 200 Kč včetně DPH.



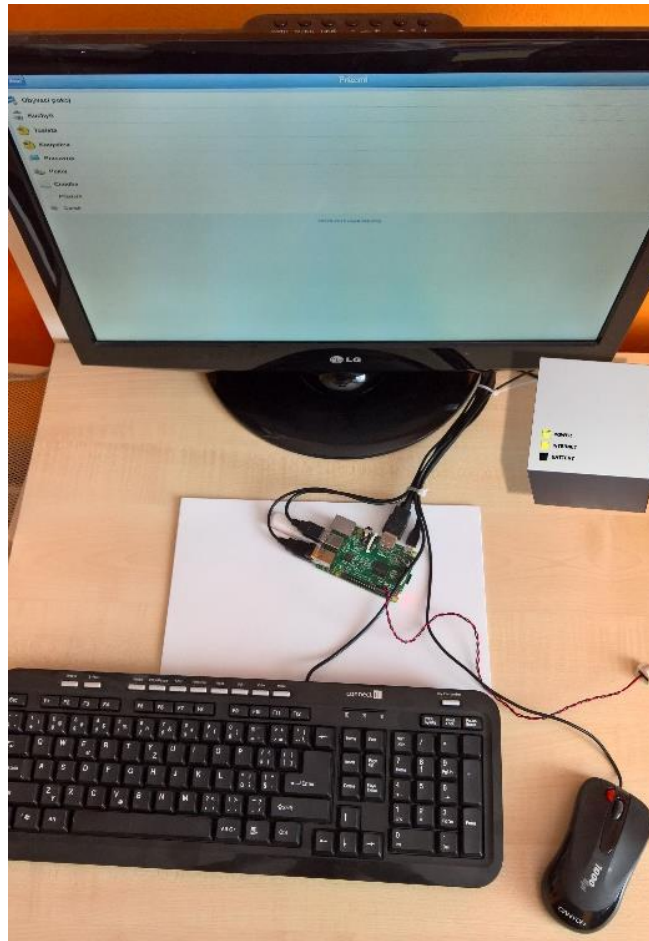
Obrázek 11 - Termostatická hlavice eQ-3 MAX!+ nainstalována na radiátor



Obrázek 12 - Umístění radiátorů

Tyto termostaty napájí dvě baterie AA, pro komunikaci s konfiguratorem využívají frekvence 868 MHz. Rozsah nastavitelných teplot je mezi 5 – 55 °C.

Aby termostaty komunikovaly v síti, bylo mimo jiné potřeba zakoupit konfigurator termostatických hlavic eQ-3 MAX! Cube Lan Gateway v hodnotě okolo 1 700 Kč včetně DPH, se kterým se spárují termostatické hlavice. Tento konfigurator je vybaven LAN rozhraním pro zapojení do sítě v domácnosti. Na obrázku číslo 13 je tento konfigurator vyobrazený spolu se zapojeným Raspberry Pi.



Obrázek 13 - Konfigurator termostatických hlavice eQ-3 MAX! Cube Lan Gateway spolu se zapojeným Raspberry Pi

3.1.4 Bindings - vazby

Zařízení připojená k řídicí jednotce se konfiguruje pomocí vazeb - bindů. Bind je volitelný balíček, který rozšiřuje funkčnost openHABu.

Balíčky ve formátu souboru `.jar` se ukládají do adresáře `openhab/addons/` a slouží ke zprostředkování komunikace mezi zařízením a centralizovaným systémem openHAB (je to ovladač, který realizuje spojení s daným typem hardware). Dle použitých vazeb se také volí vhodné protokoly pro přenos dat.

Konfigurace těchto balíčků probíhá v souboru `openhab.cfg`, kde se konfiguruje vazby jako například IP adresa nebo port USB, ve kterém je realizované zařízení.

Pro pochopení problematiky vazeb je zde uvedena vazba pro funkci Wake on Lan. Tato funkce zprostředkovává zapnutí vypnutého počítače prostřednictvím sítě. Vazba pro funkci Wake on Lan (WoL) se skládá z tří tříd – `WolActivator`, `WolBinding`

a WolBindingConfig. Samotná funkce Wake on Lan slouží k probuzení počítače prostřednictvím LAN sítě.

Třída WolActivator je určena k samotnému probuzení vybraného nakonfigurovaného zařízení.

```
public final class WolActivator implements
org.osgi.framework.BundleActivator {

    private static org.slf4j.Logger logger;

    public WolActivator() {
        // <editor-fold defaultstate="collapsed" desc="Compiled Code">
        /* 0: aload_0
         * 1: invokespecial java/lang/Object."<init>":()V
         * 4: return
         * */
        // </editor-fold>
    }

    public void start(org.osgi.framework.BundleContext bc) throws
Exception {
        // <editor-fold defaultstate="collapsed" desc="Compiled Code">
        /* 0: getstatic
org/openhab/binding/wol/internal/WolActivator.logger:Lorg/slf4j/Logger;
         * 3: ldc WoL binding has been started.
         * 5: invokeinterface
org/slf4j/Logger.debug:(Ljava/lang/String;)V
         * 10: return
         * */
        // </editor-fold>
    }

    public void stop(org.osgi.framework.BundleContext bc) throws
Exception {
        // <editor-fold defaultstate="collapsed" desc="Compiled Code">
        /* 0: getstatic
org/openhab/binding/wol/internal/WolActivator.logger:Lorg/slf4j/Logger;
         * 3: ldc WoL binding has been stopped.
         * 5: invokeinterface
org/slf4j/Logger.debug:(Ljava/lang/String;)V
         * 10: return
         * */
        // </editor-fold>
    }
}
```

Ve třídě WolBindingConfig dochází k výběru konkrétního zařízení pomocí jeho MAC adresy a IP adresy.

```
static class WolBindingConfig {

    InetAddress address;
    byte[] macBytes;

    WolBindingConfig() {
        //<editor-fold defaultstate="collapsed" desc="Compiled
Code">
```

```

        /* 0: aload_0
        * 1: invokespecial java/lang/Object."<init>":()V
        * 4: return
        * */
        // </editor-fold>
    }
}

```

Třída `WolBinding` tvoří ovladač pro funkci probuzení zařízení po síti. Probíhá zde i volání hodnot ze třídy `WilBindingConfig`, konkrétně hodnoty `macBytes`.

```

public class WolBinding extends
org.openhab.core.events.AbstractEventSubscriber implements
org.openhab.model.item.binding.BindingConfigReader {

public WolBinding() {
}

public String getBindingType() {
}

public void receiveCommand(String itemName,
org.openhab.core.types.Command command) {
}

private void sendWolPacket(WolBindingConfig config) {
}

private static byte[] fillMagicBytes(byte[] macBytes) {
}

public void validateItemType(org.openhab.core.items.Item item, String
bindingConfig) throws
org.openhab.model.item.binding.BindingConfigParseException {
}

public void processBindingConfiguration(String context,
org.openhab.core.items.Item item, String bindingConfig) throws
org.openhab.model.item.binding.BindingConfigParseException {
}

private static InetAddress getInetAddress(String ipStr) throws
org.openhab.model.item.binding.BindingConfigParseException {
}

private static byte[] getMacBytes(String macStr) throws
org.openhab.model.item.binding.BindingConfigParseException {
}

public void removeConfigurations(String context) {
}
}

```

V `openHAB` probíhá samotná konfigurace funkce `Wake on Lan` v souboru `configurations/items/*.items` přidáním položky ve formátu:

```

Switch networkPC "Computer" (network, gOGBuero) { wol="broadcast-
IP#macaddress" }

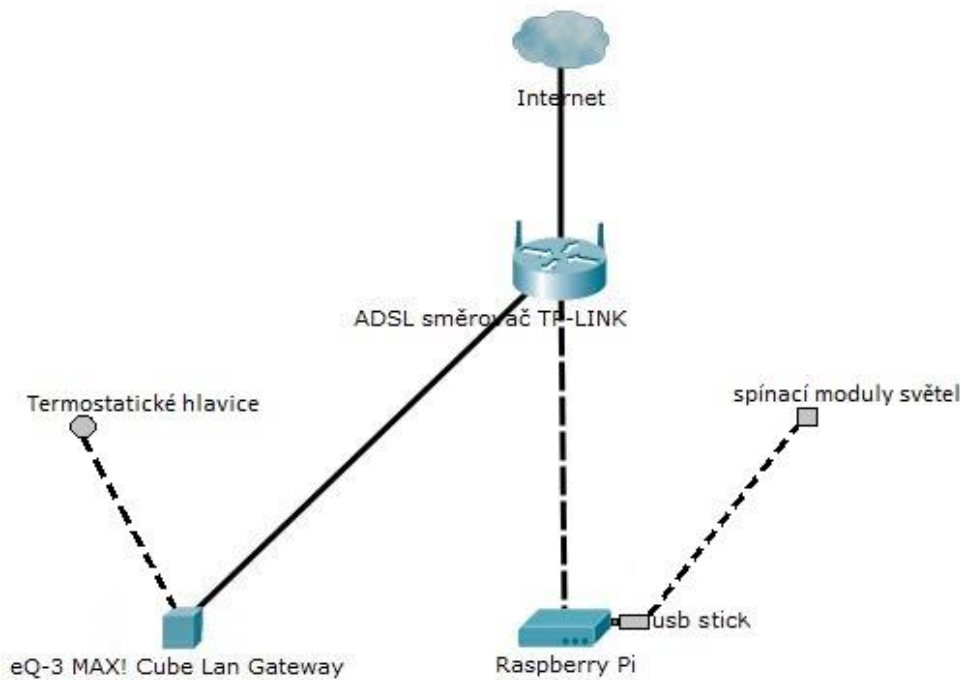
```

Do pole `<broadcast-IP>` se vyplní broadcast adresa sítě. Do pole `<macaddress>` se vyplní MAC adresa koncového zařízení, které se bude probouzet.

3.2 Realizace navrženého projektu

3.2.1 Schéma zapojení domácí sítě

Topologie zapojené sítě se zapojenými zařízeními je znázorněna na obrázku číslo 14. Středem této topologie je ADSL směrovač, který slouží i jako brána do internetu. Ten je pak prostřednictvím Wi-Fi připojen k Raspberry Pi a pomocí síťového kabelu propojen s konfiguratorem termostatických hlavice. Stejně jako ADSL směrovač a MAX! Cube, tak i Raspberry Pi je zapojeno do záložního zdroje UPS Eaton PowerWare 5110, který je dokáže při výpadku proudu po dostatečnou dobu napájet.

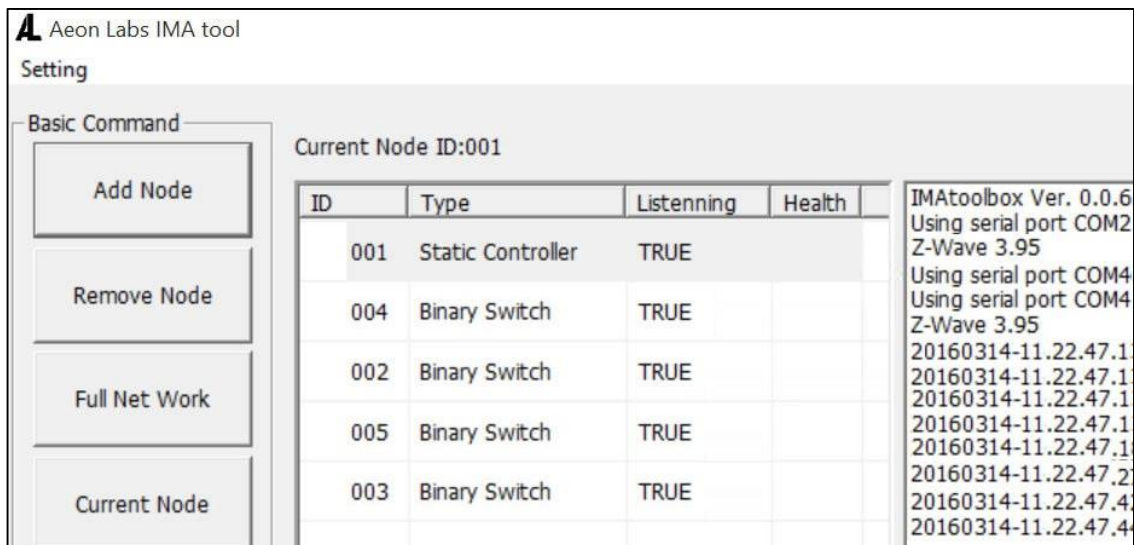


Obrázek 14 - Schéma realizované sítě

3.2.2 Prvotní konfigurace zařízení

U osvětlení je prvotní konfigurace následující. Spárování zařízení, tedy relé a Z-Wave adaptéru, probíhá pomocí nástroje Aeon Labs IMA tool, který je určen pro prostředí Windows. Nejdříve je potřeba nainstalovat ovladač pro USB modul, aby jej systém Windows správně rozpoznal. Po instalaci ovladačů a software jej otevřeme, klikneme na možnost Add Node a přidáme vyhledaná zařízení. V našem případě se jedná o USB modul a čtyři relé moduly, které jsou umístěny za vypínačem. Finální podoba vyhledaných zařízení v programu Aeon Labs IMA tool je uvedena na obrázku číslo 15. Jedná se o konkrétní nastavení realizované domácnosti. Tato konfigurace se uloží do USB adaptéru, který lze nyní zapojit do Raspberry Pi, kde se osvětlení bude následně konfigurovat v řídicím systému openHAB. ID001 patří Z-wave USB adaptéru, ID002

náleží relé v obývacím pokoji, ID003 – kuchyň, ID004 – garáž a relé pod ID005 je umístěno pod vypínačem u vstupních dveří.



Obrázek 15 - Přidaná zařízení v nástroji Aeon Labs IMA tool

Pro prvotní konfiguraci topného systému připojíme MAX! Cube do elektrické sítě dodávaným adaptérem, který má na druhé straně USB konektor pro zapojení do zařízení. Dále zapojíme do domácí sítě, kde od DHCP serveru dostane vlastní IP adresu, kterou mu nejlépe na DHCP serveru zarezervujeme dle jeho MAC adresy (konkrétní zarezervovaná IP adresa je 192.168.22.102/24). Aby byla možná komunikace mezi komponenty MAX!, je potřeba nainstalovat MAX! PC software, dostupný na webových stránkách výrobce (<http://max.eq-3.de>). Po spuštění programu budeme vyzváni k zadání IP adresy, která byla přiřazena zařízení MAX! Cube. Po jejím zadání se otevře okno v prohlížeči pod adresou <http://localhost:14368/#>, kde v průvodci proběhne spárování s termostatickými hlavicemi, které musí být zapnuté (napájené bateriemi).

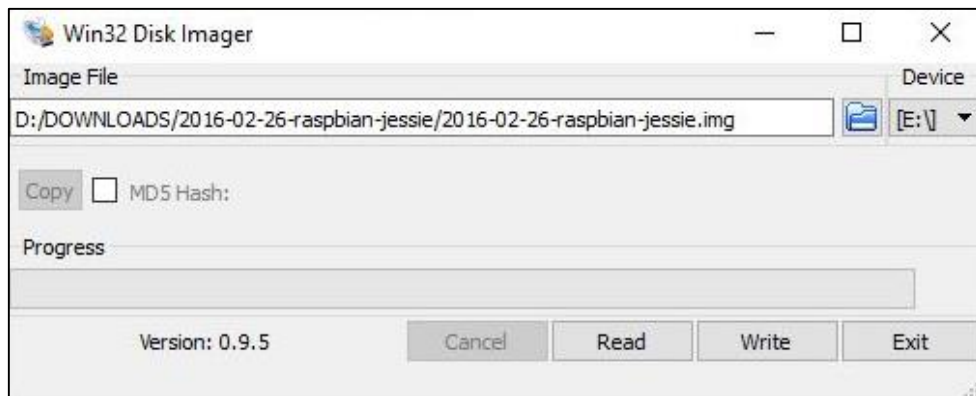
3.3 Instalace a konfigurace openHAB

Instalace aplikace openHAB na Raspberry Pi a konfigurace vazeb zařízení se skládá z několika kroků, které jsou dále podrobně popsány.

3.3.1 Příprava SD karty z Windows

Raspberry Pi nemá vlastní uložisko, ale má slot na mikro SD kartu, na kterou se operační systém instaluje. Existuje více způsobů instalace operačního systému, například balíček NOBS (New Out of Box Software). Ten v sobě ale obsahuje více distribucí zároveň. Zde je použit čistý systém Raspbian (Kernel version 4.1) dostupný z oficiálních stránek

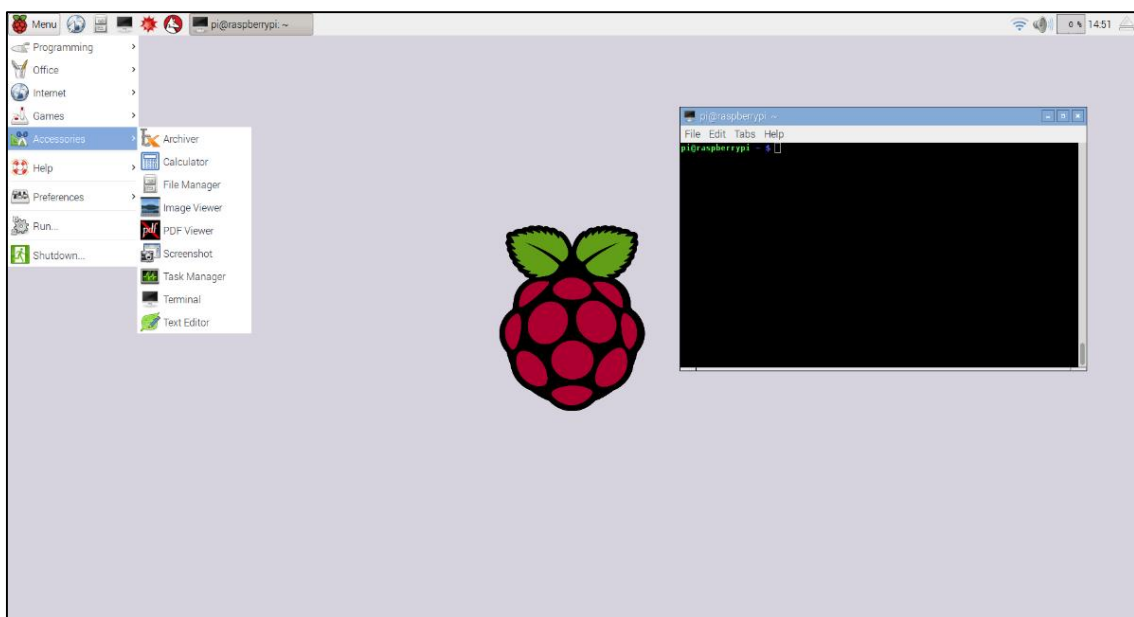
Raspberry Pi (www.raspberrypi.org). Raspbian je operační systém založený na Debianu, který je nutné mít na Raspberry Pi nainstalovaný dříve, než se začne s instalací samotné aplikace openHAB. Instalace na SD kartu probíhá tak, že se vloží SD karta (stačí 4 GB, optimálně 8 GB) do počítače a pomocí nástroje Win32 Disk Imager (obrázek číslo 16) se vybere stažený image soubor a SD karta, kam má být systém zapsán. Po stisknutí tlačítka Write dojde k zápisu systému na paměťovou kartu.



Obrázek 16 - Zapsání systému Raspbian na SD kartu programem Win32 Disk Imager

3.3.2 Spuštění Raspbian a instalace openHAB

Po připojení periférií (klávesnice, myš, monitor, Wifi dongle) k Raspberry Pi zapojíme i napájecí adaptér. Po jeho zapojení se zařízení spustí a nabojuje operační systém Raspbian (obrázek číslo 17).

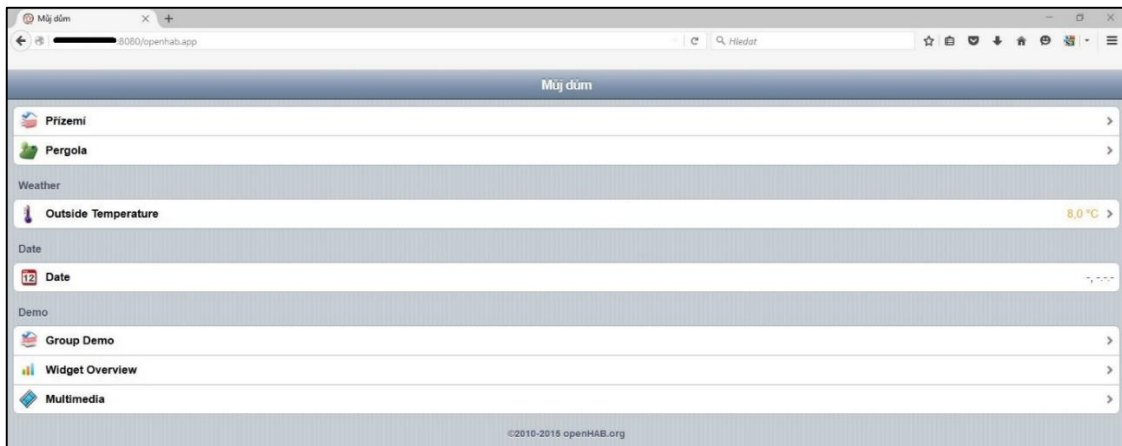


Obrázek 17 - Základní obrazovka Raspbianu

Po úspěšném spuštění systému se připojíme k internetu. Pro připojení k síti vybrat buďto kabelové připojení ethernetovým kabelem, nebo pomocí USB Wifi dongle. V tomto případě je použita druhá volba. Připojení k wifi síti se provede v pravém rohu obrazovky, kde si vybereme odpovídající síť, zadáme (pokud jsou) přihlašovací údaje, potvrdíme a počkáme na připojení. Příkazem `ifconfig` zjistíme aktuální síťové nastavení. Později se nám bude hodit statická IP adresa, jež se nakonfiguruje v souboru `/etc/network/interfaces`, kde si do vybraného rozhraní nastavíme statickou IP adresu (v tomto případě 192.168.22.101/24). Dále si otevřeme terminál. V terminálu získáme administrátorská práva příkazem `sudo su`. Následně je možné zkontrolovat dostupnost systémových aktualizací terminálovým příkazem `apt-get update`. Pokud se podařilo najít aktualizace, poté příkazem `apt-get upgrade` nainstalujeme všechny vyhledané balíčky, a s tím také aktualizace. Nyní provedeme restart příkazem `reboot`.

V tuto chvíli je vše připraveno pro instalaci samotného openHABu. Jelikož je openHAB založen na Java technologii, je potřeba ověřit, zda je v Raspbianu nainstalována odpovídající verze. Zjistíme to příkazem `java -version`. Pokud je nainstalována, dostaneme odpověď. V tomto případě je nainstalována verze 1.8.0.65. Pokud není, je potřeba Javu nainstalovat a to přidáním do repozitářů příkazem `add-apt-repository ppa:openjdk-r/ppa`, načtením aktualizací příkazem `apt-get update` a vlastní instalací příkazem `apt-get install openjdk-8-jre`. Instalaci openHABu začneme tak, že si nejdříve přidáme openHAB do seznamu ověřených balíčků, a to příkazem `wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?username=openHAB' | sudo apt-key add -`. Dále do seznamu repozitářů příkazem `echo "deb http://dl.bintray.com/openHAB/apt-repo stable main" | tee /etc/apt/sources.list.d/openHAB.list`. Nyní obnovíme seznam instalovaných balíčků příkazem `apt-get update`, poté nainstalujeme openHAB příkazem `apt-get install openHAB-runtime`. První spuštění openHABu lze provést zadáním příkazu `/etc/init.d/openHAB/start.sh`. Pro nastavení automatického spouštění openHABu po zapnutí Raspberry Pi zadáme příkaz `update-rc.d /etc/init.d/openHAB/start.sh`. Spuštěný openHAB nalezneme pod již dříve nastavenou statickou IP adresou Raspberry Pi a portem 8080. IP adresa tedy bude vypadat takto: `http://XXX.XXX.XXX.XXX:8080/`. Lze také použít

ve webovém prohlížeči localhost adresu, a to v této podobě: `http://localhost:8080/`. Tu lze použít pouze v přístupu do rozhraní ze zařízení Raspberry Pi. Prostředí openHAB se spuštěnou demoverzí je na následujícím obrázku (číslo 18). Dále si stáhneme do adresáře `addons` balíček `sbindy`, a to příkazem `wget https://bintray.com/artifact/download/openHAB/bin/distribution-1.8.1-addons.zip`. Tento zip soubor v tomto adresáři rozbalíme příkazem `unzip distribution-1.8.1-addons.zip`



Obrázek 18 - Prostředí openHAB po instalaci

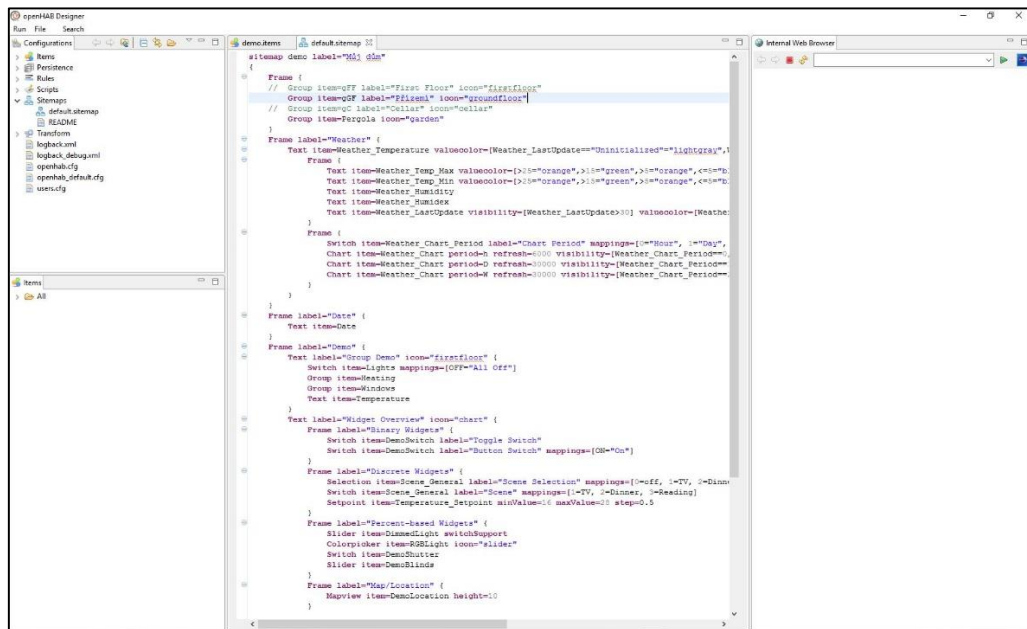
V našem případě, kdy bude ovládání osvětlení zprostředkováno Z-Wave USB adaptérem, je potřeba nastavit jeho statické nastavení, aby se po restartu zařízení nezměnil jeho název. Konfigurace probíhá v souboru `/etc/udev/rules.d/50-usb-serial.rules`, který již buď existuje, nebo si jej vytvoříme. V tomto souboru jsou nastavená pravidla pro USB porty. Abychom tyto atributy zjistili, zadáme příkaz `sudo udevadm info --attribute-walk --path=/sys/bus/usb-serial/devices/ttyUSB0`, dostaneme výpis atributů, a ty v požadovaném formátu uložíme jako pravidlo. Pravidlo pro Z-Wave USB adaptér:

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",  
SYMLINK+="USBzwave", GROUP="dialout", MODE="0666"
```

3.3.3 Konfigurace openHAB runtime a vazeb

Konfigurace openHABu probíhá pouze na úrovni textových souborů, a to pomocí jakéhokoliv editoru pro úpravu textu. Pro komplexnější konfiguraci slouží nástroj openHAB Designer, který je možné nainstalovat na platformy Windows, OS X, Linux. Tato aplikace využívá vývojové prostředí Eclipse.

OpenHAB designer (obrázek číslo 19) je samostatná Eclipse aplikace pro konfiguraci openHAB runtime. Výhoda oproti textovým editorům spočívá v kontrole syntaktických chyb, dále obsahuje zvýraznění syntaxe a dokáže automaticky dokončovat klíčová slova.



Obrázek 19 - Prostředí openHAB Designer

Vlastní konfigurace openHABu je rozvržena do více souborů, ve kterých se nastavuje například vzhled položek v uživatelském rozhraní, pravidla, přidání nové položky. Konfigurační soubory se nacházejí v adresáři openhab/configurations. Veškeré tyto konfigurace jsou rozčleněné do následujících sedmi tříd:

- Items
- Persistence
- Rules
- Scripts
- Sitemap
- Actions
- Transform

V adresáři /configurations se kromě výše uvedených položek nachází i jiné konfigurační soubory. Soubor logback.xml automaticky vypíše stav serveru do okna konzole. Pro konfiguraci vazeb a balíčků slouží soubor openhab.cfg, soubor

`user.cfg` je pro nastavení autentizačních údajů (jméno a heslo obsluhujících uživatelů).

Již po instalaci jsme stáhli `addons.zip` balíček, který obsahoval bindy. V souboru `openhav.cfg` si je musíme nakonfigurovat. Pro MAX!Cube se konfiguruje pět položek:

- IP adresa, tu jsme již dříve nastavili na 192.168.22.102
- port je nastaven defaultně na 62910, pokud není potřeba, neměnit
- interval obnovení (v ms) defaultně nastaven na 10 sekund, tedy hodnota 10000
- exkluzivní mód se používá, pokud se do MAX!Cube připojujeme mobilním zařízením
- maximální počet požadavků na připojení se defaultně stanovuje na 1000

```
##### MAX!Cube Binding #####
maxcube:ip=192.168.22.102
maxcube:port=62910
maxcube:refreshInterval=10000
maxcube:exclusive=false
maxcube:maxRequestsPerConnection=1000
```

V MAX! PC software, který byl zřízen již pro prvotní nastavení konfiguratoru termostatických hlavic, si vyhledáme a vypíšeme jednotlivá sériová čísla hlavic, která budeme dále potřebovat pro konfiguraci jednotlivých termostatů. V tomto případě se jedná o tato sériová čísla:

```
maxcube = "< KIC9665851 (obývací pokoj); KIC9665852 (pracovna);
KIC9665858 (pokoj); KIC9665863 (kuchyň); KIC9665864 (koupelna);
KIC9665866 (chodba); KIC9665880 (WC); KIC9665881 (předsíň)>".
```

Pro osvětlení realizované domácnosti se v `openhav.cfg` nastavují bindy v položce Z-Wave Binding, a to dva parametry:

- port, ve kterém je Z-Wave USB modul nainstalován, v tomto případě `ttyUSBzwave`
- hodinu, ve kterou se obnovují síťové uzly, doporučená hodnota je 2 (ve 2 hodiny)

```
##### Z-Wave Binding #####
zwave:port=/dev/ttyUSBzwave
zwave:healtime=2
```

- Items (položky)

V adresáři /configurations/items se pro konfiguraci Items nacházejí soubory s koncovkou .items. Items soubory jsou určeny k vytváření nových položek a jejich propojení s vazbami. Každá položka může být jiného typu. V Items se konfigurují také jednotlivé místnosti, které mají jako nadřazenou třídu položku ze Sitemap, v tomto případě Přízemí, tak, jak je znázorněno na obrázku číslo 20. Aktuální výčet položek je v tabulce číslo 3. (Turner, 2016)



Obrázek 20 - Prostředí openHAB po konfiguraci místností v Items

Položky se definují následující syntaxí:

```
itemtype itemname ["labeltext"] [<iconname>] [(group1, group2, ...)]
[{{bindingconfig}}
```

- Itemtype – značí typ položky
- Itemname – jedinečný název položky
- Labeltext – štítek, tento text se zobrazuje v popisu konkrétní položky

- Iconname – lze si vybrat png obrázek ze složky /webapps/images/, velikost obrázku musí být 32 x 32 pixelů
- Group – položka může být členem několika skupin
- Bindingconfig – nejdůležitější položka, definující veškerou funkčnost položky

Tabulka 3 - Seznam konfiguračních možností Items

Název položky	Popis položky	Akceptované datové typy	Akceptované příkazy
Call	Položky týkající se fce telefonu	Call	-
Color	Používá se pro barevné hodnoty, například pro LED osvětlení	OnOff, Percent, HSB	OnOff, Percent, IncreaseDecrease,
Contact	Používá se pro senzory, které říkají „otevřeno“ nebo „zavřeno“. Například u dveří.	OpenClosed	-
DateTime	Ukládá časové razítko včetně časového pásma	DateTime	DateTime
Dimmer	Přijímá příkazy ON/OFF. Lze nastavovat %, například u osvětlení.	OnOff, Percent	OnOff, Percent IncreaseDecrease,
Group	Položka se vnoří do jiné položky - seskupování	Dle nadř. pol.	Dle nadř. pol.
Location	Používá se na GPS informace	Point	Point
Number	Dec. hod., pro všechny senz. – teplota, jas, vítr	Decimal	Decimal
Rollershutter	Ovládá rolety – nahoru, dolů, zastavení	UpDown, Percent	UpDown, StopMove, Percent
String	Používá se pro vložení řetězce (String) nebo DateTime	String, DateTime	String
Switch	Vypínač, používá se u osvětlení, detekce přítomnosti	OnOff	OnOff

V našem konkrétním případě je konfigurace následující (příklad pro skupinu obývací), na obrázku 21 je znázorněno rozhraní.

Osvětlení – uvedená konfigurace je pro obývací pokoj, pro ostatní pokoje, kde je nainstalované řídicím systémem ovládané osvětlení, je realizace stejná:

```
Switch Light_GF_Obyvak_Svetla "Světla" <Lights> (GF_Obyvak, Lights)
{zwave="USBzwave:002,command=SWITCH_BINARY"}
```

Topný systém – nastavení topení zap/vyp (v rules je nastavené pravidlo pro ovládání), Uvedená konfigurace je pro obývací pokoj. Pro ostatní pokoje, kde je nainstalovaný topný systém, je realizace totožná:

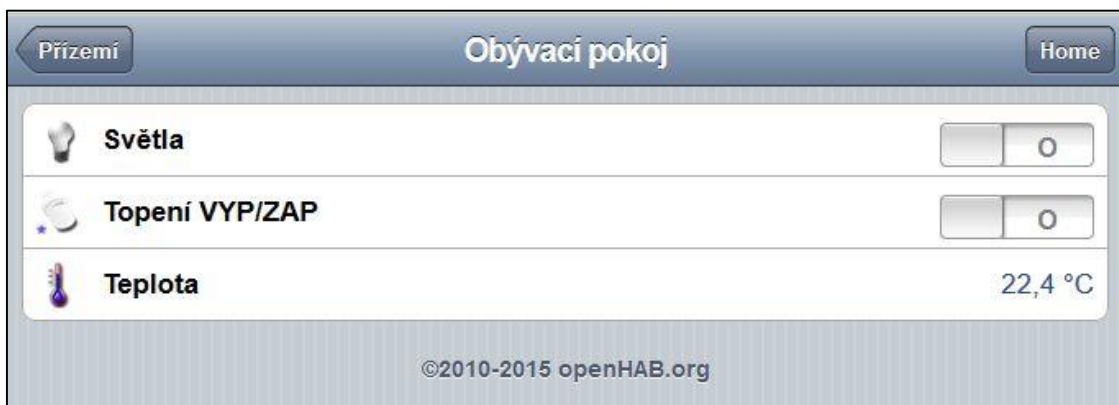
```
Switch Temperature_GF_Obyvak "Teplota [MAP(maxcube.map):%s]"
<temperature> (Temperature, GF_Obyvak) { maxcube=" KIC9665851" }
```

Topný systém – aktuální teplota. Uvedená konfigurace je pro obývací pokoj. Pro ostatní pokoje, kde je nainstalovaný topný systém, je realizace stejná:

```
Number Temperature_GF_Obyvak "Teplota [%.1f °C]" (Temperature,
GF_Obyvak) { maxcube=" KIC9665851:type=actual" }
```

Topný systém – stav baterie. Uvedená konfigurace je pro obývací pokoj. Pro ostatní pokoje, kde je nainstalovaný topný systém, je realizace stejná:

```
Number battery_GF_Obyvak "Stav baterie [%s]" <battery> (GF_Obyvak,
settings) { maxcube=" KIC9665851:type=battery"}
```



Obrázek 21 - Prostředí openHAB po konfiguraci zařízení v Items

- Persistence

Persistence ukládá v reálném čase data do vybraného uložště. Mezi podporované uložště patří relační databáze, round-robin databáze a cloudové služby. Příklady použitelných databází:

- db4o – objektová databáze vytvořená na Java
- InfluxDB – open-source distribuovaná databáze
- MongoDB – NoSQL databáze
- RRD4J – Java databáze, pouze pro numerické hodnoty. (Martinez, 2016)

Konfigurační soubory databáze s koncovkou `.persist` nalezneme ve složce `/configurations/persistence`. Tyto soubory lze editovat pomocí openHAB Designeru. Konkrétně se konfiguruje, které položky a kdy se mají ukládat. K tomu slouží Strategies. Syntaxe vypadá takto:

```

Strategies {
  <strategyName1> : "<cronexpression1>"
  <strategyName2> : "<cronexpression2>"
  ...
  default = <strategyNameX>, <strategyNameY>
}

```

Parametr `strategyName1` uvádí náš název strategie. `Cronexpression1` uvádí četnost ukládání. Každá položka, kterou chceme ukládat, musí mít přiřazenou strategii následující syntaxí:

```

Items {
  <itemlist1> [-> "<alias1>"] : [strategy = <strategy1>,
<strategy2>, ...]
  <itemlist2> [-> "<alias2>"] : [strategy = <strategyX>,
<strategyY>, ...]
  ...
}

```

- **Rules**

Pro konfiguraci Rules (pravidel) jsou soubory s koncovkou `.rules` umístěny ve složce `/configuration/rules`. Veškerá pravidla obsažená v konfiguračním souboru mají mezi sebou vztahy, díky kterým je možná vzájemná výměna proměnných, nebo k nim lze navzájem přistupovat.

U topného systému je hromadně nakonfigurována teplota, na kterou se od určité hodiny mají otevřít termostatické hlavice. V tomto řešení jsou vytvořena pravidla pro noc a den. Teplota na večer a noc je nastavena na dvacet dva stupňů a teplota od ranních hodin do večera je nastavena na dvacet stupňů. Syntaxe pravidla je následující:

```

rule "Noc"
  when
    Time cron "0 0 19 * * ?" //s, min, hod, dny, měs, dny v týd, roky
  then
    sendCommand (Heating_Max_Valve, 22 )
  end
  rule "Den"
  when
    Time cron "0 0 8 * * ?" //s, min, hod, dny, měs, dny v týd, roky
  then
    sendCommand (Heating_Max_Valve, 20 )
  end

```

Dále je zde nakonfigurované pravidlo pro kontrolu stavu baterií v jednotlivých místnostech:

```

import org.openhab.core.library.types.DecimalType

val String mailTo = "stanislavdvo@gmail.com"
val int lowBatteryThreshold = 10 //nastaveni nizke hranice

rule "Kontrola_stavu_baterie"
when Time cron "0 0 6 * * ?" //probiha v 6 hodin rano
then
    if (! Battery.allMembers.filter([state <
lowBatteryThreshold]).empty) {
        val message = ("Baterie v termostatu je vybitá!!!")//musím
zjistit jaky je to termostat
        sendMail(mailTo, "Nizky stav baterie", message)
    }
end

```

- Skripts

Pro definování skriptů se v programu openHAB využívá jazyk Java s nadstavbou Xtend. Konfigurační soubory uložené ve složce /configuration/scripts mají koncovku .scripts. Vyvolání skriptu probíhá příkazem callscript, například callscript (název skriptu).

- Sitemap

Sitemap (mapa stánek) slouží pro konfiguraci grafického rozhraní. Jsou zde informace o tom, které prvky a kde se mají zobrazit. Ve složce /configuration/sitemap se nacházejí soubory s koncovkou .sitemap, které mají v sobě tyto informace obsažené.

Pro vytvoření rozhraní slouží v openHAB více grafických prvků. Pro každý z těchto prvků je potřeba minimálně jeden povinný parametr, který je ale u každého prvku jiný. Většinou je vyžadován item parametr, díky kterému definujeme položku, kterou prvek zobrazí. Vybrané prvky požadují webovou adresu, na které je uložený například obrázek. Níže v tabulce číslo 4 jsou uvedeny grafické prvky. (GitHub, 2016)

Tabulka 4 - Výčet grafických prvků

Prvek	Popis prvku
Colorpicker	Umožní uživateli vybrat barvu
Chart	Přidá objekt s časem zaznamenaných dat
Frame	Rozdělí prvky rozhraní do rámečků
Group	Vykreslí prvky dané skupiny
Image	Vykreslí obrázek
List	Zobrazí stav položek

Prvek	Popis prvku
Selection	Povoluje výběr -> menu
Setpoint	Umožní po krocích změnu hodnot (+/-)
Slider	Vykreslí posuvník
Switch	Vykreslí přepínač
Text	Vypíše textové pole
Video	Zobrazí video

Syntaxe konfiguračního souboru musí být uvozena slovem `sitemap`, za kterou se uvádí jméno mapy stránek. Příklad:

```
sitemap name_map{
Frame [label="<labelname>"] [icon="<icon>"] [item=<item>]
{
    [additional sitemap elements]
}
}
```

V našem případě budeme v rozhraní potřebovat ovládat osvětlení a teplotu. Rozhraní si přizpůsobíme tak, že vytvoříme položku přízemí, pod kterou budou začleněny jednotlivé místnosti a v místnostech budou možnosti ovládání osvětlení či nastavení teploty. V rozhraní vytvoříme i položku s datem. Podoba souboru `sitemap` je níže a na obrázku číslo 22 je znázorněno rozhraní `openHABu`.

```
sitemap demo label="Můj dům"
{
    Frame {
        Group item=gGF label="Přízemí" icon="groundfloor"
    }
    Frame label="Date" {
        Text item=Date
    }
}
```



Obrázek 22 - Prostředí `openHAB` po konfiguraci `sitemap`

V sitemap se dále konfiguruje pro topný systém rozsah teplot v jednotlivých místnostech. Pro příklad je zde uveden rozsah teplot v obývacím pokoji, který je tedy nastavený mezi 19 a 22 stupni:

```
Setpoint Temperature_GF_Obyvak=Heating_Max_Valve step=0.5 minValue=19
maxValue=22
```

- **Actions**

Actions (akce) jsou staticky importované metody psané v jazyce Java. Dělí se do dvou základních skupin: Core Actions a Add-on Actions, tedy akcí jádra a akcí doplňků.

Akce jádra lze separovat do více bloků. Nejprve na akce, které jsou provázané se sběrnici událostí:

- `sendCommand(String itemName, String commandString)` – odešle příkaz vybrané položce
- `postUpdate(String itemName, String stateString)` – aktualizace stavu položky
- `group.send(Command command_to_send)` – odešle příkaz (ON/OFF, atd.) každé položce ve skupině. (Cocula, 2016)

Dále je zde skupina s událostmi týkajícími se zvuků:

- `setMasterVolume(float volume)` – nastavení hlasitosti
- `increaseMasterVolume(float percent)` – zvýšení hlasitosti o X procent
- `decreaseMasterVolume(float percent)` – snížení hlasitosti o X procent
- `float getMasterVolume()` – vrátí hodnotu hlasitosti v rozmezí mezi 0-1
- `playSound(String filename)` – přehraje určitý soubor
- `playStream(String url)` – přehraje audio soubor ze zadané url adresy
- `say(String text)` – přečte daný text prostřednictvím Text-to-speech
- `say(String text, String voice)` – Text-to-speech s vybraným hlasem
- `say(String text, String voice, String device)` – Text-to-speech s vybraným hlasem na určité zařízení

Další skupinou jsou akce časovače, http akce, nebo také logovací akce, které logují na více úrovních, konkrétně na ladící, chybové, výstražné a informační úrovni.

Akce doplňků jsou různé, liší se podle vlastního určení daného doplňku. Například doplněk MAIL pro odesílání emailů pomocí SMTP (Simple Mail Transfer Protocol) odešle email na zadané emailové adresy. Syntaxe příkazů pro odeslání emailů je následující:

- `sendMail(String to, String subject, String message)` – odešle email přes SMTP
- `sendMail(String to, String subject, String message, String attachmentUrl)` – odešle email s přílohou
- `sendMail(String to, String subject, String message, List<String> attachmentUrlList)` – odešle email s jednou nebo více přílohami (dostupné od verze openHAB 1.8). Příklad syntaxe:

```
import java.util.List
...
val List<String> attachmentUrlList = newArrayList(
    "http://some.web/site/snap.jpg&param=value",
    "http://192.168.22.101/data.txt",
    "file:///tmp/201601011031.jpg")
sendMail("you@email.net", "Test", "This is the message.",
attachmentUrlList)
```

Konfigurace odesílání emailu se nachází v souboru `openhbab.cfg` v sekci `# Mail Action configuration #`. Zde se konfiguruje SMTP server odesílatele emailu, port pro komunikaci se SMTP serverem, jméno a heslo pro spojení se SMTP serverem, emailová adresa odesílatele a zda jsou ke komunikaci využité autentizační mechanismy. Tato konfigurace je znázorněna na obrázku číslo 23.

```

##### Mail Action configuration #####
#
# The SMTP server hostname, e.g. "smtp.gmail.com"
#mail:hostname=smtp.gmail.com //slouží pro zadání smtp serveru

# the SMTP port to use (optional, defaults to 25 (resp. 587 for TLS))
#mail:port=587 //port který využívá smtp server

# the username and password if the SMTP server requires authentication
#mail:username=stanislavdvorak@gmail.com //jméno uživatele smtp serveru
#mail:password=***** //heslo

# The email address to use for sending mails
#mail:from=stanislavdvorak@gmail.com //emailová adresa odesílatele

# set to "true", if TLS should be used for the connection
# (optional, defaults to false)
#mail:tls=true //zabezpečení spojení

# set to "true", if POP before SMTP (another authentication mechanism)
# should be enabled. Username and Password are taken from the above
# configuration (optional, default to false)
#mail:popbeforesmtp= //zde se nastaví hodnota true, pokud je na serveru jiný
autentizační mechanismus

```

Obrázek 23 - Konfigurace odesílání emailů

- Transform

V případě, že potřebujeme informace z webového prostředí předat další aplikaci externímu programu, je potřeba hodnoty transformovat do srozumitelné formy. Ve složce /configuration/transform jsou uloženy konfigurační soubory. Spojení %s slouží k předání hodnoty, která má být transformována do externího programu.

U topného systému předáváme z openHABu hodnoty termostatickým hlavicím. Termostatické hlavice mají rozsah hodnot 5 – 55 °C s krokováním po 0,5 s tím, že při hodnotě 5 se termostatická hlavice uzavře a při hodnotě 55 je hlavice otevřená na maximum. Toto je nadefinované v adresáři transform v souboru maxcube.map takto:

```

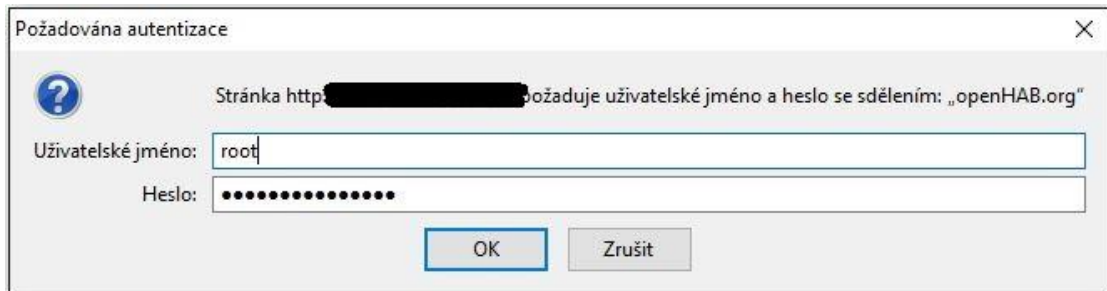
5=OFF
5.5=5.5 °C
...
54.5=54.5 °C
55=On

```

- Zabezpečení

Zabezpečení openHABu je realizováno na úrovni autentizace uživatelů. Realizace tohoto řešení probíhá prostřednictvím dvou konfiguračních souborů – openhab.cfg a users.cfg. Přihlašovací dialog je vyobrazen na obrázku číslo 24.

- V souboru `openhAB.cfg` autentizaci povolíme tak, že v části `General configurations` změníme hodnotu položky `security:option` na hodnotu `ON`.
- V souboru `users.cfg` vytvoříme uživatele a vygenerujeme mu heslo. Doporučuje se vytvoření silného hesla. Takové heslo by se mělo skládat z minimálně osmi znaků za použití velkých a malých písmen, číslic a speciálních znaků. Syntaxe vypadá takto: `root=password`.



Obrázek 24 - Přihlašovací okno

4 Závěr

Závěrem bych chtěl říci, že motivací k napsání této práce byla myšlenka vytvoření funkčního systému, který bude skutečně využíván a který bude možné v budoucnu dále rozšiřovat o další zakoupená zařízení. V této práci jsem se zabýval problematikou inteligentního domu, a to od jeho návrhu po realizaci ve skutečném domě.

Cíle mé práce, tedy návrh řešení, výběr řídicí jednotky, řídicího systému a vlastní realizace projektu, byly naplněny. Povedlo se mi tedy vytvořit funkční řešení, které vychází z řídicí jednotky, za kterou je díky malým rozměrům a nízké spotřebě vybrán mini počítač Raspberry Pi 2 Model B. Dále vychází z centralizovaného řídicího systému openHAB. Díky této kombinaci řídicí jednotky a řídicího systému je možné ovládat systém osvětlení v obývacím pokoji, kuchyni, garáži a před hlavním vstupem do domu. Automatizovaný je také topný systém, konkrétně regulace teplot v místnostech, ve kterých jsou instalované radiátory. Topný systém se povedlo nakonfigurovat tak, že během topné sezóny při provozu kotle se v místnostech udržují teploty mezi devatenácti a dva a dvaceti stupni Celsia.

Inteligentní systém domácnosti lze rozšířit o další prvky, jelikož centralizovaný systém openHAB je velmi přizpůsobivý a díky široké komunitě uživatelů má i velké množství použitelných zařízení.

Mnou vytvořený systém lze pro snížení nákladů během topné sezóny doplnit konkrétně o akumulční nádrže a solární panely. Tyto prvky by se při správném řešení nechaly využít nejen během topné sezóny, ale i pro řešení ohřevu vody v bojleru.

Citovaná literatura

Anonymous (2004). Open Source. *ABC Linuxu*. Retrieved [04-04-2016] from <http://www.abclinuxu.cz/slovník/open-source>

Anonymous (2015). Lekce 35 – Arduino – komunikace přes RS485 až na vzdálenost 1200m. *Arduino8*. Retrieved [03-02-2016] from <http://www.arduino8.cz/lekce-35-arduino-komunikace-pres-rs485-az-na-vzdalenost-1200m/>

Conrad (2015). Bezdrátová programovatelná termostatická hlavice eQ-3 MAX!+. *Conrad*. Retrieved [01-02-2016] from <http://www.conrad.cz/bezdratova-programovatelna-termostaticka-hlavice-eq-3-max.k519588>

Domoticz (2016). Domoticz Wiki Manual. *Domoticz*. Retrieved [02-03-2016] from https://www.domoticz.com/wiki/Domoticz_Wiki_Manual

Domotique-store (2015). Câbler et inclure/exclure le Qubino 1 relais contact sec "Flush 1D Relay" - ZMNHND1. *Domotique-store*. Retrieved [04-04-2016] from http://tutoriels.domotique-store.fr/content/117/311/fr/c%C3%A2bler-et-inclure_exclure-le-qubino-1-relais-contact-sec-flush-1d-relay-_zmnhnd1.html

GitHub (2016). Architecture. *Home Assistant*. Retrieved [04-04-2016] from <https://home-assistant.io/developers/architecture/>

Hunt, D. (2013). Add a 9-pin Serial Port to your Raspberry Pi in 10 Minutes. *Davidhunt*. Retrieved [04-04-2016] from <http://www.davidhunt.ie/add-a-9-pin-serial-port-to-your-raspberry-pi-in-10-minutes/>

Kabelová, A. & Dostálek, L. (2008). *Velký průvodce protokoly TCP/IP a systémem DNS* (5th ed.). Brno: Computer Press.

Kaspersky Lab (2015). Co je brána firewall? *Kaspersky Lab*. Retrieved [02-03-2016] from <http://www.kaspersky.com/cz/internet-security-center/definitions/firewall>

Olmr, V. (2005). HW server představuje - Sériová linka RS-232. *Vyvoj.HW*. Retrieved [04-04-2016] from <http://vyvoj.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>

Raspberry Pi (2015). GPIO: MODELS A+, B+, RASPBERRY PI 2 B AND RASPBERRY PI 3 B. *Raspberry Pi*. Retrieved [04-04-2016] from <https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>

Raspberry Pi (2015). RASPBERRY PI 2 MODEL B. *Raspberry Pi*. Retrieved [04-04-2016] from <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

Raspberry Pi (2015). RASPBERRY PI 2 ON SALE NOW AT \$35. *Raspberry Pi*. Retrieved [04-04-2016] from <https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/>

The smartest house (2015). Aeotec by Aeon Labs Z-Stick Z-Wave Plus Gen5 ZW090-A. *The smartest house*. Retrieved [04-04-2016] from <http://www.thesmartesthouse.com/products/aeon-labs-z-stick-gen5-z-wave-plus>

Trtík, J. (2009). *Návrh elektroinstalace rodinného domu s využitím Inteligentních prvků* (Bachelor thesis, Brno: Vysoké učení technické v Brně, fakulta elektrotechniky a komunikačních technologií).

Všetečka, R. (2015). Estonci umí posílat data světlem. Li-fi je stokrát rychlejší než wi-fi. *Technet cz*. Retrieved [04-04-2016] from http://technet.idnes.cz/li-fi-posilani-dat-lampickou-d13-/hardware.aspx?c=A151125_130708_hardware_vse

Zdařil, T. (2009). *PŘIJÍMAČ DÁLKOVÉHO OVLÁDÁNÍ*. (Bachelor thesis, Brno: Vysoké učení technické v Brně, fakulta elektrotechniky a komunikačních technologií).

Z-Wave (2016). ABOUT Z-WAVE. *Z-WAVE*. Retrieved [04-04-2016] from <http://www.z-wave.com/about>

Turner, S. (2016). Explanation of items. *GitHub*. Retrieved [04-04-2016] from <https://github.com/openHAB/openHAB/wiki/Explanation-of-items>

Martinez, I. (2016). Persistence. *GitHub*. Retrieved [04-04-2016] from <https://github.com/openhab/openhab/wiki/Persistence>

GitHub (2016). Explanation of Sitemaps. *GitHub*. Retrieved [04-04-2016] from <https://github.com/openhab/openhab/wiki/Explanation-of-Sitemaps>

Cocula, J. (2016). Actions. *GitHub*. Retrieved [04-04-2016] from <https://github.com/openhab/openhab/wiki/Actions>

Seznam obrázků

Obrázek 1 - Schéma funkcí inteligentního domu	4
Obrázek 2 - Raspberry Pi 2 Model B.....	7
Obrázek 3 - Rozložení GPIO na Raspberry Pi 2 Model B	7
Obrázek 4 - Schéma zapojení SMART TV do sítě LAN	8
Obrázek 5 - Modul rozšiřující Raspberry PI o RS232.....	8
Obrázek 6 - Schéma zapojení SMART TV do sítě LAN prostřednictvím Wi-Fi.....	9
Obrázek 7 - Půdorys domu	14
Obrázek 8 - Umístění osvětlení	15
Obrázek 9 - Schéma zapojení relé Z-Wave Plus – Qubino ZMNHND1	15
Obrázek 10 - USB modul ZW090	16
Obrázek 11 - Termostatická hlavice eQ-3 MAX!+ nainstalována na radiátor	16
Obrázek 12 - Umístění radiátorů	17
Obrázek 13 - Konfigurátor termostatických hlavic eQ-3 MAX! Cube Lan Gateway spolu se zapojeným Raspberry Pi.....	18
Obrázek 14 - Schéma realizované sítě.....	21
Obrázek 15 - Přidaná zařízení v nástroji Aeon Labs IMA tool	22
Obrázek 16 - Zapsání systému Raspbian na SD kartu programem Win32 Disk Imager	23
Obrázek 17 - Základní obrazovka Raspbianu.....	23
Obrázek 18 - Prostředí openHAB po instalaci.....	25
Obrázek 19 - Prostředí openHAB Designer	26
Obrázek 20 - Prostředí openHAB po konfiguraci místností v Items	28
Obrázek 21 - Prostředí openHAB po konfiguraci zařízení v Items	30
Obrázek 22 - Prostředí openHAB po konfiguraci sitemap	33
Obrázek 23 - Konfigurace odesílání emailů	36
Obrázek 24 - Přihlašovací okno.....	37

Seznam tabulek

Tabulka 1 – HW výbava Raspberry Pi 2 Model B	6
Tabulka 2 - porovnání řídicích systémů	13
Tabulka 3 - Seznam konfiguračních možností Items.....	29
Tabulka 4 - Výčet grafických prvků	32