


Automatizace domácnosti pomocí Raspberry Pi s využitím dálkového ovládání a měření

Bc. David Fogl

Diplomová práce
2016

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

*** Nascanované zadání, strana 1 ***

*** Nascanované zadání, strana 2 ***

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomové práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Cílem práce je vytvořit řídicí systém pro domácnost na bázi Raspberry Pi umožňující měření teploty ve všech pokojích i venku, ovládání světel pomocí RF transmitteru a senzoru pohybu, ovládání žaluzií, atd. Dále bude výsledkem GUI pro PC i mobil pro ovládání všech zařízení.

Klíčová slova: Raspberry Pi, NodeMCU, lua, python, automatizace, Internet věcí, MQTT, WiFi

Abstract

The goal of this thesis is to create manageable system controlling home environment via Raspberry Pi with remote modules for measuring temperature in all rooms and also on the outside, remote controlling of lights via RF transmitters and motion sensors, controlling the venetian blinds and others. This system will be controllable using GUI for both desktop and mobile devices.

Keywords: Raspberry Pi, NodeMCU, lua, python, automation, Internet of Things, MQTT, WiFi

Rád bych poděkoval Ing. Tomáši Sysalovi za vedení diplomové práce a čas strávený při konzultacích. Rovněž bych rád poděkoval své přítelkyni za toleranci při vypracování této práce a trpělivost s neustálými experimenty s "inteligentním" ovládáním domácnosti.

OBSAH

ÚVOD	11
I TEORETICKÁ ČÁST	11
1 INTELIGENTNÍ ŘÍZENÍ DOMÁCNOSTI	13
1.1 MOŽNOSTI AUTOMATIZACE BĚŽNÝCH DOMÁCÍCH ZAŘÍZENÍ.....	14
1.2 ZABEZPEČENÍ DOMÁCNOSTI	15
1.3 KONTROLA DŮLEŽITÝCH PRVKŮ DOMÁCNOSTI.....	15
1.3.1 Osvětlení	16
1.3.2 Žaluzie a rolety	16
1.3.3 Měření a sběr senzorických dat.....	16
1.3.4 Vytápění	17
1.3.5 Klimatizace.....	17
1.3.6 Efektivní ovládání	18
2 POROVNÁNÍ SOUČASNÝCH ŘEŠENÍ NA TRHU.....	19
2.1 KOMERČNÍ PRODUKTY.....	19
2.1.1 Loxone	19
2.1.2 Fibaro	22
2.1.3 Control4.....	24
2.2 VÝHODY A NEVÝHODY REALIZOVANÉHO ŘEŠENÍ.....	27
2.3 POROVNÁNÍ CENY JEDNOTLIVÝCH ŘEŠENÍ.....	28
3 BEZDRÁTOVÉ TECHNOLOGIE	29
3.1 WiFi	30
3.1.1 Vrstvy síťového modelu	30
3.1.2 Pakety.....	33
3.1.3 Topologie WiFi sítí.....	34
3.2 INTERNET OF THINGS.....	36
3.3 MQTT	37
3.3.1 Broker.....	37
3.3.2 Topic	37
4 VZDÁLENÝ PŘÍSTUP A ZABEZPEČENÍ INTELIGENTNÍHO SYS- TÉMU	39
4.1 POUŽITÍ MOBILNÍ SÍTĚ.....	39
II PRAKTICKÁ ČÁST.....	40
5 POŽADAVKY NA ŘEŠENÍ	42

5.1	FUNKČNÍ POŽADAVKY	42
5.2	NEFUNKČNÍ POŽADAVKY	43
6	NÁVRH ARCHITEKTURY INTELIGENTNÍ DOMÁCNOSTI	44
7	MODULY PRO BEZDRÁTOVOU KOMUNIKACI	45
7.1	RASPBERRY PI	45
7.1.1	Raspberry Pi - Model A	45
7.1.2	Raspberry Pi - Model B a B+	45
7.1.3	Raspberry Pi 2	45
7.1.4	Raspberry Pi 3	46
7.1.5	GPIO	46
7.2	ARDUINO	49
7.2.1	Arduino Uno	49
7.2.2	Arduino Due	50
7.2.3	Arduino Shields	50
7.2.4	Zhodnocení	51
7.3	NODEMCU	52
7.3.1	NodeMCU v1	52
7.3.2	NodeMCU v2	52
7.3.3	GPIO	53
8	VYTVOŘENÍ BEZDRÁTOVÝCH MODULŮ	55
8.1	IMPLEMENTACE OBSLUŽNÝCH SKRIPTŮ PRO NODEMCU	55
8.1.1	Inicializace modulu	55
8.1.2	Připojení modulu k lokální WiFi	55
8.1.3	Komunikace s centrálním uzlem	56
8.1.4	Uspání modulu na specifikovaný čas	57
8.2	MODUL PRO MĚŘENÍ TEPLoty	58
8.2.1	Senzor DS18B20	58
8.2.2	Návrh a implementace bezdrátového senzoru DS18B20	59
8.2.3	Senzor DHT-11 a DHT-22	62
8.2.4	Návrh a implementace bezdrátového senzoru DHT-11	62
8.3	MODUL PRO MONITOROVÁNÍ VLHKOSTI PŮDY	65
8.3.1	Senzor Produino LM393	65
8.3.2	Návrh a implementace bezdrátového senzoru	66
8.4	MODUL PRO DETEKCI POHYBU	68
8.4.1	Návrh a implementace detekčního bezdrátového modulu	68
8.5	MODUL PRO OVLÁDÁNÍ DOMÁCÍHO OSVĚTLENÍ	71

8.5.1	Patice umožňující bezdrátovou rádiovou komunikaci	71
8.5.2	Nová generace bezdrátových žárovek	71
8.5.3	Implementace systému Milight	75
8.6	MODUL PRO REGULACI OKENNÍCH ŽALUZÍ	78
8.6.1	Krokový elektromotor 28BYJ-48	79
8.6.2	Návrh a implementace regulátoru žaluzií	79
9	KAMEROVÝ SYSTÉM.....	82
9.1	INSTALACE KAMERY DO RASPBERRY PI	82
9.2	IMPLEMENTACE STREAMOVACÍ SLUŽBY	84
9.3	VZDÁLENÝ PŘÍSTUP.....	86
9.3.1	Mobilní aplikace pro vzdálený přístup	86
10	IMPLEMENTACE SERVEROVÉ APLIKACE OPENHAB	87
10.1	ARCHITEKTURA.....	87
10.1.1	Sběrnice událostí	88
10.1.2	Datové uložení	88
10.1.3	Automatizační logika	88
10.1.4	Konzole	88
10.1.5	Logování stavů	88
10.1.6	Vazby	89
10.2	ZABEZPEČENÍ.....	89
10.2.1	Nastavení zabezpečení.....	89
10.3	KONFIGURACE.....	89
10.3.1	Položky	90
10.3.2	Mapa stránek	91
10.3.3	Pravidla	92
10.3.4	Skripty	93
10.3.5	Akce	93
10.3.6	Perzistence	94
10.3.7	Transformace	94
10.4	NÁSTROJE PRO ÚPRAVU KONFIGURACE.....	95
10.5	INSTALACE OPENHAB NA RASPBERRY PI	96
10.5.1	Instalace Java	96
10.5.2	Instalace OpenHab	96
10.5.3	Konfigurace OpenHab	97
10.6	UŽIVATELSKÉ ROZHRANÍ	99
10.7	INTEGRACE S BEZDRÁTOVÝMI MODULY PROTOKOLEM MQTT	103

10.7.1	Instalace Mosquitto	103
10.7.2	Nastavení bezdrátových modulů	103
10.7.3	Konfigurace OpenHab položek	104
10.8	VYTVOŘENÍ PRAVIDLA PRO KONTROLU SVĚTEL	105
10.8.1	Detektor pohybu	105
10.8.2	Implementace pravidla	105
ZÁVĚR	108
SEZNAM POUŽITÉ LITERATURY	110
SEZNAM OBRÁZKŮ	115
SEZNAM TABULEK	117
SEZNAM PŘÍLOH	118

ÚVOD

Práce se věnuje návrhu a implementaci inteligentní domácnosti s využitím jednoduchých bezdrátových modulů s cílem monitorování (sběr veličin) a ovládání aktivních prvků domácnosti. Měřené veličiny jsou zpracovávány centrální jednotkou a na základě výsledných hodnot je vykonána přednastavená akce. Typy akcí může uživatel konfigurovat formou uživatelského rozhraní a vytvářet tak sadu inteligentních pravidel, která budou reagovat na podmínky či problémy, které mohou v domácnosti nastat. Obsah práce se dělí na dvě hlavní části.

První část práce se zabývá popisem inteligentního domu a způsoby vzdáleného ovládání jednotlivých prvků domácnosti. Analýza současných řešení na trhu, srovnání výhod a nevýhod v závislosti na instalaci a provoz těchto systémů je přiblížen v druhé kapitole. Další část je věnována bezdrátovým technologiím používaných při komunikaci jednotlivých modulů s centrální stanicí. Popisujeme jak běžnou komunikaci pomocí WiFi sítí, tak pomocí tzv. RF komunikace (rádiové vlny do 1GHz). Důležitým tématem je také bezpečnost inteligentního systému a komunikace centrální stanice jak se vzdálenými moduly, tak s uživatelem mimo domácí zabezpečenou síť.

Praktická část práce se zabývá návrhem a realizací bezdrátových modulů, implementací systému a uživatelského rozhraní na centrální stanici. Nejdůležitějším prvkem celého systému je bezdrátový modul, který je vybrán z dostupných produktů na trhu na základě splnění několika hlavních kritérií. Proces výběru z možných bezdrátových modulů je popsán v kapitole sedm. Velmi důležitou funkcionalitou systému je také možnost jeho vzdáleného ovládání a monitorování. Návrh této funkcionality je rozebrán v kapitole osm. V kapitole devět se věnujeme vytváření jednotlivých modulů senzorické sítě a modulů pro ovládání vybraných prvků domácnosti. Implementace samotné serverové aplikace, propojení celého systému a uživatelské ovládání je řešeno v kapitole jedenáct a dvanáct.

V závěru práce je zhodnocen vytvořený systém a srovnání s komerčními produkty z druhé kapitoly. Podíváme se také na možnosti dalšího rozšiřování systému, které nebyly zahrnuty do této práce.

I. TEORETICKÁ ČÁST

1 INTELIGENTNÍ ŘÍZENÍ DOMÁCNOSTI

Název „Inteligentní dům“ či „Chytrá domácnost“ je pojem, který je skloňován již od 80 let minulého století. Pojem „inteligentní budova“ se poprvé objevil v USA na přelomu 80. a 90. let minulého století a vyjadřoval budovu s nadstandardním komfortem[1]. S technologickým vývojem od 80 let minulého století se mění i pohled na komfort a skutečný význam spojení „inteligentní budova“. V dnešní době, kdy je běžné, že domácnosti jsou vybaveny různými druhy elektroniky - myšleno hlavně stolní i mobilní počítače, chytré televizory schopné automaticky nahrávat televizní vysílání, bezdrátové ovládání garážových vrat a zabezpečení domácnosti proti nebezpečí zevnitř i vně, toto všechno jsou technologie, které v dnešní době bereme jako samozřejmé posouvají význam inteligentních budov dále.

Pod tímto pojmem si v současnosti představujeme plně automatizovanou domácnost, kde ovládání jednotlivých prvků domácnosti je možné skrze jednoduché uživatelské prostředí. Je kladen vysoký důraz na ekonomickou a ekologickou stránku domácnosti ústící v kontrolu a regulaci úniku tepla i například ohřev vody pomocí solárních panelů. Inteligentní budova by měla být soběstačná jednotka řídící chod domácnosti, monitorovat stav jednotlivých prvků domácnosti a upozorňovat či přímo řešit problémy, které mohou nastat.

Domácí zařízení připojená do inteligentního systému, pomocí bezdrátových technologií či LAN, jsou propojena do jedné společné sítě řízené specializovaným systémem nazývaným „Řídící jednotka“. Tento systém sbírá informace z připojených zařízení a na základě vnitřní logiky je schopen reagovat na události, které v systému mohou nastat. Jednotlivé činnosti, které by jinak musel vykonávat a vyhodnocovat uživatel domácnosti jsou automaticky řízeny a ovládány bez potřeby uživatelského zásahu.

Příklad inteligentní domácnosti může být třeba následující: Při příjezdu z práce k domu je nám automaticky otevřena brána na příjezdové cestě (uživatel je rozpoznán pomocí svého mobilního telefonu a polohy GPS), stejně tak se nám automaticky odemknou vchodové dveře domu. Při vstupu do domu se ve vstupní hale automaticky rozsvítí světlo, při vstupu do domácnosti nás nepřekvapí vysoká či nízká teplota, díky automatické regulaci teploty je v domě přesně takové teplo jaké si přejeme. Při ukládání ke spánku pomocí mobilního zařízení jednoduše uvedeme celý systém do nočního klidu, jsou automaticky vypnuta všechna světla, automaticky uzamčeny vchodové dveře a případně i spuštěny další bezpečnostní prvky. Ráno nás probudí lehce hrající oblíbená hudba, v konvici nám vře horká voda na kávu a v garáži se mezitím nahřívá interiér auta pro pohodlnou cestu do práce. Při cestě do garáže se můžeme podívat do zrcadla a vidíme předpověď počasí na aktuální den. Jakmile vyjedeme z pozemku, celý systém je opět uveden do klidu a všechny bezpečnostní prvky jsou aktivovány.

Opět tento příklad musíme brát s nadhledem. I když jednotlivé technologie již existují, je problematické je spojit do jednoho uceleného systému, který je možné ovládat z jednoho místa. Jednotlivé domácí technologie mají odlišné výrobcem dané komunikační kanály a rozhraní. V některých případech komunikační rozhraní vůbec nemají. Proto je problematické do již zavedené domácnosti instalovat inteligentní systém - jednotlivá domácí zařízení nejsou pro takovýto systém připraveny. Abychom takovouto domácnost mohli automatizovat a její jednotlivé prvky do systému přidat, je třeba zvolit přístup přes aktivní ovládání jednotlivých prvků, vytvářet moduly schopné komunikovat po různých komunikačních rozhraních či pořídit zcela nové technologicky vyspělejší zařízení.

1.1 Možnosti automatizace běžných domácích zařízení

Jedním z nejdůležitějších parametrů pro zapojení zařízení do domácí automatizace je možnost zařízení komunikovat s ostatními prvky domácnosti. V zásadě nejjednodušší formou automatizace domácího zařízení je spuštění či vypnutí zařízení na uživatelský příkaz, který je vyvolán vzdáleně. Moderní systémy a zařízení jsou většinou vybaveny určitou formou komunikačního API a tedy je možné s nimi komunikovat z jiných zařízení. Patří sem například garážové systémy, které mohou být ovládány pomocí ovladače rádiových vln (použitelné do 50-100 metrů od zařízení) nebo pomocí mobilní sítě GSM z libovolné vzdálenosti, kde je mobilní signál. Dalším zástupcem jsou vytápěcí systémy, které jsou vybaveny elektrokotlem nebo plynovým kotlem. Většina kotlů je řízena termostatem, který můžeme opět ovládat pomocí mobilní sítě GSM a pomocí SMS zpráv vytápění zapínat, vypínat nebo udržovat vybranou teplotu.

Ostatní běžná zařízení v domácnosti ale nativní podporu pro komunikaci s jinými zařízeními postrádají. Pokud i tyto zařízení chceme vzdáleně ovládat, je nutné pořízení speciálních zařízení, které komunikovat dokáží a přitom jsou schopni fyzicky ovládat dané zařízení. Nejpoužívanějšími zařízeními pro tento účel jsou dálkově ovládané zásuvky, které jsou jednoduché na instalaci a umožní nám ovládat přívod elektrické energie do zařízení dálkovým ovladačem či pomocí jiné bezdrátové technologie. Dalším příkladem jsou dálkově ovládatelné objímky pro žárovky. Opět velmi jednoduché na instalaci a ovládání pomocí dálkového rádiového ovladače. Tímto způsobem však nelze ovládat všechna zařízení v domácnosti a pro některé je nutné vytvořit specifické moduly, které dané zařízení umožní vzdáleně ovládat. Zde může být dobrým příkladem ovládání okenních žaluzií, které pro automatizaci vytvářeny v dřívější době nebyly. U moderních systémů žaluzií je kromě ručního ovládání přidávána i možnost elektrického pohonu a ovládání pomocí dálkového ovladače nebo vypínače.

1.2 Zabezpečení domácnosti

Důležitou částí inteligentní domácnosti je i systém zabezpečení proti vnějším ale i vnitřním vlivům. Z pohledu vnějších vlivů se jedná o zamezení pohybu po objektu neoprávněným osobám, přístupový systém do budovy, výstražná zařízení, kamerové systémy a možnost notifikací v případě jakékoli zaznamenané události formou SMS zpráv, mobilních aplikací a pod. Jedním z ochranných prvků může být i simulace pohybu osob v domácnosti v době, kdy jsou uživatelé mimo domov či na dovolené. Systém v nepravidelných intervalech simuluje aktivitu v domácnosti formou postupného rozsvěcení světel, spuštění televize a jiných zařízení. Z pohledu vnitřní bezpečnosti systém umožňuje detekovat kouř, detekovat kvalitu ovzduší (např. oxid uhličitý a vlhkost vzduchu), hlídat ratolesti a domácí mazlíčky.

Pro běžné domácnosti (rodinné objekty) je doporučováno použití těchto základních bezpečnostních prvků spojených do jednoho centrálního zabezpečovacího systému nazývaného také jako EZS (Elektronický zabezpečovací systém).

1. Dveřní a okenní spínače
2. Signalizační (výstražná) zařízení
3. Detekce pohybu
4. Kamerový systém
5. Detektor kouře
6. Detektor kvality ovzduší

Elektrický zabezpečovací systém lze brát jako soubor kontaktů, detektorů, čidel, tísňových hlásičů, vyhodnocovacích ústředen, prostředků poplachové signalizace, přenosových zařízení a ovládacích zařízení, které hromadně utváří jeden celek, který detekuje narušení objektu a následně pak, na určeném místě, signalizuje (opticky nebo akusticky) informaci o narušení střeženého objektu nebo prostoru[2]. Některé z těchto bezpečnostních prvků jsou implementovány a popisovány dále v této práci.

1.3 Kontrola důležitých prvků domácnosti

Mezi důležité prvky domácnosti řadíme zařízení, která mají přímý vliv na chod domácnosti, ovlivňují kvalitu života nebo chceme sledovat chod ekonomicky náročných zařízení. Všechna tato zařízení mají přímý dopad na naši domácnost a tudíž je vhodné je mít zapojené, monitorované a ovladatelné inteligentním systémem.

1.3.1 Osvětlení

Typ osvětlení a možnost automatizovaného ovládání je jedním z představitelů pojmu inteligentní domácnosti. Osvětlení je napojeno na centrální systém a na základě uživatelských požadavků, automatizovaných pravidel nebo aktivací některých senzorů (např. senzor pohybu) jsme schopni osvětlení v domácnosti plně přizpůsobit uživatelským potřebám. Výhodou inteligentního systému osvětlení je efektivita, snížení energetické náročnosti a zjednodušení uživatelských aktivit.

Instalace inteligentního systému osvětlení může probíhat ve dvou variantách. Systém je tvořen už s ideou inteligentního ovládání a elektroinstalace osvětlení je napojena na sběrníkový systém, který umožňuje jednoduchou kontrolu nad všemi připojenými zařízeními. Tento systém přináší spoustu výhod a pokud plánujeme novostavbu či renovaci domácnosti, investovat do tohoto systému se určitě vyplatí. Jasnou nevýhodou je nemožnost zapojení elektroinstalace stávajícího osvětlení bez invazivního zásahu. V tomto případě přichází jako možná druhá varianta a tou je použití vzdáleně ovladatelných objímek a novou generaci WiFi žárovek. Jednotlivé ovládací prvky je možné taktéž připojit k centrálnímu systému se stejnou funkcionalitou jako u prvního řešení.

1.3.2 Žaluzie a rolety

Důležitým článkem pro zapojení žaluzií a rolet do inteligentní domácnosti je možnost ovládat jejich pozici pomocí elektromotorků a možností vzdáleného ovládání. Samotné ovládání může být buď plně automatické - reagující na intenzitu venkovního světla či nastavených pravidel, nebo manuální pomocí bezdrátového ovladače, mobilního telefonu nebo pomocí webového rozhraní centrálního systému. Výhodou automatického nastavení jsou například letní dny, kdy na základě domácí teploty systém automaticky zatáhne žaluzie a umožní tak lepší regulaci domácí teploty. Druhým příkladem může být systém budíčku, kdy se v přednastavený čas ráno postupně začnou otevírat žaluzie pro snadnější a příjemnější probuzení.

Samozřejmostí je vytváření kalendářů a denních či nočních režimů pro plnou automatizovanou kontrolu systému žaluzií. Napojením na centrální a bezpečnostní systém je možné simulovat přítomnost uživatele i v době jeho nepřítomnosti.

1.3.3 Měření a sběr senzorických dat

Jedním z důležitých metod automatizace domácnosti je sběr relevantních dat. Mezi ty řadíme:

1. Teplota vzduchu a vody
2. Vlhkost vzduchu

3. Vlhkost půdy
4. Složení vzduchu

Všechna tato data určují, jak se bude inteligentní systém chovat. Jsou velmi důležitá pro definování automatizovaných pravidel, na základě kterých jsou řízeny a vykonávány vybrané akce. Při návrhu a umístování senzorů je třeba brát zřetel na funkci místa kam senzor umístujeme, jaké jsou důležité hodnoty pro sledování a jaké mohou nastat akce při překročení vybraných naměřených hodnot.

Sběr dat probíhá z velkého množství různých senzorů a jsou ukládána na centrální server k vyhodnocování. Z těchto dat následně systém generuje přehledné grafy dostupné na webovém rozhraní, dochází k upozorňování uživatelů při překročení nastavených hranic nebo systém automaticky řídí jiné připojené zařízení na základě naměřených hodnot.

1.3.4 Vytápění

Nejdůležitějším prvkem inteligentní domácnosti je automatizované vytápění. Protože se jedná o jeden ze základních systémů a také jeden ze systémů s nejvyšší energetickou náročností, je velmi vhodné zapojit vytápění domácnosti do centralizovaného řešení. Centrální systém umožňuje regulovat teplotu v domácnosti na základě naměřených senzorických dat, automaticky řídí zapínání a vypínání topných těles v každé místnosti nebo také řídí hlavní termostat.

Aby bylo možné vytápění řídit, je zapotřebí propojit řídicí jednotky vytápění s centrálním systémem inteligentní domácnosti.

V případě plynových či elektrických kotlů je za potřeba mít termostat schopný komunikovat s dalším zařízením, ve většině případů moderní termostaty komunikují na technologiích GSM, RF, Bluetooth nebo WiFi. Všechny tyto komunikační technologie je možné jednoduše napojit do systému inteligentní domácnosti.

Pokud máme v domácnosti centrální vytápění je potřeba ke každému topnému tělesu pořídit termostatické hlavice. Starší digitální hlavice umožňují lokální naprogramování teploty pro libovolnou část dne. Novější typy hlavic umožňují napojení na centrální stanici, kterou je možné připojit do lokální WiFi sítě a kontrolovat tak tepelná tělesa z centrálního systému.

1.3.5 Klimatizace

Obdobným způsobem jako vytápění je možné řídit klimatizaci v domácnosti. Výhodou klimatizací je již zabudované vzdálené ovládání a tedy zapojení klimatizační jednotky

do inteligentní domácnosti je velmi jednoduché. Důležitým aspektem pro výběr klimatizace je její výkon a kvalita. V závislosti na velikosti domácnosti, systému zateplení, počtu oken nebo osob je třeba vybrat dostatečně výkonnou klimatizaci. Ty jsou dostupné na trhu v několika formách - mobilní klimatizace nebo zabudované klimatizace.

Podobně jako u vytápění je klimatizace v inteligentní domácnosti řízena centrálním systémem na základě naměřených senzorických dat a nastavených pravidel. Nejen že klimatizace v letních dnech umožňuje regulovat teplotu, jednou z jejích mnoha funkcí je i zlepšování kvality vzduchu - čištění a udržování vlhkosti vzduchu.

Zapojením klimatizace do centrálního systému získáme jednoduchou kontrolu nad ochlazením či vytápěním domácnosti. Nejenže jsme schopni vzdáleně ovládat klimatizaci, systém automaticky reaguje na aktuální stav a udržuje stabilní tepelné prostředí. Systém umožňuje detekci otevření okna či dveří a automaticky vypne klimatizaci. Stejný princip nastane případně zapnutí centrálního vytápění.

1.3.6 Efektivní ovládání

Propojením všech výše popsaných zařízení a systémů do jednoho centrálního systému získáváme jednoduchý přístup a kontrolu nad celou domácností. Díky jednotnému přístupu ke všem zařízením jsme schopni vytvářet tematické režimy a scény, které přesně padnou aktuální náladě uživatele. Pod jedním tlačítkem se ukrývá nastavení celé domácnosti.

Příklad režimů a scén:

1. **Ranní probuzení** - Ráno je uživatel probuzen postupným roztahováním žaluzií, slunečním svitem a hrající hudbou. V domácnosti se mezitím roztáhnou všechny žaluzie. Při vstupu do koupelny se automaticky rozsvítí světlo a začne hrát oblíbená rádiová stanice.
2. **Uzamčení domácnosti** - Při odchodu z domácnosti se automaticky vypnou všechny nepotřebné spotřebiče, ztlumí se žaluzie, vypne klimatizace, zhasnou všechna světla a zapnou se všechny bezpečnostní prvky.
3. **Příchod do domácnosti** - Před příchodem do domácnosti systém automaticky vytopí nebo zchladí domácnost na požadovanou teplotu, vypne všechny bezpečnostní prvky, v závislosti na aktuálním času zatáhne žaluzie a rozsvítí hlavní místnosti.
4. **Kino** - Systém zapne televizi, zatáhne všechny žaluzie a ztlumí světla.
5. **Noční režim** - Uvedení systému do klidového režimu, vypne všechny světla a zapne venkovní bezpečnostní prvky.

2 POROVNÁNÍ SOUČASNÝCH ŘEŠENÍ NA TRHU

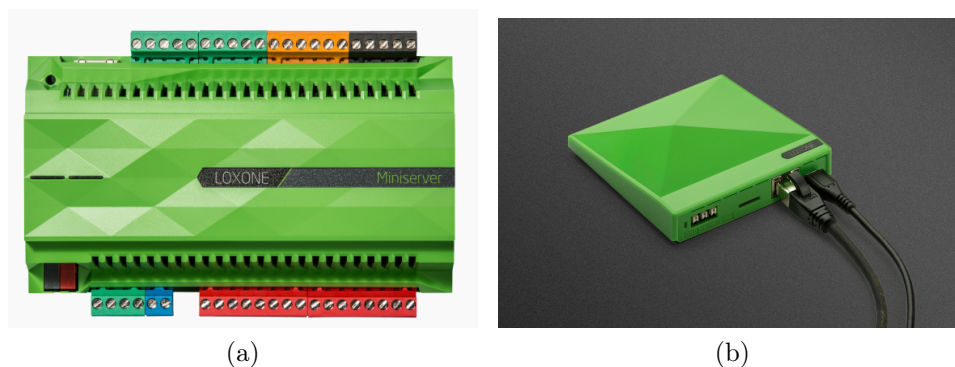
V současné době je na trhu více různých systémů pro inteligentní domácnost. V této práci se zaměřujeme na možnost využití bezdrátové kontroly, tedy nepatří sem ovládní domácnosti pomocí PLC¹⁾, které je používáno hlavně u novostaveb a je nutná speciální elektroinstalace. Nejedná se tedy o bezdrátový systém, který je jednoduše integrovatelný do běžných domácností a kterého chceme v této práci dosáhnout.

2.1 Komerční produkty

2.1.1 Loxone

V Loxone inteligentní domácnosti můžete ovládat úplně vše, od vytápění až po systém fotovoltaiky, přes tlačítka nebo pohodlně z mobilního telefonu. A co je nejlepší: Vaše domácnost ví, co má v danou chvíli udělat[3].

Společnost nabízí jak automatizaci na bázi PLC Miniserver[4] (Obr. 2.1a), který je jejich hlavní doménou, tak automatizaci pomocí Miniserver GO[4] (Obr. 2.1b), který neobsahuje PLC spoje a je tak vhodnější pro bezdrátovou komunikaci s externími zařízeními.



Obr. 2.1 Loxone Miniserver a Miniserver GO

Dalšími produkty jsou tzv. Loxone AIR a jejich účelem je propojení běžných domácích prvků do sítě. Do této produktové řady patří:

- **Smart Socket Air** (Obr. 2.2a) - bezdrátová zásuvka s integrovaným teplotním čidlem a elektroměrem pro měření spotřeby
- **Multi Extension Air** (Obr. 2.2b) - PLC rozšíření, které komunikuje bezdrátově s Loxon systémem

¹⁾PLC - programovatelný logický automat je malý průmyslový počítač, který řídí připojené jednotky pomocí digitálních a analogových vstupů na základě cyklických programů.

- **RGBW Dimmer Air** (Obr. 2.2c) - bezdrátový modul na bázi relátek pro ovládání světel
- **Loxone Remote** (Obr. 2.2d) - bezdrátový ovladač s 5 konfigurovatelnými tlačítky
- **IR Control Air** (Obr. 2.2e) - infračervený vysílač a přijímač pro ovládání televize či klimatizace
- **Detektor kouře Air** (Obr. 2.2f) - bezdrátový detektor kouře
- **Senzor teploty & vlhkosti Air** (Obr. 2.2g) - bezdrátový modul pro měření teploty a vlhkosti vzduchu
- **Touch Air** (Obr. 2.2h) - kapacitní dotykové tlačítko s 5 zónami a integrovaným čidlem pro měření teploty a vlhkosti vzduchu
- **Nano IO Air** (Obr. 2.2i) - bezdrátový modul na bázi relátek a digitálních vstupů

A mnoho dalších modulů k nalezení na webovém portálu <http://shop.loxone.com/cscz/>.

Loxone dodává i vlastní systém pro ovládání všech Loxon zařízení - Loxone OS. Ovládání je možné přes webové rozhraní a pomocí mobilních zařízení Android nebo iOS.

Na webových stránkách společnosti také najdeme vzorovou kalkulaci: Dům 4 + KK, užitná plocha domu 130m², 1x garáž, 1x chodba, 1x WC + koupelna, 1x obývací pokoj spojený s kuchyní, 2x dětský pokoj, 1x ložnice. Ovládány budou následující zařízení[5]:

přístupový systém, měření teplot, žaluzie, stmívání světel, spínání světel, LED pásky (bílá), LED RGB pásky, garáž, pohyb, topení, zásuvky, ventilátor, zvlaha, čerpadlo, zámek

Vypočítaná cena bez instalace a konfigurace systému je **57 685 Kč s DPH**.



Obr. 2.2 Loxone Air produktová řada

2.1.2 Fibaro

Fibaro nabízí snadnou instalaci bez nutnosti sekání do zdí a nevyžaduje tahání stovek metrů kabelů[6]. Produkty společnosti Fibaro jsou založeny na miniaturních modulech, které mohou být snadno instalovány bez nutnosti speciální elektroinstalace.

Mezi základní prvky inteligentní domácnosti Fibaro patří Řídící jednotka založená na tzv. Z-Wave protokolu²⁾. Řídící jednotka (Obr. 2.5) obsahuje systém pro řízení domácnosti Fibaro System Home Center 2, který umožňuje vzdálenou kontrolu nad všemi připojenými zařízeními pomocí webového rozhraní, nastavování scén a automatických pravidel. Taktéž je možné ovládat systém ze systému Android a iOS pomocí specializovaných aplikací.



Obr. 2.3 Řídící jednotka Fibaro

Do produktové řady také patří mnoho bezdrátových modulů, např.:

- Detektor kouře (Obr. 2.4a)
- Spínané zásuvky (Obr. 2.4b)
- Pohybové čidlo (Obr. 2.4c)
- Senzor na okna a dveře (Obr. 2.4d)
- Hlavice na radiátory (Obr. 2.4e)
- Ovladače na zeď (Obr. 2.4f)

²⁾Z-Wave je protokol pro bezdrátovou komunikaci určenou pro domácí automatizace a je založen na principu senzorických sítí[7]

Dále pak snímače, spínací moduly, teplotní čidla, bezdrátové ovladače a další dostupné na webové stránce <https://www.yatun.cz/produkty/smarthome-fibaro>.



Obr. 2.4 Fibaro produkty

Tab. 2.1 Přehled cen jednotlivých produktů Fibaro

Název	Cena bez DPH [Kč]
Řídící jednotka	13365
Detektor kouře	1458
Spínaná zásuvka	1458
Pohybové čidlo	1296
Senzor na okna a dveře	107
Hlavice na radiátor	1566
Ovladač na zeď	1242

2.1.3 Control4

Podobně jako předchozí produkty nabízí i společnost Control4 bezdrátové řešení inteligentní domácnosti. Integrací světla, hudby, bezpečnosti a teploty – dokonce i tabletů (Apple, Android) a chytrých telefonů (Apple, Android) – vytváří Control4 z bydlení osobitý zážitek, který navíc maximálně zvyšuje úspory, pohodlí a duševní pohodu obyvatel domu[8].

Základním prvkem systému Control4 je řídicí jednotka, kterých Control4 nabízí rovnou několik. Rozdíl mezi jednotlivými jednotkami je v jejich výkonu, množství portů, integrovaných zařízení (audio) a množstvím řízení chytrých zařízení. Dle přiložených specifikací ale každá řídicí jednotka umožňuje připojit až 50 zařízení. Každá jednotka obsahuje také vybavena rozhraním pro komunikaci protokolem ZigBee³⁾ a nabízí ovládací menu pro televizecontr. Cena jednotek začíná cca na 14 810 Kč a končí na 59 240 Kč bez DPH. Cenový rozdíl je opravdu výrazný.



Obr. 2.5 Řídicí jednotka Control4 HC-250

Dále jsou nabízeny dotykové panely přímo napojené na centrální systém Control4. Na výběr jsou nepřenosné (Obr. 2.6a) i přenosné (Obr. 2.6b) panely o velikosti 7" a 10" a v barevném provedení bílá a černá. Cena panelů se pohybuje od 19940 Kč do 29620 Kč bez DPH.

Dalším produktem je Bezdrátový Music Bridge (Obr. 2.7), který umožňuje přehrávat a sdílet hudbu z mobilních zařízení či počítače v systému Control4. Podporuje technologie AirPlay, Bluetooth a DLNA. Cílovým zařízením může být libovolný panel nebo řídicí jednotka připojená k televizi či audio systému. Cena za tuto krabičku je 6610 Kč bez DPH.

Ovládání světla je řešeno vskutku neobvykle. Jedná o zabudovatelné bezdrátové stmívače/vypínače s relátky, které obsahují několik konfigurovatelných kláves pro nastavení určité scény (Obr. 2.8). Cenově jeden tento bezdrátový stmívač s klávesnicí vychází na 5130 Kč bez DPH a vypínače/stmívače bez klávesnice jsou za 2730 Kč bez DPH.

Systém Control4 nenabízí žádné další bezdrátová čidla a moduly, využívá bezdrátových produktů společnosti Nyce, která vytváří bezdrátové moduly komunikující pomocí

³⁾ZigBee je bezdrátová komunikační technologie vystavěná na standardu IEEE 802.15.4.[10]



(a)



(b)

Obr. 2.6 Control4 dotykové panely



Obr. 2.7 Bezdrátový Music Bridge



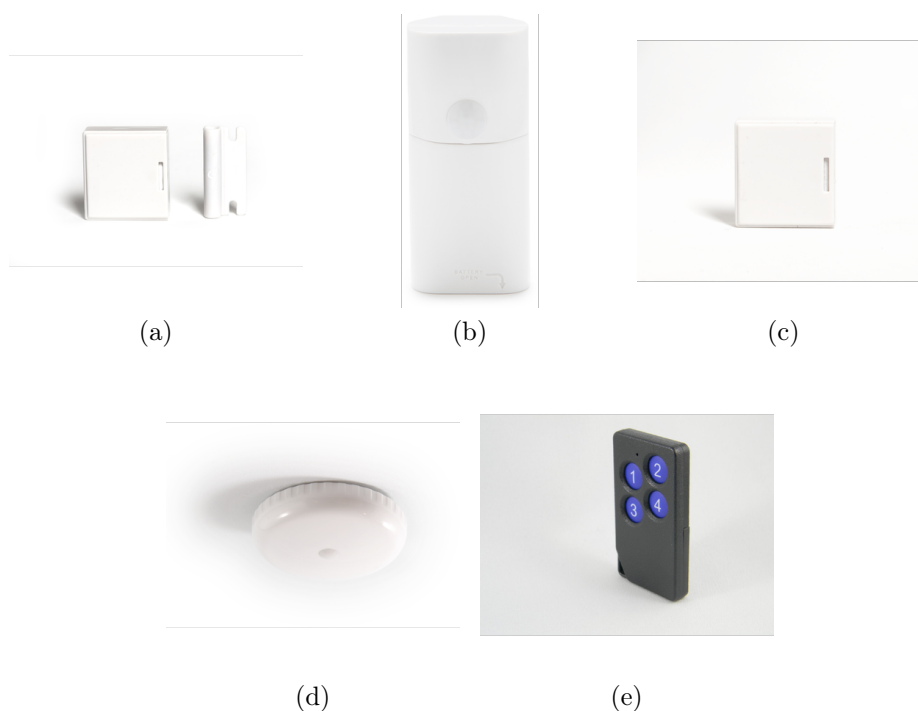
Obr. 2.8 Bezdrátová klávesnice pro ovládání světla

ZigBee. Mezi tyto produkty patří:

- **Magnetický kontakt** (Obr. 2.9a)
- **Pohybové čidlo** (Obr. 2.9b)
- **Garážový senzor náklonu** (Obr. 2.9c)
- **Stropní pohybové čidlo** (Obr. 2.9d)
- **Bezdrátový tlačítkový ovladač** (Obr. 2.9e)

Tab. 2.2 Přehled cen jednotlivých produktů Nyce

Název	Cena bez DPH [Kč]
Magnetický kontakt	1900
Pohybové čidlo	3600
Garážový senzor náklonu	1900
Stropní pohybové čidlo	3600
Bezdrátový tlačítkový ovladač	1900



Obr. 2.9 Nyce produkty

2.2 Výhody a nevýhody realizovaného řešení

Systém inteligentního ovládání domácnosti navrhovaný v této práci je postaven na třech základních myšlenkách.

Použití bezdrátových modulů

Tedy možnost automatizace domácnost neinvazivní formou. Mnoho inteligentních domácích systémů je postaveno na použití PLC a zbudováním speciální elektroinstalace, kde většina zařízení a domácích prvků je spojena kabelem za pomoci relé či jiných integračních metod. Tyto systémy jsou výhodné hlavně v případech kdy stavíme a nebo renovujeme dům či byt a pokládáme novou elektroinstalaci. Pokud ale chceme vytvořit inteligentní domácnost v novém domě nebo bytě, je tento systém velmi nákladnou volbou.

Metoda použití bezdrátových systémů a modulů je vhodná pro všechny typy domácností, bez ohledu na aktuální stav. Není nutné jakkoli invazivně zasahovat a měnit domácnost. Pokud nechceme pořizovat nové zařízení do domácnosti, za účelem jejich automatizace, můžeme využít i jiných metod jak tyto zařízení přidat do inteligentního systému. Některé z těchto metod jsou uvedeny dále v této práci.

Použití Open Source produktů a aplikací

Tímto máme na mysli použití produktů a aplikací, které jsou vyvíjeny formou komunitních projektů a jsou volně dostupné k používání. Obrovskou výhodou těchto komunitních projektů je jejich uživatelská základna, ve které můžeme s uživateli komunikovat, diskutovat naše problémy či chlubit se ve formě návodů našimi úspěchy. Pokud jsme i technicky založení a můžeme danému projektu něco nového přinést, máme možnost participace na vývoji a dále systém rozvíjet.

Cena

Použitím Open source projektů a vlastních modulárních řešení automatizace jsme nejen schopni upravovat si systém přesně podle našich potřeb, ale také na inteligentní domácnosti oproti komerčním produktům značně ušetřit. K tomu aby si ale běžný uživatel mohl inteligentní systém sám postavit je nutná alespoň základní znalost elektrotechniky a programování. Většina vytvářených modulů je založena na vytváření logických plošných spojů s mnoha elektronickými komponentami. Jakmile je modul správně sestaven, je třeba ve většině případů naprogramovat logiku celého modulu a upravit inteligentní systém tak, aby nově vytvořený modul přijal.

Postavit svou vlastní inteligentní domácnost je běh na dlouhou trať a vyžaduje aby byl člověk technicky zdatný.

2.3 Porovnání ceny jednotlivých řešení

Pokud bychom chtěli porovnat řešení inteligentní domácnosti navrhované v této práci s komerčními produkty, museli bychom určitým způsobem monetizovat náš čas strávený při návrhu a následné implementaci celého inteligentního systému. Jednoduchou metodou ale můžeme porovnat cenu samotného centrálního serveru, který je srdcem každé inteligentní domácnosti. Porovnání cen řídicích jednotek jednotlivých komerčních řešení a námi navrhovaného v Tab. 2.3.

Tab. 2.3 Porovnání ceny řídicích jednotek

Produkt	Cena bez DPH [Kč]
Loxone	8263
Fibaro	13365
Control4	14810 až 59240
Navrhované řešení	1156

3 BEZDRÁTOVÉ TECHNOLOGIE

Mezi bezdrátové technologie řadíme takové formy komunikace, které umožňují přenášení dat mezi jednotlivými účastníky komunikace bez použití kabelového média. Jedná se o přenášení dat formou elektromagnetických změn prostředí, tedy šíření pomocí analogových signálů. Abychom mohli přenášet data tímto prostředím, je třeba digitální data zakódovat pomocí amplitudové, frekvenční nebo fázové modulace. Samotná implementace přenosu dat se nachází na fyzické vrstvě ISO/OSI modelu.

Bezdrátové sítě se dále rozdělují do následujících kategorií:

- **WPAN**[11] (Wireless PAN) neboli bezdrátová osobní síť propojuje malý počet zařízení v malé oblasti. Jedná se například o bezdrátové propojení mobilních zařízení pomocí technologie bluetooth.
- **WLAN**[12] (Wireless LAN) je označení pro místní bezdrátovou síť, která propojuje dvě a více zařízení za účelem přenosu dat či přístupu do intranetu nebo internetu. Obvykle se jednotlivá zařízení připojují přes jeden nebo více přístupových bodů. Tato technologie je více popsána v další kapitole.
- **WWAN**[13] (Wireless WAN) je typ rozlehlá bezdrátové sítě propojující větší oblasti o velikosti vesnic a měst. Bezdrátové připojení je obvykle zprostředkováno pomocí mikrovlnné linky point-to-point na frekvenci 2,4 GHz.
- **WMAN**[14] (Wireless MAN) označuje bezdrátovou metropolitní síť, která spojuje několik desítek až stovek měst.
- **Mobilní síť** jedná se o bezdrátové technologie pro mobilní zařízení. Dříve pouze technologie GSM[15], v posledních letech přibývá pokrytí i pro novou technologii LTE[16] zaměřenou hlavně na přenos dat.

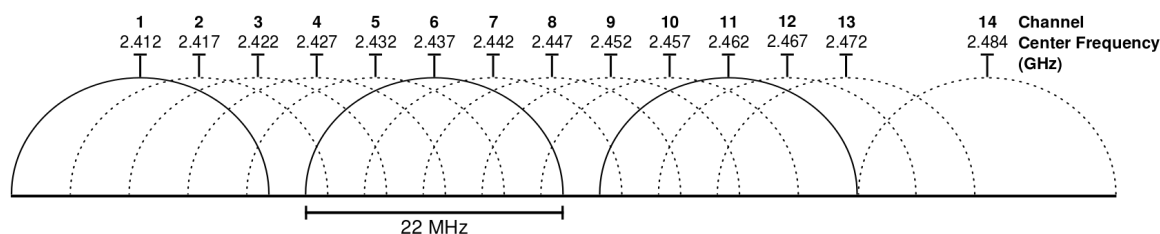
Bezdrátové sítě se musí dělit o frekvence s dalšími bezdrátovými technologiemi jako jsou televizní a rádiové vysílání a proto je v porovnání s kabelovými systémy šířka pásma velmi malá. Český telekomunikační úřad (ČTU) národní instituce dohlížející a spravující bezdrátové sítě. Pro bezdrátové datové sítě je vymezeno pásmo 2,4 GHz (2400 - 2483,5 MHz). Dalšími povolenými pásmy jsou:

- 27 MHz, toto pásmo je volně přístupné a je určeno pro rádiově řízené modely
- 433 MHz
- 868 MHz

Dále se v této práci budeme zaměřovat na bezdrátovou technologii WLAN (neboli WiFi) a směřovat náš vývoj do tzv. internetu věcí (popsán v další kapitole). V některých případech se můžeme při vytváření inteligentní domácnosti setkat také s termínem bezdrátových senzorických sítí (Wireless Sensor Network). Tyto technologie v době minulé jedinou možností jak komunikovat s fyzickými senzory. S novými moderními technologiemi umožňující jednoduché a energeticky přijatelné připojení senzorů k síti WiFi je tato technologie vhodná pouze na propojení drobných senzorů, u kterých je vyžadován extrémně nízký energetický provoz.

3.1 WiFi

WiFi neboli Wireless Lan (WLAN) je označení pro sadu standardů IEEE 802.11¹⁾ popisujících bezdrátovou komunikaci v počítačových sítích. Standard IEEE 802.11 byl publikován mezinárodním standardizačním institutem v roce 1997. Tato technologie využívá tak zvaného „bezlicenčního frekvenčního pásma“, proto je ideální pro budování levné, ale výkonné sítě bez nutnosti pokládky kabelů[12]. Ve standardu 802.11 je zahrnuto celkem šest druhů modulací pro zasílání rádiového signálu používající stejný protokol. Nejpoužívanější modulace jsou definované v dodatcích s písmeny a[20], b[21], g[22]. Dodatek 802.11n přidává další techniku modulace. Standard 802.11 se stejně jako pásma rozhlasového a televizního vysílání dělí do 13 kanálů vzájemně posunutých o 5 MHz a speciálního 14. kanálu posunutého o 12 MHz od 13. kanálu (viz Obr. 3.1).



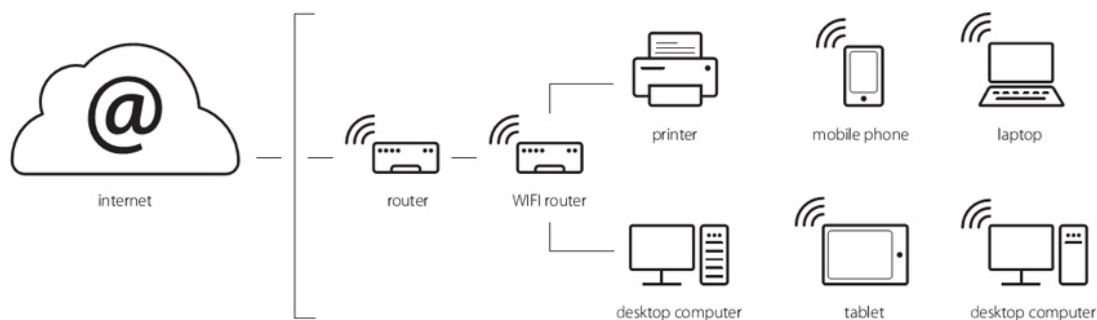
Obr. 3.1 Grafická reprezentace Wi-Fi kanálů v pásmu 2.4 GHz[17]

Základem WiFi sítě jsou dvě koncová zařízení, která spolu chtějí bezdrátově komunikovat. Spojí se spolu spojí buď přímo nebo za pomoci přístupových bodů zvaných AP (Access Point). Přístupový bod umožňuje propojení bezdrátových stanic na různých sítích - jak jiná WiFi nebo na síti LAN.

3.1.1 Vrstvy síťového modelu

Bezdrátové sítě jsou podobně jako ostatní síťové protokoly a zařízení založené na referenčním modelu ISO/OSI[19]. Výše zmiňované standardy 802.11 jsou definovány na

¹⁾IEEE 802.11 je standard pro Wi-Fi s dalšími doplňky pro lokální bezdrátové sítě (Wireless LAN, WLAN) vyvíjený 11. pracovní skupinou IEEE LAN/MAN standardizační komise (IEEE 802).[17]



Obr. 3.2 Schéma komunikace na WiFi[18]

první a druhé vrstvě tohoto modelu (viz. Obr. 3.3).

Spojová vrstva	LLC					
	IEEE 802.11 MAC					
Fyzická vrstva	IEEE 802.11 IR	IEEE 802.11 DSSS	IEEE 802.11 FHSS	IEEE 802.11a OFDM	IEEE 802.11b HR-DSSS	IEEE 802.11g OFDM

Obr. 3.3 Referenční model ISO/OSI pro Wi-Fi[19]

Fyzická vrstva

je nejnižší položenou vrstvou ISO/OSI modelu, která realizuje vysílání a příjem dat bezdrátovým prostředím[26]. Přenosové mechanismy pro standard 802.11 společně s přenosovými rychlostmi jsou uvedeny v tab. 3.1.

Tab. 3.1 Přenosové mechanismy pro standard 802.11[19]

Standard	Frekvence	Podporovaná přenosová rychlost [Mbps]
802.11	2,4 GHz; DFIR	1; 2
802.11b	2,4 GHz	1; 2; 5,5; 11
802.11g	2,4 GHz	6; 9; 12; 18; 24; 36; 48; 54
802.11a/h	5,2 - 5,5 GHz	6; 9; 12; 18; 24; 36; 48; 54

Ve všech standardech 802.11 je fyzická vrstva rozdělena na následující dvě podvrstvy[27]

- **PLCP (Physical Layer Convergence Procedure)** - v této podvrstvě se k datovým rámcům MAC (Medium Access Control) podvrstvy přikládají informace o použitém přenosovém mechanismu a modulaci. Výsledkem je datový rámeček,

který je nezávislý na typu fyzické vrstvy. Do této podvrstvy je implementována rovněž funkce CCA (Clear Channel Assessment), která poskytuje odezvu pro MAC vrstvu o připravenosti přenosového média.

- **PMD (Physical Medium Dependent)** - tato podvrstva odpovídá za přenos dat mezi jednotlivými vysílači a přijímači. Z podvrstvy PLCP jsou data v závislosti na použitém přenosovém mechanismu ve vysílači vysílána do bezdrátového prostředí, kde jsou na straně přijímače pomocí PMD přijímána a předávána podvrstvě PLCP.

Spojová vrstva

realizuje kódování a přenos informací. Mezi její hlavní operace patří umožňování vyšším vrstvám přistupovat k přenosovému médium pomocí rámců (paketů) a řízení vysílání a příjmu dat s detekcí chyb[25]. Je podobně jako fyzická vrstva rozdělena na dvě podvrstvy:

- LLC (logica Link Control) - řídí komunikaci se síťovou vrstvou a jejími protokoly (např. IP, IPX)
- MAC (Media Access Control) - provádí adresaci, příjem a vysílání signálů v závislosti na přenosovém médium

Mezi základní vlastnosti podvrstvy MAC patří:

- **Potvrzování zpráv** - používají se tzv. ACK (acknowledgement) a NAK (negative acknowledgment) zprávy zasílané za účelem zjištění připravenosti/nepřipravenosti příjemce obdržet danou informaci nebo pro potvrzení obdržení zprávy ve správném/chybném stavu
- **Fragmentace paketů** - z důvodu nebezpečného přenášení dat bezdrátovým médiem je praktičtější zasílat malé segmenty dat a v případě chybného přenosu opakovat přenos pouze vybraných segmentů než velkého celku. Velikost fragmentů je určována parametrem MTU (Maximal Transmission Unit). Nevýhodou velké fragmentace paketů je snížení rychlosti z důvodu navýšení ACK/NAK potvrzovacích zpráv.
- **RTS/CTS (Request To Send/Clear To Send)** - metoda je využívána v situaci, kdy jedna stanice chce přistupovat k médium na kterém už komunikuje jiná stanice. Aby nedocházelo ke kolizím, používají se řídicí pakety RTS a CTS, které předchází kolizi při komunikaci přístupových bodů a koncových stanic.

- **CRC** - cyklický kontrolní součet je využíván k detekování chyb během přenosu a k ukládání dat. Každý přenášený paket je opatřen kontrolním součtem CRC a pokud dorazí paket s odlišnou hodnotou CRC než jakou uvádí, je zničen a stanice si zažádá o znovu zaslání daného paketu.
- **Mód šetření energie (Power Saving)** - jedná se o metodu šetření energie, kdy stanice zašle přístupovému bodu zprávu a AP od té doby ukládá zprávy určené pro tuto stanici. Stanice se periodicky probouzí a pomocí beacon packetu se dozvídá zda jsou pro ni na AP uloženy data. Pokud jsou data připraveny, stanice je přijímá.
- **Skenování** - může být aktivní kdy stanice aktivně zasílá tzv. probe request a všechny přístupové body (AP), které tento paket obdrží zasílají zpátky probe response. V případě pasivního skenování zasílá AP tzv. beacon pakety s informacemi o daném AP. Tyto pakety stanice přijímá a rozhoduje se dle přijatých informací, které AP si vybere.
- **Autentizace** - ve standardu 802.11 se využívají dva typy autentizace: Open system authentication a Shared key authentication.
- **Asociace** - metoda synchronizace stanice a přístupového bodu. Stanice zasílá association request jenž obsahuje identifikátor SSID a podporovanou přenosovou rychlost. AP odpovídá pomocí association response a po úspěšné synchronizaci začne přenos dat.
- **WEP (Wired Equivalent Privacy)** - jedná se dnes již o zastaralé zabezpečení WiFi sítí, které lze snadno prolomit. Stanice zakóduje tělo každého paketu a příjemce tento paket pomocí svého klíče dekoduje.

Struktura MAC hlavičky u WiFi je zobrazena na obr. 3.4. Jak můžeme vidět, hlavička u bezdrátových sítí je výrazně složitější než u Ethernetových sítí.

3.1.2 Pakety

Ve standardu 802.11 jsou pakety rozděleny do následujících kategorií:

- **Řídící** pakety jsou spojené s přístupováním k médiu (MAC). Jde například o výše popsané pakety RTS a CTS, ACK a NAK.
- **Management** pakety slouží např. k odhlášení uživatele ze sítě nebo pakety typu Beacon.
- **Datové** pakety slouží pro přenášení samotných dat.

802.11 MAC header (WLAN)						
Frame Control	Duration ID	Address 1	Address 2	Address 3	Sequence Control	Address 4
2 Bytes	2 Bytes	6 Bytes	6 Bytes	6 Bytes	2 Bytes	6 Bytes

802.3 MAC header (Ethernet)		
Dest. Address	Source Address	Type or Length
6 Bytes	6 Bytes	2 Bytes

Obr. 3.4 Porovnání MAC hlaviček u WiFi a Ethernetu[23]

3.1.3 Topologie WiFi sítí

V bezdrátových sítích existují následující tři typy sítí[28]

- Ad hoc
- BSS (Basic Service Set)
- ESS (Extended Service Set)

Příklad propojení stanic typem Ad-Hoc, BSS a ESS je zobrazen na obr. 3.5

Ad-Hoc

je typ sítě, kde komunikují koncové stanice v režimu peer-to-peer. Tedy všechny stanice si jsou v síti rovny a není používán přístupový bod. Aby spolu stanice mohli komunikovat je nezbytné, aby stanice byly ve vzájemném dosahu. Výhodou Ad hoc spojení je jednoduchost ve vytvoření spojení mezi více stanicemi bez nutnosti přístupového bodu. Hlavní nevýhodou je omezená rychlost přenosu dat (11 Mbps) a nutnost zůstat se stanicí v dosahu ostatních stanic.

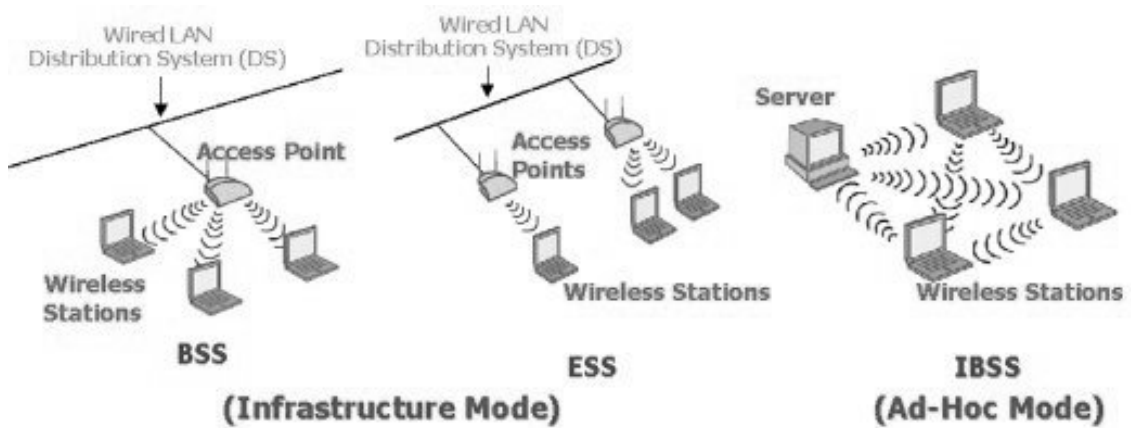
BSS (Basic Service Set)

je další typ bezdrátové sítě, kdy koncové stanice komunikují s jedním přístupovým bodem. K tomuto přístupovému bodu se stanice připojují po úspěšné asociaci a autorizaci. Jakákoli změna nastavení znamená změnu pro všechny koncové stanice. Stejně

tak pokud bude přístupový bod vytížen a dojde ke snížení propustnosti dat, bude komunikace všech stanic zpomalena a bude docházet k latencím.

ESS (Extended Service Set)

je posledním typem zapojení bezdrátové sítě a dochází k ní při sloučení dvou a více BSS bezdrátových sítí. Důležitým parametrem je dostupnost všech přístupových bodů a společného pojmenování sítě.



Obr. 3.5 Porovnání Ad-hoc, BSS a ESS[24]

moduly.

3.3 MQTT

MQTT (MQ Telemetry Transport) je tzv. M2M²⁾ komunikační protokol pro Internet věcí postavený nad TCP/IP[31]. Jedná se o přenos malých senzorických dat mezi senzory a infrastrukturou, která tyto data zpracovává a případně také publikuje. Tento protokol vytvořili v roce 1999 Andy Stanford-Clark (IBM) a Arlen Nipper (Eurotech; nyní Cirrus Link) jako jednoduchý protokol pro monitorování ropovodu v poušti. Základem bylo mít protokol, který bude přenášet velmi malá data a bude energeticky velmi nenáročný - senzory bylo třeba propojit se satelitem a v této době to bylo velice nákladné.

3.3.1 Broker

Protokol používá tzv. publish/subscribe (zapiš/odebírej) metodu, která umožňuje na jedné straně zapisování dat na server a z druhé strany přijímání dat klientem[31]. Tomuto serveru se říká MQTT Broker a má na starost pasivní sběr dat od senzorů, které data nahrávají do tzv. Topiců, a zasílání aktivních notifikací klientům, kteří se k těmto Topicům přihlásí.

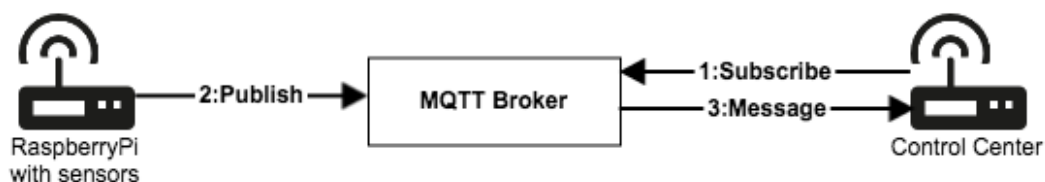
Zde je vidět jasný rozdíl od protokolu HTTP, který pracuje metodou request/response. Tedy klient si vyžádá data a server mu je poskytne. MQTT broker poskytuje data automaticky pokud se klient jednou k vybranému Topicu přihlásí.

Abychom mohli protokol MQTT používat, je nutné senzor připojit do sítě WiFi, popřípadě data získaná ze senzorů pomocí bezdrátové senzorické sítě přenést na centrální server připojený do sítě internet. Tento připojený senzor/server kontaktuje server v internetu zvaný MQTT Broker a pomocí definovaného protokolu se zaregistruje. Následně senzor po naměření senzorických dat odesílá naměřená data svému Broker serveru. K tomuto Broker serveru můžeme také zaregistrovat čtecí zařízení (mobilní telefon, serverovou aplikaci či libovolného klienta), které bude od Broker serveru přijímat vybraná naměřená data. Postup komunikace mezi senzorem, MQTT brokerem a klientem je zobrazeno na obr. 6.1

3.3.2 Topic

Topicem rozumíme jednoduchý řetězec znaků. Tento řetězec může být hierarchicky rozdělen pomocí dopředného lomítka. Například pro data z teplotního senzoru bude vytvořený topic **domácnost/obývací-pokoj/teplota**. Pokud se klient zaregistruje pro tento topic, budou mu chodit informace o teplotě získané ze senzoru v obývacím

²⁾M2M (machine-to-machine) - stroj-stroj komunikace



Obr. 3.7 Komunikace senzoru a klienta pomocí MQTT[31]

pokoji. Pokud v naší domácnosti budeme mít více teplotních senzorů v různých částech domu, např. **domácnost/kuchyně/teplota** a budeme chtít získávat data od obou, je možné při registraci k topicům používat tzv. wildcards, tedy speciální znaky s danou funkcionalitou. V tomto případě použijeme znak **+** abychom se zaregistrovali k topicu s názvem **domácnost/+/teplota** a získávali bychom záznamy teplot ze všech topiců, které 3 úrovně hierarchie s 1. úrovní „domácnost“ a 3. úrovní *teplota*. Dalším speciálním znakem je **#**, který nám dovoluje zaregistrovat všechny úrovně pod námi vybranou úrovní. Například **domácnost/#** nám zaregistruje všechny topiky, které začínají s textem „domácnost“.

4 VZDÁLENÝ PŘÍSTUP A ZABEZPEČENÍ INTELIGENTNÍHO SYSTÉMU

Inteligentní domácnost je možné nejen ovládat z domu samotného či jeho blízkého okolí, ale také z jakéhokoli místa na světě odkud je možné se k systému připojit. Přistupovat vzdáleně k systému můžeme například pomocí mobilních technologií (GSM sítě) nebo pomocí internetu. Nejjednodušším způsobem pro komunikaci domácnosti a uživatele na dálku je použitím SMS či MMS. V případě určité události může systém zaslat uživateli SMS zprávu nebo MMS zprávu s fotkou, když kamerový systém zaznamená pohyb. Stejně tak jako systém zasílá zprávy uživateli může uživatel zasílat SMS zprávy systému. V tomto případě se bude jednat o nastavovací zprávy měnící určitým způsobem chování systému. Tyto zprávy budou mít zpravidla přesně definovanou strukturu zpráv, ale v případě odcizení mobilního zařízení může být systém narušen. Ovládání domácnosti pomocí SMS zpráv také není velmi praktické a navíc je zpoplatněné. Je dobré tedy využívat SMS zprávy pouze jako notifikace události nebo v případech, kdy jiná komunikace není možná.

Dalším způsobem ovládání domácnosti je pomocí webového rozhraní z libovolného zařízení připojeného do sítě internet. V tomto případě jsme přímo napojeni na centrální server a jsme schopni celou domácnost ovládat. Opět i v tomto případě je nutné myslet na bezpečnost. Prostředí internetu je daleko nebezpečnější než mobilní síť GSM a navíc přes webové rozhraní je daleko snadnější přístup k systému včetně prohlížení bezpečnostních kamer. Proto je velmi důležité aby veškerá komunikace mezi webovým rozhraním a uživatelem byla šifrována a samotný přihlašovací proces k systému nebyl postaven jen na znalosti přihlašovacích údajů ale například na použití jednorázových hesel a generátorů s využitím SMS zpráv.

4.1 Použití mobilní sítě

Pro ovládání domácího inteligentního systému pomocí mobilní sítě GSM je nutné připojit k systému mobilní telefon se SIM kartou. Nejjednoduššími telefony pro automatizace jsou s operačním systémem Android. Tyto telefony je díky otevřené architektuře možné ovládat pomocí ADB (Android Debug Bridge)[32] aplikace připojit k libovolnému operačnímu systému a mobilní zařízení pomocí jednoduchých příkazů ovládat.

Abychom byli schopni odesílat SMS pomocí ADB příkazů, není třeba mobilní zařízení tzv. rootovat. Pokud bychom chtěli i zprávy z mobilního telefonu vyčítat, je nutné mít povolený root abychom mohli přečíst obsah SMS databáze v telefonu.

Tímto způsobem můžeme inteligentní systém nastavit pro pravidelné vyčítání SMS zpráv a v případě obdržení zprávy, za předpokladu že bude mít správnou strukturu, zprávu přečíst a vykonat určitý příkaz. Opačně jsme schopni i na základě definované

události zaznamenané centrálním systémem odeslat SMS zprávu uživateli.

II. PRAKTICKÁ ČÁST

5 POŽADAVKY NA ŘEŠENÍ

5.1 Funkční požadavky

Na inteligentní systém řízení domácnosti jsou v této práci kladeny následující požadavky.

1. Vzdálená správa systému

- k systému je možné přistupovat z libovolného počítače a mobilního zařízení v domácnosti
- přístup mimo domácnost je povolen pouze se znalostí přístupových údajů
- systém umožňuje konfigurovat a monitorovat všechna připojená zařízení
- možnost vytváření pravidel pro automatické ovládání systému (časová i událostmi řízená)
- systém umožňuje zasílání automatických notifikací při změně stavu jednotlivých zařízení (GSM, email a mobilní aplikace)

2. Měření teploty

- měření teploty v libovolné místnosti i vně domácnosti

3. Měření vlhkosti prostředí

- měření vlhkosti v libovolné místnosti i vně domácnosti
- měření vlhkosti půdy pokojových rostlin

4. Detekce pohybu

- ve vybraných místnostech je detekován pohyb osob a zvířat

5. Ovládání světel

- rozsvícení a zhasnutí světel v libovolné místnosti bez přítomnosti uživatele v dané místnosti
- regulace intenzity osvětlení v libovolné místnosti

6. Regulace žaluzií

- plné ovládání vnitřních žaluzií (plné i částečné vytahování a zatahování)

7. Kamerový systém

- monitorování libovolného místa v domácnosti s možností streamování a záznamu

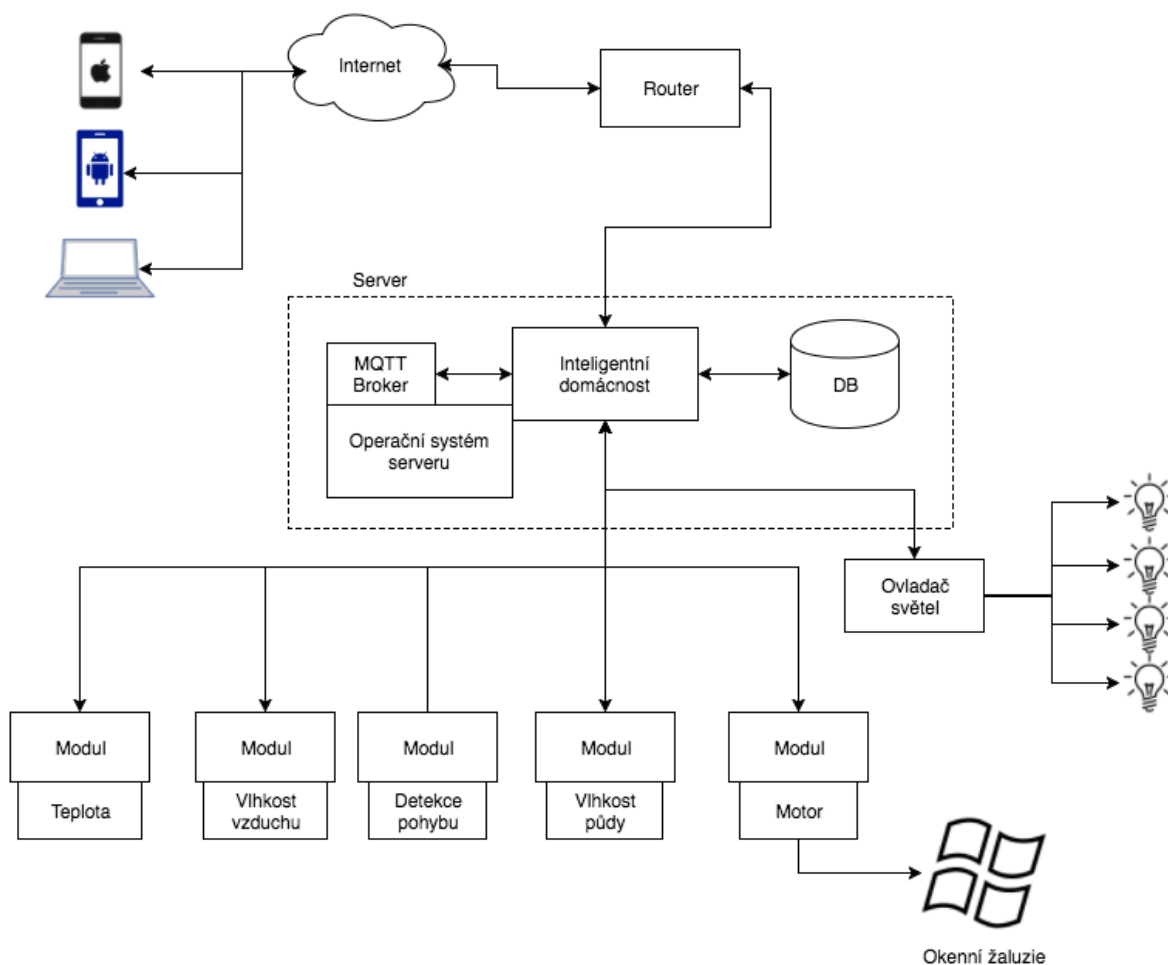
8. Ovládání u hlavních dveří

- možnost ovládání celého systému z fixního místa
- „odchodové tlačítko“ pro přepnutí celého systému do režimu opuštěná domácnost

5.2 Nefunkční požadavky

1. K systému je možné přidat libovolné další zařízení komunikující na podporovaných komunikačních protokolech (REST, MQTT).
2. Doba odezvy na příkaz odeslaný do systému nesmí trvat déle než 5 sec (od zpracování žádosti do požadované akce systému).
3. Systém nesmí zasílat false-positive a false-negative notifikace.
4. V případě havárie systému musí systém umožňovat vzdálenou možnost obnovy bez ztráty měřených dat a konfigurace připojených zařízení.

6 NÁVRH ARCHITEKTURY INTELIGENTNÍ DOMÁCNOSTI



Obr. 6.1 Návrh architektury inteligentní domácnosti

Dle funkčních a nefunkčních požadavků je navržena architektura. Inteligentní domácnost, MQTT Broker a Databáze jsou provozovány na serveru. K inteligentnímu systému se připojují jednotlivé moduly, na kterých jsou připojené senzory pro měření teploty, vlhkosti vzduchu, detekce pohybu, vlhkost půdy a modul pro ovládání okenních žaluzií. Dále je k systému připojen systém pro ovládání domácích světel.

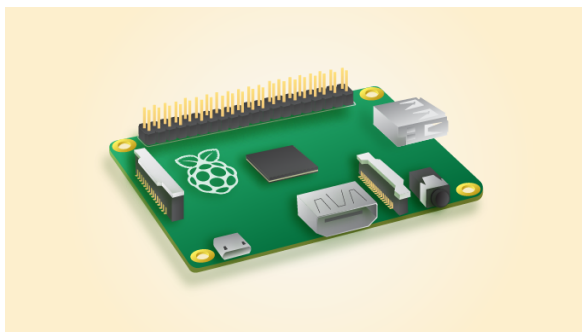
7 MODULY PRO BEZDRÁTOVOU KOMUNIKACI

7.1 Raspberry Pi

Raspberry Pi je plnohodnotný, miniaturní, jednodeskový počítač vyvinutý britskou nadací Raspberry Pi Foundation s cílem podpořit výuku informatiky na středních školách [33]. Konfiguraci Raspberry Pi není možné měnit, všechny moduly jsou naletovány na jednom plošném spoji. Jako operační systém je používán GNU/LINUX a je možné si vybrat z několika distribucí (např. Debian, Ubuntu a další). Systém Windows momentálně toto zařízení nepodporuje.

7.1.1 Raspberry Pi - Model A

První verze Raspberry pi (Model A) se objevila na trhu již v únoru 2012 a byl ve své době revolučním mini počítačem s velmi nízkou cenou (přibližně 900 Kč). Model A byl osazený SoC Broadcom BCM2835 integrující procesor ARM11 s frekvencí 700 MHz a grafickým procesorem umožňujícím dekódování FullHD videa v reálném čase. Na desce byla také dostupná 256 MB operační paměť (sdílená s grafickou kartou), výstup HDMI a RCA pro připojení externích monitorů, USB 2.0 port, audio konektor Jack 3,5 mm a slot pro připojení SD/MMC karty.



Obr. 7.1 Raspberry Pi (Model A)[34]

7.1.2 Raspberry Pi - Model B a B+

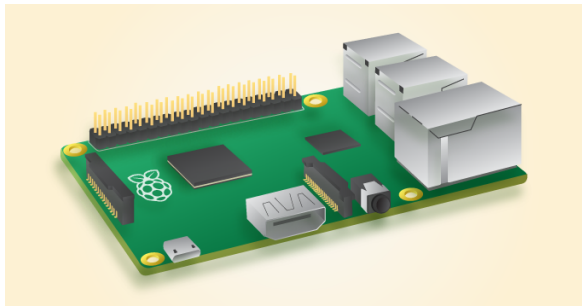
V dubnu téhož roku vychází Model B, který dostává dva USB 2.0 porty a ethernetový adaptér 10/100 s konektorem RJ45 a dočká se zvýšení operační paměti na 512 MB.

V červenci 2014 se objevuje vylepšený model B+, který obsahuje oproti modelu B další 2 USB 2.0 porty (celkem 4) a 1 GB operační paměti.

7.1.3 Raspberry Pi 2

V únoru 2015 vychází nový model Raspberry Pi 2, který je osazený novým SoC Broadcom BCM2836 a obsahuje čtveřici procesorových jader s taktem 900 MHz, posílenou

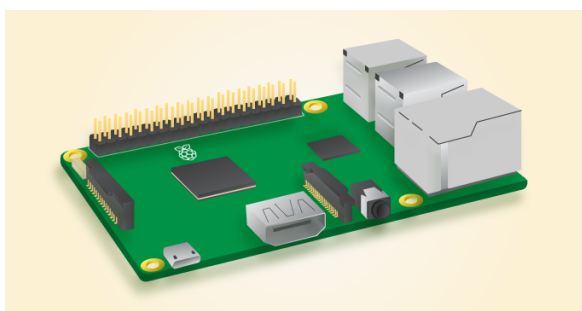
jednotkou SIMD 1GB operační paměti. Osazen je grafický procesor VideoCore IV, podporující OpenGL ES 2.0, 1080p30, MPEG-4[33]. Tento model je v této práci používán jako centrální řídicí jednotka inteligentního domácího systému.



Obr. 7.2 Raspberry Pi 2[34]

7.1.4 Raspberry Pi 3

Poslední dostupnou verzí je Raspberry Pi 3, které se na trhu objevuje v únoru 2016. Tento model je vybaven 64 bitovým 4-jádrovým CPU ARM Cortex-453 o frekvenci 1,2 GHz. Novinkou je integrovaný WiFi a Bluetooth modul.



Obr. 7.3 Raspberry Pi 3[34]

7.1.5 GPIO

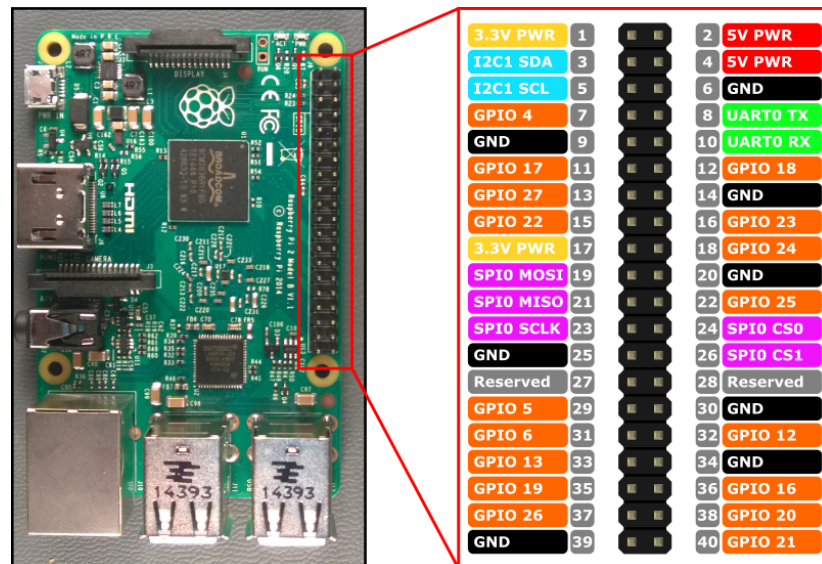
Vstupně výstupní piny (General Purpose Input Output - GPIO) slouží k připojení jakýchkoli zařízení, které pracují s logickými signály (dvoustavovými logická 0 a 1). Raspberry Pi 2 na své desce obsahuje následující vstupně výstupní piny:

1. 17 vstupně výstupních pinů
2. SPI bus pin (Serial Peripheral Interface)
3. I2C bus (Inter-Integrated Circuit)

Kromě vstupně výstupních pinů nalezneme i následující piny určené pro napájení připojených zařízení:

1. dva 5 V napájecí piny
2. dva 3,3 V napájecí piny
3. osm pinů pro uzemnění

Rozložení a umístění jednotlivých pinů je zobrazen na (Obr. 7.4).



Obr. 7.4 GPIO [35]

Každý z vstupně výstupních pinů může být nastaven v těchto módech:

1. vstupní pin, který umožňuje detekovat, zda je k němu připojena
 - logická 0 (napětí v rozsahu 0 - 0,8 V)
 - logická 1 (napětí v rozsahu 2 - 3,3 V)
2. výstupní pin, který umožňuje nastavit svou hodnotu na
 - logickou 0 (napětí v rozsahu 0 - 0,4 V)
 - logickou 1 (napětí v rozsahu 2,4 - 3,3 V)

Následující příklad ukazuje otevření pinu číslo 5 a zapsání logické jedničky na výstup (kód je psaný v jazyce Python).

```
1 import RPi.GPIO as GPIO
2
3 // inicializace GPIO controlleru
4 GPIO.setmode(GPIO.BOARD)
5
6 // nastavime GPIO 5 do vystupniho modu
7 GPIO.setup(5, GPIO.OUT)
8
9 // nastavime HIGH hodnotu pro pin (logicka 1), opak LOW (logicka 0)
10 GPIO.output(5, GPIO.HIGH)
11
12 // vycistení GPIO do defaultního stavu
13 GPIO.cleanup()
```

7.2 Arduino

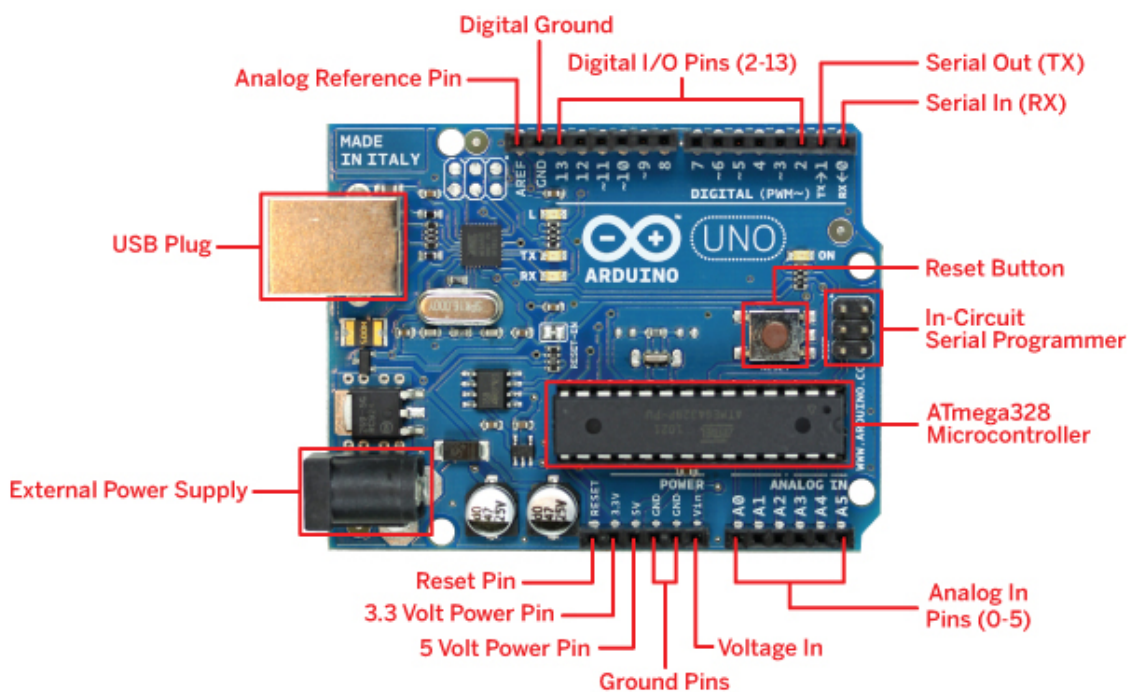
Arduino je otevřená platforma s grafickým vývojovým prostředím, které vychází z prostředí Wiring (podobný projekt jako Arduino, tedy deska s mikrokontrolerem a IDE)[36]. Vývojové desky obsahují 8 bitové mikrokontroléry od společnosti Atmel a existuje jich poměrně velký počet - např. ATmega8, ATmega168, ATmega328, ATmega1280 a ATmega2560.

Oproti Raspberry Pi není Arduino koncipováno jako stolní počítač a neobsahuje tedy žádný operační systém. Obslužné programy je třeba vytvořit na jiném zařízení a do Arduina jej posléze nahrát. Výhodou tohoto řešení je nižší spotřeba a oproti Raspberry Pi je možné tyto moduly napájet pomocí baterií.

7.2.1 Arduino Uno

Jedná se o základní verzi Arduina, která vyšla v roce 2010. Tato verze poskytuje 14 vstupně výstupních digitálních pinů a 6 analogových pinů. Modul je vybaven procesní jednotkou ATmega328P s frekvencí 16 MHz, 2 KB SRAM, 32 KB flash paměti, rozměry 68,6 x 53,4 mm a hmotností 25 g. Pro svůj chod potřebuje Uno stabilní napětí 5 V.

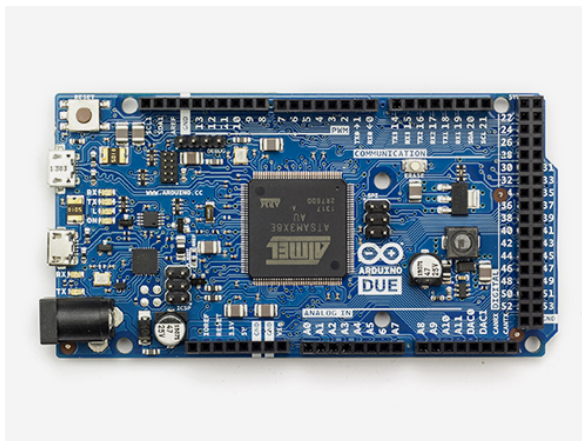
Rozložení a umístění jednotlivých komponent a pinů je zobrazeno na (Obr. 7.5).



Obr. 7.5 Arduino Uno [37]

7.2.2 Arduino Due

Další verze Arduina vydaná v roce 2012 obsahuje řadu vylepšení oproti svému předchůdci. Nový procesor AT91SAM3X8E taktuje na 84 MHz dále nový modul obsahuje 96 KB SRAM, 512 KB flash paměť, snížil se počet digitálních I/O pinů na 12 a 2 poze analogové, rozměry 101,52 x 53,3 mm a hmotnost 36 g. Modul nově potřebuje stále napětí 3,3 V.



Obr. 7.6 Arduino Due [38]

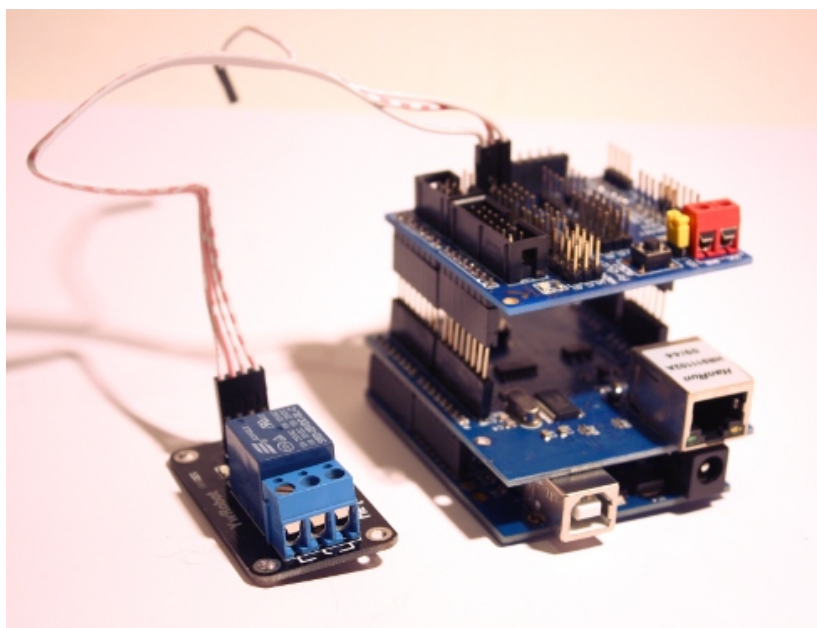
7.2.3 Arduino Shields

Hlavním cílem platformy Arduino je možnost jednoduše připojovat další moduly, tzv. Shiedly, které jsou připojovány přes standardizované patice. Spojením více modulů do jednoho vytvoříme například modul přístupný po LAN/WiFi, který bude zapínat či vypínat žárovku v akváriu.

Pro vytvoření takového modulu potřebujeme:

1. Arduino Uno - základní procesní jednotka s běžícím web serverem
2. Ethernet shield - deska opatřená ethernetovou RJ45 přípojkou
3. Sensor shield - deska pro ovládání připojených modulů (v tomto případě Relay modul)
4. Relay module - modul pro zapínání/vypínání elektrického proudu připojeného zařízení (napájení žárovky v akváriu)

Spojení všech shieldů je zobrazeno na (Obr. 7.7).



Obr. 7.7 Arduino Shields [39]

7.2.4 Zhodnocení

Modulárnost Arduina je jednou z jeho velkých výhod. Spojením jednotlivých shieldů dohromady vznikají komplexní moduly schopné řídit či monitorovat libovolné prvky domácnosti. Další výhodou je otevřená a rozšířená komunita s velkým množstvím návodů a naprogramovaných řešení.

Na druhou stranu za velikou nevýhodu řadím bohužel vyšší cenu. Abychom si postavili modul schopný komunikovat na síti, je třeba spojit základní jednotku arduino due (cca 280 Kč) a WiFi shield (cca 120 Kč). Dohromady tedy cca 400 Kč. Ve srovnání s jinými moduly - např. NodeMCU (které zastane tu samou funkci jako spojní obou modulu) a cenou 90 Kč ukazuje na výrazný cenový rozdíl v případě, že bychom chtěli mít v každé místnosti inteligentní domácnosti jeden bezdrátový modul snímající teplotu.

7.3 NodeMCU

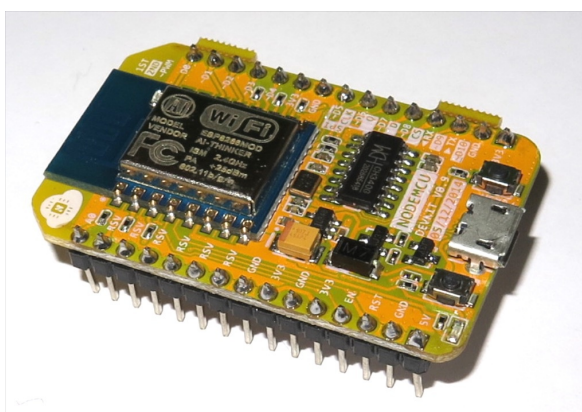
NodeMCU je otevřená IoT platforma, která se vyvinula z projektu ESP8266¹⁾ vložením procesní jednotky na plošný spoj společně s micro USB portem (pro napájení i komunikaci) a 10 I/O piny (6 digitálních, 3 analogové a 1 ADC). NodeMCU umožňuje nahrát vlastní upravené SDK²⁾ a které upravuje chování celého modulu.

Nejpoužívanějšími firmwary pro NodeMCU postavených na SDK ESP8266 NONOS SDK 1.4.0 vyvinuté společností Espressif jsou:

1. Nodemcu-firmware (skriptovací jazyk Lua)
2. Arduino IDE (programovací jazyk C++)

7.3.1 NodeMCU v1

První generace využívá procesní jednotku ESP-12, obsahuje 20 KB SRAM a 4 MB flash paměť. Integrovaný wifi modul podporuje 802.11b/g/n, devkit nodemcu v0.9.



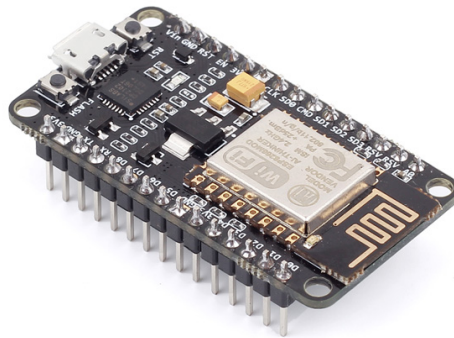
Obr. 7.8 NodeMCU v1 [40]

7.3.2 NodeMCU v2

Druhá generace používá nový procesor ESP-12E a devkit nodemcu v1.0. Nová verze zaznamenala také zúžení plošného spoje se zachováním stejného počtu I/O pinů.

¹⁾ESP8266 je nízko nákladový miniaturní procesor. V srpnu roku 2014 vznikl na základě tohoto chipu ESP-01 modul, který obsahuje integrovanou WiFi anténu a umožňuje tak jednoduchou a levnou možnost komunikace pro moduly.

²⁾Software development kit (SDK nebo devkit) je typická sada vývojových nástrojů umožňující vytváření aplikací pro jisté softwarové balíčky, frameworky, platformy, počítačové systémy, herní konzole, operační systémy nebo podobnou platformu[43].



Obr. 7.9 NodeMCU v2 [41]

7.3.3 GPIO

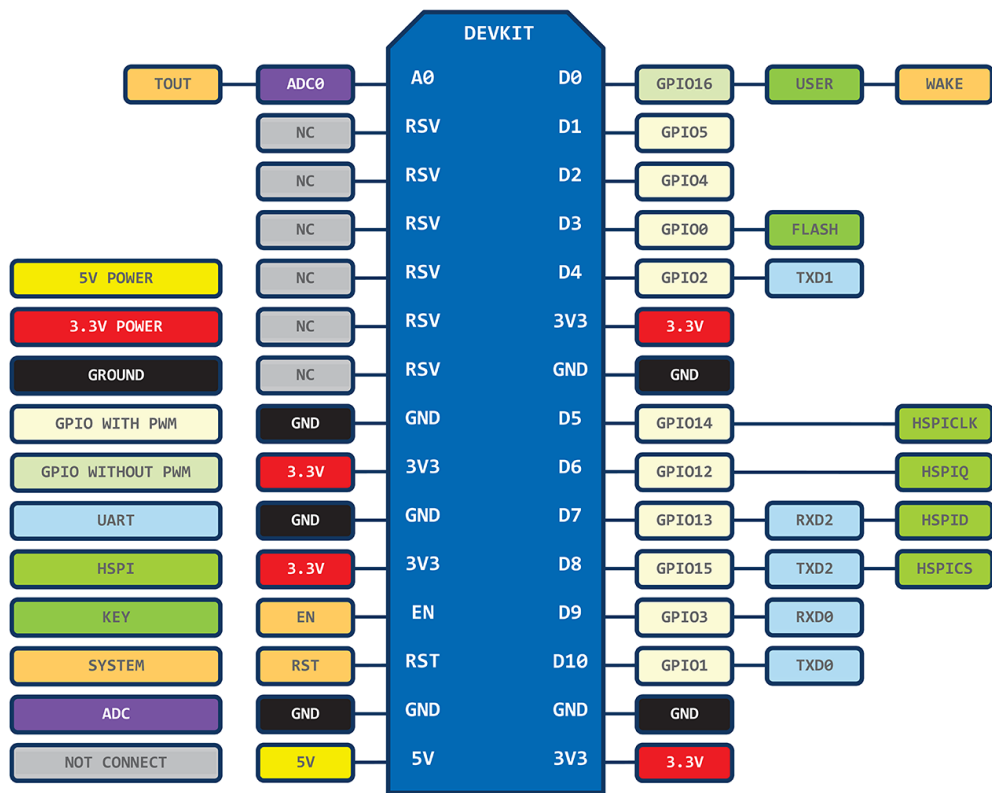
Podobně jako u Raspberry Pi nebo Arduina obsahuje i NodeMCU I/O piny pro komunikaci s dalšími zařízeními. Rozdělení pinů je následující:

1. 6 digitálních
2. 3 analogové
3. 1 ADC
4. 4 3,3 V piny
5. 1 5 V pin
6. 4 uzemňovací piny

Následující příklad ukazuje otevření pinu číslo 1 a zapsání logické jedničky na výstup (kód je psaný v jazyce LUA). Kód je velmi podobný přístupu k pinům i Arduina i Raspberry Pi.

```
1 pin = 1
2 gpio.mode(pin,gpio.OUTPUT)
3 gpio.write(pin,gpio.HIGH)
```

Kompletní dokumentace všech LUA funkcí je dostupná na adrese:
<http://nodemcu.readthedocs.io/en/dev/>



8 VYTVOŘENÍ BEZDRÁTOVÝCH MODULŮ

8.1 Implementace obslužných skriptů pro NodeMCU

Při vytváření a programování jednotlivých modulů se v mnoha případech opakují následující kroky:

1. aktivace modulu ze spánkového režimu
2. připojení modulu k WiFi
3. vykonání akce specifického modulu
4. komunikace s centrálním uzlem
5. uspání modulu na specifikovaný čas

Pro jednotlivé kroky je třeba implementovat obslužný LUA skript, který danou funkcionalitu vykonává. V této sekci je popsána implementace těchto základních kroků, které jsou používány téměř ve všech vytvářených modulech.

8.1.1 Inicializace modulu

Inicializace modulu probíhá nastartováním nahraného firmwaru (používáme základní s jazykem LUA) a vyhledáním inicializačního skriptu „init.lua“. Pokud je tento skript dostupný, je automaticky spuštěn. V případě automatizovaného spouštění tento skript odkazuje v našem případě na jiný skript „wifi.lua“, který připojí modul k lokální přednastavené WiFi. V dalším kroku je odkaz na skript, který je specifický pro funkci modulu „specific.lua“ a následné uspání modulu skriptem „sleep.lua“.

```
1  -- pripoji se k preddefinovane wifi
2  dofile("wifi.lua")
3
4  -- spusti specifickou akci modulu
5  dofile ("specific.lua")
6
7  -- uspi modul na definovany cas
8  dofile ("sleep.lua")
```

8.1.2 Připojení modulu k lokální WiFi

Modul se opakovaně snaží připojit k nadefinované WiFi v souboru „wifi.config“, ze kterého vyčte SSID a heslo. Pokud se mu nepodaří připojit k WiFi nebo nedostane IP adresu od DHCP serveru, zopakuje pokus o připojení po 10 sec intervalech. Pokud se mu ani po těchto pokusech nepodaří připojit, je modul uspán na 10 minut. Po probuzení modulu se opět spouští skript „init.lua“.

Skript „wifi.lua“ (viz příloha „Naprogramované aplikace a skripty“, Code 1.1/strana 119) je postavený na existenci jedné WiFi v domácnosti. Pokud bychom v jedné domácnosti měli větší počet WiFi sítí, například z důvodu neprostopnosti WiFi sítě přes zdi nebo velká vzdálenost mezi WiFi routerem a bezdrátovým modulem, je možné skript jednoduše rozšířit o více konfiguračních souborů (wifi1.config ... wifiN.config) a následné rotování mezi jednotlivými soubory. Taktéž by bylo možné po úspěšném připojení označit jeden z konfiguračních souborů jako hlavní a dynamicky tak vybírat nejlepší WiFi a připojovat se vždy k bodu, který byl vybrán v předchozím běhu.

8.1.3 Komunikace s centrálním uzlem

Pro komunikaci s centrálním uzlem v případě notifikačních modulů používáme jednoduché HTTP rest api, kde specifikujeme ID modulu a data, které potřebujeme přenést a pomocí metody POST odešleme data centrálnímu serveru. Ten data zpracuje, odešle modulu odpověď HTTP 200 a dle implementovaného mechanismu uloží do databáze či spustí jinou předdefinovanou akci. Pokud centrální aplikace není schopná zprávu přijmout, odešle chybový kód HTTP 400 a uloží informace o neproběhlé transakci do logů.

Implementace skriptu je v tomto případě velmi jednoduchá. V jazyce LUA existují obslužné metody pro HTTP komunikaci, jediné co musíme nadefinovat je adresa příjemce, obsah restové zprávy a zkontrolovat přijetí zprávy.

Syntaxe pro přijetí dat z modulu je na centrálním serveru implementována následovně:

Tab. 8.1 Syntaxe REST api - POST data

Method	POST
URI	api/v1/data

Accepts

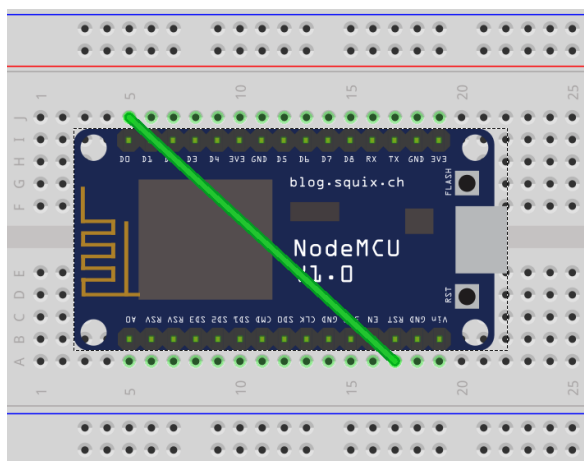
modulID	parametr s unikátním identifikátorem modulu
dataType	temperature, state
data	řetězec znaků

Skript „send_data.lua“ pro odesílání dat pomocí HTTP POST requestu na centrální server je přiložen v příloze (viz příloha „Naprogramované aplikace a skripty“, Code 1.2/strana 119)).

8.1.4 Uspání modulu na specifikovaný čas

Aby bezdrátové moduly mohli sloužit svému účelu, je nutné se zamýšlet nad napájením jednotlivých modulů a minimalizaci spotřeby energie. U některých modulů je nutný nepřetržitý provoz - jedná se o senzory a moduly, které čekají na spouštěcí událost. Například sensor pohybu, ovládání žaluzií, kamerový systém a pod. Druhá skupina sensorů, tzv. notifikáčnů, pracují na principu měření fyzikálních veličin v pravidelných časových intervalech. V časech kdy zařízení nevyčítá žádná data, je možné vstoupit do tzv. sleep módu, kdy modul spotřebovává téměř zanedbatelné množství energie a po zadaném časovém intervalu se probudí do operativního stavu a vykoná svou úlohu.

Pro tento spací mód existuje u modulu NodeMCU funkcionalita nazývaná „`dsleep()`“. Jedná se o funkci jádra, která po uplynutí definovaného času (u současného SDK max. 71 minut, u firmware vydaných před 5. lednem 2016 max. 35 minut) vyšle resetovací signál z PIN32 (RST) a na PIN8 (GPIO16 nebo taktéž označovaný na desce D0) se pokusí tento signál vyčíst. Pro správné fungování této funkcionality je proto nezbytné, aby piny RST a D0 byly propojeny. Schéma zapojení je zobrazené na obr. 8.1.



Obr. 8.1 Propojení pinů RST a D0 u NodeMCU - funkce `dsleep()`

Pro aktivaci spánkového režimu používám na modulech skript „`sleep.lua`“ (viz příloha „Naprogramované aplikace a skripty“, Code 1.3/strana 120). Dokumentace k funkci `node.dsleep()` k nalezení na url: <https://nodemcu.readthedocs.io/en/dev/en/modules/node/#nodedsleep>.

8.2 Modul pro měření teploty

Existují jednoduché senzory měřící pouze teplotu a pokročilejší senzory umožňující zaznamenávat i vlhkost vzduchu. Vlhkost vzduchu nás může zajímat v případě, kdy chceme udržovat v domácnosti konstantní vlhkost vzduchu například automatickými zvlhčovači. Doporučovaná vlhkost vzduchu v domácnosti by se měla pohybovat okolo 40 až 60% v závislosti na konkrétní místnosti. Vyšší vlhkost je obvykle v koupelně, kuchyni, ale i ložnici. Nežádoucí je vyšší vlhkost v místech, kde máme potraviny a oblečení. Hrozí jejich ohrožení plísní. Měření vlhkosti nám tedy může pomoci k zabránění vytváření plísní nebo naopak upozornit na problém nedostatečného zvlhčení vzduchu vedoucí ke zdravotním obtížím (pálení očí nebo možná bolest v krku).

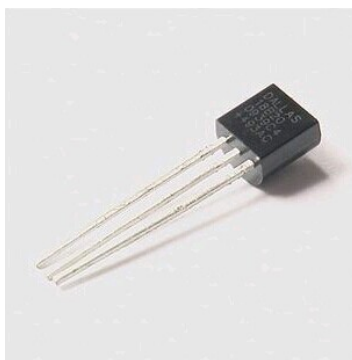
Jsou ale místa, kde nás vlhkost vzduchu nezajímá a postačuje nám pouhé měření teploty. V tomto případě můžeme sáhnout po levnějším senzoru. V našem případě senzor DS18B20. Pokud nás v místě měření zajímá i vlhkost, zvolíme senzor DHT11 nebo dražší ale přesnější DHT22.

8.2.1 Senzor DS18B20

Běžně užívaným senzorem pro měření teploty je DS18B20 (Obr. 8.3). Specifikace senzoru DS18B20 jsou znázorněny v následující tabulce (Tab. 8.2).

Tab. 8.2 Specifikace DS18B20

Teplotní rozpětí	-55°C až 125°C
Přesnost	$\pm 0.5^\circ\text{C}$ při teplotách -10°C až 85°C
Rozlišení	9 bit
Napájecí napětí	3,0 - 5,5 V
Odezva na změnu	750 ms
Rozměry	14,47 x 4,58 mm
Cena senzoru	16 Kč



Obr. 8.2 Senzor DS18B20

Tento senzor je jedním z nejlevnějších teplotních senzorů na trhu pro použití uvnitř i vně. Teplotní rozpětí senzoru se pohybuje v běžných hodnotách a přesnost senzoru je taktéž akceptovatelná pro běžné měření teplot v domácnosti.

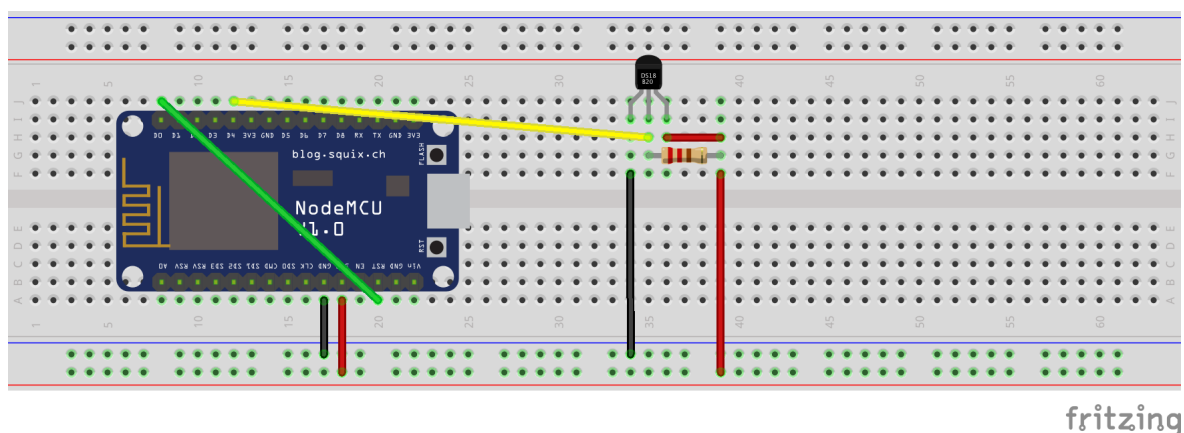
8.2.2 Návrh a implementace bezdrátového senzoru DS18B20

Seznam potřebných součástek k vytvoření bezdrátového teplotního senzoru je znázorněn v tabulce (Tab. 8.3).

Tab. 8.3 Seznam součástek a vyčíslení ceny

Součástka	Cena za kus	Počet	Cena
NodeMCU v1.0	90 Kč	1	90 Kč
DS18B20	16 Kč	1	16 Kč
4,7 k Ω resistor	5 Kč	1	5 Kč
Celková cena			111 Kč

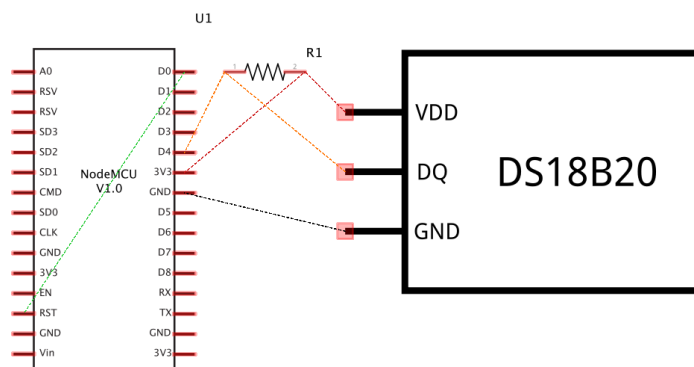
V aplikaci Fritzing byl vytvořen návrh k propojení jednotlivých součástek a základní otestování návrhu modelu (Obr. 8.3). Aplikace nám taktéž vygeneruje schéma elektronického zapojení modelu (Obr. 8.4).



Obr. 8.3 Senzor DS18B20

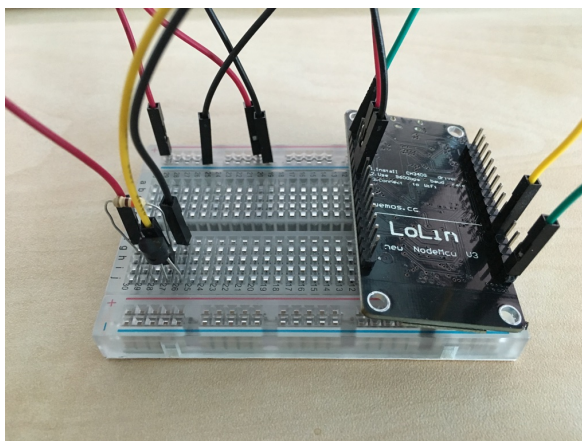
Na následujících fotografiích můžeme vidět sestavený bezdrátový model. Na prvním snímku (obr. 8.5) je zobrazeno zapojení modulu pomocí desky plošných spojů - jedná se o první sestavení dle schémat vytvořených aplikací Fritzing. Na snímku (obr. 8.6a) vidíme model bez použití desky plošných spojů a na posledním snímku (obr. 8.6b) je bezdrátový model napájen 750 mA baterií. Jedná se o základní testovací model, který je možné umístit kdekoli v domácnosti, kde je přítomen WiFi signál. Dle implementace programové části odesílá model informace o teplotě na nastavený koncový bod, který informace o teplotě zpracovává a případně vykonává další akce na základě těchto naměřených hodnot.

DS1
DS18B20

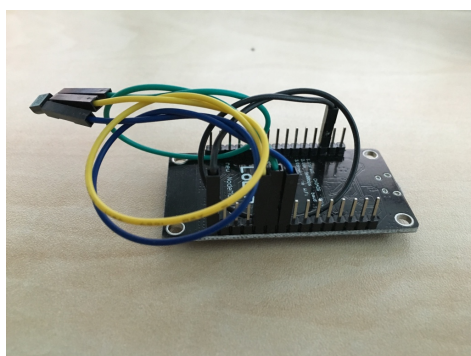


fritzing

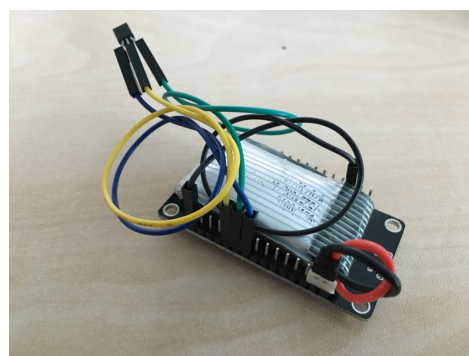
Obr. 8.4 Senzor DS18B20



Obr. 8.5 Kompletace bezdrátového modulu za použití desky plošných spojů



(a)



(b)

Obr. 8.6 Testovací bezdrátový modul bez baterie a s baterií

Aby vytvořený modul plnil svou funkci, tedy vyčítal aktuální teplotu a odesílal data na centrální server, je třeba na implementovat skript v jazyce LUA. Účelem tohoto skriptu bude:

1. aktivace modulu ze spánkového režimu („init.lua“)
2. připojení modulu k WiFi („wifi.lua“)
3. načtení hodnot z teplotního senzoru („temp_ds18b20.lua“)
4. odeslání vypočtené teploty na server („data_send.lua“)
5. uspání modulu na specifikovaný čas („sleep.lua“)

Modul uspáváme po vykonání akce z důvodu ušetření energie z připojené baterie. Nepotřebujeme měřit teplotu každých několik vteřin, a proto měříme v pravidelných intervalech v řádu několika minut. V mezičase měření může být modul „vypnutý“ a nespotebovat stejné množství energie jako za jeho plném chodu.

Abychom mohli přepočítat hodnotu ze senzoru, který data zasílá jako rozdíl vstupního a výstupního napětí, je v našem případě vhodné použít knihovnu funkcí předpřipravených pro komunikaci s naším senzorem. Tato knihovna je dostupná na webovém portálu GitHub v repozitáři `nodemcu/nodemcu-firmware`¹⁾. Tato knihovna spolu s ostatními skripty je k nalezení v přílohách této práce.

Vyčítání hodnot ze senzoru

Pro načtení hodnot z teplotního senzoru použijeme výše zmíněnou knihovnu `ds18b20`. Ze zapojení senzoru k modulu musíme vyčíst snímací piny - v tomto případě pin D4 (GPIO 02), na kterém budeme vyčítat aktuální teplotu. Skript je k nahlédnutí v příloze (viz příloha „Naprogramované aplikace a skripty“, Code 1.4/strana 120).

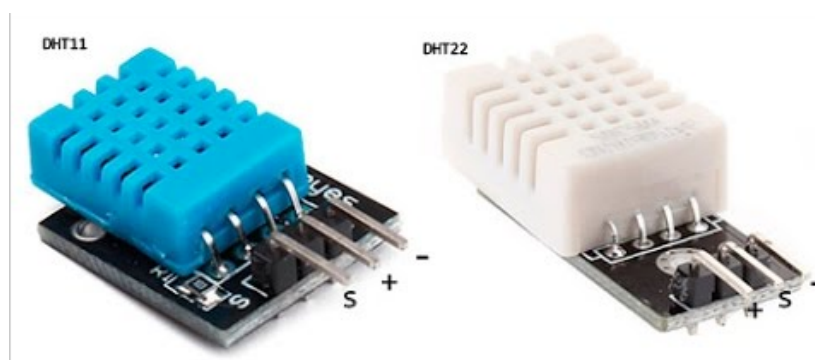
¹⁾<http://github.com/nodemcu/nodemcu-firmware/blob/master/luasrc/modules/ds18b20/ds18b20.lua>

8.2.3 Senzor DHT-11 a DHT-22

Dalšími běžně užívanými senzory pro měření teploty a vlhkosti vzduchu jsou DHT-11 a vyšší řada DHT-22 (Obr. 8.7). Specifikace obou senzorů jsou znázorněny v následující tabulce (Tab. 8.4).

Tab. 8.4 Specifikace DHT-11 a DHT-22

	DHT-11	DHT-22
Teplotní rozpětí	0°C až 50°C	-40°C až 125°C
Přesnost	±2°C	±0,5°C
Vlhkostní rozpětí	20% až 80%	0% až 100%
Přesnost	±5%	±2-5%
Vzorkování	1Hz (každou 1 sec)	0,5Hz (každé 2 sec)
Napájecí napětí	3,0 - 5 V	3,0 - 5 V
Rozměry	15,5 x 12 x 5,5 [mm]	15,1 x 25 x 7,7 [mm]
Cena senzoru	22 Kč	64 Kč



Obr. 8.7 Senzor DHT-11 a DHT-22

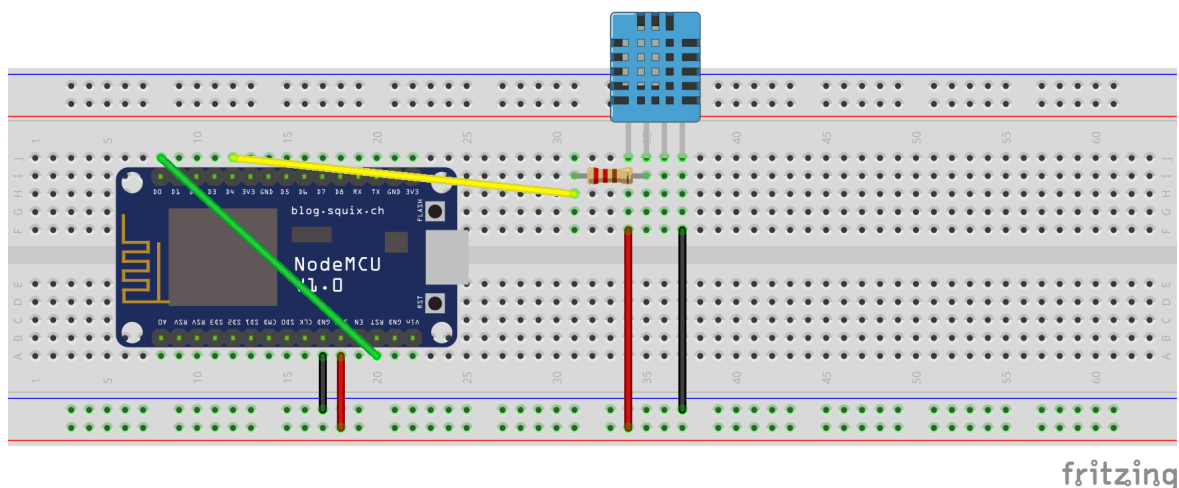
8.2.4 Návrh a implementace bezdrátového senzoru DHT-11

Seznam potřebných součástek k vytvoření bezdrátového teplotního senzoru je znázorněn v tabulce (Tab. 8.5).

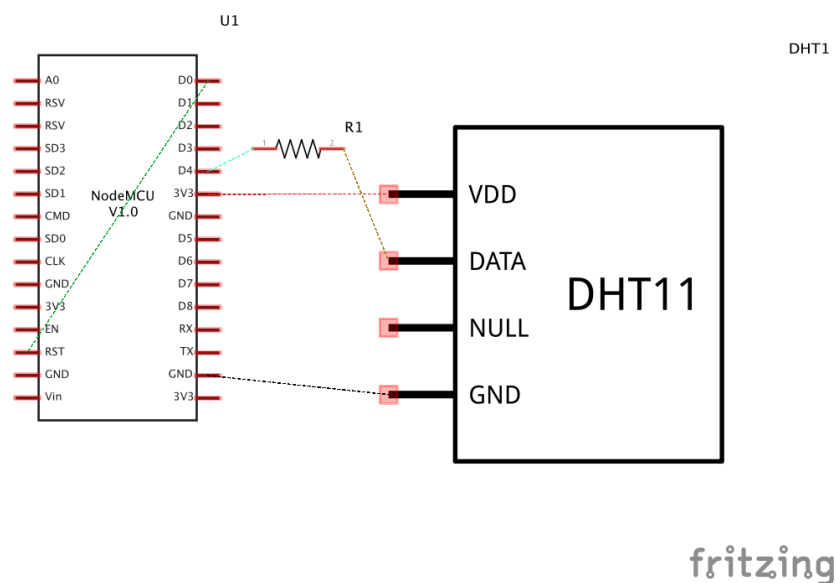
Tab. 8.5 Seznam součástek a vyčíslení ceny

Součástka	Cena za kus	Počet	Cena
NodeMCU v1.0	90 Kč	1	90 Kč
DHT-11	22 Kč	1	22 Kč
4,7 kΩ resistor	5 Kč	1	5 Kč
Celková cena			117 Kč

V aplikaci Fritzing byl vytvořen návrh k propojení jednotlivých součástí a základní otestování návrhu modelu (Obr. 8.8). Aplikace nám taktéž vygeneruje schéma elektronického zapojení modelu (Obr. 8.9).



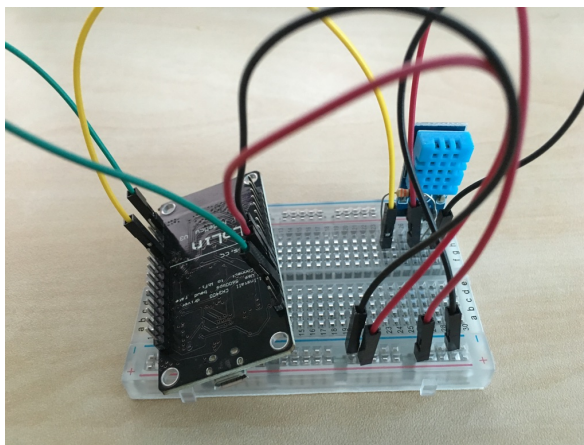
Obr. 8.8 Senzor DHT-11



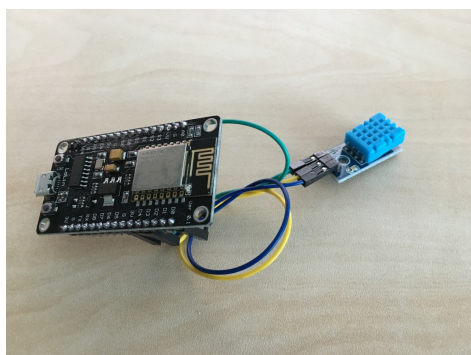
Obr. 8.9 Senzor DHT-11

Vyčítání hodnot ze senzoru

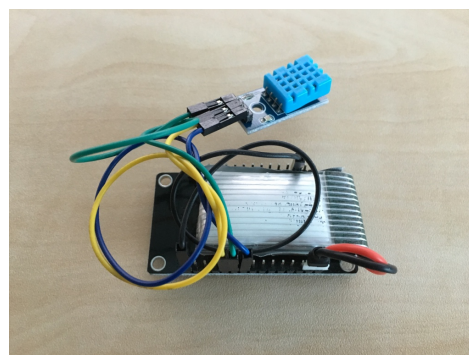
Pro načtení hodnot ze senzoru použijeme integrovanou knihovnu „dht“. Ze zapojení senzoru k modulu musíme vyčíst snímací piny - v tomto případě pin D4 (GPIO 02), na kterém budeme vyčítat aktuální teplotu a vlhkost. Data jsou odesílány opět pomocí skriptu „data_send.lua“. Skript je k nalezení v příloze (viz příloha „Naprogramované aplikace a skripty“, Code 1.5/strana 120).



Obr. 8.10 Kompletace bezdrátového modulu za použití desky plošných spojů



(a)



(b)

Obr. 8.11 Testovací bezdrátový modul bez baterie a s baterií

8.3 Modul pro monitorování vlhkosti půdy

Jedním z dalších velmi užitečných senzorů pro domácnosti jsou senzory vlhkosti půdy. Snad každá domácnost obsahuje různé druhy rostlin (od bylin až po kaktusy), které vyžadují pravidelné zalívání a udržování dostatečné vlhkosti půdy. Z mé osobní zkušenosti to vypadá tak, že používám jako detektor sucha jednu vybranou květinu, která je z naší domácnosti nejvíce náchylná na sucho. Jakmile se listy začnou být „vláčné“ a ohnuté směrem dolů, zpozorním a zaliji preventivně všechny květiny. Toto je samozřejmě velmi špatný systém, protože různé květiny potřebují různou vlhkost půdy a některým může tento styl velmi nevyhovovat.

Pokud máme bezdrátový senzor umístěný v květináči, v pravidelných intervalech modul zasílá aktuální informace o vlhkosti půdy. Tyto informace jsou sbírány a vyhodnocovány na centrálním serveru a v případě překročení určité předem definované hranice je uživatel domácnosti upozorněn formou notifikace (např. upozorněním přímo na uživatelský mobil). Jednou z dalších možností je nejen sbírání dat o vlhkosti půdy, ale případné automatické zalívání v případě potřeby. Rozšířením tohoto modulu o zásobník na vodu a jednoduchého malého čerpadla připojeného taktéž k modulu bychom získali autonomní systém, který by mohl fungovat např. měsíc bez nutnosti uživatelského zásahu - pouze v případě vyčerpání vody v zásobníku by musel uživatel vodu doplnit. Toto rozšíření ale není součástí této práce.

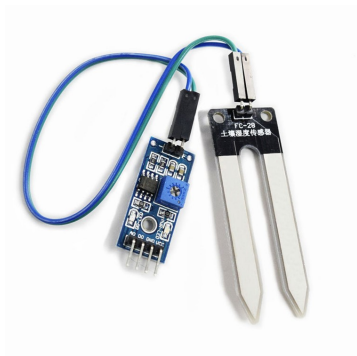
8.3.1 Senzor Produino LM393

Existuje mnoho senzorů pro detekci vlhkosti půdy, pro tuto práci byl vybrán cenově výhodný senzor Produino LM393 (Obr. 8.25). Specifikace senzoru je znázorněna v následující tabulce (Tab. 8.6).

Tab. 8.6 Specifikace Produino LM393

Digitální rozpětí	0 a 1
Analogové rozpětí	0 až 1024
Napájecí napětí	3,3 - 5,0 V
Rozměry	43 x 14 x 8 mm
Cena senzoru	47 Kč

Tento senzor je jedním z nejlevnějších senzorů pro měření vlhkosti v půdě na trhu. Umožňuje vyčítání hodnot jak v digitální podobě tak v analogové. Digitální interpretace je ale velmi nepřesná a ukazuje nám pouze, zdali vlhkost půdy překročila přednastavenou hranici. K posunutí hranice slouží digitální potenciometr ovladatelný pomocí křížového závitu na plošném spoji senzoru. Modul NodeMCU nám umožňuje vyčítat



Obr. 8.12 Senzor Produino LM393

i analogové hodnoty pomocí pinu A0 (ADC0) a jsme schopni si tak vyčítat přesné hodnoty ze senzoru, na které můžeme vytvořit libovolný počet hraničních hodnot.

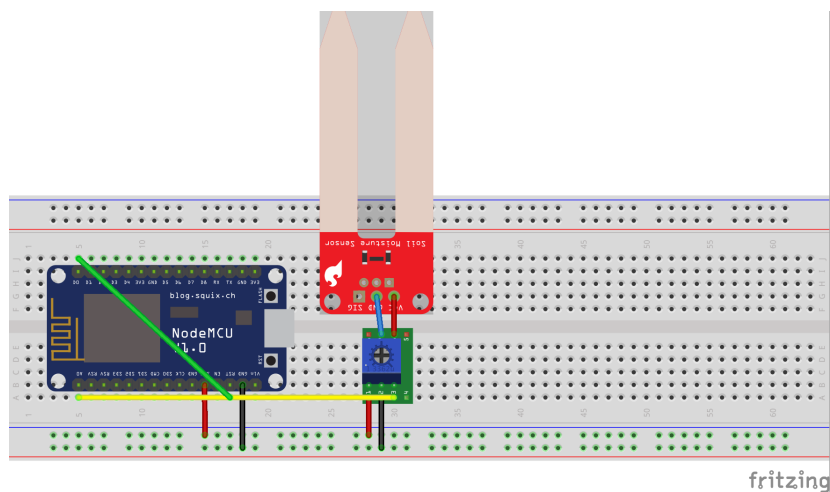
8.3.2 Návrh a implementace bezdrátového senzoru

Seznam potřebných součástek k vytvoření bezdrátového senzoru pro měření vlhkosti půdy je znázorněn v tabulce (Tab. 8.7).

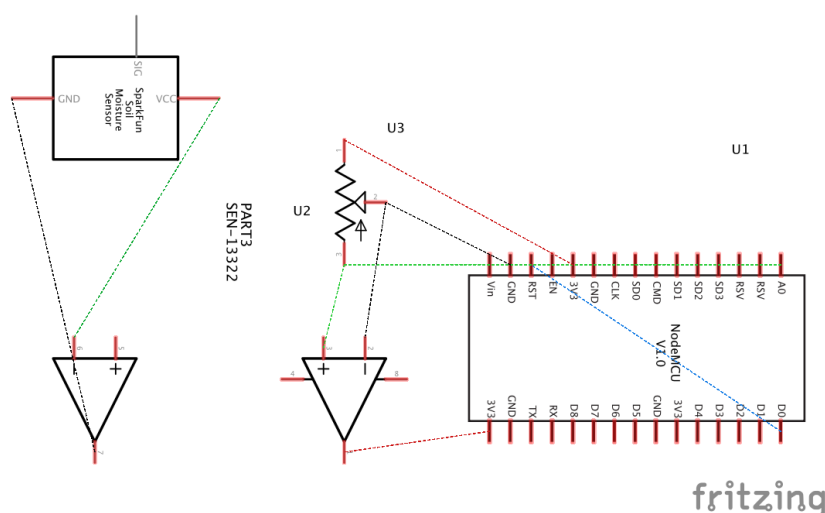
Tab. 8.7 Seznam součástek a vyčíslení ceny

Součástka	Cena za kus	Počet	Cena
NodeMCU v1.0	90 Kč	1	90 Kč
Produino LM393	47 Kč	1	47 Kč
Celková cena			137 Kč

V aplikaci Fritzing byl vytvořen návrh k propojení jednotlivých součástek a základní otestování návrhu modelu (Obr. 8.13). Aplikace nám taktéž vygeneruje schéma elektronického zapojení modelu (Obr. 8.9).



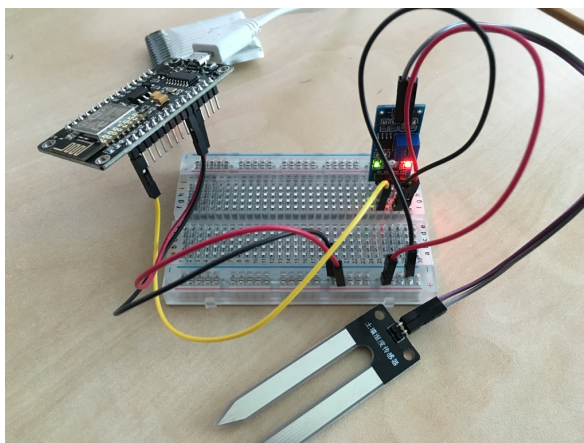
Obr. 8.13 NodeMCU a Produino senzor



Obr. 8.14 Schema zapojení NodeMCU a Produino LM393

Vyčítání hodnot ze senzoru

Pro načtení hodnot ze senzoru použijeme analogový pin A0, ze kterého vyčteme hodnotu 0 až 1024, která bude přemapována na aktuální vlhkost půdy. Data jsou odesílány opět pomocí skriptu „data_send.lua“ s typem dat „Moisture“. Skript je k nalezení v příloze (viz příloha „Naprogramované aplikace a skripty“, Code 1.6/strana 121). Modul je opět navržen jako notificační a jsou opět propojeny piny RST a D0 umožňující spací režim modulu. Vlhkost půdy domácích rostlin se na rozdíl od teploty nemění tak často a není třeba modul probouzet každých 10 minut a proto na modulu nastavíme dlouhou dobu spánku - 60 min (maximum je cca 71 min). Životnost modulu na baterii se takto výrazně prodlouží.



Obr. 8.15 Kompletace bezdrátového modulu za použití desky plošných spojů

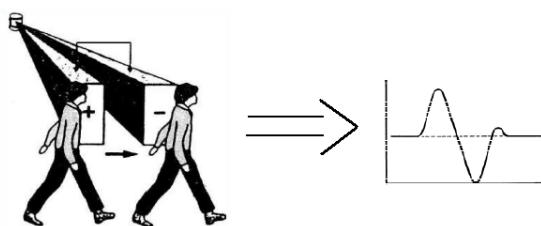


Obr. 8.16 Bezdrátový modul měřící vlhkost v květináči kaktusu

8.4 Modul pro detekci pohybu

Jedním z nejpoužívanějších senzorů v téměř každé budově či domácnosti je pohybový senzor. Tyto senzory jsou nejčastěji používány pro automatická vchodová světla, senzory světla na toaletách a dalších uzavřených místech bez oken.

Senzor funguje na principu pyroelektrického jevu, při kterém se pyroelektrické materiály deformují při změnách teploty. Pyroelektrické materiály tvoří podmnožinu piezoelektrických materiálů. Změna teploty, která vyvolá deformaci, tedy díky piezoelektrickému jevu indukuje na povrchu materiálu elektrický náboj. Na povrch pyroelektrického materiálu je optickou soustavou promítán obraz okolí. Pokud v okolí nastane tepelná změna, např. projde člověk, je materiál změnou teploty v části povrchu deformován a je možné detekovat indukovaný náboj na jeho povrchu[44].



Obr. 8.17 Generování impulsu při detekci pohybu[45]

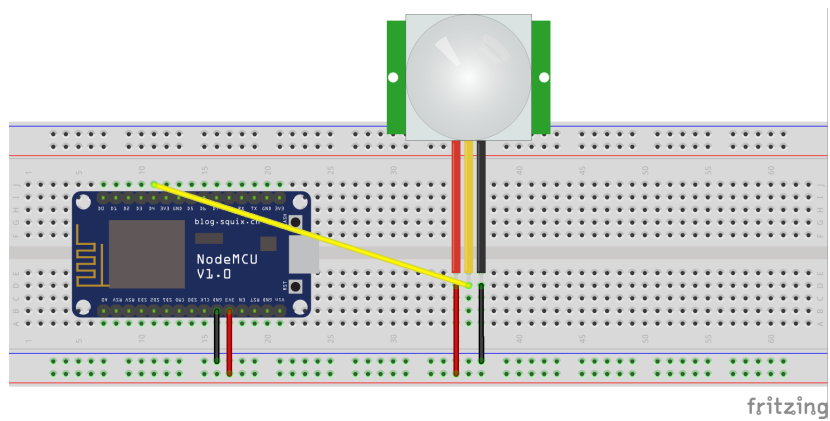
8.4.1 Návrh a implementace detekčního bezdrátového modulu

Seznam potřebných součástí k vytvoření bezdrátového senzoru pro detekci pohybu je znázorněn v tabulce (Tab. 8.9).

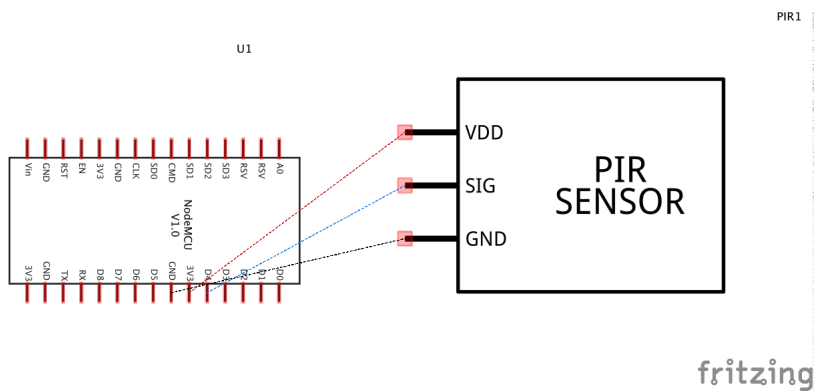
V aplikaci Fritzing byl vytvořen návrh k propojení jednotlivých součástí a základní otestování návrhu modelu (Obr. 8.24). Aplikace nám taktéž vygeneruje schéma elektronického zapojení modelu (Obr. 8.20).



Obr. 8.18 Senzor HC-SR501[47]



Obr. 8.19 NodeMCU a HC-SR501



Obr. 8.20 Schema zapojení NodeMCU a HC-SR501

Tab. 8.8 Specifikace HC-SR501[46]

Napájecí napětí	5 - 20 V
Úhel záběru	120° na vzdálenost 7 metrů
Nastavitelné zpoždění	5 až 200 sec
Provozní teplota	-15 až 70° C
Rozměry	32 x 24 x 23 mm
Cena senzoru	25 Kč

Tab. 8.9 Seznam součástek a vyčíslení ceny

Součástka	Cena za kus	Počet	Cena
NodeMCU v1.0	90 Kč	1	90 Kč
HC-SR501	25 Kč	1	25 Kč
Celková cena			115 Kč

Vyčítání hodnot ze senzoru

Pro načtení hodnot ze senzoru použijeme digitální pin D4, který nastavíme do módu „gpio.INT“ a pomocí funkce „gpio.trig()“ uvedeme pin D4 do módu automatické detekce. V tomto módu je na pinu D4 nastavené čekání na událost (tzv. callback) a v případě obdržení signálu 1 vyvolá modul přednastavenou funkci. V našem případě proběhne notifikace centrální stanice o detekci pohybu pomocí skriptu „data_send.lua“ s typem zasílaných dat „Motion“. Skript je k nalezení v příloze (viz příloha „Naprogramované aplikace a skripty“, Code 1.7/strana 121). Na rozdíl od předchozích senzorů se nejedná o pouhý notifikační senzor ale o senzor detekční. V tomto případě není chtěné, aby byl senzor v režimu spánku jako u notifikačních senzorů.

Jedna z možností jak ušetřit energii, ovšem za cenu zvýšené odezvy senzoru, je vzdálené spouštění modulu z centrálního serveru a v čase kdy není aktivovaný detekční senzor uvádíme modul do režimu spánku. Zde pak záleží na implementaci a na funkci modulu. V případě, že bychom rádi modul propojili s osvětlením tmavé místnosti je nutné aby modul neustále pracoval a byl připojený k centrálnímu serveru, který ovládá osvětlení v místnosti. Pokud ale funkce modulu umožňuje delší prodlevu, je možné modul uvádět do režimu spánku, například po dobu jedné minuty a poté kontaktovat centrální server zda má začít detekovat pohyb. V záporném případě by se opět uvedl do režimu spánku.

8.5 Modul pro ovládání domácího osvětlení

Ovládání domácího osvětlení vyžaduje nejen vytvoření modulu schopného komunikovat bezdrátově s mobilním zařízením, ale také bezdrátovou možnost ovládat světla samotná. Všechny předchozí modely pouze sbírali fyzikální data ze svého prostředí a se svým okolím dále nekomunikovaly. Pokud ale chceme vzdáleně ovládat světla, je třeba řešit problém s čím u světel komunikovat. Běžné domácí světlo je připojeno ke zdroji 220 V a je ovládáno dotykovým spínačem umístěným na zdi nebo na přívodném kabelu (v případě domácích lamp). Provedl jsem detailní analýzu současných řešení pro ovládání domácího osvětlení s přihlédnutím na neinvazivní řešení - tedy řešení, které nebudou vyžadovat jakékoli stavební úpravy. Dalším požadavkem byla možnost tlumení intenzity světla. Z této analýzy vyvstalo několik potenciálních metod jak problém řešit.

1. objímky umožňující bezdrátovou rádiovou komunikaci
2. nová generace bezdrátových žárovek

8.5.1 Patice umožňující bezdrátovou rádiovou komunikaci

Jedná se o velmi jednoduchý a účinný způsob, jak ovládat osvětlení v domácnosti. Toto řešení je velmi levné a za cenu cca 460 Kč můžeme pořídit 4 objímky s bezdrátovým RF ovladačem. Hlavní nevýhodou tohoto řešení je ale velmi špatná integrace do inteligentního systému. Objímky sice můžeme ovládat i za pomoci libovolného RF transmitteru a ne pouze za pomoci dodávaného ovladače, bohužel jsou ale patice s každým ovladačem spárovány a pro ovládání pomocí vlastního RF transmitteru jsme nuceni zachytit komunikaci každého patice se spárovaným ovladačem. Ač je toto řešení jedním z nejlevnějších, je také nejnáročnější na implementaci a vyžaduje vysokou úroveň znalosti radiových technologií.

8.5.2 Nová generace bezdrátových žárovek

V posledních několika letech se vedle radiově řízených patic objevují i WiFi řešení pro ovládání domácích světel. Na rozdíl od pouhého vložení patice mezi objímku a žárovku pracujeme s upravenou žárovkou, která obsahuje rádiový přijímač ve svém těle. Tento systém ale obsahuje další komponentu nazývanou WiFi Hub, který přijímá jednoduché UDP pakety a na základě nich vysílá rádiový signál pro ovládání jednotlivých žárovek. Toto řešení je velmi snadné na implementaci a ovládání pomocí libovolného zařízení komunikujícího přes WiFi. Cenově tento systém vychází draž než u předchozího řešení, dostáváme ale systém, který je velmi jednoduché integrovat do naší inteligentní domácnosti.

V současnosti existují dva hlavní výrobci těchto systémů:

1. Milight

- WiFi Hub (Obr. 8.21a)
 - Cena: 550 Kč
- Žárovka 6W a 9W (Obr. 8.21b)
 - Cena 6W: 230 Kč
 - Cena 9W: 437 Kč



(a)



(b)

Obr. 8.21 Produkty Milight - WiFi box[48] a žárovka[49]

2. Phillip Hue

- WiFi Bridge (Obr. 8.22a)
 - Cena: 1380 Kč
- Žárovka bílá a barevná (Obr. 8.22b)
 - Cena bílé: 345 Kč
 - Cena barevné: 1380 Kč

Dalším zástupcem „inteligentních“ žárovek je systém **LIFX**. Tento systém odebrává nutnost používání rádiové komunikace a tedy i překladače ve formě WiFi hub. Žárovka je sama schopná komunikovat přímo na WiFi a ovládání celého systému se tak ještě více zjednoduší. Bohužel cenově se samotná žárovka řadí do vyšší třídy. Jedná se o relativně novou technologii a s přibývajícím výrobci můžeme do budoucna očekávat snížení ceny.



(a)



(b)

Obr. 8.22 Produkty Phillips Hue - WiFi bridge[50] a žárovka[51]

Na trhu jsou aktuálně od LIFX celkem čtyři typy žárovek, které se liší hlavně v podporovaném barevném spektru a maximální svítivosti.

1. White 800 (Obr. 8.23a)
 - Cena: 920 Kč
2. White 900 (Obr. 8.23b)
 - Cena: 920 Kč
3. Color 1000 (Obr. 8.23c)
 - Cena: 1380 Kč
4. Color 1000 BR30 (Obr. 8.23d)
 - Cena: 1380 Kč



(a)



(b)



(c)



(d)

Obr. 8.23 Žárovky LIFX [52]

8.5.3 Implementace systému Milight

Pro tuto práci jsem si vybral systém Milight, který je nejlevnějším ze systémů žárovek nové generace. Tento systém je jednoduše ovladatelný z libovolného zařízení připojeného k WiFi, kterým jsme schopni zasílat UDP pakety. Jelikož se jedná o UDP, komunikace se zařízením je bez možnosti získání odpovědi či aktuálního stavu žárovky. Jeden WiFi hub může obsluhovat maximálně 4 žárovky. Pokud bychom potřebovali ovládat více žárovek, budeme potřebovat další WiFi hub.

Pro přístup k vybrané žárovce a nastavení požadovaného stavu musíme nejprve spárovat žárovku s WiFi boxem. Toto probíhá uvedením WiFi boxu do stavu párování a zapnutím žárovky. Ta po úspěšném spárování dvakrát problikne a od toho okamžiku jsme schopni žárovku vzdáleně ovládat.

Jelikož se jedná o systém, který komunikuje přímo po WiFi implementace obsluhy světel je provedena přímo na centrálním serveru. Pokud bychom chtěli spojit například modul pro detekci pohybu a rozsvícením světla ve vybrané místnosti, vytvoříme si stavové pravidlo na centrálním serveru, které při obdržení notifikace o zaznamenání pohybu vyšle signál žárovce. Stejně tak pokud modul v nastaveném čase nezaznamená pohyb v místnosti, centrální stanice automaticky vypne žárovku.

Implementace Milight kontroleru je založena na python knihovně milight, která je volně dostupná na webovém portálu GitHub[53]. Tato knihovna umožňuje ovládat jednoduchými příkazy všechny typy Milight žárovek. Pro inicializaci kontroleru je třeba znát IP adresu WiFi hubu a mít spárované všechny Milight žárovky. Knihovna milight je používána následovně:

Navázání spojení s WiFi hubem je vytvořeno pomocí parametrizovaného konstrukturu „MiLight“ s parametry IP adresa, port a inicializací žárovky pomocí konstrukturu „LightBulb“ se specifikováním připojených žárovek - rgb, rgbw a white.

```
1 import milight
2
3 controller = milight.MiLight({'host': 'x.x.x.x', 'port': 8899},
4                               wait_duration=0)
5 light = milight.LightBulb(['rgbw', 'white', 'rgb'])
```

Následně je možné ovládat žárovky přes vytvořenou instanci žárovky „LightBulb“. Pro zapnutí žárovek slouží metoda „on(id)“, kde id je identifikátor žárovky (číslo 1-4) nebo „all_on()“, která zapne všechny žárovky. Naopak pro vypnutí slouží metody „off(id)“ a „all_off()“. Pokud použijeme id=0 u metod on()/off(), je provedena změna na všech žárovkách stejně jako při použití funkcí all_on()/all_off(). Nastavení odešleme přes instanci controller WiFi hubu, který rozešle příkazy žárovkám.

```

1 -- zapnout
2 controller.send(light.on(1))
3 controller.send(light.all_on())
4
5 -- vypnout
6 controller.send(light.off(1))
7 controller.send(light.all_off())

```

Každá žárovka má určité parametry, které je možné měnit. U všech typů Milight žárovek nalezneme funkci `brightness`²⁾ pro změnu intenzity svícení, u žárovek typu White nalezneme funkci `warmness`³⁾, která nastavuje zda bude odstín studený nebo naopak teplý. Dalším rozdělením je na základě barevného modelu RGB⁴⁾ a HLS⁵⁾.

Tab. 8.10 Nastavení barvy

Funkce	Popis
<code>milight.color_from_hls(0.5, 0.5, 0.5)</code>	model HLS
<code>milight.color_from_rgb(255, 0, 0)</code>	model RGB
<code>milight.color_from_hex('#00ff00')</code>	hexadecimální vyjádření barvy

Tab. 8.11 Funkce pro ovládání žárovek typu White

Funkce	Popis
<code>brightness(percent, id)</code>	změna intenzity světla
<code>warmness(percent, id)</code>	změna barevného odstínu - studená, teplá
<code>night(id)</code>	nastavení nočního režimu

Tab. 8.12 Funkce pro ovládání žárovek typu RGB/GRBW

Funkce	Popis
<code>brightness(percent, id)</code>	změna intenzity světla
<code>color(milight.color(), id)</code>	nastavení barvy žárovky

Dále je možné využívat speciálních funkcí pro opakované příkazy, např. postupné snižování svítivosti žárovky a postupné zvyšování v cyklu.

²⁾světlost

³⁾hřejivost

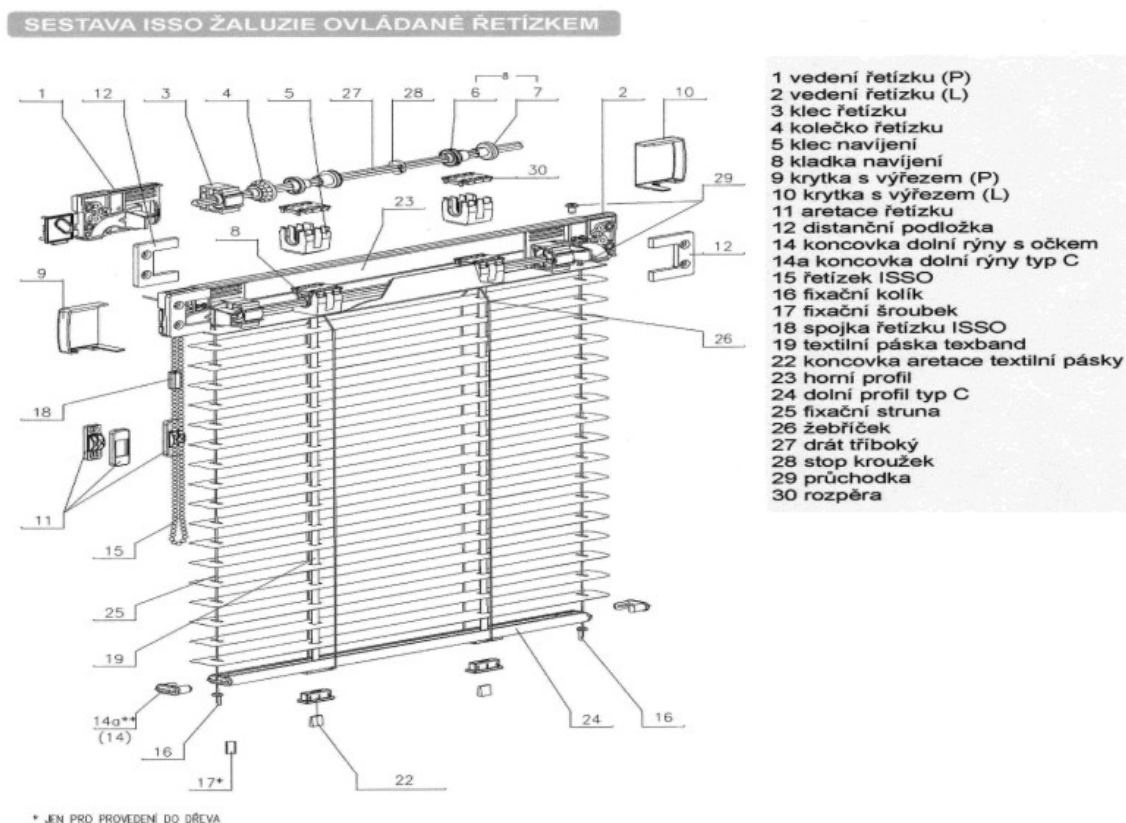
⁴⁾RGB - červená (red), zelená (green), modrá (blue). Model založený na míchání tří základních barev.

⁵⁾HLS - odstín barvy (Hue), světlost (Lightness) a sytost (Saturation). Tento model je používán ve foto editorech a odpovídá lidskému vnímání a popisu barev.

```
1 import time
2 commands = light.fade_up() + light.fade_down()
3
4 -- 5x zopakujeme prikaz fade_up a fade_down
5 key = controller.repeat(commands, reps=5)
6
7 time.sleep(10)
8
9 -- zastavime opakovany efekt
10 controller.cancel(key)
```

8.6 Modul pro regulaci okenních žaluzií

Poslední z vytvářených modulů pro inteligentní domácnost je automatizovaný regulátor okenních žaluzií. Existuje mnoho systémů žaluzií, v této práci si všímáme aktuálně nejrozšířenějšímu systému vnitřních horizontálních žaluzií dodávaném do běžných plastových oken. Tento systém je ovládán pomocí kuličkového řetízku, který otáčí ovládací hřídel žaluzie.



Obr. 8.24 Systém horizontálních žaluzií[54]

Nejjednodušší způsob jak systém žaluzií ovládat je přímé ovládání hřídele. U některých typů oken je tato metoda problematická a může být vyžadována invazivní montáž. Hřídel je zakrytovaná plechovým krytem a není tak možné vložit do ní motorek, který by hřídel ovládal. Proto jsem zvolil druhou metodu ovládání žaluzií a to pomocí řetízku, který po zatažení otáčí přes klec řetízku hřídel. Nevýhodou tohoto způsobu je montáž modulu na konec řetízku, tedy je třeba větší síly pro zatažení, minimálně 12V motorek, a z toho důvodu je potřeba modul napájet adaptérem.

8.6.1 Krokový elektromotor 28BYJ-48

Jedním z mála krokových elektromotorků s provozním napětím 12 V je elektromotor 28BYJ-48, který je v nižší cenové hranici do 200 Kč. Krokové motorky jsou vhodné k ovládání žaluzií z důvodu řízeného otočení motorku o přesně daný počet kroků. Na jedno celé otočení hřídele motorku je potřeba 64 kroků.

Tab. 8.13 Specifikace MOTS1 12V[46]

Provozní napětí	12 V
Úhel kroku	5.625°
Počet kroků	64
Převodový poměr kroků	1/64
Maximální rychlost otáčení	90 pps
Hlučnost	40dB
Rozměry	35 x 28 x 19 mm
Cena senzoru	154 Kč



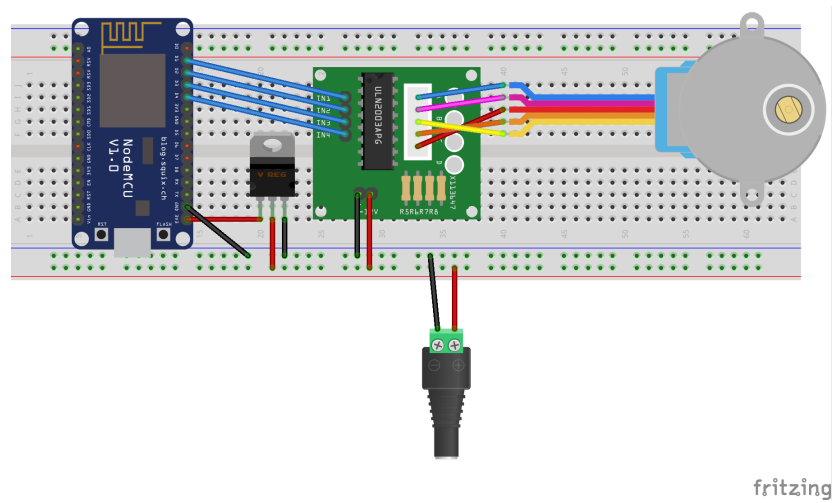
Obr. 8.25 elektromotor MOTS1

8.6.2 Návrh a implementace regulátoru žaluzií

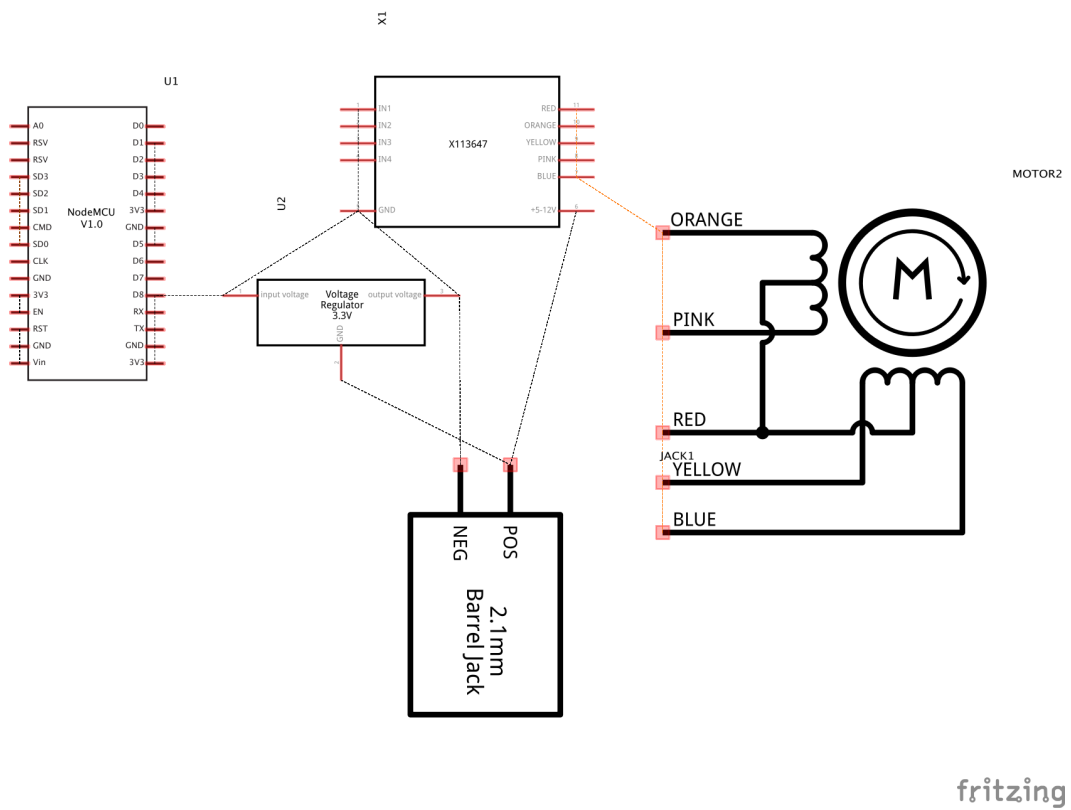
Seznam potřebných součástí k vytvoření modulu pro ovládání žaluzií je znázorněn v tabulce (Tab. 8.14).

V aplikaci Fritzing byl vytvořen návrh k propojení jednotlivých součástí a základní otestování návrhu modelu (Obr. 8.26). Aplikace nám taktéž vygeneruje schéma elektronického zapojení modelu (Obr. 8.27).

Krokový elektromotor je zapojený k modulu pomocí 4 pinů - D1, D2, D3 a D4, které jsou s elektromotorem propojeny přes procesní jednotku ULN2003A. Tento chip



Obr. 8.26 NodeMCU a elektromotor MOTS1 12V



Obr. 8.27 Schema zapojení NodeMCU a elektromotoru MOTS1 12V

Tab. 8.14 Seznam součástí a vyčíslení ceny

Součástka	Cena za kus	Počet	Cena
NodeMCU v1.0	90 Kč	1	90 Kč
Elektromotor MOTS1 12V	154 Kč	1	154 Kč
Elektromotor drive board	41 Kč	1	41 Kč
Napájecí adaptér	164 Kč	1	164 Kč
Regulátor napětí	18 Kč	1	18 Kč
Celková cena			467 Kč

se stará o správný převod impulzů obdržných z našeho modulu NodeMCU k následnému otáčení hřídele elektromotoru. Pro správné otočení hřídele je nutné aktivovat jednotlivé piny ve správném pořadí aby se vykonal jeden krok hřídele. Celý proces je zaznamenán v Tab. 8.15. Jak zde vidíme, pro vykonání jednoho kroku po směru hodinových ručiček je třeba poslat 8 po sobě jdoucích signálů. Pokud bychom chtěli otočit hřídel o plných 360° musíme zaslat celkem 512 po sobě jdoucích signálů v přesném pořadí. Poslední řádek v tabulce slouží pro zastavení. Pokud chceme otáčet hřídeli proti směru hodinových ručiček, je třeba matici (Tab. 8.15) upravit a postupovat od spodu. Tedy vykonávat kroky v tomto pořadí: 6, 5, 4, 3, 2, 1, 0, 7.

Tab. 8.15 Vykonání jednoho kroku po směru hodinových ručiček

Krok	D1	D2	D3	D4
0	0	0	0	1
1	0	0	1	1
2	0	0	1	0
3	0	1	1	0
4	0	1	0	0
5	1	1	0	0
6	1	0	0	0
7	1	0	0	1

Výsledný kód pro otáčení hřídele je k nahlédnutí v příloze (viz příloha „Naprogramované aplikace a skripty“, Code 1.8/strana 122).

9 KAMEROVÝ SYSTÉM

Jedním z hlavních bezpečnostních prvků pro zabezpečení domácnosti jsou rozhodně kamerové systémy. Mohou sloužit nejen proti vnějším vlivům ale také jako možnost z jednoho místa monitorovat různé části domácnosti, zvířata či dětské pokoje bez nutnosti fyzické přítomnosti. Většina kamerových systémů taktéž umožňuje funkci detekce pohybu a v případě, že je pohyb zaznamenán, kamerový systém umí notifikovat uživatele, zaslat fotografii při detekci pohybu či spustit nahrávání oné události.

Existuje obrovské množství kamer a kamerových systémů které můžeme používat pro zabezpečení naší domácnosti. Největší důraz při výběru kvalitní kamery je třeba klást na její funkci, na její umístění a také na možnosti komunikovat s dalšími systémy. Pro venkovní kameru budou nutné naprosto jiné parametry než pro kameru vnitřní.

V této práci je rozebírána Raspberry Pi kamera, která je jednoduše integrovatelná se zařízením Raspberry Pi a dosahuje vynikajících kvalit u domácích projektů.

Tab. 9.1 Raspberry Pi Camera

Senzor	OV5647 5 megapixel
Rozlišení	1080p
Úhel záběru	75,7°
Napájecí napětí	3,3V
Speciální funkce	Night Vision ¹⁾
Rozměry	25 x 24 x mm
Cena senzoru	460 Kč

9.1 Instalace kamery do Raspberry Pi

Kamera se skládá ze dvou částí - čočky umístěné na plošném spoji a datového kabelu (Obr. 9.1). Pokud chceme kameru používat i v noci, je doporučováno ke kameře dokoupit dva LED generátory infračerveného světla, které se uchytí šroubky na plošný spoj kamery každý z jedné strany (Obr. 9.1). Bez těchto přídavných světel nebude kamera schopna kvalitně zaznamenávat obraz ve tmě.

Instalace kamery k Raspberry Pi je velmi jednoduchá. Stačí pouze zasunout datový kabel do připravených zdírek správným směrem - tedy obnaženými kontakty směrem ke plošnému spoji kamery a u Raspberry Pi obnaženými kabely směrem ke konektoru HDMI (viz Obr. 9.2).

Dalším krokem je spuštění Raspberry Pi a zpřístupnění kamery. V terminálu spustíme příkaz „sudo raspi-config“, v otevřeném nastavení systému nalezneme možnost „Enable camera“ a potvrdíme entrem. Poté ukončíme nastavení možností „finish“ a re-

startujeme Raspberry Pi. Oficiální postup připojení kamery k Raspberry Pi je dostupný na <https://www.raspberrypi.org/documentation/usage/camera/README.md>.

Po restartování systému můžeme kameru vyzkoušet. Spustíme terminál a vložíme příkaz „`raspistill -o test.jpg`“. Pokud jsme instalační postup provedli správně, vygeneruje se nám snímek z kamery. Test jsem prováděl v práci a na Obr. 9.3 můžeme vidět výhled z okna na vedlejší budovu.



Obr. 9.3 Test kamery

9.2 Implementace streamovací služby

Pro streamování živého záznamu kamery jsem zvolil opensource aplikaci RPi Cam Control, která je dostupná na GitHubu pod jménem RPi_Cam_Web_Interface[56]. Jedná se o webové rozhraní k Raspberry Pi, které umožňuje vzdálený přístup ke kameře včetně přístupu k nastavení kamery. Rozhraní taktéž podporuje integraci s detekčními systémy, popřípadě využití detekčního systému v kameře. Další užitečnou funkcionalitou je vytváření kalendáře, kde může uživatel nastavit různé kamerové režimy pro libovolné časové úseky (viz. Obr. 9.4).

Time Offset: 0 Sunrise: 04:08 Sunset: 19:43 Current: 19:48 Period: Dusk			
Period	Motion Start	Motion Stop	Period Start
Night	<input type="text" value="ca 1"/>	<input type="text" value="ca 0"/>	<input type="text" value="md 1;em night"/>
Dawn	<input type="text"/>	<input type="text"/>	<input type="text" value="md 1;em night"/>
Day	<input type="text" value="ca 1"/>	<input type="text" value="ca 0"/>	<input type="text" value="md 1;em auto"/>
Dusk	<input type="text"/>	<input type="text"/>	<input type="text" value="md 0;em night"/>

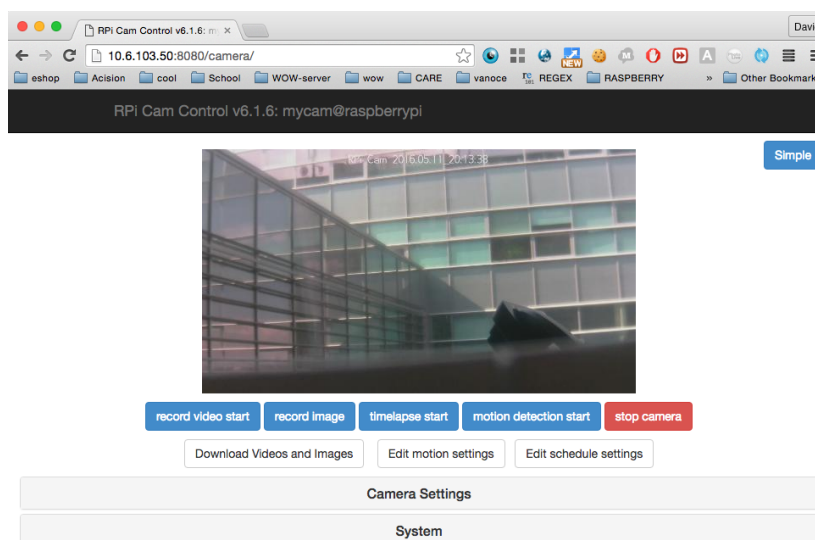
Obr. 9.4 Kalendář webového rozhraní kamery Raspberry Pi

Jednoduchým nastavením jsme schopni pokrýt libovolnou část dne a nastavit pro dobu kdy nejsme doma detekční systém, který při detekci pohybu spustí nahrávání a odešle notifikaci s fotografií na email.

Samotná instalace je velmi jednoduchá a je zapotřebí spustit pouze tyto 4 příkazy:

```
1 git clone https://github.com/silvanmelchior/RPi_Cam_Web_Interface.git
2 cd RPi_Cam_Web_Interface
3 chmod u+x *.sh
4 ./install.sh
```

Během instalačního procesu je uživatel dotazován na základní nastavení celého webového rozhraní. Toto nastavení je možné kdykoli po instalačním procesu změnit editací konfiguračního souboru „config.txt“. Následně, pokud jsme tak již neučinili během instalace, můžeme systém zapnout pomocí skriptu „start.sh“. Interface je pak dostupný na námi nastavené adrese, např. `http://10.6.103.50:8080/camera`. Pokud instalace proběhla úspěšně, měli bychom na dané webové adrese vidět podobný obsah jako na Obr. 9.5.



Obr. 9.5 Webového rozhraní RPi Cam Control

Jak můžeme vidět na Obr. 9.5, systém umožňuje velmi snadno ovládat kameru a s tím i celé streamování popřípadě i záznam. V základním nastavení je kvalita streamovaného videa nastavena na 10%, což výrazně snižuje velikost přenášených dat a obraz je streamován plynule. Pokud začneme obraz nahrávat, je automaticky zvolena

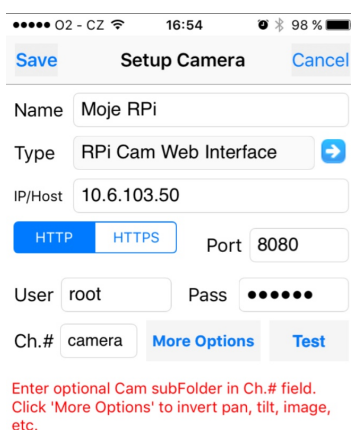
nejvyšší kvalita záznamu. Opět všechny parametry je možné jednoduše modifikovat v nastavení kamery.

9.3 Vzdálený přístup

Pokud chceme ke streamu přistupovat i mimo domov, je z hlediska bezpečnosti nutné přístup kontrolovat pomocí uživatelského jména a hesla. Dalším bezpečnostním prvkem může být i změna základního portu 80 na jiný. Toto vše je možné měnit v průběhu instalace. Jakmile máme nastavené přihlašovací údaje a změněný port, restartujeme webové rozhraní a zkusíme se přihlásit. Následně je nutné se přihlásit na domácí router a nastavit přesměrování našeho vybraného portu na IP adresu Raspberry Pi. Jakmile je toto hotovo, můžeme se zkusit přihlásit např. z mobilního telefonu, který není připojený do naší domácí sítě.

9.3.1 Mobilní aplikace pro vzdálený přístup

Přístup ke streamu je možný i za pomoci mobilní aplikace „IP Cam Viewer Lite“. Tato aplikace je dostupná jak pro mobilní systém Android²⁾ tak iOS³⁾. Po instalaci aplikace je třeba přidat novou IP kameru - Raspberry Pi, type: „Rpi Cam Web Interface“ a nastavit přístupové údaje ke kameře (viz. Obr 9.6). Do kolonky „Ch.#“ je nutné vložit poslední část webové adresy kamery.



Obr. 9.6 Nastavení mobilní aplikace IP Cam Viewer Lite

²⁾Android - <https://play.google.com/store/apps/details?id=com.rcreations.ipcamviewer>

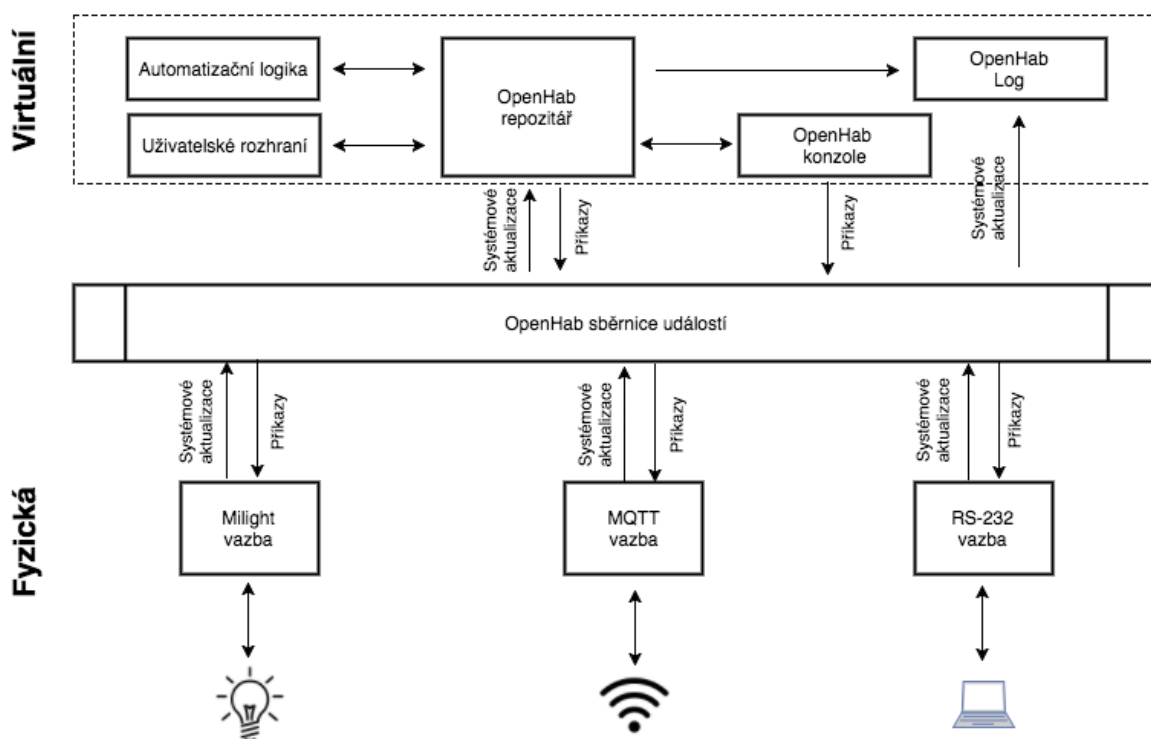
³⁾iOS - <https://itunes.apple.com/us/app/ip-cam-viewer-lite/id397322405?mt=8>

10 IMPLEMENTACE SERVEROVÉ APLIKACE OPENHAB

OpenHAB (Open Home Automation Bus) je serverová Java aplikace určená pro domácí automatizace. Poskytuje jak serverový backend, který může běžet na všech používaných platformách (Windows, Linux a OS X), tak frontend ve formě webového prostředí a nativních mobilních aplikací pro Android i iOS. Jedná se o projekt s otevřeným kódem, tedy aplikaci je možné pod licenci EPL používat zdarma. Právě tímto se OpenHab stává výborným kandidátem pro začínající projekty domácí automatizace a vytváření inteligentní domácnosti.

10.1 Architektura

Backend aplikace je velmi nenáročný a je možné ho bez problému používat i na méně výkonných zařízeních (např. Raspberry Pi nebo Arduino). Systém je modulární a je možné ho jednoduše rozšiřovat na základě addonů, které fungují jako rozhraní pro externí zařízení. Mezi těmito rozšířeními tak můžeme najít napojení (binding) na systém Milight pro ovládání žárovek, napojení na MQTT broker v obou režimech publish/-subscribe, emailový klient, bluetooth a další. Blokové schéma architektury OpenHab je zobrazeno na Obr. 10.1.



Obr. 10.1 Blokové schéma architektury OpenHab

10.1.1 Sběrnice událostí

Sběrnice událostí (Obr. 10.1) je hlavní služba aplikace OpenHab, která umožňuje komunikaci mezi jednotlivými prvky architektury. Každé zařízení je schopné využít sběrnici pro aktualizaci nebo oznámení změny stavu. Dalšími událostmi jsou tzv. příkazy, které mění stav nebo spouští akce na připojeném zařízení. Všechny bindings (vazby) komunikují přímo se sběrnici. Tato nízká vazba mezi jednotlivými svazky umožňuje dynamickou povahu.

10.1.2 Datové uložště

Neboli repozitář (Obr. 10.1) slouží pro ukládání aktuálních stavů všech připojených prvků a stará se o synchronizaci mezi všemi připojenými balíčky. Všechna obdržená data je možné ukládat do databáze, na disk nebo do cloudových systémů aby byla zajištěna perzistentnost i po restartování systému.

10.1.3 Automatizační logika

Základem pro inteligentní domácnost je možnost automaticky reagovat na aktuální stavy připojených zařízení. Dle přednastavených pravidel (Automation rules) je možné automaticky ovládat všechna připojená zařízení. Příkladem může být reakce na přijatou změnu stavu od pohybového senzoru, kdy v případě detekce pohybu a splnění dané časové podmínky (mezi 20:00 a 7:00) aktivujeme osvětlení odesláním příkazu pro žárovku Milight.

10.1.4 Konzole

Pomocí konzole (Obr. 10.1) je možné ovládat OpenHab nejen pomocí grafického uživatelského rozhraní, ale také pomocí příkazové konzole. Mezi tyto příkazy patří:

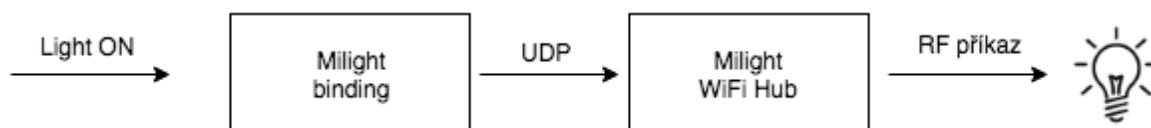
- odesílání příkazů
- vyčítání aktuálních stavů

10.1.5 Logování stavů

OpenHab Log (Obr. 10.1) slouží pro ukládání dat do souborů (logů) nebo do databází. Výchozí databází je RRD4J (Round Robin Database for Java), která slouží hlavně pro vykreslování grafů v uživatelském prostředí. Další možností je napojení na databázi SQL nebo jiné. Pokud ukládáme data do souborů, můžeme tyto soubory nalézt v adresáři v `openhab_home/logs`.

10.1.6 Vazby

Vazby (z anglického slova Bindings) jsou určeny pro komunikaci s externími zařízeními. V každé vazbě jsou definovány funkce a komunikační protokol pro komunikaci a ovládání externího zařízení pomocí jednotných příkazů ze sběrnice. Příkladem vazby může být balíček Milight, který umožňuje komunikaci s Milight WiFi hub. Ze sběrnice přijde do balíčku obecný příkaz pro spuštění určité žárovky a z balíčku odchází příkaz určený pro Milight WiFi Hub. Komunikace je znázorněna v Obr. 10.2.



Obr. 10.2 Komunikace vazby Milight s externím zařízením

10.2 Zabezpečení

K zabezpečení komunikace s OpenHab jsou dostupné dva mechanismy[58]. Komunikace klientských aplikací se serverem pomocí protokolu HTTPS (Hypertext Transfer Protocol Secure) a šifrování pomocí SSL (Secure Sockets Layer). Druhou možností je autentizace uživatelů vůči službě JAAS (Java Authentication and Authorization Service), která umožňuje řídit přístup uživatelů na základě uživatelských loginů a hesel. Pro pokročilé nastavení je možné využít nastavení seznamu pravidel přístupu ACL¹⁾.

10.2.1 Nastavení zabezpečení

Je možné volit ze tří typů zabezpečení:

- ON zabezpečení je globálně povoleno
- OFF zabezpečení je vypnuto
- EXTERNAL pouze požadavky, které přichází z externí sítě musí být autorizovány

10.3 Konfigurace

Konfigurace aplikace OpenHab je založena pouze na textových souborech. Úprava těchto souborů je tedy velmi jednoduchá a je možné ji provádět odkudkoli. Existují ale i speciální nástroje pro jednodušší editaci uživatelského prostředí - OpenHab Designer.

¹⁾ACL (Access Control list) - je v oblasti počítačové bezpečnosti seznam oprávnění připojený k nějakému objektu (např. souboru). Seznam určuje, kdo nebo co má povolení přistupovat k objektu a jaké operace s ním může provádět[57].

Konfigurace a tedy i konfigurační soubory jsou rozděleny na celkem sedm skupin[59]:

- Položky (items)
- Mapa stránek (sitemap)
- Pravidla (rules)
- Skripty (scripts)
- Akce (actions)
- Perzistence (persistent)
- Transformace (transformation)

Konfigurační soubory jsou všechny uloženy v adresáři **/configurations**, kde jsou rozděleny do jednotlivých skupin. V tomto adresáři nalezneme také konfigurační soubory pro autorizaci uživatelů (**user.cfg**) a nastavení pro vazby a balíčky (**openhabs.cfg**).

10.3.1 Položky

Položky jsou nastavovány v konfiguračním souboru **/configurations/items** a obsahují příponu **.items**. V těchto konfiguračních souborech vytváříme položky a napojujeme je na vazby. Položky můžeme rozdělovat do virtuálních skupin, které hrají roli v aplikování pravidel a jsou dle skupin zobrazovány na uživatelském rozhraní.

Syntaxe pro přidání položky je následující[59]:

Typ jméno [štítek] [ikona] [(skupiny)] [konfigurace]

- **Typ** značí druh položky (viz. Tab 10.1).
- **Jméno** označuje název proměnné, kterou můžeme volat v dalších konfiguračních souborech pro přístup k této položce.
- **Štítek** označuje text, pod kterým bude položka viditelná v uživatelském rozhraní.
- **Ikona** slouží pro přidělení ikony, která bude společně se štítkem viditelná v uživatelském rozhraní.
- **Skupiny** virtuálně rozdělují položky do jedné nebo více skupin. S těmito skupinami můžeme dále pracovat v jiných konfiguračních souborech pro jednodušší přístup k podobným položkám.
- **Konfigurace** nastavuje vazby na externí zařízení (adresa, doba obnovení).

Tab. 10.1 Seznam podporovaných položek

Název	Popis	Typy příkazů
Call	Přístup k funkcím telefonu	-
Color	Informace o barvách (RGB)	OnOff, IncreaseDecrease, Percent, HSB
Contact	Stav kontaktu dveří/oken	-
DateTime	Obsahuje datum a čas	DateTime
Dimmer	Obsahuje hodnotu stmívače v %	OnOff, IncreaseDecrease Percent
Group	Sdružuje položky do jedné skupiny	funkce položek
Number	Ukládá hodnoty ve formátu čísla	Decimal
Rollershutter	Typicky používané pro žaluzie	UpDown, StopMove Percent
String	Text	String
Switch	Přepínač světel	OnOff

10.3.2 Mapa stránek

Mapy stránek slouží pro definici elementů a rámců uživatelského rozhraní. Jsou nastavovány v konfiguračním souboru `/configurations/sitemap` a obsahují příponu `.sitemap`. Každý element vyžaduje parametr `item`, který definuje jakou položku ma element zobrazovat. V Tab. 10.2 jsou uvedeny všechny podporované elementy.

Tab. 10.2 Seznam podporovaných položek

Název prvku	Popis
Colorpicker	Výběr barvy (RGB)
Chart	Zobrazí graf z dat uložených v rrd4j databázi
Frame	Rozdělí prvky uživatelského rozhraní do rámečků
Group	Zobrazí všechny položky náležící do skupiny
Image	Zobrazí obrázek
List	Zobrazí stav položek
Switch	Slouží pro zobrazení přepínače
Selection	Zobrazí menu pro výběr
Setpoint	Zobrazí tlačítka plus a minus pro nastavení hodnot
Slider	Zobrazí posuvník
Text	Zobrazí text a stav položek nebo vstup do další úrovně
Video	Zobrazí video
Webview	Zobrazí internetovou stránku

Syntaxe a použití jednotlivých elementů je dostupná na stránkách oficiální dokumentace <https://github.com/openhab/openhab/wiki/Explanation-of-Sitemaps>.

10.3.3 Pravidla

Pravidla jsou nastavovány v konfiguračním souboru `/configurations/rules` a obsahují příponu `.rules`. Pravidla psaná v jednom konfiguračním souboru sdílí definované proměnné.

Struktura pravidel je rozdělena do tří částí:

- import knihoven (Imports)
- deklarace proměnných (Variable declarations)
- pravidla (rules)

Import knihoven

Pro import knihovny se používá klíčové slovo `#import`, za kterým následuje jméno knihovny (používá se stejný přístup jako v jazyce Java).

Deklarace proměnných

Sekce určená pro deklaraci proměnných. Používáme buď statické proměnné začínající klíčovým slovem `val` a nebo proměnlivé proměnné uvozujeme slovem `var`.

Pravidla

Sekce pravidel začíná klíčovým slovem `when`, která určuje vstupní prohlídku pro vykonávání pravidla. Pokud je pravidlo splněno, pokračuje se v další sekci `then`, ve které je vlastní tělo pravidla. Pravidlo je ukončeno klíčovým slovem `end`.

Následující příklad pravidla je ukáзка pravidla závislého na aktuálním čase - pokud je více než 19 hodin, spustí světlo v obývacím pokoji.

```
rule "zapni svetlo"
when
    Time cron "0 0 19 ? * *"
then
    Svetlo0byvak.sendCommand(ON)
end
```

V těle pravidla je možné používat běžné programovací rozhodovací podmínky (`if`, `switch`) podobně jako v jazyce Java. Existují tři základní typy pravidel:

- časově závislá pravidla reagující na určitý čas
- stavová pravidla závislá na změnu stavu položek (použití cronu nebo Quartz)

- závislá na stavu systému - např. vypnutí/zapnutí

10.3.4 Skripty

Podobně jako ostatní konfigurace jsou skripty nastavovány v konfiguračním souboru `/configurations/scripts` ale na rozdíl od předchozích neobsahují příponu. Používaným jazykem je Xtend, který je nekompileovaný. Jedná se o odnož Javy, která umožňuje psát výstižný a stručný kód.

Syntaxe pro spuštění skriptu je následující:

```
callscript(jméno skriptu)
```

Po vykonání skriptu je navržena hodnota posledního výrazu, která může být i nulová.

10.3.5 Akce

Akce jsou předdefinované funkce psané v jazyce Java, které jsou staticky importovány a mohou být používány v pravidlech nebo skriptech. Jsou definovány dva druhy akcí - Akce jádra a Akce doplňků (addons).

Akce jádra

Jedná se o akce spojené s událostmi sběrnice, které umožňují zasílat příkazy typu:

```
sendCommand(položka, příkaz)
```

a aktualizace:

```
postUpdate(položka, stav)
```

kteřé dané položce zašle specifikovaný stav.

Dalším typem akcí je ovládání systémového zvuku, přehrávání audio souborů či internetových streamů. Zajímavostí je funkce převodu textu na řeč. Do této kategorie také patří akce spojené s logováním zpráv, zasílání HTTP požadavků, časové akce a jiné. Přesný popis akcí je dostupný na oficiálních stránkách dokumentace

<https://github.com/openhab/openhab/wiki/Actions>.

Akce doplňků

Každá vazba obsahuje své vlastní akce, které je možné používat. V předchozí kapitole byl popisován scénář přijetí požadavku ze sběrnice na spuštění Milight žárovky. Přijatý požadavek byl vazbou zpracován a byla spuštěna akce, která zaslala příkaz Milight Wifi Hubu. Příklad dalších akcí pro různé vazby je opět uveden v oficiální dokumentaci.

10.3.6 Perzistence

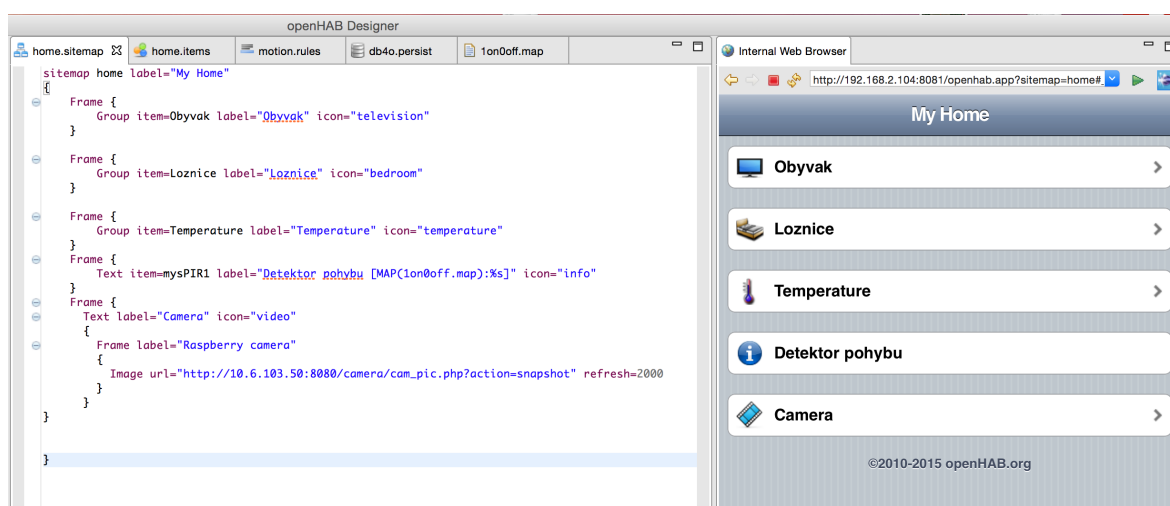
Jedná se o metody ukládání dat do souborů nebo databází. OpenHab nabízí jak možnost ukládání do jednoho systému tak do více najednou. Pokud chceme používat externí databázi nebo jiné zálohovací řešení, je nutné doinstalovat podobně jako u vazeb balíčků, který bude umožňovat s tímto externím řešením komunikovat. Perzistence jsou nastavovány v konfiguračním souboru `/configurations/persistence` a pokud chceme využívat určitou formu zálohování, je nutné na této cestě vytvořit konfigurační soubor typu `.persist`. Plný list podporovaných databází a serverů je dostupný na stránkách oficiální dokumentace <https://github.com/openhab/openhab/wiki/Persistence>.

10.3.7 Transformace

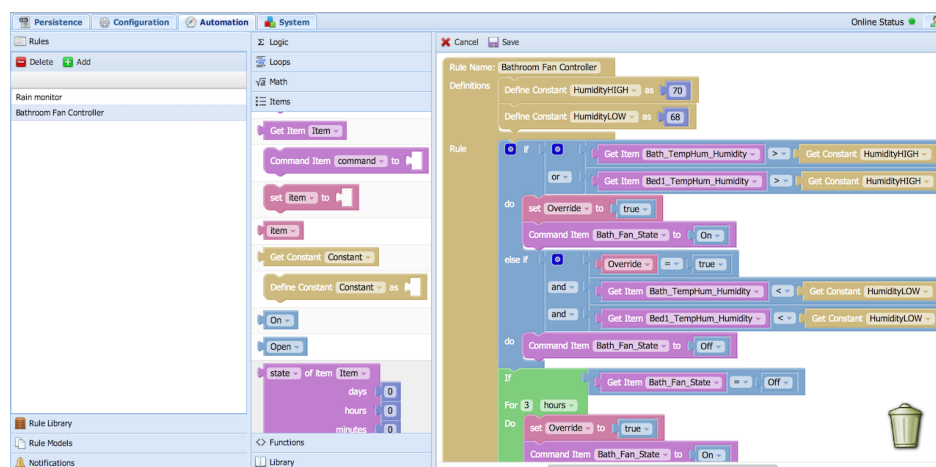
Transformace je nutná v případě, kdy potřebujeme zpracovat externí data do interního formátu. Příkladem může být transformace z webového rozhraní, kdy chceme z webové stránky o počasí přečíst předpověď a na základě této předpovědi vykonávat určitá pravidla nebo skripty. Transformace jsou nastavovány v konfiguračním souboru `/configurations/transformation` a mohou být použity buď ve formátu JSON (JavaScript Object Notation) nebo XML (eXtensible Stylesheet Language). Další možností transformací je mapování hodnot nebo rozsahů.

10.4 Nástroje pro úpravu konfigurace

Jedním z vynikajících nástrojů na úpravu konfigurace, pokud se nespokojíme s běžnými textovými editory, je OpenHab Designer. Tento nástroj je postavený na tzv. Eclipse RCP (Rich Client Platform), který umožňuje syntax highlighting (obarvování textu) specifických příkazů OpenHab. V této aplikaci můžeme konfigurovat libovolné OpenHab konfigurační soubory a navíc umožňuje grafický nástroj pro jednoduché vytváření uživatelského prostředí. Dále umožňuje jednoduché vytváření pravidel s kontrolou syntaxe a funkčnosti. V tuto dobu je ve vývoji aplikace pojmenovaná Rule Designer[60], která umožňuje vytvářet pravidla podle grafických flow chartů (viz. Obr. 10.4). Tento způsob vytváření pravidel je pro mnohé uživatele mnohem pochopitelnější pokud nemají zkušenosti s programováním.



Obr. 10.3 OpenHab Designer



Obr. 10.4 Rule Designer

10.5 Instalace OpenHab na Raspberry Pi

Instalace OpenHub na Raspberry Pi je velmi jednoduchá a rychlá. Nainstalovaná verze obsahuje vše potřebné pro spuštění jednoduché Demo stránky, díky které si uživatel vyzkouší základní ovládání celého systému a postupně začne systém upravovat dle svých potřeb.

Tato instalace předpokládá, že na Raspberry Pi je nainstalovaný funkční systém Debian nebo Raspbian. Výhodou je přístup pomocí SSH. Před zahájením jakékoli instalace je dobrým zvykem aktualizovat instalovaný systém.

```
sudo apt-get update
sudo apt-get upgrade
```

10.5.1 Instalace Java

Java by měla být součástí instalace operačního systému. V případě, že příkaz:

```
java -version
```

vrací error je nutné Javu doinstalovat ručně následujícím příkazem:

```
sudo apt-get install oracle-java7-jdk
```

10.5.2 Instalace OpenHab

Nejjednodušší metodou instalace OpenHabů je za pomoci příkazu **apt-get**, který vyhledává aplikace v definovaných repozitářích. Musíme tedy přidat do našeho systému odkaz na OpenHab repozitář a následně můžeme nainstalovat openhab-runtime balíček.

```
wget -q0 - 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab'\
|sudo apt-key add -
echo "deb http://dl.bintray.com/openhab/apt-repo stable main" \
| sudo tee /etc/apt/sources.list.d/openhab.list
sudo apt-get update
sudo apt-get install openhab-runtime
sudo update-rc.d openhab defaults
```

Bohužel během instalace se může stát, že některé soubory budou mít jako vlastníka uživatele „root“ a nebudou tak přístupné uživatelem „openhab“. Tento stav opravíme následujícími dvěma příkazy, které nastaví správného vlastníka:

```
sudo chown -hR openhab:openhab /etc/openhab
sudo chown -hR openhab:openhab /usr/share/openhab
```

Samba

Nepovinným ale doporučovaným krokem je nainstalovat na Raspberry Pi aplikaci Samba, která nám umožní z lokální sítě přistupovat na souborový systém Raspberry Pi.

Toto je velmi výhodné z hlediska zjednodušení konfigurace OpenHab, který je založený na konfiguračních souborech.

Sambu nainstalujeme pomocí příkazu:

```
sudo apt-get install samba samba-common-bin
```

následně je třeba nastavit Sambu aby umožňovala přístup k OpenHab složkám. Toto uděláme pomocí editace souboru vybraným textovým editorem (osobně používám aplikaci vim).

```
sudo vim /etc/samba/smb.conf
```

Je nutné odkomentovat řádek a změnit řádek **wins support** na následující:

```
wins support = yes
```

a přidat tyto řádky na konec konfiguračního souboru:

```
[OpenHAB Home]
comment= OpenHAB Home
path=/usr/share/openhab
browseable=Yes
writeable=Yes
only guest=no
create mask=0777
directory mask=0777
public=no
[OpenHAB Config]
comment= OpenHAB Site Config
path=/etc/openhab
browseable=Yes
writeable=Yes
only guest=no
create mask=0777
directory mask=0777
public=no
```

Posledním krokem je nastavením hesla uživatele openhab pro Sambu

```
sudo smbpasswd -a openhab
```

a restartování Samba servisy.

```
sudo service smbd restart
```

Nyní se můžeme na server připojit z jiného počítače a jednoduše konfigurovat OpenHab soubory.

```
smb://openhab@raspberry_ip_address
```

10.5.3 Konfigurace OpenHab

Nyní máme nainstalovaný OpenHab na Raspberry Pi a musíme serverovou aplikaci nakonfigurovat abychom ji mohli poprvé spustit. Toto uděláme použitím Demo balíčku,

který nám nastaví základní konfigurační soubory a vytvoří jednoduché uživatelské rozhraní s několika položkami a rámcimi.

Demo konfiguraci můžeme stáhnout z internetové stránky OpenHab²⁾, kde ji nalezneme pod jménem **Demo setup**. Po stažení balíček rozbalíme a nalezneme dvě složky **addons** a **configuration** (Obr. 10.5).

Name	^	Date Modified	Size	Kind
▼ addons		Yesterday 09:48	--	Folder
org.openhab.binding.http-1.8.2.jar		21 Mar 2016 19:50	20 KB	Java archive
org.openhab.binding.ntp-1.8.2.jar		21 Mar 2016 19:50	10 KB	Java archive
org.openhab.persistence.exec-1.8.2.jar		21 Mar 2016 19:56	6 KB	Java archive
org.openhab.persistence.logging-1.8.2.jar		21 Mar 2016 19:56	7 KB	Java archive
org.openhab.persistence.rrd4j-1.8.2.jar		21 Mar 2016 19:56	625 KB	Java archive
▼ configurations		Yesterday 23:36	--	Folder
▶ items		Yesterday 09:48	--	Folder
▶ persistence		Yesterday 09:48	--	Folder
▶ rules		Yesterday 09:48	--	Folder
▶ scripts		Yesterday 09:48	--	Folder
▶ sitemaps		Yesterday 09:48	--	Folder
README.TXT		21 Mar 2016 19:44	657 bytes	text

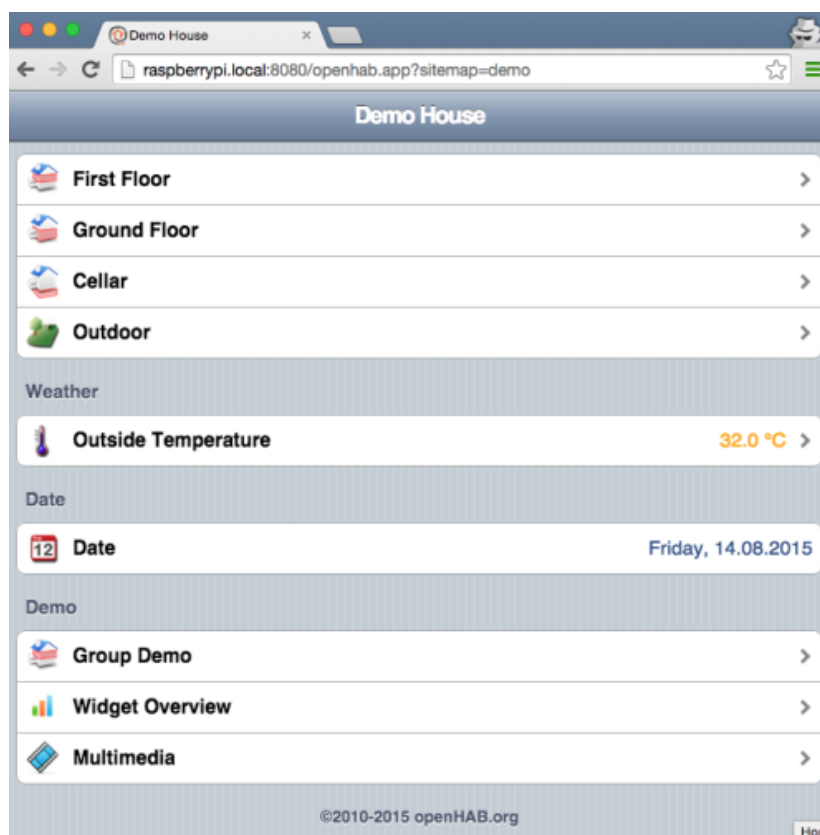
Obr. 10.5 OpenHab Demo soubory

Pomocí nastavené Samby, popřípadě jinou metodou, nahrajeme soubory ze složky „addons“ do adresáře **OpenHAB Home** a obdobně nahrajeme soubory ze složky „configuration“ do adresáře **OpenHAB Config** (přepíšeme původní soubory pokud jsme dotázáni).

Poté vložíme do prohlížeče následující adresu a měli bychom vidět stejný obsah jako na Obr. 10.6.

`http://raspbery_ip_address:8080/openhab.app?sitemap=demo`

²⁾ // www.openhab.org/getting-started/downloads.html



Obr. 10.6 OpenHab Demo běžící aplikace

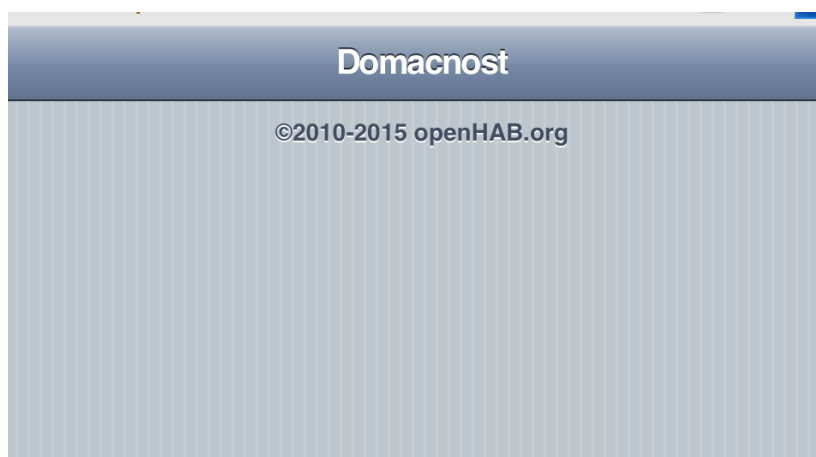
10.6 Uživatelské rozhraní

Když už máme OpenHab nainstalovaný a otestovánu základní funkčnost, můžeme se pustit do implementace našeho vlastního uživatelského rozhraní. Pro tento krok budeme potřebovat vytvořit dva nové konfigurační soubory **home.sitemap** pro mapu stránky a **home.items** pro položky (je nutné vytvořit konfigurační soubory v patřičných složkách). Můžeme použít jak libovolné textové editory tak výše popisovaný OpenHab Designer, který nám vytváření konfigurací značně usnadňuje.

Jakmile máme oba soubory vytvořeny, pustíme se do nastavování mapy, která bude určovat strukturu a vzhled naší aplikace. Pro inicializaci mapy musíme do souboru vložit tuto strukturu:

```
sitemap home label="Domacnost"  
{  
}  
}
```

Tímto se nám vytvoří holá stránka s pojmenováním Domacnost (Obr. 10.7). Přidáme několik místností, do kterých budeme postupně přidávat položky. Tyto místnosti jsou v mapách označovány jako „Frames“.



Obr. 10.7 OpenHab vytváření mapy - inicializace

```
sitemap home label="Domacnost"
{
  Frame {
    Group item=ObyvaciPokoj label="ObyvaciPokoj" icon="television"
  }
  Frame {
    Group item=Loznice label="Loznice" icon="bedroom"
  }
  Frame {
    Group item=Pracovna label="Pracovna" icon="computer"
  }
  Frame {
    Group item=Kuchyn label="Kuchyn" icon="kitchen"
  }
  Frame {
    Group item=Koupelna label="Koupelna" icon="bath"
  }
}
```



Obr. 10.8 OpenHab vytváření mapy - místnosti

Po uložení a obnovení stránky by aplikace měla vypadat jako na Obr. 10.8. Názvy ikonek je třeba volit dle ikon obsažených v adresáři:

```
OpenHAB Home: webapps/images
```

Pokud bychom chtěli použít jiné obrázky je možné je do tohoto adresáře nahrát ale je nutné dodržet daný formát.

Jakmile máme připravenou stránku, můžeme si do vytvořených skupin (štítků) přidat jednotlivé položky. Tzn. budeme konfigurovat položky v souboru **home.items**.

Jako první nadefinujeme skupiny dle použitých štítků. Je dobré si přidat i další skupiny dle funkcionality položek, např. skupina Svetla.

```
Group All
Group Loznice
Group ObyvaciPokoj
Group Pracovna
Group Kuchyn
Group Koupelna
Group Svetla
```

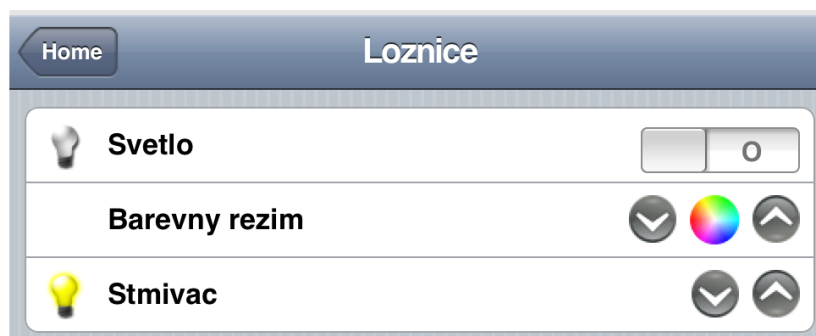
Dále si přidáme položky ovládající světla v ložnici. Tyto položky budou ovládat světla RGBW Milight a tyto žárovky nám umožňují nastavovat:

- Zapnout - Vypnout (nastavit bílou barvu)
- RGB barvy
- Stmívání

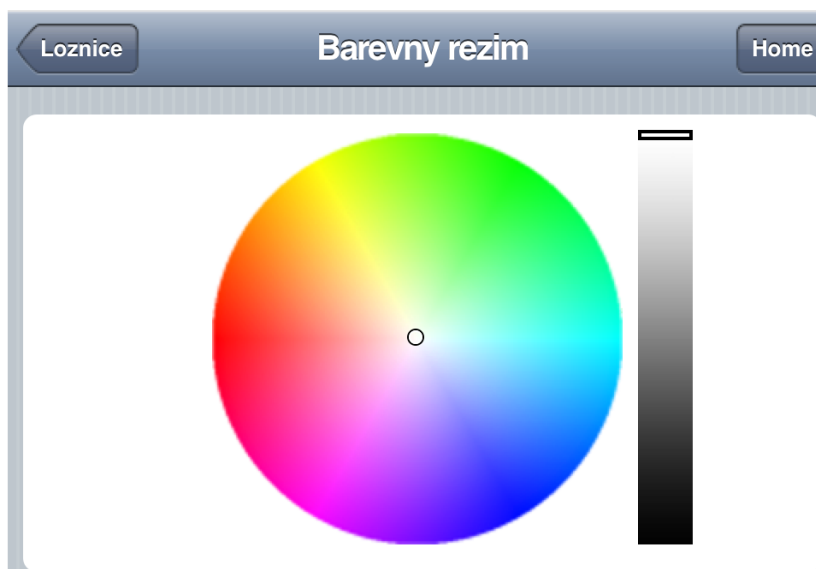
Vytvoříme tedy adekvátní položky s funkcionalitou Milight RGBW žárovek. Každá žárovka má své označení, RGBW žárovky mají označení 6-10, kde identifikátor 6 slouží pro ovládání všech žárovek najednou. Identifikátory jednotlivých žárovek jsou určovány při párovacím procesu s Milight WiFi Hubem, který identifikátory přiděluje postupně od 7 do 10.

```
Switch Loznice1 "Svetlo" (Loznice, Svetla)
    {milight="bridge;7;whiteMode"}
Color LozniceBarva „Barevny rezim" (Loznice, Svetla)
    {milight="bridge;7;rgb"}
Dimmer LozniceStmivac "Stmivac" (Loznice, Svetla)
    {milight="bridge;7;brightness;27"}
```

Po uložení bychom měli dostat uživatelské rozhraní jako na Obr. 10.9 a v případě zakliknutí položky „Barevny rezim“ můžeme měnit barvu žárovky (Obr. 10.10).



Obr. 10.9 OpenHab přidávání položek - ložnice



Obr. 10.10 OpenHab přidávání položek - výběr barvy

10.7 Integrace s bezdrátovými moduly protokolem MQTT

Při výrobě modulů jsme kladli velký důraz na komunikaci pomocí univerzálních protokolů. Jedním z těchto protokolů byl MQTT, který je popisovaný v třetí kapitole. Abychom mohli tento protokol používat, je třeba vytvořit MQTT Broker server, který bude přijímat do Topiců data ze senzorů a externí aplikace se k němu mohou registrovat a odebírat data v jednotlivých Topicích. Jedním z doporučovaných MQTT Broker aplikací je tzv. Mosquitto[61]. Tento broker je vhodný hlavně díky své jednoduchosti a nenáročnosti. Používá MQTT verzi protokolu 3.1 a 3.1.1. Jedná se navíc o Open source projekt, taktéž licencovaný pomocí EPL a plně zdarma.

10.7.1 Instalace Mosquitto

Postupujeme stejným způsobem jako při instalaci OpenHab. Přidáme si repositář Mosquitto a aplikaci nainstalujeme pomocí **apt-get**.

```
curl -O http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
sudo apt-key add mosquitto-repo.gpg.key
rm mosquitto-repo.gpg.key
cd /etc/apt/sources.list.d/
sudo curl -O http://repo.mosquitto.org/debian/mosquitto-repo.list
sudo apt-get update
sudo apt-get install mosquitto mosquitto-clients python-mosquitto
```

Pomocí těchto málo příkazů jsme nainstalovali Mosquitto MQTT broker. Dalším krokem je konfigurace vazby MQTT v konfiguračním souboru

```
/etc/openhab/configurations/openhab.cfg
```

v sekci MQTT. Zde odkomentujeme a zmeníme následující řádky:

```
mqt:broker.url=tcp://raspberrypi_ip_address:1883
mqt:broker.clientId=openhab
```

a následně restartujeme OpenHab příkazem:

```
sudo /etc/init.d/openhab restart
```

10.7.2 Nastavení bezdrátových modulů

Moduly máme již připravené pro komunikaci protokolem MQTT. Jediné co potřebujeme nastavit jsou názvy topiců a cesta k MQTT broker. Na příkladu modulu snímající teplotu a vlhkost vzduchu předvedeme jak nastavit dané topicy a jak se k nim přihlásit ze strany OpenHab.

Název topicu

Při vytváření topicu je nutné si uvědomit s jakými úrovněmi bude naše serverová aplikace pracovat. V případě běžné domácnosti nám postačí 3 - 4 úrovně:

pojmenování domácnosti / pojmenování místnosti / pojmenování měřené veličiny

Pokud tedy chceme ukládat data o teplotě a vlhkosti vzduchu z obývacího pokoje, zaregistrujeme modul k následujícím dvěma topicům:

```
home/obycipokoj/teplota
home/obycipokoj/vlhkost
```

Modul bude následně s každým měřením teploty a vlhkosti vzduchu odesílat data následujícím způsobem:

```
home/obycipokoj/teplota21
home/obycipokoj/vlhkost30
```

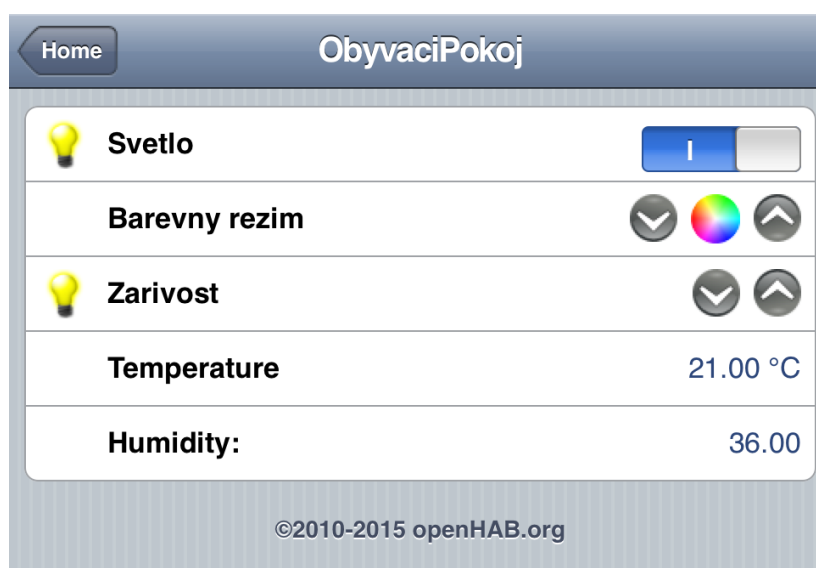
10.7.3 Konfigurace OpenHab položek

Abychom mohli vyčítat data o teplotě a vlhkosti vzduchu z MQTT brokeru, musíme vytvořit adekvátní položky, které se budou odkazovat na výše uvedené topic. Do konfiguračního souboru s položkami **home.items** přidáme následující dva řádky:

```
Number Temp "Teplota [%2f °C]" (ObyvaciPokoj, Teplota)
    {mqtt="<[mosquitto:home/obycipokoj/teplota:state:default]"}
Number Humi "Vlhkost: [%2f ]" (ObyvaciPokoj, Vlhkost)
    {mqtt="<[mosquitto:home/obycipokoj/vlhkost:state:default]"}

```

Jakmile soubor s položkami uložíme, je automaticky nahrán do systému a je zavolána metoda Subscribe na oba topic. Protože jsme přidali obě položky do skupiny „ObyvaciPokoj“, na webové i mobilní aplikaci (Obr. 10.11) nyní uvidíme jak položku Teplota tak položku Vlhkost s aktuálními naměřenými daty.



Obr. 10.11 OpenHab aplikace - obyci pokoj

10.8 Vytvoření pravidla pro kontrolu světel

Jednou z možností jak jsou ovládány světla bez fyzického kontaktu je za pomoci detektoru pohybu. Praxe je taková, že pokud je detekován pohyb a světla jsou vypnutá, systém rozsvítí světla na určitý časový okamžik. Pokud během tohoto okamžiku nezaznamená pohyb, jsou světla opět vypnuta. Pokusíme se takovéto pravidlo přenést do naší domácnosti.

10.8.1 Detektor pohybu

Nejprve je důležité přidat položku, která bude odkazovat na senzor pohybu. Tento senzor opět komunikuje pomocí MQTT a v případě zaznamenání pohybu, zapíše do topicu

```
home/events/215745/status1
home/events/215745/status0
```

kde číselná část označuje identifikátor modulu, status 1 v případě detekce pohybu a 0 v případě, že v uplynulé minutě nebyl zaznamenán pohyb.

Položka v **home.items** bude vypadat následovně:

```
Number Pohyb1          "Pohyb1"          (Sensor)
    {mqtt="[mosquitto:home/events/215745/status:state:default]"}
```

Abychom si aktuální stav mohli prohlédnout i v aplikaci, přidáme si textovou položku i do **home.sitemap**. Abychom ale neviděli v aplikaci číslo 1 nebo 0, vytvoříme si také transformovací mapu, která bude převádět číslo 1 na ON a číslo 0 na OFF. Tento soubor nazveme **1on0off.map** a bude uložen v adresáři transformací.

```
1=ON
0=OFF
ON=1
OFF=0
```

Nyní můžeme přidat nový Frame do mapy, kde číselná hodnota bude nahrazena výše popsanou transformací.

```
Frame {
    Text item=Pohyb1
        label="Detektor pohybu [MAP(1on0off.map):%s]"
        icon="info"
}
```

Na Obr. 10.12 můžeme vidět vytvořený Frame pro Detektor pohybu.

10.8.2 Implementace pravidla

Nyní vytvoříme pravidlo („motion.rules“ v adresáři **rules**), které bude kontrolovat hodnotu senzoru „mysPIR1“. Pokud bude detekován pohyb a světlo bude přepínačem „ObyvacíPokoj1“ vypnuté (tzn. základní stav světla je Off), zapneme světlo pod názvem



Obr. 10.12 OpenHab aplikace - detektor pohybu

„ObyvaciPokojAuto“ a nastaví se plná svítivost. Tento název je pouze virtuální a slouží pro rozdělení, zda bylo světlo spuštěno automaticky nebo uživatelem. Pokud nebude zaznamenán pohyb v první pátou minutu (nastavení pomocí cronu), stlumíme světlo na 50% a pokud ani po uplynutí dalších 5 minut nebude detekován pohyb, bude světlo vypnuto.

```
var Number counter = 0
var Number lastCheck = 0
rule "ObyvaciPokojOn"
when
    Item Pohyb1 changed
then
    if (ObyvaciPokoj1 == ON) {
        logInfo("motion.rules","ObyvaciPokoj1 is set to ON")
    }
    else
    {
        counter = counter + 1
        sendCommand(ObyvaciPokojAuto,ON)
        sendCommand(ObyvaciPokojDimmer, "100");
        logDebug("motion.rules", "ObyvaciPokoj1 state:" + Loznice)
    }
end
rule "ObyvaciPokojOff"
when
    Time cron "0 */5 * * * ?"
then
    if (ObyvaciPokoj1 == ON) {
        logInfo("motion.rules","ObyvaciPokoj1 is set to ON")
    }
    else if (lastCheck == counter) {
        counter = 0
        lastCheck = -1;
        sendCommand(ObyvaciPokojAuto, OFF)
    } else {
        sendCommand(ObyvaciPokojDimmer, "50");
        lastCheck = counter
    }
end
```

ZÁVĚR

Prvním úkolem na práci bylo vytvořit literární rešerši na téma možnosti inteligentního řízení domácnosti. V první části práce je definována inteligentní domácnost a očekávaná funkcionalita inteligentního systému. Následně jsou představeny vybrané komerční produkty, které fungují na bázi bezdrátových modulů. Je provedeno cenové srovnání těchto produktů a představena možnost vlastního řešení inteligentního systému.

Dalším úkolem bylo provést návrh systému pro řízení domácnosti řídicí jednotkou na bázi Raspberry Pi. Bylo provedeno srovnání jednotlivých možností pro řídicí jednotky a uvedeny důvody pro výběr minipočítače Raspberry Pi jako vhodného kandidáta. Obdobným způsobem bylo provedeno srovnání bezdrátových modulů a vybrán modul NodeMCU. Následně byla provedena analýza a definování funkcionality inteligentního systému. Byly zdefinovány potřebné senzorické moduly pro měření fyzikálních dat v domácnosti, provedena analýza a následná implementace. Také byly zdefinovány, zanalyzovány a na implementovány moduly pro ovládání aktivních prvků domácnosti jako jsou okenní žaluzie a domácí světla.

V poslední části práce je provedena analýza inteligentního systému OpenHab a následně byla provedena implementace tohoto systému na řídicí jednotku Raspberry Pi. V této části byla také provedena základní konfigurace systému a připojení všech bezdrátových modulů k systému OpenHab. Dále bylo vytvořeno uživatelské rozhraní mapující jednotlivé části domácnosti a rozdělení bezdrátových modulů do patřičných skupin pro jednoduchou správu a monitoring. Vytvořili jsme taktéž pravidlo pro automatické spouštění světla napojeného na detektor pohybu.

Navržený a implementovaný systém splňuje všechny funkční i nefunkční požadavky definované na tento systém. Existuje ale stále mnoho dalších domácích prvků, které by bylo možné do inteligentního systému přidat. Mezi dalšími nápady pro rozšiřování tohoto inteligentního systémů jsou:

- IR modul pro ovládání televize a domácího audio systému
- modul pro ovládání domácích ventilátorů
- implementace hlasového ovládání
- detekce a sledování domácích mazlíčků

Současná implementace inteligentního systému může ve své funkčnosti konkurovat komerčním řešením. I když je navrhované řešení postaveno jako nízko nákladové, kopíruje základní funkcionalitu komerčních řešení a umožňuje uživateli ovládat základní prvky domácnosti. Výhodou systému je univerzální kompatibilita jak s deskopovými tak mobilními zařízeními. Současně je řešení snadno rozšiřitelné a je možné zbývající domácí

aktivní prvky zapojit do inteligentního systému. Na rozdíl od většiny komerčních produktů je toto řešení postaveno na nové technologii MQTT, která je koncipována jako univerzální protokol pro komunikaci v Internetu věcí. Je tedy očekávána kompatibilita s novými zařízeními pro Internet věcí, které tak bude možné jednoduše zapojit do inteligentního systému.

SEZNAM POUŽITÉ LITERATURY

- [1] HRBÁČEK, Martin. *Inteligentní rodinný dům IV*. [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://bit.ly/24ndolw>.
- [2] MICHÁLEK, Libor. *Komplexní zabezpečení objektů*. [Online] 2011. [cit: 2016-05-03]. Dostupné z WWW: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=40017.
- [3] *Loxone chytrý dům* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.loxone.com/cscz/chytry-dum/prehled.html>.
- [4] *Loxone chytrý dům - Miniserver* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.loxone.com/cscz/produkty/miniserver/miniserver.html>.
- [5] *Loxone chytrý dům - vzorová kalkulace* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.loxone.com/cscz/koupit/vzorova-kalkulace.html>.
- [6] *Systém Fibaro* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.fibaro.com/cz>.
- [7] *Z-Wave* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://en.wikipedia.org/wiki/Z-Wave>.
- [8] *Systém Control4* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.control4.cz/>.
- [9] *ZigBee* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://cs.wikipedia.org/wiki/ZigBee>.
- [10] *ZigBee* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://cs.wikipedia.org/wiki/ZigBee>.
- [11] *Personal area network Wikipedia.org* [Online]. [cit: 2016-05-03]. Dostupné z WWW: https://en.wikipedia.org/wiki/Personal_area_network.
- [12] *WiFi Wikipedia.org* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://cs.wikipedia.org/wiki/Wi-Fi>.
- [13] *Wireless WAN Wikipedia.org* [Online]. [cit: 2016-05-03]. Dostupné z WWW: https://en.wikipedia.org/wiki/Wireless_WAN.
- [14] *IEEE 802.16 Wikipedia.org* [Online]. [cit: 2016-05-03]. Dostupné z WWW: https://en.wikipedia.org/wiki/IEEE_802.16.

- [15] *Global System for Mobile Communications Wikipedia.org* [Online].[cit: 2016-05-03]. Dostupné z WWW: https://cs.wikipedia.org/wiki/Global_System_for_Mobile_Communications.
- [16] *Long Term Evolution Wikipedia.org* [Online].[cit: 2016-05-03]. Dostupné z WWW: https://cs.wikipedia.org/wiki/Long_Term_Evolution.
- [17] *Standard IEEE 802.11 Wikipedia.org* [Online].[cit: 2016-05-03]. Dostupné z WWW: https://cs.wikipedia.org/wiki/IEEE_802.11.
- [18] *NETIA WiFi (WLAN)* [Online].[cit: 2016-05-03]. Dostupné z WWW: http://biznes.netia.pl/netia_dla_korporacji,rozwiwania,324,325,21669,0,netia-wifi-wlan.html.
- [19] *Fyzická vrstva Wi-Fi* [Online].[cit: 2016-05-03]. Dostupné z WWW: http://pandatron.cz/?699&fyzicka_vrstva_wi-fi&mobile=1.
- [20] *IEEE: 802.11a - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification: High Speed Physical Layer in the 5 GHz Band* [online]. [cit: 2016-05-03]. Dostupné z WWW: <http://standards.ieee.org/getieee802/802.11.html>.
- [21] *IEEE: 802.11b - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification: High Speed Physical Layer Extension in the 2,4 GHz Band* [online]. [cit: 2016-05-03]. Dostupné z WWW: <http://standards.ieee.org/getieee802/802.11.html>.
- [22] *IEEE: 802.11g - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification: Further Higher Data Rate Extension in the 2, 4 GHz Band* [online].[cit: 2016-05-03]. Dostupné z WWW: <http://standards.ieee.org/getieee802/802.11.html>.
- [23] *802.11 WLAN Packets and Protocols* [Online].[cit: 2016-05-03]. Dostupné z WWW: http://www.wildpackets.com/resources/compendium/wireless_lan/wlan_packets.
- [24] *IEEE 802.11b - Operation Modes* [Online].[cit: 2016-05-03]. Dostupné z WWW: <http://www.fedu.uec.ac.jp/~thavisak/Tech-Link/IEEE802.11b/IEEE80211b.htm>.
- [25] *Linková vrstva Wikipedia.org* [Online].[cit: 2016-05-03]. Dostupné z WWW: https://cs.wikipedia.org/wiki/Linkov%C3%A1_vrstva.
- [26] *Linková vrstva Wikipedia.org* [Online].[cit: 2016-05-03]. Dostupné z WWW: https://cs.wikipedia.org/wiki/Fyzick%C3%A1_vrstva.

- [27] *Fyzická vrstva Wi-Fi* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://access.feld.cvut.cz/view.php?cisloclanku=2008050006>.
- [28] *LEJTNAR, Michal., Komparativní analýza ad hoc a direct spojení v bezdrátových sítích* [Online]. [cit: 2016-05-03]. Dostupné z WWW: https://theses.cz/id/1t8516/Lejtnar_BP.pdf.
- [29] *THE INTERNET OF THINGS EXPLAINED* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.i-scoop.eu/internet-of-things/>.
- [30] *How the Internet of Everything Will Change the World* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://blogs.cisco.com/digital/how-the-internet-of-everything-will-change-the-worldfor-the-better-infographic>.
- [31] *MQTT 101* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.hivemq.com/blog/how-to-get-started-with-mqtt>.
- [32] *Android Debug Bridge* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://developer.android.com/studio/command-line/adb.html>.
- [33] *Raspberry Pi Wikipedia.org* [Online]. [cit: 2016-05-03]. Dostupné z WWW: http://cs.wikipedia.org/wiki/Raspberry_Pi.
- [34] *Raspberry Pi Products* [Online]. Dostupné z WWW: <https://www.raspberrypi.org/products/>.
- [35] *GPIO* [Online]. Dostupné z WWW: <https://ms-iot.github.io/content/en-US/win10/samples/PinMappingsRPi2.htm>.
- [36] *Arduino* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://cs.wikipedia.org/wiki/Arduino>.
- [37] *Arduino Uno* [Online]. Dostupné z WWW: <http://richarduino.com/specification-arduino-uno/>.
- [38] *Arduino Due* [Online]. Dostupné z WWW: <https://www.arduino.cc/en/Main/ArduinoBoardDue>.
- [39] *Arduino Shields* [Online]. Dostupné z WWW: <http://www.hobbyist.co.nz/?q=web-switch-tutorial>.
- [40] *NodeMCU v1* [Online]. Dostupné z WWW: <http://cktplus.com/NodeMCU-Lua-WiFi-Internet-of-Things-IoT-Development-Board>.

- [41] *NodeMCU v2* [Online]. Dostupné z WWW: <http://www.seeedstudio.com/depot/NodeMCU-v2-Lua-based-ESP8266-development-kit-p-2415.html>.
- [42] *NodeMCU GPIO* [Online]. Dostupné z WWW: <https://github.com/nodemcu/nodemcu-devkit>.
- [43] *SDK* [Online]. [cit: 2016-05-03]. Dostupné z WWW: https://cs.wikipedia.org/wiki/Software_development_kit.
- [44] *Pasivní infračervený senzor* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://senzory.aspone.cz/pir.htm>.
- [45] *Generování impulsu při detekci pohybu* [Online]. Dostupné z WWW: <http://vyvoj.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>.
- [46] *Specifikace senzoru HC-SR501* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://www.mpja.com/download/31227sc.pdf>.
- [47] *Specifikace senzoru HC-SR501* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://robotstore.cz/obchod/arduino/hc-sr501-pir-senzor-detekce-pohybu-arduino-modul/>.
- [48] *Milight WiFi Box* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.milight.com/milight-wifi-receiver-bridge-3-0-controller-box/>.
- [49] *Milight Light Bulb* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.milight.com/milight-5w-rgbw-e14-screw-iphone-ios-android-controlled-light/>.
- [50] *Phillips Hue Bridge* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www2.meethue.com/en-gb/productdetail/philips-hue-bridge>.
- [51] *Phillips Hue Light Bulb* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www2.meethue.com/en-gb/productdetail/philips-hue-white-extension-bulb-a19>.
- [52] *LIFX products* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://www.lifx.com/products>.
- [53] *GitHub python-milight* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://github.com/McSwindler/python-milight>.
- [54] *System horizontálních žaluzií* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://zaluzie.tymos.cz/dily.html>.
- [55] *Infračervené LED* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://goo.gl/ZuvUZ0>.

-
- [56] *Github RPi_Cam_Web_Interface* [Online]. [cit: 2016-05-03]. Dostupné z WWW: https://github.com/silvanmelchior/RPi_Cam_Web_Interface.
- [57] *Access control list* [Online]. [cit: 2016-05-03]. Dostupné z WWW: https://cs.wikipedia.org/wiki/Access_control_list.
- [58] *OpenHab security* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://github.com/openhab/openhab/wiki/Security>.
- [59] *OpenHab configuration* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://github.com/openhab/openhab/wiki/Configuring-the-openHAB-runtime>.
- [60] *OpenHab Rule Designer* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <https://github.com/cdjackson/HABmin/wiki/Rule-Designer:-Overview>.
- [61] *Mosquitto message broker* [Online]. [cit: 2016-05-03]. Dostupné z WWW: <http://mosquitto.org/>.

SEZNAM OBRÁZKŮ

2.1	Loxone Miniserver a Miniserger GO	19
2.2	Loxone Air produktová řada	21
2.3	Řídící jednotka Fibaro	22
2.4	Fibaro produkty	23
2.5	Řídící jednotka Control4 HC-250	24
2.6	Control4 dotykové panely	25
2.7	Bezdrátový Music Bridge	25
2.8	Bezdrátová klávesnice pro ovládání světla	26
2.9	Nyce produkty	27
3.1	Grafická reprezentace Wi-Fi kanálů v pásmu 2.4 GHz[17]	30
3.2	Schéma komunikace na WiFi[18]	31
3.3	Referenční model ISO/OSI pro Wi-Fi[19]	31
3.4	Porovnání MAC hlaviček u WiFi a Ethernetu[23]	34
3.5	Porovnání Ad-hoc, BSS a ESS[24]	35
3.6	Růst komunikace internetu věcí[29]	36
3.7	Komunikace senzoru a klienta pomocí MQTT[31]	38
6.1	Návrh architektury inteligentní domácnosti	44
7.1	Raspberry Pi (Model A)[34]	45
7.2	Raspberry Pi 2[34]	46
7.3	Raspberry Pi 3[34]	46
7.4	GPIO [35]	47
7.5	Arduino Uno [37]	49
7.6	Arduino Due	50
7.7	Arduino Shields [39]	51
7.8	NodeMCU v1 [40]	52
7.9	NodeMCU v2 [41]	53
7.10	NodeMCU GPIO [42]	54
8.1	Propojení pinů RST a D0 u NodeMCU - funkce dsleep()	57
8.2	Senzor DS18B20	58
8.3	Senzor DS18B20	59
8.4	Senzor DS18B20	60
8.5	Kompletace bezdrátového modulu za použití desky plošných spojů	60
8.6	Testovací bezdrátový modul bez baterie a s baterií	60
8.7	Senzor DHT-11 a DHT-22	62
8.8	Senzor DHT-11	63
8.9	Senzor DHT-11	63

8.10	Kompletace bezdrátového modulu za použití desky plošných spojů	64
8.11	Testovací bezdrátový modul bez baterie a s baterií	64
8.12	Senzor Produino LM393	66
8.13	NodeMCU a Produino senzor	66
8.14	Schema zapojení NodeMCU a Produino LM393	67
8.15	Kompletace bezdrátového modulu za použití desky plošných spojů	67
8.16	Bezdrátový modul měřící vlhkost v květináči kaktusu	68
8.17	Generování impulsu při detekci pohybu[45]	68
8.18	Senzor HC-SR501[47]	69
8.19	NodeMCU a HC-SR501	69
8.20	Schema zapojení NodeMCU a HC-SR501	69
8.21	Produkty Milight - WiFi box[48] a žárovka[49]	72
8.22	Produkty Phillips Hue - WiFi bridge[50] a žárovka[51]	73
8.23	Žárovky LIFX [52]	74
8.24	Systém horizontálních žaluzií[54]	78
8.25	elektromotor MOTS1	79
8.26	NodeMCU a elektromotor MOTS1 12V	80
8.27	Schema zapojení NodeMCU a elektromotoru MOTS1 12V	80
9.1	Raspberry Pi Camera modul [55]	83
9.2	Instalace kamery k Raspberry Pi	83
9.3	Test kamery	84
9.4	Kalendář webového rozhraní kamery Raspberry Pi	84
9.5	Webového rozhraní RPi Cam Control	85
9.6	Nastavení mobilní aplikace IP Cam Viewer Lite	86
10.1	Blokové schéma architektury OpenHab	87
10.2	Komunikace vazby Milight s externím zařízením	89
10.3	OpenHab Designer	95
10.4	Rule Designer	95
10.5	OpenHab Demo soubory	98
10.6	OpenHab Demo běžící aplikace	99
10.7	OpenHab vytváření mapy - inicializace	100
10.8	OpenHab vytváření mapy - místnosti	100
10.9	OpenHab přidávání položek - ložnice	102
10.10	OpenHab přidávání položek - výběr barvy	102
10.11	OpenHab aplikace - obyvací pokoj	104
10.12	OpenHab aplikace 0 detektor pohybu	106

SEZNAM TABULEK

2.1	Přehled cen jednotlivých produktů Fibaro	23
2.2	Přehled cen jednotlivých produktů Nyce	26
2.3	Porovnání ceny řídicích jednotek	28
3.1	Přenosové mechanismy pro standard 802.11[19]	31
8.1	Syntaxe REST api - POST data	56
8.2	Specifikace DS18B20	58
8.3	Seznam součástí a vyčíslení ceny	59
8.4	Specifikace DHT-11 a DHT-22	62
8.5	Seznam součástí a vyčíslení ceny	62
8.6	Specifikace Produino LM393	65
8.7	Seznam součástí a vyčíslení ceny	66
8.8	Specifikace HC-SR501[46]	70
8.9	Seznam součástí a vyčíslení ceny	70
8.10	Nastavení barvy	76
8.11	Fukce pro ovládání žárovek typu White	76
8.12	Fukce pro ovládání žárovek typu RGB/GRBW	76
8.13	Specifikace MOTS1 12V[46]	79
8.14	Seznam součástí a vyčíslení ceny	81
8.15	Vykonání jednoho kroku po směru hodinových ručiček	81
9.1	Raspberry Pi Camera	82
10.1	Seznam podporovaných položek	91
10.2	Seznam podporovaných položek	91

SEZNAM PŘÍLOH

P I. Naprogramované aplikace a skripty

PŘÍLOHA P I. NAPROGRAMOVANÉ APLIKACE A SKRIPTY

```
1 function connect(ssid,pass)
2   wifi.sta.disconnect()
3   wifi.setmode(wifi.STATION);
4   wifi.sta.config(ssid,pass);
5   wifi.sta.connect()
6   tmr.delay(1000000)   -- wait 1,000,000 us = 1 second
7   print(wifi.sta.status())
8   return wifi.sta.getip()
9 end
10 --main
11 file.open("wifi.config", "r")
12 ssid = file.readline()
13 pass = file.readline()
14 file.close()
15 local i = 0
16 repeat
17   ip = connect(ssid,pass)
18   i = i + 1
19 until (ip ~= nil or i>5)
20 if ip == nil
21 then
22   dofile("sleep.lua")
23 end
```

Code 1.1 wifi.lua

```
1 -- send_data.lua
2 function get_modul_id()
3   return node.chipid()
4 end
5 -- vycte IP adresu serveru ze souboru "server.config"
6 function get_server_ip()
7   file.open("server.config", "r")
8   ip = file.readline()
9   file.close()
10  return ip
11 end
12 function sendData(data,type)
13   server_ip = get_server_ip()
14   modul_id = get_modul_id()
15   -- vytvor spojeni se serverem
16   conn=net.createConnection(net.TCP, 0)
17   conn:on("receive", function(conn, payload) print(payload) end)
18   conn:connect(80,server_ip)
19   -- vytvoreni POST requestu
20   post = "POST /api/v1/data?modulId="..modul_id.."&dataType="..type.."&data
21         ="..data.." HTTP/1.1\r\nHost:"..server_ip.." \r\nAccept: */*\r\nUser-
22         Agent: Mozilla/4.0 (compatible; esp8266 Lua; Windows NT 5.1)\r\n\r\n"
23   -- odeslani POST requestu
24   conn:send(post)
25   conn:on("sent",function(conn)
26     conn:close()
27   end)
28 end
```

Code 1.2 send_data.lua

```

1 -- cas je udavan v mikro sekundach, v tomto pripade 600 sec = 1 min
2 sleep_time = 600 * 1000 * 1000
3 node.ds.sleep(sleep_time)

```

Code 1.3 sleep.lua

```

1 -- temp_ds18b20.lua
2 require('ds18b20')
3 require('data_send')
4 function get_temp(gpio)
5     -- nastav vycitaci pin
6     ds18b20.setup(gpio2)
7     -- ziskej adresu pripojeneho senzoru
8     addrs=ds18b20.addrs()
9     -- vycti teplotu ze senzoru
10    t=ds18b20.read(addrs[1], ds18b20.C)
11    -- pokud se teplota nepodari spravne vycist, metoda trm.alarm otestuje
    ziskanou hodnotu a v pripade chyby nacte hodnotu znovu
12    tmr.alarm (2, 500, 1, function ()
13        if (t==nil) then
14            t=ds18b20.read(addrs[1], ds18b20.C)
15        else
16            tmr.stop (2)
17            return t
18        end
19    end)
20    return t
21 end
22 --main
23 temp = get_temp(4)
24 send_data(temp,"temperature")

```

Code 1.4 temp_ds18b20.lua

```

1 -- temp_dht11.lua
2 require('data_send')
3 pin = 4
4 status,temp,humi,temp_decimial,humi_decimial = dht.read(pin)
5 if( status == dht.OK )
6 then
7     print("DHT Temperature: "..temp.." ; ".."Humidity:"..humi)
8 elseif( status == dht.ERROR_CHECKSUM )
9 then
10    print( "DHT Checksum error." );
11 elseif( status == dht.ERROR_TIMEOUT )
12 then
13    print( "DHT Time out." );
14 end
15 --main
16 send_data(temp.." ; "..humi,"temp_hum")

```

Code 1.5 temp_dht11.lua

```

1 require(data_send)
2 value = adc.read(0)
3 -- pouze informativni rozdeleni, o vysledne se stara centralni server
4 if (value > 400)
5 then
6   print("nizka vlhkost: "..value)
7 elseif (value >250)
8 then
9   print("stredni vlhkost: "..value)
10 else
11   print("vysoka vlhkost: "..value)
12 end
13 -- odesilame data typu "moisture" v cistem nemapovanem stavu
14 send_data(value,"moisture")

```

Code 1.6 moisture.lua

```

1 require("data_send")
2 pin = 4
3 gpio.mode(pin,gpio.INT,gpio.PULLUP)
4 function motion()
5   send_data(1,"motion")
6 end
7 gpio.trig(pin, "up",motion)

```

Code 1.7 motion.lua

```

1 --motor.lua
2 -- pouzivame piny D1, D2, D3 a D4
3 pins = {1,2,3,4}
4 --id pro timer
5 timer = 2
6 step_states = {
7   {0,0,0,1},
8   {0,0,1,1},
9   {0,0,1,0},
10  {0,1,1,0},
11  {0,1,0,0},
12  {1,1,0,0},
13  {1,0,0,0},
14  {1,0,0,1},
15 }
16 -- pocet radku matice step_states
17 num_states = 8
18 -- posledni pouzity radek z matice step_states
19 step_state = 0
20 -- zpozdeni v ms mezi jednotlivymi kroky
21 delay = 10
22 -- smer otaceni: 1 po smeru, -1 proti smeru hod. rucicek
23 direction = 1
24 -- kolik kroku jeste mame vykonat
25 stepsleft = 0
26 function make_step()
27   -- uprava radku matice dle nastaveného smeru otaceni
28   step_state = step_state + direction
29   if step_state > num_states then
30     step_state = 1;
31   elseif step_state < 1 then
32     step_state = num_states
33   end
34   -- vykonani jednoho kroku
35   for i = 1, 4, 1 do
36     gpio.write(stepper_pins[i], step_states[step_state][i])
37   end
38   -- kontrola zda mame mame vykonat jeste nejake kroky
39   stepsleft = stepsleft-1
40   if stepsleft > 0 then
41     tmr.alarm(timer, delay, 0, make_step )
42   else
43     step_stopstate()
44   end
45 end
46 function step_move(steps, direction, delay)
47   tmr.stop(timer)
48   -- chceme aby jsme vykonali vzdy celou jednu otacku o 1/64
49   stepsleft = steps * 8
50   direction = direction
51   delay = delay
52   make_step()
53   stop()
54 end
55 --funkce pro zastaveni otaceni a uvedeni motorku do klidoveho stavu
56 function stop()
57   tmr.stop(timer)
58   stepsleft = 0
59   -- nastaveni 0 na vsehch pouzivanych pinech
60   for i = 1, 4, 1 do
61     gpio.write(stepper_pins[i], 0)
62   end
63 end
64 --main
65 --priklad pouziti funkce - plne otoceni hridele po smeru hod rucicek za
66   7,680 sec
67 step_move(64, 1, 15)

```

Code 1.8 motor.lua