



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**BEZDRÁTOVÁ KOMUNIKACE MEZI DOMÁCÍ ELEK-
TRONIKOU**

WIRELESS COMMUNICATION FOR HOME AUTOMATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VÁCLAV SEZIMA

VEDOUcí PRÁCE

SUPERVISOR

Doc. Dr. Ing. PETR HANÁČEK

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Sezima Václav**

Obor: Informační technologie

Téma: **Bezdrátová komunikace mezi domácí elektronikou
Wireless Communication for Home Automation**

Kategorie: Počítačové sítě

Pokyny:

1. Prostudujte dostupné materiály o problematice domácích komunikačních sítí.
2. Navrhněte systém pro dálkové bezdrátové řízení domácích spotřebičů pomocí osobního počítače, PDA nebo mobilního telefonu. Kritériem je cenová náročnost, funkcionality a dosah řešení. Jako příklad využití můžeme uvést dálkové ovládání audiovizuální techniky apod.
3. Vyberte vhodnou dostupnou technologii pro implementaci ovládání spotřebičů (Wifi, Bluetooth, IR či jiné).
4. Navržené řešení implementujte. Vytvořte podrobnou technickou dokumentaci řešení.
5. Zhodnoťte práci, nastiňte budoucí vývoj.

Literatura:

- Podle pokynů vedoucího

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Hanáček Petr, doc. Dr. Ing.**, UITS FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá problematikou bezdrátové komunikace mezi přístroji a senzory v inteligentní domácnosti. Cílem je návrh a implementace zařízení zvané brána, které tvoří uzel pro posílání zpráv a příkazů mezi zařízeními v domácnosti a kontrolérem, který slouží k jejich ovládání z centrálního místa, například počítače nebo mobilu. Součástí práce je také vytvořit několik senzorů pro testovací a demonstrační účely. Brána a senzory jsou postavené na vývojových deskách WEMOS a Arduino a podporují několik různých bezdrátových technologií pro komunikaci mezi sebou a kontrolérem.

Abstract

This bachelor's thesis deals with the issue of wireless communication among devices and sensors for home automation. The goal is to design and implement a device known as the gateway that is passing messages and commands between sensors and controller that controls them from central point, e.g. computer or mobile phone. A part of this thesis is also a creation of several sensors for the purpose of testing and demonstration. The gateway and sensors are based on development boards WEMOS and Arduino and supports multiple wireless technologies for communication with among themselves and controller.

Klíčová slova

Inteligentní domácnost, bezdrátová komunikace, WEMOS, ESP32, Arduino, radiokomunikace, infračervená komunikace, WiFi, MQTT

Keywords

Home automation, wireless communication, WEMOS, ESP32, Arduino, radiocommunication, infrared communication, WiFi, MQTT

Citace

SEZIMA, Václav. *Bezdrátová komunikace mezi domácí elektronikou*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Dr. Ing. Petr Hanáček

Bezdrátová komunikace mezi domácí elektronikou

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Dr. Ing. Petra Hanáčka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Václav Sezima
15. května 2018

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce, panu doc. Dr. Ing. Petru Hanáčkoví, za odborné vedení, rady a připomínky při tvoření této práce.

Obsah

1	Úvod	4
2	Teorie	5
2.1	Inteligentní domácnost	5
2.1.1	Teorie senzorů	6
2.1.2	Teorie kontroléru	7
2.1.3	Teorie brány	7
2.2	Existující řešení	8
2.2.1	Open-source řešení	8
2.2.2	Komerční řešení	10
3	Návrh řešení	12
3.1	Vývojové desky	12
3.1.1	Deska Arduino Nano	12
3.1.2	Deska WEMOS LOLIN32	14
3.2	Senzorové moduly	15
3.2.1	Teplotní čidlo DS18B20	15
3.2.2	Detektor pohybu HC-SR501	16
3.2.3	Relé JQC-3FF-S-Z	16
3.3	Bezdrátové technologie	17
3.3.1	Rádiová komunikace	17
3.3.2	Infračervená komunikace	19
3.3.3	WiFi + MQTT	21
4	Implementace řešení	23
4.1	Hardware	23
4.1.1	Zapojení senzorů	24
4.1.2	Zapojení brány	27
4.2	Firmware	28
4.2.1	Kód Arduino Nano	28
4.2.2	Kód WEMOS LOLIN32	30
5	Testování	33
5.1	Dosah komunikace	33
5.2	Spolehlivost komunikace	34
5.3	Kontrolér openHAB	35
6	Závěr	37

Literatura	38
Přílohy	41
A Obsah CD	42

Seznam obrázků

2.1	Obecné schéma zapojení a komunikace zařízení v inteligentní domácnosti . . .	5
2.2	Topologie sítí	6
3.1	Arduino UNO (převzato z [5])	13
3.2	Arduino Nano (klon)	13
3.3	WEMOS LOLIN32	14
3.4	Teplotní čidlo DS18B20	15
3.5	PIR motion (piny a čidlo)	16
3.6	PIR motion (potenciometry a pouzdro)	16
3.7	Relé štít	17
3.8	Rádio 2,4 GHz tranceiver nRF24L01+	18
3.9	Rádio 433 MHz přijímač SRX882 (vlevo) a vysílač STX882 (vpravo)	19
3.10	Formát zprávy podle NEC protokolu (převzato z [18])	20
3.11	Příklad dvou opakovacích kódu po poslané zprávě NEC protokolem (převzato z [18])	21
4.1	Schéma zapojení implementovaného řešení	23
4.2	Zapojení senzoru s teplotním čidlem a nRF24L01+ modulem	24
4.3	Zapojení senzoru s detektorem pohybu a nRF24L01+ modulem	25
4.4	Zapojení senzoru s relé štítem a nRF24L01+ modulem	25
4.5	Zapojení senzoru s relé štítem a moduly STX882 a SRX882	26
4.6	Zapojení senzoru s relé štítem a moduly STX882 a SRX882	26
4.7	Zapojení senzoru s relé štítem a moduly STX882 a SRX882	27
4.8	Zapojení brány s komunikačními moduly	28
5.1	openHAB - Basic UI	35
5.2	openHAB - HABPanel	36

Kapitola 1

Úvod

Žijeme v digitální době, kdy nás obklopuje čím dál více elektroniky, která nám má usnadnit náš život. Dříve to bylo pouze v práci, pro větší efektivnost a rychlost. V poslední době se však taková chytrá elektronika dostává i do našich domovů, aby nám umožnila zautomatizovat běžné denní úkony a my tak měli více volného času a méně námahy. Proto se také tomuto procesu říká domácí automatizace, díky kterému můžeme ovládat velkou část domu z pohodlí gauče pomocí chytrého mobilu nebo dokonce nastavit zařízení tak, aby sama reagovala na nějaké vnější podněty. Taková domácnost se pak nazývá „chytrá“ nebo „inteligentní“. Co tento pojem znamená a jaká existují v současnosti řešení je uvedeno v kapitole **2 Teorie**.

V praxi je taková inteligentní domácnost tvořena zpravidla třemi prvky: senzory, bránou a kontrolérem. Sensory označují koncová zařízení, která vykonávají nějakou činnost a mají se dát ovládat. Kontrolér na druhé straně je řídicí program poskytující uživatelské rozhraní pro snadné a přehledné ovládání těchto zařízení a zobrazení jejich dat. Mezi nimi pak stojí brána, která se chová jako takový prostředník a jejím úkolem je směřovat data mezi senzory a kontrolérem v obou směrech.

Tato práce se zabývá vytvořením právě takové brány spolu s několika jednoduchými senzory. Kapitola **3 Návrh řešení** popisuje, z čeho se tyto prvky skládají a jak komunikují mezi sebou a kontrolérem. Následně pak v kapitole **4 Implementace řešení** je uvedena jejich realizace jak po hardwarové, tak softwarové stránce.

Mezi hlavní parametry brány patří podpora množství komunikačních technologií a senzorů, se kterými dokáže komunikovat. Co je ale důležité, je spolehlivost takové komunikace a samozřejmě její dosah. Hlavně tyto vlastnosti jsou otestovány v kapitole **5 Testování**.

Nakonec kapitola **6 Závěr** zhodnocuje výsledné řešení a provedení, jak po fyzické, tak programové stránce. Zkoumány jsou výsledky z testování, cenová náročnost brány a možná budoucí rozšíření.

Kapitola 2

Teorie

Tato část se zabývá vysvětlením teorie a pojmů z oblasti inteligentní domácnosti. V podkapitole **2.1 Inteligentní domácnost** je popsán koncept inteligentní domácnosti, čím je tvořena a co jednotlivé části dělají. Dále v podkapitole **2.2 Existující řešení** je uveden současný stav existujících řešení, na jaké druhy se dělí, čím se liší a jaké jsou jejich výhody a nevýhody.

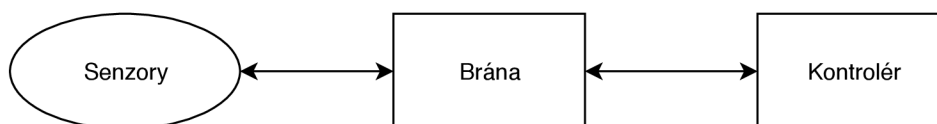
2.1 Inteligentní domácnost

Pojem inteligentní domácnost (angl. Smart Home) není žádnou novinkou. V minulosti se tak označoval dům vybavený nějakým „chytrým“ zařízením, které nahrazovalo a automatizovalo určitou manuální činnost. Dříve se takovými zařízeními myslely například pračky nebo myčky nádobí, které dnes považujeme za samozřejmost. Takže tento pojem je s námi již dlouho, co se ale mění, je jeho význam, respektive, co všechno pod něj spadá.

V dnešní době vzniká mnoho takových „chytrých“ zařízení. Aby ale byl také dům „inteligentní“, nestačí ho jenom naplnit takovými přístroji a nechat je, aby si sami dělali pouze svoji práci. Ta inteligentní část totiž spočívá právě v interakci mezi těmito přístroji a také samotným domem. Nejčastějším příkladem inteligentní domácnosti totiž bývá právě ovládání jednotlivých částí domu, ať už například osvětlení, otevírání/zavírání oken a dveří nebo třeba ovládání teploty pomocí termostatu. A všechno toto se děje na dálku, ať už pomocí chytrého mobilního telefonu nebo v poslední době i hlasové aktivace, například s použitím Amazon Echo nebo Google Home.

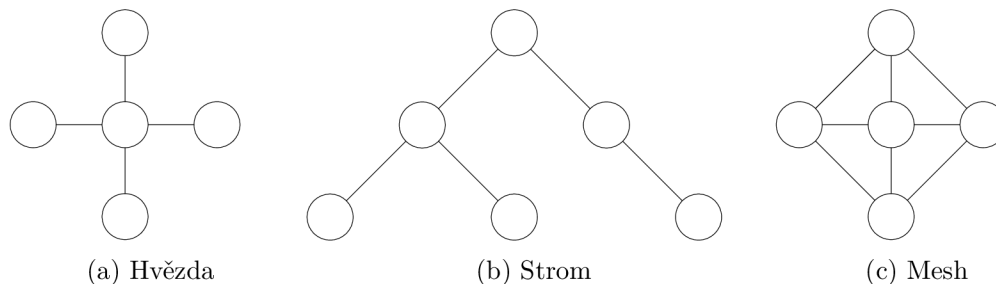
Inteligentní domácnost je tedy tvořena takovou sítí navzájem propojeným a komunikujícími zařízeními. A tato síť se zpravidla skládá z těchto tří prvků:

- Sensory - což jsou koncová zařízení,
- Kontrolér - který tvoří mozek sítě a slouží pro ovládání a zobrazování dat od sensorů,
- Brána - která stojí mezi senzory a kontrolérem a směruje data.



Obrázek 2.1: Obecné schéma zapojení a komunikace zařízení v inteligentní domácnosti

Sítě se dále dělí na různé typy podle topologií, tedy jak jsou daná zařízení propojena. Taková základní topologie je hvězdice (viz obr. 2.2a), kde jedno zařízení uprostřed sítě tvoří přístupový bod, ke kterému se ostatní zařízení připojují a komunikace probíhá vždy přes tento centrální prvek. Hvězdicové topologie jsou pak použity v tzv. stromové topologii (viz obr. 2.2b), kdy prostřední přístupové body jednotlivých hvězdicových topologií jsou propojeny do tvaru připomínající strom. Další častou topologií je tzv. mesh (viz obr. 2.2c), kdy všechna zařízení jsou propojena stylem každý s každým. V této topologii jsou si všechna zařízení rovna v tom smyslu, že nemají nějakou určenou roli; jsou tak na stejné úrovni a komunikují spolu navzájem.



Obrázek 2.2: Topologie sítí

V této práci je použita pouze hvězdicová topologie, ale za to několikrát. Protože brána podporuje několik různých komunikačních technologií, tvoří tak několik samostatných hvězdic, pro každou technologii jednu, vždy s bránou uprostřed jako přístupovým bodem. Navíc i celá inteligentní domácnost je takovou hvězdicovou topologií, s bránou také jako centrálním prvkem a zbytek tvoří senzory a kontrolér.

2.1.1 Teorie senzorů

Jak už bylo zmíněno, senzory jsou souhrnným názvem pro všechna koncová zařízení patřící do inteligentní domácnosti. A nejde jenom o běžné spotřebiče jako televize nebo pračka, ale lze sem zařadit v podstatě jakékoliv zařízení, které se dokáže připojit k síti inteligentní domácnosti a přijímat/posílat/zpracovávat data. K tomu je zapotřebí nějaký procesor pro práci s daty a komunikační modul pro zajištění komunikace s bránou. Oba tyto prvky lze dnes sehnat v malé a levné podobě, a proto je možné udělat téměř jakékoliv zařízení „inteligentní.“

Z tohoto důvodu ale pojem senzory zahrnuje spoustu velice různých zařízení, která tak můžeme rozdělit do několika následujících kategorií podle toho, s jakými daty pracují:

- Měřicí senzory - slouží k měření nějaké fyzikální veličiny, např. teploty, vlhkosti vzduchu, slunečního záření apod.,
- Bezpečnostní zařízení - pro zajištění ochrany domácnosti. Sem patří třeba kamery, detektory pohybu a elektrické zámky na dveřích,
- Domácí spotřebiče - jako je televize, domácí kino, audio systém nebo také chytré pračky a lednice,
- Ostatní - v podstatě jakékoliv zařízení, které se dokáže připojit do sítě chytré domácnosti a posílat/přijímat data, což mohou být třeba další počítače, servery nebo dokonce i roboti.

Dále můžeme senzory dělit podle jejich způsobu připojení do sítě. Tady jsou možné dva přístupy, drátově nebo bezdrátově. Při drátové komunikaci jsou tak senzory připojeny do sítě přímo pomocí nějakého kabelu vedeného do routeru, serveru, brány nebo jiného zařízení, které slouží jako přístupový bod do sítě. Výhodou takového zapojení je pak spolehlivost stabilního připojení a v závislosti na použitém kabelu a rozhraní může být komunikace i rychlejší. Nevýhodou je pak hlavně nedostatek mobility, protože kabely mají fixní délku, která jednak omezuje dosah, a pokud jsou vedeny ve stěnách, tak s nimi nejde hýbat téměř vůbec. To samozřejmě nemusí být nevýhodou pro všechna zařízení, jako například kamery nebo různé měřicí přístroje, jejichž místo působení se nemění.

Bezdrátové komunikace řeší tyto nevýhody kabelů, tedy zajištění pohybu a zvětšení vzdálenosti, na kterou může zařízení komunikovat se sítí. Co se ale týče ostatních vlastností, to už záleží na jednotlivých bezdrátových technologiích, neboť jich je mnoho a fungují na různých principech, a tak mají i různé silné a slabé stránky. Mezi nejpoužívanější technologie patří rádiové vlny, které se dělí na několik frekvenčních pásem, a infračervené záření, které lze najít například v dálkových ovladačích. Podrobněji jsou tyto technologie popsány v kapitole **3.3 Bezdrátové technologie**, kde je také uvedeno, jak jsou použity v této práci.

2.1.2 Teorie kontroléru

Kontrolér funguje jako jakási řídicí jednotka celého systému inteligentní domácnosti. Zpravidla jde o komplexní program, který lze nainstalovat na dedikovaný počítač nebo server. Dokonce tak nemusí být kontrolér ani ve stejné síti, ale někde na cloudu, a se zbytkem chytré domácnosti je pak spojen přes Internet. V popředí poskytuje uživatelské rozhraní pro ovládání senzorů a zobrazení jejich dat a v pozadí se stará o komunikaci se senzory a posílání/přijímání dat. Můžou existovat i fyzická řešení, kdy má kontrolér svůj vlastní hardware. Tato možnost bývá většinou součástí kompletních řešení chytrých domácností.

Při práci s daty v pozadí staví kontroléry na dvou základech: předmětech (angl. Items, Things, Devices) a pravidlech (angl. Rules, Scripts). Předměty představují senzory a drží si jejich aktuální stav, který je pak zobrazován v uživatelském rozhraní. Stěžejní jsou ale právě pravidla, která představují tu „inteligentní“ část domácnosti, neboť ony rozhodují, co mají senzory dělat na základě zadaných podmínek. Může tak například existovat pravidlo, že při snížení teploty se má v dané místnosti zapnout topení. Když tedy stav předmětu, který představuje teplotní senzor, spadne pod určitou hranici, kontrolér sám automaticky pošle příkaz jinému předmětu, který zastupuje topení, a v pozadí se pošle tento příkaz danému senzoru pomocí určené komunikační technologie.

Kontroléry jsou schopné udržet si stav předmětů i v případě výpadku nebo plánovaného vypnutí (tzv. persistence). Dále můžou také logovat historii těchto stavů a datum jejich změny pro případné další potřeby a použití, jako například tvoření grafů nebo řešení chyb.

Kontrolér může se senzory komunikovat přímo nebo přes bránu. Přímá komunikace bývá většinou u kompletních řešení, kdy jsou senzory a kontrolér od jednoho výrobce a fungují pouze spolu. Řešení s bránou se naopak hodí v případě, kdy je potřeba sloučit více senzorů od různých výrobců a nejsou navzájem kompatibilní.

2.1.3 Teorie brány

Jak bylo uvedeno výše, brána je zařízení stojící mezi senzory a kontrolérem. Jejím hlavním smyslem je být jakýsi prostředník mezi nekompatibilními senzory a kontroléry. Tato další úroveň abstrakce také umožňuje nezávislost senzorů na kontroléru a naopak. Lze tak snadno vyměnit kontrolér bez zásahu do senzorů; samozřejmě za předpokladu, že nový kontrolér

je schopný komunikovat s bránou. Stejně tak na druhé straně lze měnit senzory, pokud se dokážou dorozumět s bránou.

Brány lze tak hlavně rozlišovat podle množství podporovaných komunikačních technologií a senzorů a kontrolérů, se kterými dokáže komunikovat. Proto je brána tvořena jak hardwarovou částí, která se stará o komunikaci mezi zařízeními, tak i softwarem, který řídí tok dat a rozhoduje, kam data chodí.

2.2 Existující řešení

V současnosti existuje velice mnoho řešení, ať už senzorů, kontrolérů nebo bran. A také existuje množství kategorií, do kterých je lze rozdělit:

- Podle uzavřenosti - může být řešení otevřené (je použit otevřený hardware a software) nebo uzavřené (komerční řešení),
- Podle úplnosti - mohou obsahovat všechny prvky (tj. senzory, bránu i kontrolér) nebo se zaměřují pouze na vývoj jednoho typu zařízení,
- Podle komunikační technologie - s jakou technologií pracují nebo kolik jich podporují,
- Podle systému - na jakém systému běží (týká se kontrolérů), ať už Linux, macOS a Windows nebo mobilní Android a iOS.

Samozřejmě tyto kategorie se můžou navzájem prolínat a tvořit různé kombinace, např. otevřené neúplné řešení, podporující jednu technologii a ovládané pouze z Android aplikace. Zbytek této podkapitoly popisuje rozdíly hlavně mezi otevřenými a komerčními řešeními, protože to je takové základní a největší dělení.

2.2.1 Open-source řešení

Open-source neboli otevřená řešení používají volně dostupný hardware a software nebo jej sami vytvářejí a dovolují používat volně dál. Hlavní předností (po které se také nazývají) je právě otevřenost těchto řešení, tedy že v případě hardwaru jsou k dispozici schémata, diagramy, použité součástky a všechny další věci potřebné k vytvoření stejného výrobku. Stejně tak v případě otevřeného softwaru jsou dostupné zdrojové kódy a algoritmy. Hlavní výhodou takového řešení je tedy cena, protože není potřeba kupovat licenci a všechny zdroje jsou zadarmo. Samozřejmě v případě hardwaru je zde cena materiálu a součástek, ale samotná schémata a návrhy jsou k dispozici bezplatně. Z toho také vyplývá další výhoda, a to je možnost měnit a upravovat, čímž se může vzít takové existující řešení a specializovat jej pro vlastní potřeby.

Bohužel nejsou tato řešení perfektní a mají i své nedostatky. Všechny vychází z toho, že jsou open-source řešení vyvíjena otevřenou komunitou lidí, kteří to dělají dobrovolně ve svém volném čase, jelikož jsou tato řešení zadarmo pod bezplatnými licencemi, a tudíž nevydělávají peníze. Z toho tedy plyne například to, že tato řešení nemusí být důkladně otestována a podporována jako v případě komerčních řešení vyvíjených firmami. Dalším záporem je nepohodlnost, protože zdrojové podklady jsou sice zadarmo, ale je třeba vynaložit určité úsilí a námahu pro vytvoření a zprovoznění konečného funkčního řešení. Spolehlivost takového řešení nemusí být zaručena a jakékoliv aktualizace a rozšíření je třeba dělat znovu ručně.

Open-source řešení také často nepokrývají všechny prvky inteligentní domácnosti a zaměřují se na softwarovou část. Mezi takové řešení patří například **openHAB**[20], jeden z hlavních otevřených kontrolérů s velkou komunitou uživatelů a vývojářů. Podporuje instalaci na všechny hlavní systémy a nabízí uživatelské rozhraní pomocí jak webového prohlížeče, tak aplikacemi chytrých telefonů. openHAB je napsaný v Javě, což znamená, že pro svůj běh potřebuje přítomnou Javu v systému, na kterém má běžet. Má také svou vlastní linuxovou distribuci openHABian[23], což je systém postavený na distribuci Raspbian s již přítomným openHAB kontrolérem pro jednoduchou instalaci a je určený pro platformu Raspberry Pi nebo Pine A64. Pro psaní pravidel pak openHAB používá vlastní jazyk založený na syntaxi jazyka Xtent (dříve Xbase)[24].

Předností openHAB je také velká podpora různých technologií pomocí tzv. vazeb (angl. Bindings[21]), například Samsung, LG, Panasonic TV, pro ovládání chytrých televizí od těchto výrobců, nebo ZigBee a Z-Wave, pro ovládání zařízení komunikujících přes tyto protokoly, které jsou často používány v komerčních řešení a tímto způsobem je lze ovládat přímo bez potřeby specializovaných kontrolérů a bran. Tyto vazby jsou defaultně vypnuté a musí se povolit, pokud se chce s danou technologií komunikovat a pracovat.

Jednou z obtížnějších věcí je přidávání a nastavování nových zařízení do openHABu. Původně byl systém vyvinut pro správu přes konfigurační soubory a tato možnost je primárním způsobem doposud. I když existuje možnost správy zařízení přes webový prohlížeč, tato funkce je ještě ve vývoji a neposkytuje všechny možnosti jako konfigurační soubory[22]. Proto zvláště pro začátečníky je těžší openHAB poprvé zprovoznit a nastavit.

Dalším populárním příkladem open-source kontroléru je **Domoticz**[6], který se naopak od openHAB zaměřuje hlavně na tu grafickou správu přes webové rozhraní a je tak pohodlnější pro začátečníky. Stejně jako openHAB podporuje všechny hlavní systémy. Je napsaný v C++ spolu s vlastním vestavěným webovým serverem (napsaným také v C++), který používá pro zobrazení webového rozhraní (které je napsáno v HTML5)[7].

Domoticz rozděluje zařízení, se kterými může komunikovat, na dvě skupiny: *Hardware*, což jsou zařízení, se kterými dokáže komunikovat přímo, a *Devices* jsou zařízení, ke kterým potřebuje bránu. Stejně jako openHAB podporuje velké množství různých zařízení a technologií, ale nepotřebuje k tomu vazby; ty jsou defaultně zabudovány v uvnitř.

Co se týče zbytku inteligentní domácnosti, **MySensors**[16] pokrývají jak senzory, tak brány. Jejich řešení staví na mikrokontrolerech Arduino a ESP8266 a modulech nRF24L01+, RFM69, RFM95 (LoRa) a RS485 pro komunikaci. První tři moduly jsou pro bezdrátovou rádiovou komunikaci na frekvencích 433 MHz, 868 MHz, 915 MHz a 2,4 GHz. Poslední modul slouží pro sériovou komunikaci přes kabel.

MySensors je velká otevřená komunita, která vytvořila a spravuje vlastní knihovnu MySensors, jejíž cílem je snadná tvorba senzorů pomocí již připravených příkladů pro nejběžnější zařízení a vytvoření sítě takto nazvájem komunikujících zařízení. Tato síť má stromovou topologii, kdy zařízení jsou propojena v několika vrstvách a dokážou směřovat zprávy ostatním zařízením o obou směrech; rodičům i potomkům. V případě výpadku rodiče se pak snaží zařízení najít broadcastem jiného rodiče, přes kterého by mohlo zase komunikovat s bránou nebo kontrolérem, takže je taková síť i tzv. sebe-opravující (angl. Self-healing). Tato síť může obsahovat až 254 takových zařízení, z nichž jedno se může chovat jako brána nebo kontrolér, který slouží jako přidělovač identifikačních čísel senzorů a zdroj hodin[17].

Dalším možným otevřeným řešením pro bránu je **OpenMQTTGateway**[35], která je určena pro ESP8266, ESP32, Sonoff RF Bridge a Arduino. Pro komunikaci s kontrolérem používá MQTT protokol nad Ethernetovým nebo WiFi spojením. Se senzory se dokáže

domluvit přes rádiovou komunikaci na frekvencích 433, 315, 868 a 915 MHz, Bluetooth, infračervenou komunikaci a 2G mobilní síť (GSM/GPRS).

2.2.2 Komerční řešení

Komerční řešení jsou přesným opakem open-source řešení, jejichž otevřené zdroje lze upravovat a jsou zdarma. Ty jsou zde nahrazeny uzavřeným hardwarem a softwarem, u kterých nejsou vidět ani schémata, ani kód, protože se jedná o firemní tajemství a know-how a je potřeba za ně zaplatit.

Na oplátku ale nabízejí hotová řešení v elegantních provedeních s pohodlnějším nastavením a používáním. Také je často zajištěna technická podpora a aktualizace softwaru. V mnoha případech pak jde také o kompletní řešení inteligentních domácností obsahující jak senzory, tak kontrolér a bránu.

Docela sporným bodem u komerčních řešení bývá bezpečnost a soukromí. Kontrolér totiž bývá většinou umístěn v cloudu společnosti a všechna data (a není vždy jasné, o která data se vůbec jedná) chodí z domácnosti ven do internetu. V tomto ohledu vychází lépe open-source řešení, protože lze přesně nastavit, co a kam chodí a celá inteligentní domácnost může být v jedné domácí síti, takže data ani nemusí chodit do internetu. Proto se v tomto bodě často dělá kompromis mezi tím, jak moc nám záleží na soukromí a kolik jej chceme vyměnit za pohodlí.

Komerčních řešení existuje mnohonásobně víc než otevřených, protože se jedná o poměrně nový a lukrativní trh. Pro příklad lze tak uvést jen pár společností a produktů, jako třeba **Vera**[29], která nabízí kontroléry a bezpečnostní kamery spolu s bezpečnostními službami a velkou podporou mnoha senzorů. Mezi podporované technologie patří například Z-Wave, ZigBee, VeraLink nebo Bluetooth (BLE)[30].

Veru lze ovládat přes webové rozhraní nebo mobilní aplikaci. Nabízí jak „offline“ režim, kdy kontrolér není připojen k Internetu a lze ovládat zařízení jen v domácí síti, tak i „online“ režim, kdy domácnost může být ovládána vzdáleně přes Internet a data přitom chodí přes zabezpečené cloud servery[31]. Vera také nabízí širokou technickou podporu se zárukou 12 měsíců na vlastní kontroléry a kamery a podrobné manuály a návody na jejich zprovoznění. Nejlevnější základní kontrolér začíná na 100 dolarech (cca 2 100 Kč) a nabízí pouze Z-Wave a WiFi, čímž i tak dokáže podporovat spoustu senzorů a tvoří tak docela levný a dostupný základ inteligentní domácnosti.

Co se týče komerčních senzorů, jako příklad lze uvést společnost **2GIG**[1], která se zaměřuje především na bezpečnostní zařízení jako alarmy, bezpečnostní kamery, ovládací panely, kouřové senzory, detektory pohybu a spoustu dalších. Pro komunikaci používají především Z-Wave, WiFi a proprietární RF technologie.

Ovládání 2GIG zařízení probíhá hlavně přes ovládací panely GC2[2] a GC3[3] s možností přístupu také přes webový prohlížeč. Kontrolér je tak obsažen v ovládacím panelu, ale protože zařízení používají pro komunikaci Z-Wave, lze je ovládat například i z Vera kontroléru, který 2GIG zařízení podporuje. 2GIG lze tak použít jako kompletní řešení inteligentní domácnosti, i když s cenou za GC2 pohybující se okolo 200 dolarů (cca 4 300 Kč) a za GC3 okolo 400 dolarů (cca 8 500 Kč) je možná výhodnější si pořídit Vera kontrolér, který je několikanásobně levnější a podporuje mnoho 2GIG senzorů.

Inteligentní domácnosti jsou v současnosti docela rozšířené i u nás v České Republice a několik firem nabízí kompletní řešení pro instalaci ať už při stavbě nového domu nebo do již fungující bytu. Jedním z takových projektů je například **FIBARO**[11], které je postaveno čistě na bezdrátové technologii Z-Wave. Všechny nabízené produkty lze zapojit do

zařízeného domu, většinu přímo do zásuvek a některé přímo do živého vedení elektřiny. Kontrolérem je zde FIBARO Home Center 2 nebo chudší verze Home Center Lite, kteří slouží také jako brána s přístupem do Internetu. Zařízení pak lze ovládat vzdáleně přes Internet pomocí webového rozhraní nebo mobilní aplikace.

FIBARO nabízí několik vlastních senzorů pro základní ovládání domácnosti jako jsou chytré zásuvky (se zaznamenáním spotřebované elektřiny), stmívače osvětlení, ovladače žaluzií, různé detektory nebo třeba termostatická hlavice na topení pro jeho ovládání[12]. Protože však používá Z-Wave technologii, podporuje tak i další zařízení od ostatních výrobců.

Oproti předchozím řešením je však FIBARO podstatně dražší, kdy kontrolér Home Center 2 u nás stojí 15 990 Kč (přibližně 750 dolarů) a senzory se pohybují v rozmezí 1 000-3 000 Kč (přibližně 50-150 dolarů). Vzhledem k tomu, že podporuje pouze Z-Wave, je tak možná výhodnější upřednostnit jiná řešení s větším okruhem podporovaných zařízení a senzorů.

Kapitola 3

Návrh řešení

Tato práce se zabývá hlavně realizací brány spolu s jednoduchými senzory pro demonstraci použití a testovací účely. Oba prvky jsou tvořeny jak hardwarovou, tak softwarovou částí, přičemž jsou postaveny na otevřených platformách a knihovnách. Základem hardwaru jsou vývojové desky spolu s komunikačními moduly, u softwaru pak hlavně knihovny pro ovládání daných modulů.

3.1 Vývojové desky

Vývojová deska je deska plošných spojů osazena mikrokontrolerem s vyvedenými výstupy ve formě pinů. Dále jsou zde přítomny různé podpůrné čipy a obvody, například převodník mezi USB a RS232 pro programování mikrokontroleru přes USB nebo regulátory napětí pro poskytnutí více úrovní napětí, zpravidla 3.3 a 5 V.

Vývojové desky tedy slouží k snadnému přístupu a programování mikrokontroleru a testování funkčnosti s připojenými komponentami přes vyvedené piny. Od toho také název vývojová deska, protože je určena k vývoji a otestování programů předtím, než bude mikrokontroler vložen do konečného obvodu.

Různých typů a druhů vývojových desek je neskutečné množství, protože není dán žádný předpis, jak by měla deska vypadat, jaké mít rozměry nebo co všechno obsahovat. Desky se tedy liší velikostí i tvarem, ale hlavně výbavou, ať už se jedná o různé mikrokontrolery nebo dostupné komponenty.

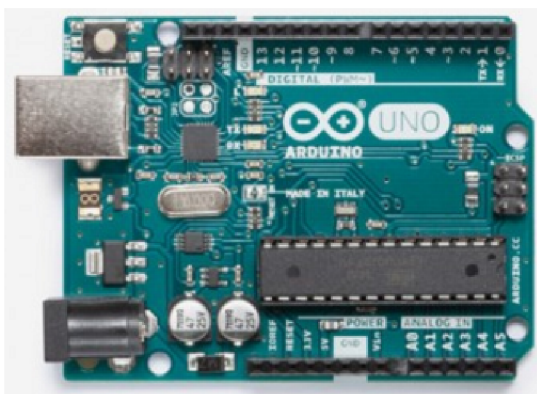
Výbavu vývojových desek lze ale také rozšířit o tzv. štíty (angl. Shield). Jedná se o další desky plošných spojů, které mají stejné rozměry a rozložení a funkčnost pinů jako vývojové desky, ke kterým jsou určeny. Tyto desky již neobsahují mikrokontroler, ale pouze další rozšiřující komponenty, které s mikrokontrolerem komunikují přes určité piny. Navíc lze většinou takové štíty skládat na sebe a zvětšovat tak množství funkcí za předpokladu, že nebudou používat stejné piny, protože by si tak jednotlivé štíty narušovaly správnou funkčnost.

3.1.1 Deska Arduino Nano

Arduino je jedna z nejnámějších platform pro vývojové desky. Jedná se o otevřenou platformu, která v sobě zahrnuje jak vývojové desky, tak programovací jazyk Arduino, obojí vycházející z platformy Wiring[34], a také vývojařský software Arduino IDE postavený na Processingu. Původně bylo Arduino vytvořeno v Itálii na Ivrea Interaction Design Insti-

tute jako jednoduchý a levný učební nástroj pro prototypování se zaměřením na studenty bez znalostí elektroniky a programování[4].

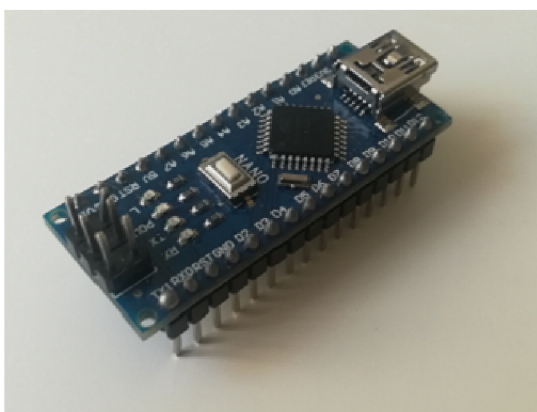
V současnosti nabízí Arduino několik různých vývojových desek, lišících se hlavně použitým mikrokontrolerem, kterých je několik různých druhů, jako třeba ATmega, ARM, Intel a Atheros. Mezi nejpobulárnější desky, a doporučenou hlavně pro začátečníky, patří zejména Arduino Uno[5] běžící na ATmega328P, což je 8bitový mikrokontroler s 32 KB paměti Flash, 2 KB SRAM a 20ti vyvedenými piny (z toho 14 digitálních vstupně/výstupních a 6 analogových vstupních).



Obrázek 3.1: Arduino UNO (převzato z [5])

Kromě obecných desek s všestranným využitím, jako výše uvedená Uno, nabízí Arduino také specializované desky pro konkrétní potřeby jako například Arduino LilyPad, která je určena k zašití do oděvů nebo k projektům vhodných k nošení na těle.

V této práci je použita deska Arduino Nano (respektive její klon, který ale má stejnou funkcionalitu), která běží podobně jako Uno na mikrokontroleru ATmega328P, což znamená, že výkonem a vnitřní výbavou jsou stejné. Čím se ale liší jsou zejména fyzické rozměry a počtem a funkcí pinů. Nano má o jeden digitální a jeden uzemňovací pin méně, na druhou stranu ale o 2 analogové piny navíc. Uno také používá k propojení s počítačem a nahrání programu USB typu B, Nano naproti tomu používá USB Mini.



Obrázek 3.2: Arduino Nano (klon)

Funkcionalitou jsou tedy tyto dvě desky víceméně stejné. Rozhodujícím faktorem pro výběr Arduino Nano ale byla právě fyzická velikost, která je oproti Uno asi 4x menší, a to

kvůli tomu, že tyto desky tvoří základ senzorů, u kterých jsou rozměry důležité. Oproti WEMOS deskám popsaných níže jsou také méně energeticky náročné a vydrží tak déle běžet na baterie, což je u senzorů také velmi důležité, protože například takový venkovní měřič teploty a vlhkosti vzduchu nejspíš nebude mít zrovna přístup k zásuvce.

3.1.2 Deska WEMOS LOLIN32

WEMOS[32] je mladá čínská společnost se zaměřením na vývoj nákladově efektivních IoT produktů. V současnosti mají v nabídce hlavně vývojové desky a rozšiřující štíty. Desky jsou postavené na mikrokontrolerech od firmy Espressif, zejména starším ESP8266 a novějším ESP32 a jejich různých verzích.

Mezi jejich nejpopulárnější produkty patří deska D1 mini postavená na starším čipu ESP-8266EX a deska LOLIN32 běžící na nejnovějším ESP-WROOM-32. Nejnovějším přírůstkem do rodiny je deska D32, která je následníkem LOLIN32. D32 běží na stejném mikrokontroleru ESP-WROOM-32 a oproti LOLIN32 nabízí jiné rozložení pinů plus vyvedení pinů USB a baterie.



Obrázek 3.3: WEMOS LOLIN32

Oproti Arduino jsou WEMOS desky o mnoho výkonnější. Oba ESP mikrokontrolery obsahují 32bitové mikroprocesory, přičemž novější ESP32[9] je dokonce dvoujádrový oproti staršímu jednojádrovému ESP8266[10]. Dále mají také několikanásobně větší paměti RAM a Flash, spolu se zabudovanou podporou WiFi (IEEE 802.11 b/g/n); ESP32 dokonce i Bluetooth 4.2 (BLE). To ale také znamená, že jsou více energeticky náročné, hlavně při neustálém provozu a používání více funkcí a komponent najednou.

Co se programování týče, WEMOS desky jsou tzv. Arduino-kompatibilní, tedy lze psát kód v Arduino IDE v Arduino jazyce a nahrát jej bez problému na WEMOS desku, aspoň tedy při použití základních funkcí a příkazů jazyka. V čem může být problém, jsou Arduino knihovny, protože Arduino a WEMOS desky běží na odlišných mikrokontrolerech a hardwarově specifické knihovny tak nebudou fungovat a musely by se upravit pro potřeby ESP čipů.

V této práci je použita starší deska LOLIN32[33], protože v době výběru byla ještě nejnovějším modelem. Vzhledem k výkonnosti a vybavenosti byla tato deska zvolena jako základ a mozek brány. U brány se očekává, že bude v domě na jednom místě a bude neustále v provozu, aby mohla směřovat data mezi senzory a kontrolérem, a proto není vyšší spotřeba ESP32 problémem, protože se může dát někam k zásuvce a mít tak stálý příjem energie. Dále zabudovaná podpora WiFi byla velice důležitá a v této práci je zásadní. Přítomnost Bluetooth je také vítaná, a i když zde není využita, vybízí k budoucímu rozšíření a přidání tak podpory pro Bluetooth zařízení a senzory.

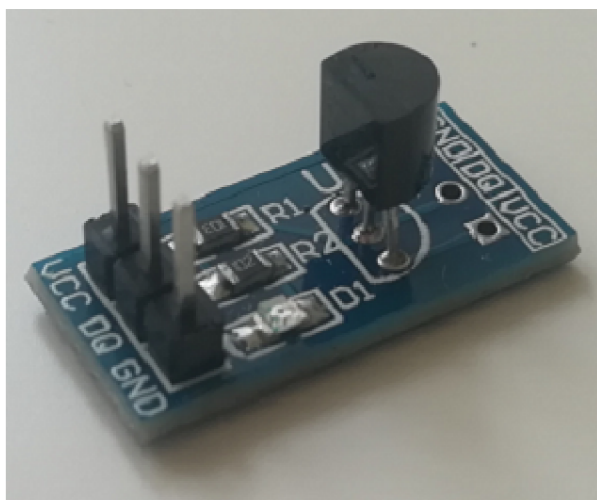
3.2 Senzorové moduly

Součástí této práce je také realizace jednoduchých senzorů pro testovací a demonstrační účely. Pro tyto účely byly vybrány 3 jednoduché moduly: teplotní čidlo, detektor pohybu a relé. Zde je uvedena základní teorie a informace každém z nich, později v kapitole 4 **Implementace řešení** je popsáno, jak jsou zapojeny a co dělají.

3.2.1 Teplotní čidlo DS18B20

Jedná se o jednoduchý digitální teploměr využívající rozhraní 1-Wire. Každý DS18B20 má unikátní 64bitový sériový kód, což dovoluje přítomnost a funkčnost více čidel na jediné 1-Wire sběrnici. Nabízí 9-12bitové měření teploty ve stupních Celsia v celkovém rozsahu -55°C až $+125^{\circ}\text{C}$, s přesností $\pm 0,5^{\circ}\text{C}$ v rozsahu -10°C až $+85^{\circ}\text{C}$. DS18B20 lze napájet 3-5,5 V, přičemž při nečinnosti odebírá 750-1000 nA, při měření pak 1-1,5 mA[8].

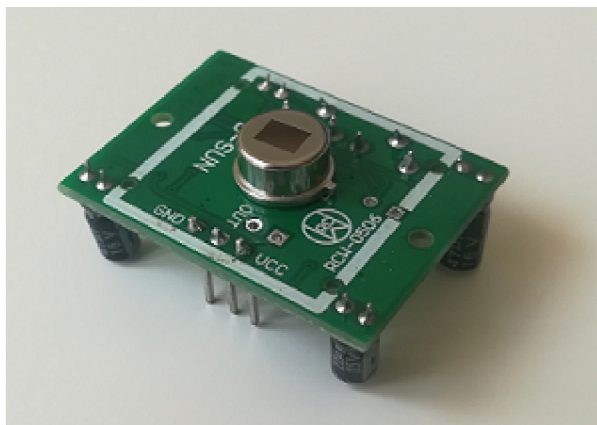
Dodává se v několika různých pouzdrech a provedeních, jako například 8pinové pouzdro SO (označené DS18B20Z), 8pinové pouzdro μSOP (označené DS18B20U) a 3pinové pouzdro TO-92 (označené pouze DS18B20), které je také použito v této práci v provedení malého plošného spoje (viz obr. 3.4). Dva piny jsou pro napájení a jeden je datový. DS18B20 ale podporuje také tzv. parazitický napájecí mód, kdy stačí pouze uzemňovací pin a datový pin, ze kterého probíhá napájení (původní napájecí pin je uzemněn).



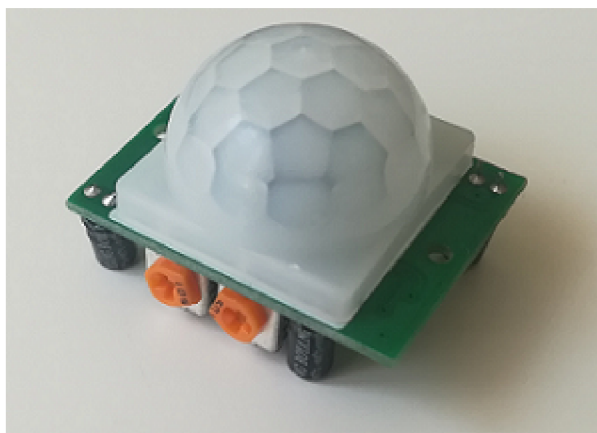
Obrázek 3.4: Teplotní čidlo DS18B20

3.2.2 Detektor pohybu HC-SR501

HC-SR501 je detektor pohybu založený na infračerveném záření. Nabízí vysokou citlivost a nízkou spotřebou. Detektor lze napájet 5-20 V a v klidovém stavu odebírá méně jak $50 \mu\text{A}$, při práci pak 65 mA. Dodává se v pouzdře s pouze 3 piny (dva pro napájení a jeden datový, kdy logická 1 znamená zaznamenaný pohyb, logická 0 klid) a 2 potenciometry; jeden pro nastavení vzdálenosti, na kterou je detekován pohyb (3-7 metrů), a druhý pro zpoždění mezi jednotlivými detekcemi (5-200 s). Pohyb je schopen detekovat pod úhlem 120° a pracuje v teplotním rozsahu od -15°C do $+70^\circ\text{C}$ [13].



Obrázek 3.5: PIR motion (piny a čidlo)



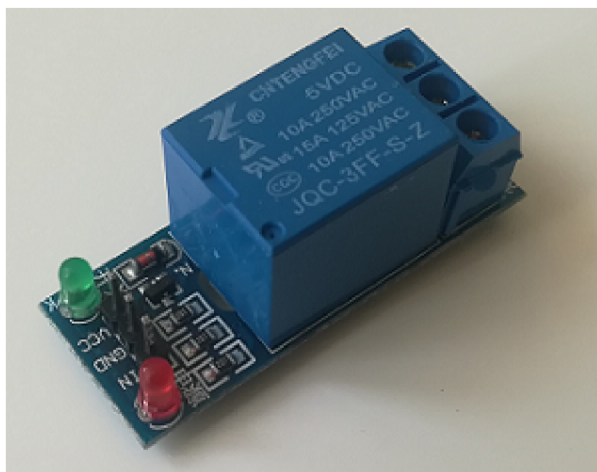
Obrázek 3.6: PIR motion (potenciometry a pouzdro)

3.2.3 Relé JQC-3FF-S-Z

Tento tzv. relé štít[14] je určen přímo pro mikrokontrolery s cílem spínání relé pomocí nízkého napětí (v tomto případě 5 V, ale existují například i 3,3V, 6V a vyšší verze). Relé je tak řízeno 5V logikou, přičemž tento model je tzv. low-level trigger neboli spíná při logické 0 a rozpíná při logické 1. Dokáže spínat napětí až do 250 V s proudem do 10 A. Lze tak zapínat/vypínat domácí zařízení, která jsou připojena do rozvodné sítě, nebo také samotné vstupy/výstupy sítě, jako světla, zásuvky a prodlužovací příklady.

Na jednom konci desky jsou 3 piny pro připojení k mikrokontroleru (dva pro napájení a jeden datový pro spínání relé). Na druhém konci jsou 3 vývody pro spínání obvodu:

- C (common) - znamená společný kontakt, který se překlápí podle toho, jestli je relé sepnuto nebo ne,
- NC (normally closed) - obvod je uzavřen/spojen s kontaktem COM v klidovém stavu (relé není sepnuto),
- NO (normally open) - obvod je otevřen/rozpojen s kontaktem COM v klidovém stavu



Obrázek 3.7: Relé štít

3.3 Bezdrátové technologie

Bezdrátové technologie jsou v této práci zásadní. Veškerá komunikace, ať už mezi senzory a bránou nebo bránou a kontrolérem, probíhá bezdrátově. To se sebou přináší určité výhody i nevýhody, jak již bylo zmíněno v teoretické části u sensorů. Hlavní výhodou je samozřejmě volnost a mobilita. Zařízení nejsou vázána na určité místo, kde jsou vyvedeny kabely, ale v případě potřeby se můžou volně přenášet, jak je potřeba. Naopak nevýhodou je spolehlivost komunikace, která může být rušena několika různými způsoby, v závislosti na použitém typu bezdrátové komunikace.

Bezdrátových technologií je totiž velké množství, ale dají se rozdělit na dvě hlavní skupiny podle fyzického způsobu přenosu informací, a to jsou rádiové vlny a infračervené záření. V těchto skupinách pak existuje mnoho technologií a protokolů, které určují, jak jsou informace kódovány a přenášeny.

3.3.1 Rádiová komunikace

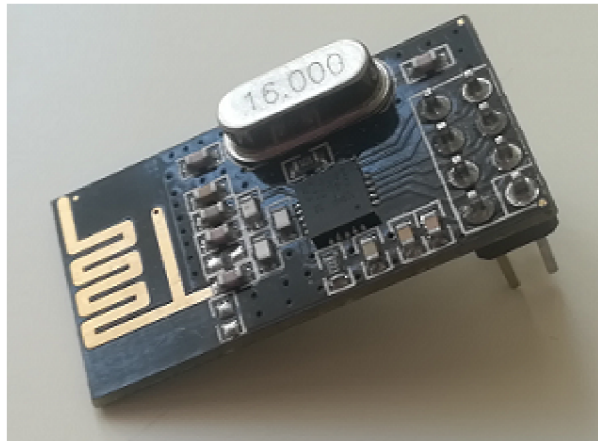
Rádiová komunikace je postavena na rádiových vlnách, což je elektromagnetické záření s vlnovou délkou od 1 mm až po 100 km s odpovídající frekvencí od 300 GHz do 3 kHz. Podle frekvence se rádiové vlny dělí na frekvenční pásma, například nízké frekvence (30-300 KHz) používané pro AM vysílání nebo ultra vysoké frekvence (300-3000 MHz) pro televizní vysílání a mobilní telefony.

Rádiové komunikace jsou silně regulovány úřadem ITU (International Telecommunication Union - Mezinárodní telekomunikační unie), který rozhoduje o využití frekvenčních pásem. Tento úřad také určil tzv. ISM pásma (Industrial, Science, Medical), která se používají pro rádiové vysílání v průmyslových, vědeckých a zdravotních oborech. Tato pásma jsou volná, což znamená, že není potřeba mít licenci k jejich použití, avšak negarantují ochranu proti rušení od ostatních zařízení používaných stejnou frekvencí. Nejběžnějšími zástupci těchto pásem jsou například mikrovlnné trouby (2,45 GHz) nebo WiFi sítě (2,4 a 5 GHz).

V rádio technologiích jsou rádiové vlny generovány uměle pomocí vysílače s připojenou anténou a na druhé straně přijímány přijímačem také s anténou. Délka antény je velmi důležitá a závisí přímo na frekvenci, kterou má vysílat nebo přijímat. Délka je dána vzorečkem $l = v/f$, kde v je rychlost šíření (ve vakuu je rychlost blízko rychlosti světla, tj. 299 792 458 m/s) a f je požadovaná frekvence, tedy například pro frekvenci 2,4 GHz by plná délka antény měla být cca 12,49 cm ($l = \frac{299\,792\,458}{2\,400\,000\,000} [m] \approx 0,1249\,m \approx 12,49\,cm$). Při této délce vzniká rezonance na dané frekvenci 2,4 GHz, která zapříčiní vysílání rádiových vln. Délku lze ale také zkrátit na polovinu nebo čtvrtinu se zachováním rezonance, což je vhodné z důvodu konstrukce antén.

V této práci se používají frekvence 433 MHz a 2,4 GHz, obě z ISM pásma, přičemž 2,4 GHz má dvě různá použití, a to jak pro komunikaci mezi bránou a senzory pomocí modulů nRF24L01+, tak pro spojení brány a kontroléru pomocí WiFi technologie. 433 MHz je poté také využita ke komunikaci mezi senzory a bránou pomocí modulů STX882 a SRX882.

nRF24L01+ [19] je jednočipový transceiver neboli vysílač a přijímač v jednom. Jak bylo zmíněno, pracuje na frekvenci 2,4 GHz a používá GFSK modulaci, což je metoda frekvenční modulace. Transceiver lze napájet napětím 1,9-3,6 V a vyznačuje se velmi nízkou spotřebou: v režimu spánku 900 nA, v pohotovostním režimu 26 μ A, při vysílání dat 11,3 mA při síle signálu 0 dBm a při příjmu 13,5 mA při rychlosti přenosu dat 2 Mbps. Dále podporuje také rychlosti 250 kbps a 1 Mbps. Při nižších rychlostech je lepší příjem, ale vyšší rychlost snižuje spotřebu a také pravděpodobnost kolizí.



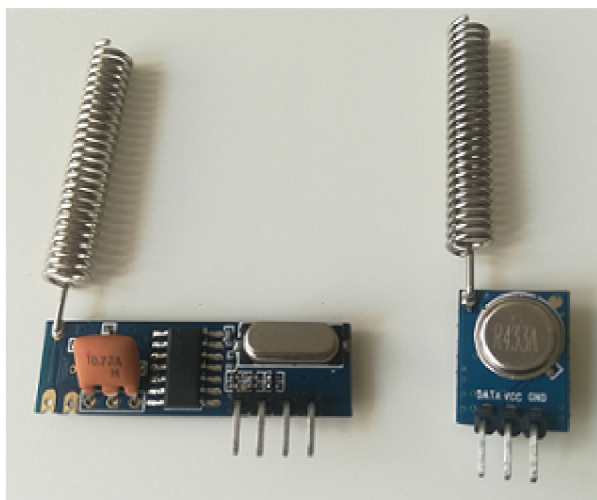
Obrázek 3.8: Rádio 2,4 GHz transceiver nRF24L01+

Tento modul je populární volbou v oblasti domácí automatizace a IoT kvůli své všestranné využitelnosti. Za nízkou cenu nabízí malé rozměry, integrovanou anténu a velké rychlosti přenosu dat. Stačí ho připojit k mikrokontroleru, ke kterému se připojuje pomocí rozhraní SPI, a který se pak stará o jeho ovládání a posílání dat.

Dalšími komunikačními moduly jsou **SRX882**[25] jako přijímač a **STX882**[26] jako vysílač na frekvenci 433 MHz. Oproti nRF24L01+ je tak funkce vysílače a přijímače rozdělena do samotných modulů. Zároveň také nemají integrovanou anténu, která je potřeba k oběma deskám připájet. Používají ASK modulaci, což je metoda amplitudové modulace.

Vysílací modul STX882 je možné napájet 1,2-6 V s doporučenou úrovní 3,3 V. Při tomto napětí umožňuje vysílat signál o síle 15 dBm, přičemž konzumuje 34 mA. V režimu spánku pak spotřebovává méně jak 10 nA. Takže při vysílání je náročnější než nRF24L01+, zato ale při nečinnosti je úspornější.

Přijímací modul SRX882 je možné napájet od 2,4-5,5 V s doporučenou úrovní 5 V. Při tomto napájení spotřebovává při příjmu dat 2,8 mA, při nečinnosti pod 2 μ A. Zase v porovnání s nRF24L01+ je tak SRX882 úspornější při práci a jen trochu náročnější při neaktivitě. Rychlost přenosu dat je v rozmezí 0,1-9,6 Kbps.



Obrázek 3.9: Rádio 433 MHz přijímač SRX882 (vlevo) a vysílač STX882 (vpravo)

Tyto moduly jsou tak pomalejší než nRF24L01+ a v závislosti na módu činnosti také náročnější. Z principu vlastností vlnových délek o různých délkách, a tedy i frekvencích, jsou však rádiové vlny o frekvenci 433 MHz více penetrující než 2,4 GHz, což se hodí v domech a zastavěných oblastech, protože tak projdou více překážkami (například stěnami) a pokryjí tak větší oblast. Proto se používá tato frekvence například u zvonků dveří, bezpečnostních alarmů nebo dálkových ovladačích (například ke dveřím do auta nebo garáže).

3.3.2 Infračervená komunikace

Další populární bezdrátovou technologií je infračervená komunikace založená na infračerveném záření, které je stejně jako rádiové vlny elektromagnetickým zářením o vlnové délce mezi 760 nm a 1 mm. Nejčastějším případem použití je v dálkových ovladačích a mobilních telefonech.

Infračervená komunikace je velice levná záležitost, protože jako vysílač stačí pouhá infračervená LED dioda a jako přijímač fotodiody. Komunikace pak probíhá jednoduše rozsvícením a zhasínáním LED diody. Bohužel nelze tak snadno kódovat informace, jako že „dioda svítí“ = logická 1 a „dioda nesvítí“ = logická 0, kvůli rušení okolního prostředí, protože například slunce nebo i obyčejná žárovka je také zdrojem infračerveného záření[36].

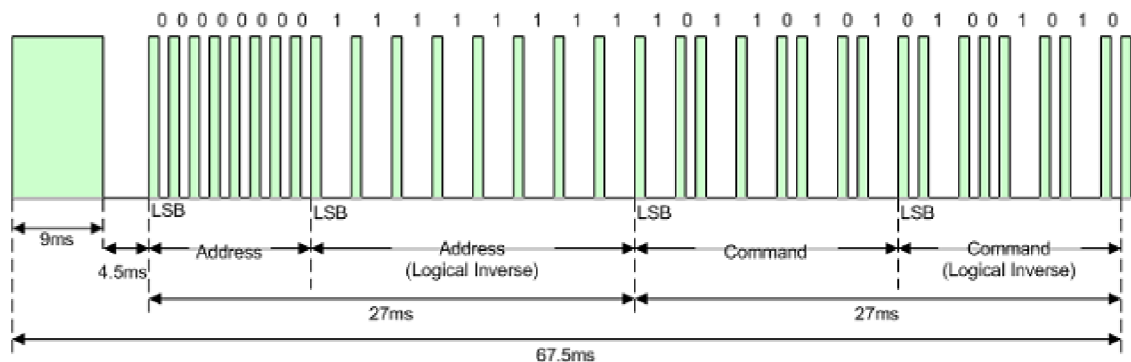
Proto je potřeba provést modulaci vysílaných dat; typicky se používá frekvence 30-56 kHz. Při modulaci logickou 1 je tedy LED dioda buzena o dané frekvenci a střídá (tento stav se anglicky nazývá „mark“ neboli „značka“), po dobu logické 0 zůstává vypnuta (angl. „space“ neboli „mezera“)[36]. Existuje několik komunikačních protokolů, které přesně definují, jak dlouho značka a mezera trvají a co znamenají. Mezi nejznámější patří například NEC, Panasonic nebo RC5 a RC6.

V této práci je použita knihovna IRRemote, která podporuje asi 20 protokolů, z čehož některé jenom pro příjem nebo jenom posílání dat. V implementaci brány je zavedena podpora následujících protokolů: LG, NEC, Samsung, Sony, Sharp a Whynter kvůli jejich podpoře jak odesílání, tak přijímání dat. Tato podpora je ale z větší části teoretická, protože prakticky odzkoušený je pouze protokol NEC, který je zde uveden jako příklad.

NEC protokol[18] používá pro kódování bitů pulzně poziční modulaci a pracuje na nosné frekvenci 38 kHz. Každá značka trvá $562,5 \mu\text{s}$. Logické bity jsou pak kódovány následovně:

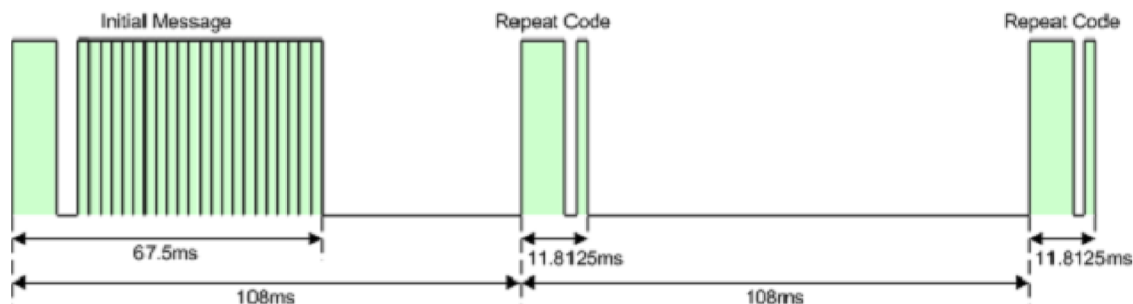
- logická 0 - značka o délce $562,5 \mu\text{s}$ následovaná stejně dlouhou mezerou, s celkovou dobou přenosu $1,125 \text{ ms}$,
- logická 1 - značka o délce $562,5 \mu\text{s}$ následovaná 3x mezerou, tedy $1,6875 \text{ ms}$ s celkovou dobou přenosu $2,25 \text{ ms}$, tedy přesně dvakrát větší jak logická 0.

Jedna zpráva je podle protokolu rozdělena na hlavičku, adresovou část, příkazovou část. Hlavičku tvoří počáteční značka trvající 9 ms , následovaná mezerou $4,5 \text{ ms}$. Následuje 8bitová adresa identifikující přijímající zařízení spolu s 8bitovou negací dané adresy. Podobně následuje i 8bitový příkaz zase s 8bitovou negací daného příkazu. Na závěr je ještě jedna značka o délce $562,5 \mu\text{s}$ označující konec zprávy. Zde je uveden příklad zprávy určené pro zařízení s adresou 00h (00000000b) a příkazem ADh (10101101b). Nejdříve se posílají nejméně významné bity (LSB):



Obrázek 3.10: Formát zprávy podle NEC protokolu (převzato z [18])

Při opakovaném posílání kódu (například při držení tlačítka dálkového ovladače) se neposílá celá zpráva znovu, ale pouze tzv. opakovací kód (angl. Repeat Code), který je složen z 9 ms značky, $2,25 \text{ ms}$ mezery a nakonec $562,5 \mu\text{s}$ značky označující konec předchozí mezery a potažmo i opakovacího kódu. Tento opakovací kód je poté posílán každých 108 ms po celou dobu, co je drženo tlačítko ovladače.



Obrázek 3.11: Příklad dvou opakovacích kódu po poslané zprávě NEC protokolem (převzato z [18])

Výhodou infračervené komunikace je tedy její jednoduchost a nízké náklady. Hlavní slabostí je ale nutnost přímé viditelnosti mezi zdrojovou LED diodou a přijímací fotodiodou. To značně omezuje rozsah možných využití a také vzdálenost, na kterou je možné komunikovat. V domácnosti je přesto ještě několik zařízení, které tento způsob komunikace využívá, jako například televize nebo audio a video systémy, a proto je podporována i v této práci. Navíc je možné si pořídit i univerzální dálkové ovladače a není problém poté v kontroléru nastavit pravidla, co má které tlačítko dělat.

V této práci je jako vysílač použita infračervená dioda **TSAL6100**[28], která vyzařuje světlo o vlnové délce 940 nm se zářivostí 170 mW/sr a úhlem 10° . Pro přijímač je použit modul **TSOP4438**[27], který je určený pro frekvenci 38 kHz a zároveň demoduluje data. Modul má pouze 3 piny, dva pro napájení a jeden pro výstup dat. Lze jej napájet v rozmezí 2,5-5,5 V a odebírá pouze 5 mA.

3.3.3 WiFi + MQTT

WiFi je asi nejznámější a celosvětově jednou z nejpoužívanějších bezdrátových technologií. Jedná se technologii postavenou na několika standardech IEEE 802.11, které specifikují implementaci bezdrátové lokální sítě (WLAN) na frekvencích 900 MHz a 2,4, 5 a 60 GHz. V této práci je použita WLAN pro navázání spojení mezi bránou a kontrolérem. Nahrazuje tak nutnost kabelu a přítomnost brány poblíž kontroléru. Tato možnost byla zvolena z důvodu přítomnosti WiFi v použitém mikrokontroleru ESP32, který tvoří základ brány. S tím ale přichází také požadavek, aby byl kontrolér připojen do WiFi sítě, protože brána nemá možnost kabelového spojení s kontrolérem.

MQTT[15] (MQ Telemetry Transport) je poté protokol pro zasílání zpráv založený na návrhovém vzoru publisher-subscriber (vydavatel-odběratel). Jedná se o extrémně jednoduchý a lehký protokol, vhodný především pro omezená zařízení a nespolehlivé sítě s malou propustností. Základ tvoří tzv. MQTT broker, který se stará o správu a přeprávu zpráv. Funguje na systému tzv. témat (angl. Topic), kdy se k danému tématu může kdokoli přihlásit a odebírat zprávy a kdykoliv někdo do daného tématu pošle nějakou zprávu, broker ji přepoše všem odběratelům. Vydavatel ani nemusí přemýšlet nad tím, jestli dané téma existuje nebo jestli jej někdo odebírá. Pokud dostane broker zprávu pro téma, které ještě neexistuje, založí si ho.

Témata lze také hierarchicky členit, přičemž se oddělují lomítkem, například „dům/ložnice/světlo“ nebo „dům/ložnice/topení“. Kdokoli se pak může přihlásit k vyššímu tématu a odebírat zprávy ze všech nižších témat, která pod něj spadají. Takže v příkladu výše by se mohl někdo přihlásit k tématu „dům/ložnice/#“ a dostávat všechny zprávy určené

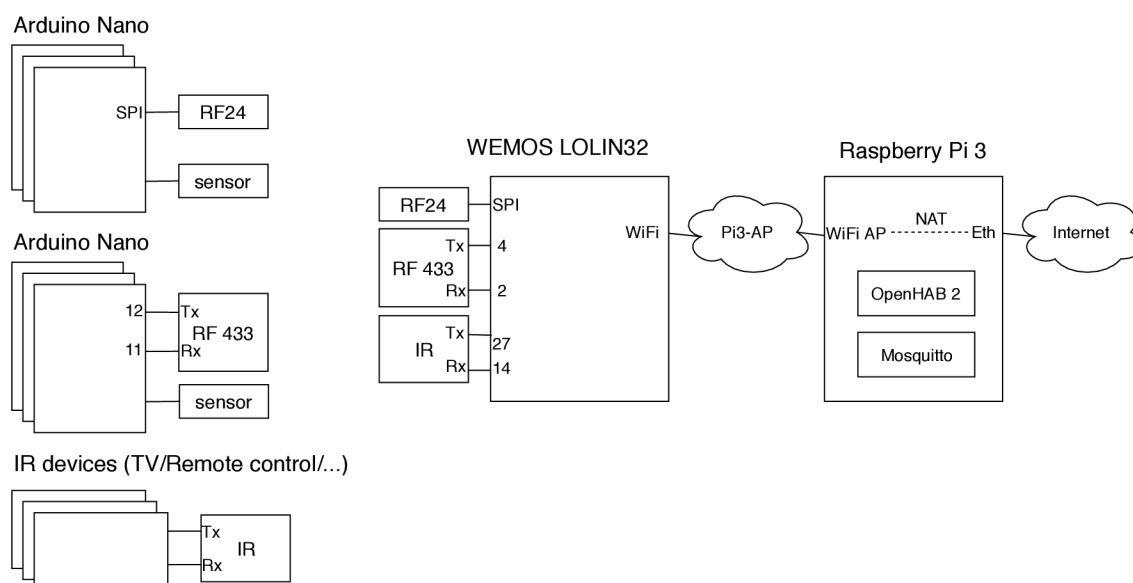
pro světlo i topení v ložnici (samozřejmě i zprávy pro samotné téma „dům/ložnice“). Speciální znak # znamená všechna témata pod danou úrovní a tento znak musí být uveden vždy jako poslední. Ještě existuje znak +, který nahrazuje pouze jednu úroveň a může být uveden kdekoliv, takže například „dům/+ /světla“ adresuje všechna světla v domě, bez ohledu na místnost.

Cílem MQTT je tak hlavně minimalizovat množství zpráv v síti a zátěž, která by byla na odesílateli zprávy, protože ten by si jinak musel pamatovat, kdo všechno má o jeho zprávu zájem a každému ji posílat samostatně. Takto ji pošle jednou brokeru, který se už o to postará sám. Tímto způsobem je pak snazší i rozšiřování a přidávání nových vydavatelů a odběratelů, protože v předchozím případě by se muselo u každého vydavatele upravit, kdykoliv by se někdo přihlásil nebo odhlásil z odběru jeho zpráv. O toto se také stará broker a vydavatel tak pořád posílá zprávy jenom jednou na jedno místo, tj. brokeru.

Kapitola 4

Implementace řešení

V této části je popsána realizace brány a senzorů. Oba prvky mají jak hardwarovou část, tak svou programovou část neboli firmware, který řídí jejich činnost.



Obrázek 4.1: Schéma zapojení implementovaného řešení

Výše je uvedeno schéma výsledného zapojení řešení (viz obr. 4.1). V levé části se nachází senzory, které komunikují s bránou uprostřed. Ta je připojena přes WiFi k síti Pi3-AP. Tuto síť vytváří Raspberry Pi 3 Model B, na kterém je nahrán systém openHABian s kontrolérem openHAB spolu s MQTT brokerem Mosquitto. Raspberry Pi zároveň tuneluje internetové připojení z kabelového připojení Ethernetovým kabelem. Je tak možné jej připojit k routeru nebo notebooku, získat přístup do Internetu a ovládat kontrolér vzdáleně.

4.1 Hardware

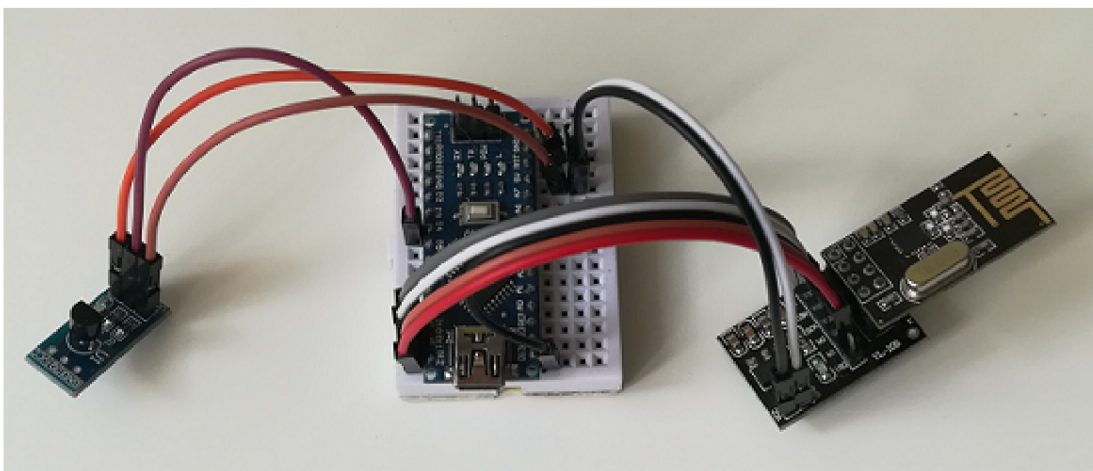
Hardware se v obou případech, jak u brány, tak u senzorů, skládá z řídicího mikrokontroleru a komunikačních modulů. Senzory ale mají navíc další moduly, v závislosti na práci, která se po nich požaduje. V této práci byly pro testovací a demonstrační účely zvoleny 3 takové moduly, které byly popsány výše.

4.1.1 Zapojení senzorů

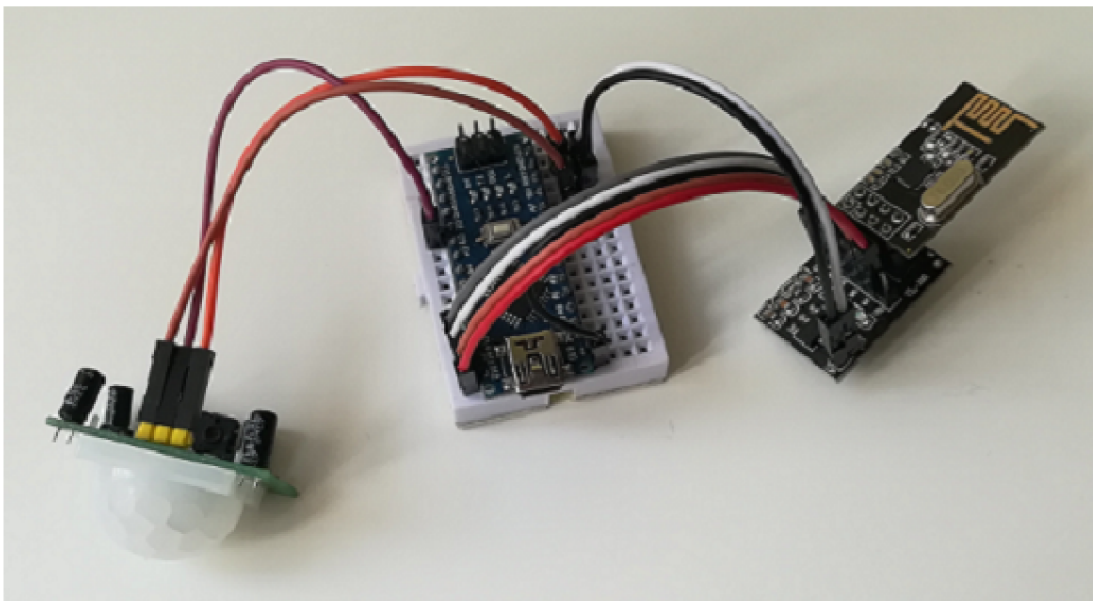
Jako řídicí část senzorů byla zvolena vývojová deska Arduino Nano. Co se týče komunikace s bránou, v této práci jsou dva druhy senzorů, kdy jeden používá rádiový modul nRF24L01+ pro 2,4 GHz komunikaci a druhý typ používá set modulů STX882+SRX882 pro 433 MHz komunikaci. Sensory s infračervenou komunikací nejsou v této práci vytvářeny, ale pro testování je použit dálkový ovladač a DVD přehrávač pro oba směry komunikace (ovladač → brána a brána → přehrávač).

Každý senzor se tedy skládá z vývojové desky s připojeným komunikačním modulem a dále jedním ze tří testovacích modulů. Tyto senzory nepodporují zapojení více modulů a mít tak jakýsi multi-senzor. Je to omezeno z programového hlediska, protože software pro senzory (uvedený pak v níže v [4.2 Firmware](#)) povoluje jenom jeden modul a jeden typ komunikace pro zjednodušení.

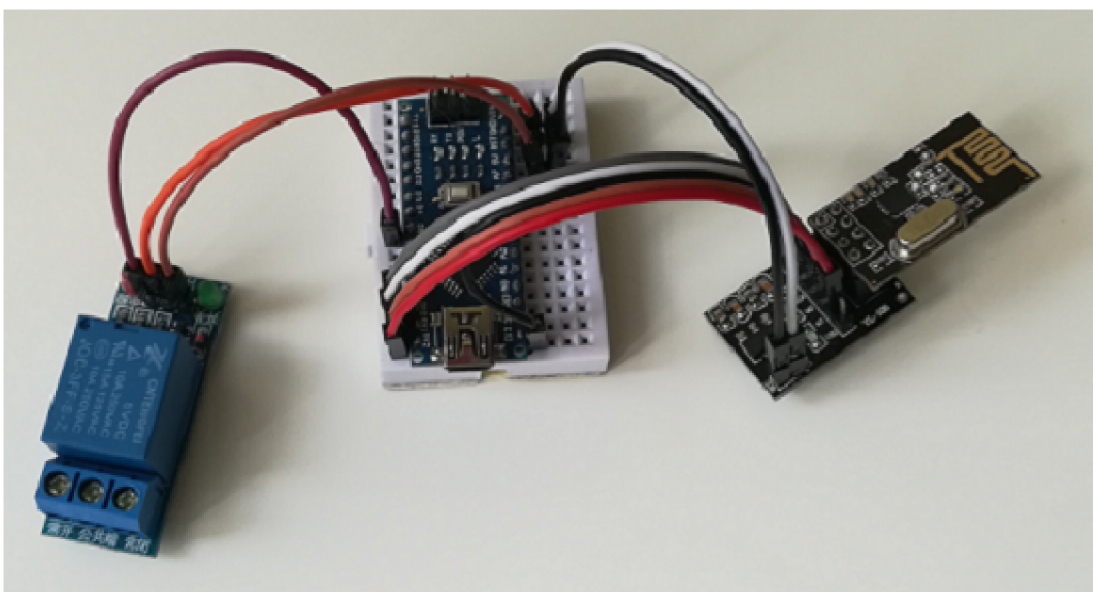
Nejdříve tedy zapojení senzoru s modulem nRF24L01+. Jelikož tento modul používá SPI rozhraní, je potřeba jej zapojit k Arduino Nano na určité piny: D10 (CS), D11 (MOSI), D12 (MISO) a D13 (SCK). Knihovna pro ovládání tohoto modulu potřebuje ještě jeden digitální pin pro povolení komunikace (CE), v tomto případě pin D9. To znamená, že zbývá 7 digitálních pinů (D2-D8) a 1 analogový pro zbylé moduly, což bohatě stačí, protože tyto senzory mohou mít jenom jeden modul a každý z testovaných modulů potřebuje jenom jeden datový pin. Testované moduly jsou pak zapojeny následovně: detektor pohybu na D2, teplotní čidlo na D4 a relé na D5. Kromě pinu D2 pro detektor pohybu (vysvětlen později v [4.2.1 Kód Arduino Nano](#)) jsou ostatní piny zvoleny náhodně a mohou být v kódu změněny.



Obrázek 4.2: Zapojení senzoru s teplotním čidlem a nRF24L01+ modulem

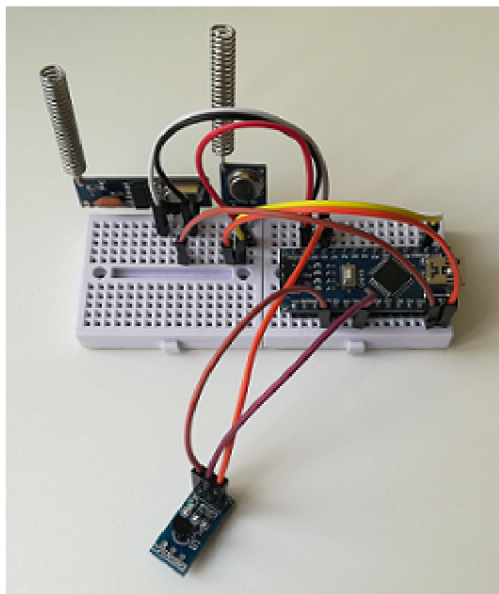


Obrázek 4.3: Zapojení senzoru s detektorem pohybu a nRF24L01+ modulem

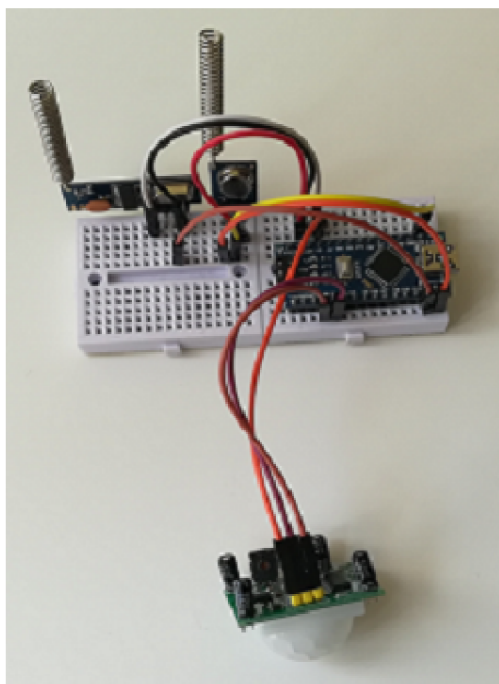


Obrázek 4.4: Zapojení senzoru s relé štítem a nRF24L01+ modulem

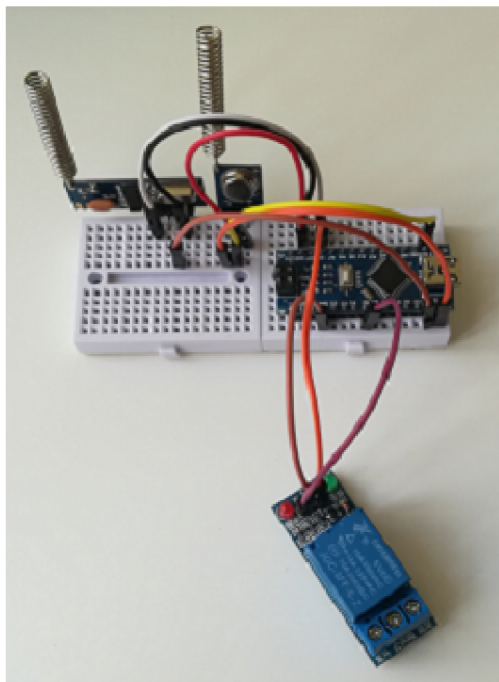
Dále mohou být senzory zapojeny s komunikačními moduly STX882 a SRX882. Tyto moduly potřebují oproti nRF24L01+ pouze 2 datové piny, každý jeden. Použitá knihovna pro ovládání těchto modulů má defaultně zvolené piny D11 (pro přijímací modul SRX882) a D12 (pro vysílací modul STX882), ale tyto piny lze v programu změnit na libovolné digitální piny. Testovací moduly jsou pak zapojeny stejně jako v předchozím případě.



Obrázek 4.5: Zapojení senzoru s relé štítem a moduly STX882 a SRX882



Obrázek 4.6: Zapojení senzoru s relé štítem a moduly STX882 a SRX882



Obrázek 4.7: Zapojení senzoru s relé štítem a moduly STX882 a SRX882

4.1.2 Zapojení brány

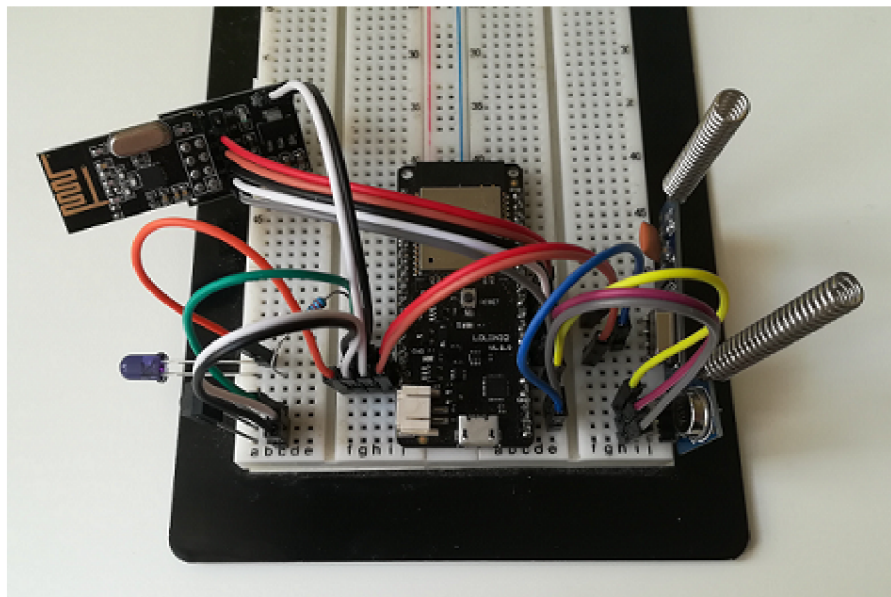
Základ brány je tvořen vývojovou deskou WEMOS LOLIN32. K ní jsou připojeny jak rádiové moduly, tak infračervená LED dioda a fotodioda pro infračervenou komunikaci, pro komunikaci se senzory. Vestavěný WiFi modul v mikrokontroleru ESP32 je pak použit pro připojení k WLAN síti a komunikaci s kontrolérem.

WEMOS LOLIN32 nabízí celkem 26 GPIO pinů, tj. obecné piny, které nemají jednu danou funkci, ale můžou jich podporovat několik. Všechny tyto piny jsou digitální, ale některé můžou fungovat i jako analogové (konkrétně 12 pinů: 12-14, 25-27, 32-36 a 39, všechny jako analogové vstupy do A/D převodníku, přičemž piny 25 a 26 můžou být také analogové výstupy z D/A převodníku).

Komunikační moduly jsou tedy připojeny následovně: nRF24L01+ potřebuje SPI rozhraní, které se na této desce nachází na pinech 5 (CS), 18 (CLK), 19 (MISO) a 23 (MOSI). Zase knihovna pro tento modul potřebuje ještě jeden pin pro povolení komunikace (CE), kterým zde byl zvolen pin 17, protože byl hned mezi 5 a 18 pinem.

Moduly STX882 a SRX882 potřebují jenom dva piny, ale defaultní piny z knihovny (pro Arduino desky piny 11 a 12) nelze tady uplatnit, protože pin 11 není ani vyvedený na desce a pin 12 nelze použít kvůli jeho vnitřnímu provedení v mikrokontroleru. Naštěstí má WEMOS LOLIN32 spoustu dalších volných pinů a pro tyto moduly tak byly vybrány piny 2 (pro přijímací SRX882) a 4 (pro vysílací STX882).

Nakonec infračervená komunikace potřebuje také jenom dva digitální piny, které lze libovolně zvolit a později v kódu upřesnit. Pro vysílací infračervenou LED diodu byl tedy vybrán pin 27 a pro přijímací modul pin 14. Mezi LED diodu a pin desky je ale potřeba přidat ještě ochranný rezistor, v tomto případě stačí jeden o hodnotě 220 Ω .



Obrázek 4.8: Zapojení brány s komunikačními moduly

Jak je vidět na zapojení (viz obr. 4.8), externě jsou potřeba zapojit pouze moduly pro komunikaci se senzory. Deska WEMOS LOLIN32, přesněji mikrokontroler ESP32 má WiFi modul zcela vestavěný a nepotřebuje z hardwarového hlediska nic přidávat. Komunikace s kontrolérem je tedy v tomto případě možná dokonce s čistou deskou bez dalších rozšiřujících modulů, záleží pouze na nahraném řídicím softwaru.

4.2 Firmware

Firmware je program napsaný pro konkrétní zařízení. Jedná se tak o specializovaný software, který je přímo určený pro použitý hardware. Protože každý hardware má vlastní firmware, je i v této práci několik různých firmwarů, jeden pro každý senzor a bránu. Liší se především knihovnami podle použitých modulů a hlavně kódem, který řídí jejich činnost.

V této práci jsou použité následující knihovny pro bezdrátovou komunikaci mezi senzory a bránou: **IRRemote** pro infračervenou komunikaci, **RadioHead** pro rádiovou komunikaci na frekvenci 433 MHz a kombinace knihoven **RF24** + **RF24Network** + **RF24Mesh** pro rádiovou komunikaci na frekvenci 2,4 GHz. Pro komunikaci brány s kontrolérem přes WiFi pomocí MQTT protokolu je pak využita knihovna **PubSubClient**.

Dále jednotlivé senzory používají různé knihovny podle připojených modulů, které ovládají. V této práci je pouze jeden takový modul, který potřebuje knihovnu, a to je teplotní čidlo DS18B20. Přesněji jsou potřeba dvě knihovny: **OneWire** pro implementaci sběrnice 1-Wire a **DallasTemperature**, která je závislá na knihovně OneWire, a která poskytuje rozhraní pro čtení dat z čidel.

4.2.1 Kód Arduino Nano

Arduino jazyk má určitou povinnou strukturu. Tak jako například v jazyce C je povinná funkce `main()`, odkud začíná program, tak i Arduino má dvě takové funkce. Funkce `setup()`, která se vykoná hned po připojení vývojové desky k napájení, a funkce `loop()`, která se zavolá po dokončení funkce `setup()` a opakuje se v nekonečné smyčce, dokud

je deska napájena a běží. Pokud je kdykoliv v průběhu vykonávání programu deska restartována, znovu se začíná funkcí `setup()` a pokračuje se opakováním funkce `loop()`.

Kdekoliv mimo tyto dvě funkce lze deklarovat a definovat pomocné uživatelské funkce a také globální proměnné, které jsou vytvářeny při běhu programu a uloženy v paměti SRAM. Samotný program je poté uložen v paměti Flash, která je nevolatilní (program v ní zůstává i po odpojení napájení).

Testované senzory, které jsou postaveny na desce Arduino Nano, mají poměrně krátký a jednoduchý firmware, viz příklad relé senzoru s komunikačním modulem nRF24L01+:

```
1  #include <SPI.h>
2  #include <RF24.h>
3  #include <RF24Network.h>
4  #include <RF24Mesh.h>
5
6  const byte rf24CEpin = 9;
7  const byte rf24CSpin = 10;
8  RF24 radio(rf24CEpin, rf24CSpin);
9  RF24Network network(radio);
10 RF24Mesh mesh(radio, network);
11
12 const uint16_t thisNode = 2;
13 const uint16_t gatewayNode = 0;
14 const byte relayPin = 5;
15 bool relayState = HIGH;
16 uint8_t buf[MAIN_BUFFER_SIZE];
17
18 void setup() {
19     pinMode(relayPin, OUTPUT);
20     digitalWrite(relayPin, relayState);
21     mesh.setNodeID(thisNode);
22     mesh.begin();
23 }
24
25 void loop() {
26     mesh.update();
27     while (network.available()){
28         memset(buf, 0, sizeof(buf));
29         RF24NetworkHeader header;
30         network.read(header, &buf, sizeof(buf));
31
32         String msg = String((char *)buf);
33         if (msg == "OFF")
34             relayState = HIGH;
35         else if (msg == "ON")
36             relayState = LOW;
37         digitalWrite(relayPin, relayState);
38     }
39 }
```

Program začíná připojením potřebných knihoven (`#include`), v tomto případě knihovny pro ovládání nRF24L01+ modulu. Následuje vytvoření objektů, které představují daný komunikační modul. Objekt `radio()` knihovny `RF24` poskytuje základní ovládání modulu přes sběrnici SPI, přičemž vstupní parametry konstruktoru jsou piny CE a CS (resp. piny D9 a D10 na desce Arduino Nano). Dále objekt `network(radio)` knihovny `RF24Network` vytváří jakousi síťovou vrstvu nad rádiovými moduly a vstupním parametrem je tedy objekt `radio`. Tyto objekty dostávají logické adresy, čímž umožňují adresování a již by se daly použít pro adresování senzorů v síti. Jenže tyto adresy jsou v osmičkovém tvaru a při průchodu branou by byla potřeba konverze do desítkové soustavy, kterou používá kontrolér. Proto je přidána ještě další vrstva knihovnou `RF24Mesh`, která sice vytváří ze sítě senzorů topologii typu mesh, ale umožňuje adresy v desítkovém tvaru, což usnadňuje poté přidělování adres v kontroléru.

Následují globální proměnné určující adresu daného senzoru (`thisNode`, která je v tomto případě 2), adresu brány (`gatewayNode`, která je vždy 0), pin, na kterém je připojeno relé (`relayPin`, zvolen pin D5), stav sepnutí relé (`relayState`, defaultně „HIGH“ neboli logická 1, což znamená, že relé není sepnuto, protože spíná na logickou 0) a buffer pro příchozí zprávu od brány (`buf [MAIN_BUFFER_SIZE]`, velikost je dána knihovnou `RF24`).

Funkce `setup()` poté obsahuje nastavení pinu pro relé (nastavení pinu jako výstupu a zapsání stavu relé, což je po spuštění defaultní logická 1). Pak je nastavena adresa tohoto senzoru pomocí objektu `mesh` a spuštěním metody `mesh.begin()` začne modul nRF24L01+ naslouchat zprávám od ostatních senzorů a brány.

Funkce `loop()` začíná metodou `mesh.update()`, která je potřeba být neustále volána, aby síť fungovala, protože v této metodě jsou zprávy přijímány a přeposílány dál. Následuje smyčka `while()` jejíž podmínkou je přítomnost nové zprávy určené pro tento senzor. Pokud je tedy podmínka splněna a přišla nová zpráva, vynuluje se buffer pomocí Arduino funkce `memset()`, vytvoří se nový objekt pro hlavičku zprávy a oba se předají metodě `network.read()`, která je naplní. Dále se převede buffer na datový typ `String`, aby mohl být porovnán obsah zprávy, a pokud zpráva obsahuje příkaz `OFF`, nastaví se stav relé na logickou 1, v případě příkazu `ON` na logickou 0 a tento stav se zapíše na výstup relé.

To je celý program pro tento senzor a ostatní senzory jsou podobné, akorát se liší v použitých knihovnách a proměnných. Princip ale mají stejný: ve funkci `setup()` nastaví piny a komunikační moduly a ve funkci `loop()` vykonávají práci a čtou nebo posílají zprávy bráně. Trochu speciální je program pro senzor s detektorem pohybu. Dříve bylo uvedeno, že potřebuje být připojen na pin D2, a to protože Arduino Nano má na tomto pinu vstup externího zdroje přerušení. Senzor tedy může být uspaný nebo vykonávat jakýkoliv jiný program a v případě zaznamenaného pohybu se probudí nebo přeruší činnost, aby poslal zprávu bráně.

4.2.2 Kód WEMOS LOLIN32

Jelikož je deska WEMOS LOLIN32 kompatibilní s Arduino jazykem, pak přinejmenším struktura firmwaru pro bránu, který je na této desce postaven, je podobná firmwaru senzorů. Oproti senzorům je však o mnoho složitější a delší. Kde předchází kód pro relé senzor měl jenom dvě základní povinné funkce a asi 40 řádků kódu (včetně mezer pro lepší přehlednost; ostatní firmware pro další senzory je na tom podobně), program pro bránu má přibližně 400 řádků kódu (zase včetně mezer a komentářů, které ale odhadem tvoří asi 100 řádků). Tento kód je pak rozdělen do několika pomocných funkcí, protože brána musí

podporovat několik různých komunikačních technologií a také formátovat zprávy mezi kontrolérem a senzory, aby jim koncoví příjemci rozuměli.

Z toho důvodu zde není uveden celý kód, ale pouze pár funkcí pro vysvětlení komunikace. Pro komunikaci s kontrolérem přes MQTT jsou použity následující funkce:

```
1 void recvMQTT(String topic, String message){
2     int i = 0;
3     while(message[i] != ':'){
4         i++;
5     }
6
7     String toNode = message.substring(0,i);
8     String command= message.substring(i + 1);
9
10    if (topic == "gateway/in/RF24"){
11        fromMQTTtoRF24(toNode, command);
12    }
13    else if (topic == "gateway/in/RF433"){
14        fromMQTTtoRF433(toNode, command);
15    }
16    else if (topic == "gateway/in/IR"){
17        fromMQTTtoIR(command);
18    }
19 }
```

Funkce `recvMQTT()` přijímá v parametru MQTT zprávu a téma, z kterého přišla. Zpráva je ve tvaru `Address:Command`, a proto je potřeba nejdříve najít pozici dělicí dvojtečky a zprávu rozdělit pomocí metody `substring` na dva `Stringy`, adresu senzoru a příkaz určený pro něj. Podle tématu se pak rozhoduje, která komunikační technologie se má použít a zavolá se příslušná funkce s předanou adresou a příkazem jako parametry.

```
1 void mqttPublish(const char* topic, String fromNode, String data)
2 {
3     String msg = String("{\}") + fromNode + String("\":") + data +
4         String("}");
5     mqttClient.publish(topic, msg.c_str());
6 }
```

V opačném směru, posílání MQTT zprávy kontroléru, je volána tato jednoduchá funkce `mqttPublish()`, jejímž parametry je téma, adresu zdrojového senzoru a posílaná data. Funkce akorát zformátuje zprávu do JSON formátu a předá ji spolu s tématem metodě `publish()` objektu `mqttClient`, který je vytvořen z knihovny `PubSubClient`, a který se stará o správu MQTT komunikace.

Pro komunikaci se senzory jsou pak podobně krátké funkce, například tyto pro rádiovou komunikaci na frekvenci 433 MHz:

```

1 void recvRF433(){
2     uint8_t len = sizeof(bufRF433);
3     uint8_t from;
4
5     while (rf433Manager.available()){
6         memset(bufRF433, 0, len);
7         rf433Manager.recvfrom(bufRF433, &len, &from);
8         fromRF433toMQTT(from, (char *)bufRF433);
9     }
10 }

```

Funkce `recvRF433()` slouží pro příjem komunikace od senzorů. Nejdříve se vytvoří pomocné proměnné, `len` pro velikost bufferu a `from` pro adresu senzoru. Následuje smyčka `while()`, jejíž podmínkou je přítomnost nové zprávy. Při splnění podmínky se tedy vyprázdní buffer a předá se jako parametr metodě `rf433Manager.recvfrom()` spolu s pomocnými proměnnými. Metoda použije proměnnou `len` jako maximální počet znaků, které má přečíst ze zprávy a naplnit jimi buffer, a proměnnou `from` naplní adresou zdrojového senzoru. Poté je zavolána funkce `fromRF433toMQTT()` s předanou adresou a zprávou jako parametry, která je pak připraví pro výše uvedenou funkci `mqttPublish()`.

```

1 void sendRF433(uint8_t toNode, String message){
2     memset(bufRF433, 0, sizeof(bufRF433));
3     memcpy(bufRF433, message.c_str(), message.length()+1);
4     rf433Manager.sendto(bufRF433, sizeof(bufRF433), toNode);
5 }

```

Funkce `sendRF433()` slouží pro posílání zprávy senzorům. V parametrech dostává adresu senzoru (`toNode`) a zprávu (`message`), kterou mu posílá. Nejdříve vyprázdní buffer od možných předchozích dat a naplní jej zprávou pomocí funkce `memcpy()`. Poté se zavolá metoda `sendto()` objektu `rf433Manager` z knihovny `RadioHead`, který se stará o rádiovou komunikaci, a jako parametry se jí předají zmíněný buffer, jeho velikost a adresa cílového senzoru, který má dostat zprávu.

Ostatní funkce pro komunikaci se senzory jsou víceméně podobné a v principu stejné: nejdříve vyprázdní buffer a naplní jej zprávou. V případě směru od senzorů ke kontroléru se tato zpráva předá funkci připravující MQTT zprávy, v opačném směru (od kontroléru k senzorům) se předá zpráva příslušným metodám objektů, které obstarávají samotnou komunikaci.

Nebyly zmíněny základní funkce `setup()` a `loop()`, protože většina funkcionality brány je právě v těch ostatních funkcích spravující komunikaci a předávání zpráv. Funkce `setup()` obsahuje pouze nastavení WiFi a MQTT pro připojení k WLAN síti a MQTT brokeru, a zapnutí komunikačních modulů, ať mohou začít komunikovat. Funkce `loop()` pak akorát kontroluje jestli nedorazila nějaká nová zpráva na některém z modulů a pokud ano, předá ji pak ostatním funkcím pro zpracování. Je zde poté ještě kontrola spojení s MQTT brokerem a v případě výpadku se zkouší znovu připojit. Jedná se o neblokující pokusy, jednou každých 10 vteřin, takže brána může i tak fungovat, ale bohužel bez MQTT spojení nemůže přijímat ani posílat zprávy kontroléru.

Kapitola 5

Testování

Tato kapitola se zabývá testováním výsledného řešení. Hlavními parametry brány a vůbec celé inteligentní domácnosti je spolehlivost a dosah komunikace mezi jednotlivými zařízeními, a proto je implementovaná brána testována především v těchto vlastnostech. V části **5.1 Dosah komunikace** je tedy testován dosah komunikace pro jednotlivé komunikační technologie. Následně v části **5.2 Spolehlivost komunikace** je pak zkoušena spolehlivost takové komunikace, tedy jestli všechno probíhá v pořádku. Nakonec v poslední části **5.3 Kontrolér openHAB** je shrnut způsob testování pomocí kontroléru openHAB.

5.1 Dosah komunikace

Nejdůležitějším parametrem jakékoliv bezdrátové komunikace je její dosah. V této práci byla použita jak rádiová, tak infračervená komunikace, které se v tomto ohledu velmi liší.

Nejprve tedy infračervená komunikace. Již z principu fungování této technologie je třeba, aby byla přímá viditelnost mezi zdrojem a cílem. U brány je zdrojem infračervená LED dioda a jako cíl byl použit DVD přehrávač. Testování probíhalo v uzavřené místnosti následujícím způsobem: DVD přehrávač byl položen na vyvýšeném místě na jedné straně místnosti a brána byla připojena k notebooku. Notebook byl připojen přes WiFi k WLAN síti s kontrolérem, jehož rozhraní bylo otevřeno ve webovém prohlížeči. Z tohoto rozhraní a následně přes bránu byly posílány stejné IR kódy, jaké jsou na originálním dálkovém ovladači k danému přehrávači. Začalo se ve vzdálenosti asi 2 metrů a postupně se s notebookem a bránou zvětšovala vzdálenost až do přibližně 7 metrů, kdy přehrávač přestal reagovat na příkazy. Z toho vyplývá, že pro vysílání z brány je maximální vzdálenost cca 7 metrů, neboť v místnosti nebyly přítomny žádné možné zdroje rušení infračerveného světla. Při testování také vyšlo najevo, že směr LED diody je velice důležitým faktorem, neboť použitá dioda TSAL6100 má vyzařovací úhel pouze 10° , a tak musí být přesněji namířena na cíl.

Co se týče použití brány jako příjemce, implementovaný modul TSOP4438 je na tom oproti vysílání o mnoho lepší. Testovací způsob byl podobný, akorát teď byla stacionární brána s notebookem a pohyblivým prvkem byl dálkový ovladač. Notebook byl umístěn na okně směrem do zahrady a vzdálenost byla postupně otestována znovu od 2 metrů až po konec zahrady, který byl vzdálen přibližně 20 metrů. Minimálně 20 metrů je tedy vzdálenost pro příjem infračerveného záření, přičemž je nutno podotknout, že i za přítomnosti rušení v podobě slunečních paprsků. Modul TSOP4438 také podporuje širší úhel příjmu, takže není potřeba na něj mířit přesně ani přímo z přední strany.

U rádiové komunikace dost záleží na frekvenci, na které operuje. Nižší frekvence mají lepší penetraci a pokrývají větší vzdálenost, což se hodí zejména v zastavěných prostorách. Záleží ale také na materiálu staveb, ne všechny můžou rádiové vlny projít. Vyšší frekvence tak mají kratší dosah, který ale vynahrazují větší propustností dat a také menšími anténami. V této práci byly použity frekvence 443 MHz a 2,4 GHz a obě byly testovány najednou následovně. Byly vytvořeny dva jednoduché senzory obsahující pouze desku Arduino a každý po jednom komunikačním modulu. Firmware senzorů poté spočíval jednoduše v tom, že každou vteřinu zvyšoval hodnotu proměnné o 1 a danou hodnotu pak posílal kontroléru. Brána byla zase připojena k notebooku a rozhraní kontroléru zobrazeno ve webovém prohlížeči. Senzory byly umístěny v obývacím pokoji 1. patra rodinného domu a notebook s bránou byl postupně přesouván po místnostech a patrech a kontrolovala se změna přijímaných hodnot od senzorů. Takto vyšlo najevo, že rádiová komunikace přes frekvenci 2,4 GHz dosahuje pouze pár metrů do sousedních místností přes jednu stěnu; přes dvě stěny již signál neprošel. Naproti tomu senzor na frekvenci 433 MHz dokázal s bránou komunikovat po celém domě napříč několika stěnami a také rozdílem pater.

Z toho vyplývá, že senzory s 433 MHz moduly můžou pokrýt celý dům a komunikovat s bránou přímo. Pro senzory s 2,4 GHz by pak bylo potřeba vytvořit tzv. opakovače (angl. Repeater), což jsou zařízení pro přeposílání zpráv a zvyšující tak dosah, nebo použít moduly s externí anténou pro zvýšení dosahu. Tyto moduly však nebyly dostupné pro otestování. Navíc právě daná anténa může být překážkou, pokud je omezený prostor při konstrukci senzoru nebo brány (například ochranné pouzdro).

5.2 Spolehlivost komunikace

Během testování dosahu komunikace byla testována také její spolehlivost, tedy jestli dochází všechny zprávy a s jakým zpožděním. Způsob testování zde byl hlavně vizuální a akustický.

V této práci jsou vytvořeny senzory s tzv. relé štítem, který má tu vlastnost, že při sepnutí nebo rozepnutí jde relé jasně slyšet (protože jde o mechanické sepnutí obvodu). Toho bylo využito při testování rádiové komunikace, které tyto senzory používají. Testování pak probíhalo tak, že přes webové rozhraní kontroléru byl poslán příkaz pro sepnutí/rozepnutí relé. Při poslání příkazu přes frekvenci 2,4 GHz byla prodleva velice malá, zvuk sepnutí relé šel slyšet prakticky hned po poslání příkazu. Také byl příkaz vykonán pokaždé, tj. vždycky došel až k cílovému senzoru. Oproti tomu senzor na frekvenci 433 MHz měl zpoždění 0,5-1 s a také ne každý příkaz byl vykonán. Záleželo na rychlosti opakování příkazů a pauz mezi nimi, protože příliš rychlé posílání příkazů za sebou znamenalo, že přibližně každý druhý příkaz byl zanedbán. Vzdálenost přitom na toto zpoždění nebo zanedbávání neměla vliv, ať už s frekvencí 443 MHz nebo 2,4 GHz, zpoždění bylo vždy stejné na jakoukoliv vzdálenost (samozřejmě v dosahu komunikace). Na vině je nejspíš použitá knihovna, které trvá dlouho připravit a poslat zprávu a na druhém konci ji zase přečíst a vykonat, protože se neposílají tak velká data, aby na to 433 MHz nestačilo.

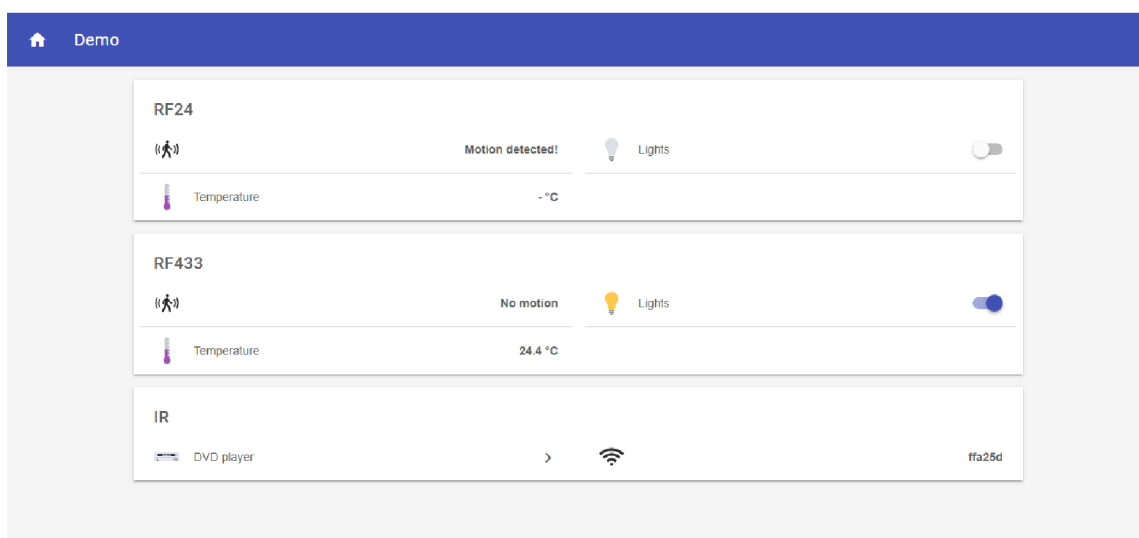
Pro vizuální testování pak byly použity senzory s detektory pohybu a rádiovou komunikací a poté příjem infračerveného kódu z dálkového ovladače. Oba tyto prvky posílaly zprávu kontroléru, který vypsál změnu ve webovém rozhraní, čímž šlo poznat, jak rychle a jestli vůbec zprávy správně chodí. U senzorů s detektorem pohybu byly stejné výsledky jako u relé štítu, tedy že senzor na frekvenci 2,4 GHz reagoval okamžitě a vždycky, druhý senzor na frekvenci 433 MHz pouze se zpožděním a výpadky.

Infračervená komunikace si naproti tomu vedla perfektně. Přijímací modul TSOP4438 byl testován posláním příkazů z dálkového ovladače z několika různých směrů a vzdále-

ností a pokaždé byl přijatý kód okamžitě vypsan v rozhraní kontroléru. Také v případě vysílání reagovala brána okamžitě a v ovládní DVD přehrávače byla stejně rychlá jako originální ovladač. To samozřejmě není překvapivé, protože infračervené záření je pořád záření s rychlostí blízkou rychlosti světla (tj. 299 792 458 m/s). Daný výsledek tak akorát znamená, že použitá knihovna je velice rychlá a nezpomaluje komunikaci.

5.3 Kontrolér openHAB

Testování probíhalo pomocí ovládní a zobrazování dat v kontroléru openHAB, který byl nainstalován na Raspberry Pi 3. Kontrolér openHAB poskytuje grafické rozhraní v několika různých provedeních pro různé způsoby ovládní. Existuje tak například základní GUI (Basic UI) pro ovládní z počítače pomocí klávesnice a myši, moderní GUI (HABPanel) pro dotykové ovládní (hodí se například na tablety) nebo tzv. Paper UI, což je konfigurační rozhraní pro nastavení openHAB (zatím ještě ve vývoji, protože openHAB byl původně nastavován přímo v konfiguračních souborech).



Obrázek 5.1: openHAB - Basic UI

Výše uvedený snímek obrazovky (viz obr. 5.1) zobrazuje uživatelské rozhraní Basic. Jedná se o demo příklad pro testování senzorů použitých v této práci. UI je rozděleno na tři části podle typu komunikační technologie. Rádiová komunikace, jak frekvence 433 MHz, tak 2,4 GHz, obě mají po třech senzorech, od každého typu jeden.

Prvním senzorem je detektor pohybu, který při zaznamenaném pohybu posílá kontroléru zprávu „1“, v klidu „0“ (což odpovídá logické 1 a 0 generované detektorem) a kontrolér na základě přijaté zprávy v pozadí pomocí pravidla zobrazí hlášku „Motion detected!“ nebo „No motion“.

Druhým senzorem je relé, které je zde prezentováno jako vypínač světel. Při zapnutí světla pošle kontrolér senzoru zprávu „ON“, při vypnutí zprávu „OFF“ a firmware relé podle této zprávy relé sepne nebo rozepne.

Třetím senzorem je teplotní čidlo, které je zde zobrazeno jako pouhý formátovaný text, který ukazuje naměřenou teplotu ze senzoru s přidanou značkou stupně Celsia.

Část pro infračervenou komunikaci má pak dva prvky: „DVD přehrávač“ představující dálkový ovladač pro DVD přehrávač (konkrétně výrobce a model GoGEN DXH365) a textový blok zobrazující přijatý infračervený kód. Jak vypadá dálkový ovladač je zobrazeno níže v jiném GUI.



Obrázek 5.2: openHAB - HABPanel

Tento snímek obrazovky (viz obr. 5.2) ukazuje uživatelské rozhraní ve stylu HABPanel. Zobrazen je zrovna rozbalený prvek „DVD přehrávač“, který představuje dálkový ovladač s několika tlačítky. V tomto případě pouze osm tlačítek nejzákladnějších operací pro ovládní videa. V pozadí kontroléru se jedná o jeden předmět s osmi stavy, přičemž každý stav má jiný, předem nastavený infračervený kód (pro výše zmíněný model přehrávače, získaný z originálního dálkového ovladače). Tento předmět lze pohodlně rozšířit o další stavy a přidat tak další tlačítka ovladače. Při stisku tlačítka se aktivuje pravidlo, které podle stavu tlačítka pošle zprávu bráně s daným kódem, která jej vyšle pomocí infra LED diody.

Kapitola 6

Závěr

Tato práce se zabývala problematikou inteligentních domácností. Byly vysvětleny pojmy a zařízení, ze kterých je taková domácnost tvořena a jak funguje. Dále byly uvedeny existující řešení. Cílem práce pak bylo vytvořit bránu a senzory, které tvoří důležitou a podstatnou část inteligentní domácnosti. Hlavními kritérii přitom byla funkcionality, dosah řešení a cenová dostupnost. Z těchto důvodů byly zvoleny otevřené hardwarové a softwarové platformy. Dalším důležitým parametrem inteligentní domácnosti je podpora několika různých komunikačních technologií, aby mohla spolu komunikovat i různá zařízení od různých výrobců postavená na různých technologiích.

Řešení je proto z hardwarové části postaveno na vývojových deskách Arduino Nano pro senzory a WEMOS LOLIN32 pro bránu, spolu s několika komunikačními moduly pro rádiovou a infračervenou komunikaci. Cena takového řešení pak vychází pro bránu přibližně 10 dolarů (cca 210 Kč). U senzorů je cenové rozmezí poměrně dost široké, v závislosti na použité desce (originál nebo klon) a také použitých sensorových modulech, které se liší podle vykonávané práce, a tudíž i cena je jiná. Senzory vytvořené v této práci se pohybují okolo 4 dolarů (cca 80 Kč) díky použitému klonu Arduino desky (originální deska by zvedla cenu o dalších 20 dolarů na 24-25 dolarů, tedy o přibližně 420 Kč na celkových 500 Kč). Cenově je tak výsledné řešení velice přívětivé a dostupné oproti existujícím řešením.

Co se týče dalšího kritéria, funkcionality, brána je funkční a dokáže komunikovat s jakýmkoliv senzorem a kontrolérem za následujících podmínek. Senzory musí podporovat rádiovou komunikaci na frekvencích 433 MHz nebo 2,4 GHz a příslušné knihovny (RadioHead nebo RF24 + RF24Network + RF24Mesh), nebo infračervenou komunikaci na frekvenci 38 kHz a protokoly LG, NEC, Samsung, Sony, Sharp nebo Whynter. Kontrolér pak musí podporovat MQTT komunikaci a umět zpracovat zprávu v JSON formátu. Z testování však vyšlo najevo, že ačkoli je brána funkční, není spolehlivá na 100 % u všech komunikačních technologií, a ne všechny zprávy dorazí do cíle.

Z testování byl také zjištěn dosah komunikace mezi bránou a senzory, který byl pro většinu komunikačních technologií přijatelný. Pouze rádiová komunikace na frekvenci 2,4 GHz poskytovala krátkou vzdálenost, a to i přestože všechny zprávy na této technologii chodily správně a okamžitě. Dosah by se tak dal vylepšit například připojením externí antény, jinak je potřeba použít několik daných komunikačních modulů navíc, které slouží pro přeposílání zpráv, aby šla pokrýt celá domácnost.

Výsledné řešení je tak funkční, avšak pouze v omezené míře a do budoucna by šlo vylepšit jak investicí do hardwaru, tak také do firmwaru, s cílem zvětšit dosah a podporu více komunikačních technologií a senzorů, například s využitím zabudovaného Bluetooth v mikrokontrolerech ESP32.

Literatura

- [1] *2GIG*. 2GIG, [Online; navštíveno 13.05.2018].
URL <https://www.2gig.com/>
- [2] *2GIG - GC2*. 2GIG, [Online; navštíveno 13.05.2018].
URL <https://www.2gig.com/products/control-panels-keypads/gc2-panel/>
- [3] *2GIG - GC3*. 2GIG, [Online; navštíveno 13.05.2018].
URL <https://www.2gig.com/products/control-panels-keypads/gc3-panel/>
- [4] *Arduino*. Arduino, [Online; navštíveno 13.05.2018].
URL <https://www.arduino.cc/en/Guide/Introduction>
- [5] *Arduino UNO*. Arduino, [Online; navštíveno 13.05.2018].
URL <https://store.arduino.cc/arduino-uno-rev3>
- [6] *Domoticz*. Domoticz, [Online; navštíveno 13.05.2018].
URL <https://www.domoticz.com/>
- [7] *Domoticz - About*. Domoticz, [Online; navštíveno 13.05.2018].
URL https://www.domoticz.com/wiki/About_Domoticz
- [8] *DS18B20*. Maxim Integrated, [Online; navštíveno 13.05.2018].
URL <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [9] *ESP32*. Espressif, [Online; navštíveno 13.05.2018].
URL https://www.espressif.com/sites/default/files/documentation/esp-wroom-32_datasheet_en.pdf
- [10] *ESP8266*. Espressif, [Online; navštíveno 13.05.2018].
URL https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- [11] *FIBARO*. FIBARO, [Online; navštíveno 13.05.2018].
URL <https://www.mojefibaro.cz/>
- [12] *FIBARO - Produkty*. FIBARO, [Online; navštíveno 13.05.2018].
URL <https://www.mojefibaro.cz/produkty/>
- [13] *HC-SR501*. [Online; navštíveno 13.05.2018].
URL <https://laskarduino.cz/vstupni-periferie-cidla/131022-pir-detektor-pohybu-hc-sr501.html>

- [14] *JQC-3FF-S-Z*. [Online; navštíveno 13.05.2018].
URL <https://www.generationrobots.com/media/JQC-3FF-v1.pdf>
- [15] *MQTT*. MQTT, [Online; navštíveno 13.05.2018].
URL <https://mqtt.org/>
- [16] *MySensors*. MySensors, [Online; navštíveno 13.05.2018].
URL <https://www.mysensors.org/>
- [17] *MySensors - Network*. MySensors, [Online; navštíveno 13.05.2018].
URL <https://www.mysensors.org/about/network>
- [18] *NEC protokol*. Altium, [Online; navštíveno 13.05.2018].
URL <https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol>
- [19] *nRF24L01+*. Nordic Semiconductor, [Online; navštíveno 13.05.2018].
URL https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf
- [20] *openHAB*. openHAB, [Online; navštíveno 13.05.2018].
URL <https://www.openhab.org/>
- [21] *openHAB - Bindings*. openHAB, [Online; navštíveno 13.05.2018].
URL <https://docs.openhab.org/addons/bindings.html>
- [22] *openHAB - Configuration*. openHAB, [Online; navštíveno 13.05.2018].
URL <https://docs.openhab.org/configuration/index.html#versatility>
- [23] *openHABian*. openHAB, [Online; navštíveno 13.05.2018].
URL <https://docs.openhab.org/installation/openhabian.html>
- [24] *openHAB - Rules*. openHAB, [Online; navštíveno 13.05.2018].
URL <https://docs.openhab.org/configuration/rules-dsl.html#the-syntax>
- [25] *SRX882*. Nice RF, [Online; navštíveno 13.05.2018].
URL <https://www.gme.cz/data/attachments/dsh.772-295.1.pdf>
- [26] *STX882*. Nice RF, [Online; navštíveno 13.05.2018].
URL <https://www.gme.cz/data/attachments/dsh.772-294.1.pdf>
- [27] *TSAL4838*. Vishay, [Online; navštíveno 13.05.2018].
URL <https://www.vishay.com/docs/82459/tsop48.pdf>
- [28] *TSAL6100*. Vishay, [Online; navštíveno 13.05.2018].
URL <https://www.vishay.com/docs/81009/tsal6100.pdf>
- [29] *Vera*. Vera, [Online; navštíveno 13.05.2018].
URL <http://getvera.com/>
- [30] *Vera - Compatibility*. Vera, [Online; navštíveno 13.05.2018].
URL <http://getvera.com/compatibility/>
- [31] *Vera - FAQ*. Vera, [Online; navštíveno 13.05.2018].
URL <http://getvera.com/faq/>

- [32] WEMOS. WEMOS, [Online; navštíveno 13.05.2018].
URL <https://www.wemos.cc/>
- [33] WEMOS LOLIN32. WEMOS, [Online; navštíveno 13.05.2018].
URL <https://wiki.wemos.cc/products:lolin32:lolin32>
- [34] Barragán, H.: *The Untold History of Arduino*. [Online; navštíveno 13.05.2018].
URL <https://arduinohistory.github.io/>
- [35] Florian: *OpenMQTTGateway*. GitHub, [Online; navštíveno 13.05.2018].
URL <https://github.com/1technophile/OpenMQTTGateway>
- [36] Růžek, M.: *Přijímač/vysílač dálkového ovládní*. FITkit, [Online; navštíveno 13.05.2018].
URL http://merlin.fit.vutbr.cz/FITkit/docs/aplikace/apps_demo_infra.html

Přílohy

Příloha A

Obsah CD

- /src/ - zdrojové kódy implementace řešení této práce
 - /src/gateway/ - zdrojový kód pro bránu
 - /src/sensors/ - zdrojové kódy pro senzory
- /thesis/ - zdrojové soubory pro vytvoření tohoto dokumentu
 - /thesis/latex/ - soubory pro systém LaTeX
 - /thesis/xsezim00.pdf - tato práce ve formátu PDF