VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019 Bc. Tomáš Resler

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH DOMÁCÍ BRÁNY PRO ZAŘÍZENÍ IOT VYUŽÍVAJÍCÍ TECHNOLOGII Z-WAVE
HOME GATE FOR IOT DEVICES USING Z-WAVE TECHNOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Tomáš Resler**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Jiří Hošek, Ph.D.**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Tomáš Resler ***ID:*** 153610
Ročník: 2 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Návrh domácí brány pro zařízení IoT využívající technologii Z-Wave

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce bude vytvoření domácí brány pro zařízení IoT komunikujících pomocí technologie Z-Wave. Výsledný prototyp bude koncipován jako rozšiřující modul pro jednodeskový počítač Raspberry Pi. Vytvořená deska bude osazena bezdrátovými moduly pro technologii Z-Wave, které budou umožňovat šifrovanou komunikaci v obou směrech. Pro usnadnění konfigurace komunikačních modulů i připojených zařízení bude deska osazena i komunikačním modulem pro Wi-Fi nebo Bluetooth a bude vytvořeno uživatelské rozhraní přístupné skrze webový prohlížeč umožňující vzdálenou správu (přidání, odebrání, konfigurace nastavení, změna šifrovacího klíče) a monitoring (výčet hodnot, interakce se zařízením) těchto zařízení. Data získaná z chytrých senzorů budou ukládána v databázi. Použitý databázový systém bude zvolen s ohledem na formát přijímaných dat a omezený výpočetní výkon použitého jednodeskového počítače. Získaná data bude možné vizualizovat v reálném čase, ale i nad databází dat. Závěrem bude navržena deska plošných spojů s komunikačními moduly pro osazení na zvoleném typu jednodeskového počítače. Funkčnost vytvořené platformy bude ověřena s komerčně dostupnými zařízeními.

DOPORUČENÁ LITERATURA:

[1] BOSWARTHICK, David, Omar ELLOUMI a Olivier HERSENT. 2012. M2M communications: a systems approach. Hoboken, N.J.: Wiley, xxiii, 308 p. ISBN 978-1-119-99475-6.

[2] Safer, smarter homes start with Z-Wave. [online] 2016 [cit. 2016-12-14] Silicon Laboratories. Dostupné z: https://www.z-wave.com/learn

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 16.5.2019

Vedoucí práce: doc. Ing. Jiří Hošek, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:
Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práce je věnována platformě Z-Wave, která se snaží o vytvoření univerzálního systému pro chytrou domácnost. V teoretické části je detailně popsána vlastní platforma, vhodné knihovny pro tvorbu aplikace i dostupná koncová zařízení. V praktické části je pomocí Z-Wave modulu, Raspberry Pi 3, dotykové obrazovky, powerbanky a těla, navrženého v SolidWorksu a vytisknutého na 3D tiskárně, sestrojen hardware brány. Do modulu ZM5304 je nahraný firmware a tento model je pomocí navržené desky plošných spojů propojen s Raspberry Pi 3. Takto sestrojená brána řídí Z-Wave síť pomocí vlastní aplikace napsané v Python wrapperu nad knihovnou OpenZWave. Součástí práce je i souhrn alternativních komerčních zařízení dostupných na českém trhu.

KLÍČOVÁ SLOVA

Z-Wave, Internet věcí, Chytrá domácnost, Brána, Home Assistant, OpenHAB, OpenZWave, Python, Kivy, SQLite, 3D tisk, SolidWorks, Bezpečnost, Raspberry Pi 3, ZM5304

ABSTRACT

This dissertation deals with the platform Z-Wave. This platform tries to create an universal system for smart home. The theoretical part of the work describes in details the platform itself as well as the applicable libraries for programming of applications and the available equipments. The practical part presents the new custom gateway. The hardware of this gateway is designed with utilization of the Z-Wave module, Raspberry Pi 3 microcomputer, touchscreen, powerbank and the body of the system. The body has been designed in the Solid Works CAD system and printed on a 3D printer. The needed firmware was loaded into the Z-Wave module ZM5304 and this module has been interconnected with the Raspberry Pi 3 through the printed circuit board of the custom design. The gateway drives OpenZWave network through the new own application written with help of Python wrapper of the liberary OpenZWave and other open source components. The part of the work is also the list and description of the alternative commercial gateways available on the Czech market.

KEYWORDS

Z-Wave, IoT, SmartHome, Gateway, Home Assistant, OpenHAB, OpenZWave, Python, Kivy, SQLite, 3D printing, SolidWorks, Security, Raspberry Pi 3, ZM5304

RESLER, Tomáš. *Návrh domácí brány pro zařízení IoT využívající technologii Z-Wave*. Brno, 2019, 74 s [cit. 2019-05-16]. Dostupné také z: `https://www.vutbr.cz/studenti/zav-prace/detail/118218`. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Jiří Hošek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Návrh domácí brány pro zařízení IoT využívající technologii Z-Wave“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu PhDr. Janu Krejčímu a doc. Ing. Jiřímu Hoškovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora

Tato práce vznikla jako součást klíčové aktivity KA6 - Individuální výuka a zapojení studentů bakalářských a magisterských studijních programů do výzkumu v rámci projektu OP VVV Vytvoření double-degree doktorského studijního programu Elektronika a informační technologie a vytvoření doktorského studijního programu Informační bezpečnost, reg. č. CZ.02.2.69/0.0/0.0/16_018/0002575.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

Obsah

Úvod **12**

1 Internet věcí **13**

2 Technologie Z-Wave **14**
2.1 Z-Wave Protokol . . . 14
2.1.1 Z-Wave síť . . . 14
2.1.2 Identifikační kód . . . 15
2.1.3 Třídy příkazů . . . 15
2.1.4 Správa energie . . . 16
2.1.5 Bezpečnostní vrstva . . . 17
2.2 Komunikace . . . 17
2.2.1 Párování zařízení . . . 18
2.2.2 Výměna dat Z-Wave . . . 18
2.3 Z-Wave Aliance . . . 18
2.3.1 Plné členství . . . 19
2.3.2 Přidružené členství . . . 19
2.3.3 Integrované členství . . . 19
2.4 Z-Wave Plus . . . 19
2.4.1 Série 700 . . . 19

3 Dostupné knihovny **21**
3.1 Z-Wave>Me . . . 21
3.2 Home Assistant . . . 22
3.2.1 Hass.io . . . 22
3.3 OpenHAB . . . 22
3.3.1 Z-Wave vazba . . . 22
3.3.2 OpenHABian . . . 23
3.4 OpenZWave . . . 23
3.4.1 Knihovna . . . 23
3.4.2 Aplikace . . . 24
3.4.3 Nejnovější verze 1.6 . . . 24
3.4.4 Licence . . . 25
3.4.5 Python obal pro OpenZWave . . . 25
3.5 Z-Wave software . . . 26
3.5.1 SDK pro kontrolery . . . 26
3.5.2 SDK pro koncová zařízení . . . 27

3.5.3 Pomocné vývojové aplikace a knihovny . . . 29
3.6 Výběr knihovny . . . 31
3.7 Další použité knihovny . . . 31
3.7.1 Python-OpenZWave . . . 31
3.7.2 Kivy . . . 32
3.7.3 SQLite . . . 32

4 Z-Wave zařízení na českém trhu 34
4.1 Brány . . . 34
4.1.1 Fibaro . . . 34
4.1.2 Fifthplay . . . 35
4.1.3 Zipato . . . 35
4.1.4 Vera Control . . . 35
4.1.5 Devolo . . . 36
4.1.6 Popp . . . 36
4.2 Senzory . . . 36
4.2.1 Pohybové senzory . . . 37
4.2.2 Senzory kouře . . . 37
4.2.3 Ostatní senzory . . . 38
4.3 Akční členy . . . 38
4.4 Shrnutí . . . 39

5 Zařízení využitá k vytvoření Z-Wave sítě 40
5.1 Raspberry Pi 3 . . . 40
5.1.1 Externí 40 PIN GPIO . . . 40
5.1.2 UART . . . 41
5.2 Z-Wave Static Controller `(ACC-UZB2-E)` . . . 41
5.3 Z-Wave modul ZM5304 s anténou . . . 42
5.3.1 Konfigurace vývodů . . . 42
5.3.2 Programovací mód . . . 42
5.4 CP2102-MINI . . . 43
5.5 Aeotec MultiSensor 6 - ZW100 . . . 43
5.5.1 Aktualizace firmwaru . . . 44
5.6 Fibaro Wall Plug - FGWPE-102 . . . 45
5.6.1 Ovládání . . . 46
5.6.2 Třídy . . . 46

6 Realizace Z-Wave sítě 48
6.1 Instalace Raspian OS (Operating System) . . . 48
6.1.1 NOOBs . . . 48

6.1.2 Pomocí obrazu disku 48
6.2 Nahrání firmwaru do ZM5304 49
6.3 Připojení ZM5304 k Raspberry Pi 3 pomocí UART 50
6.4 Instalace OpenZWave 51
6.4.1 Control Panel 51
6.4.2 Připojení modulu ZM5304 a Multisenzoru 6 52
6.4.3 Vyčtení dat z Multisenzoru 53
6.5 Řešení pomocí jiných knihoven 53
6.5.1 Instalace Home Assistant 53
6.5.2 Instalace OpenHABian 53

7 Vývoj vlastní brány **55**
7.1 Návrh těla brány v programu SolidWorks 55
7.2 Konstrukce brány 56
7.2.1 Instalace dotykové LCD obrazovky 57
7.3 Vývoj vlastní aplikace 57
7.3.1 Python OpenZWave 57
7.3.2 PyCharm 58
7.3.3 Kivy 58
7.3.4 Import knihoven 59
7.3.5 Grafické rozhraní v Kivy 60
7.3.6 Nastavení Z-Wave komunikace 60
7.3.7 Připojení k síti 61
7.3.8 Zachytávání signálů 61
7.3.9 Připojení uzlu do sítě 61
7.3.10 Vyčtení dat ze zařízení a ovládání zásuvky 62
7.3.11 Lambda funkce 63
7.3.12 SQLite a Phyton 63
7.3.13 Zobrazení grafu v Kivy 65
7.4 Uživatelské rozhraní aplikace 65
7.4.1 Obrazovka pro připojení k síti 65
7.4.2 Hlavní obrazovka 65
7.4.3 Obrazovka informací o uzlu 66
7.4.4 Obrazovka grafu 67

8 Závěr **68**

Literatura **69**

Seznam symbolů, veličin a zkratek **73**

Seznam obrázků

2.1 Smíšená Z-Wave topologie 15
2.2 Z-Wave formát rámců 18
3.1 Uživatelské rozhraní Z-Wave>Me 21
3.2 SDK pro kontrolery 26
3.3 Z-Wave Programmer 30
5.1 40 kolíkové GPIO 41
5.2 Schéma vývodů modulu ZM5304 (pohled z vrchu) 42
5.3 Schéma CP2102n-mini - vrchní pohled 43
5.4 Multisenzor 6 44
5.5 Aeon Labs OTA 45
5.6 Wall Plug 46
6.1 Realizovaná Z-Wave síť 48
6.2 Propojení ZM5304, CP2102n a PC 49
6.3 Deska plošných spojů pro ZM5304 50
6.4 Open zwave control panel 52
6.5 Home Assistant 54
6.6 OpenHAB 54
7.1 Tělo navrženého tabletu (vrchní pohled) 55
7.2 Konstrukce navržené brány (spodní pohled) 56
7.3 PyCharm interpretr 59
7.4 Obrazovka pro připojení k síti 66
7.5 Hlavní obrazovka 66
7.6 Obrazovka informací o uzlu 67
7.7 Obrazovka grafu vybrané hodnoty vybraného uzlu 67

Seznam tabulek

4.1 Přehled bran na českém trhu 34
4.2 Přehled pohybových senzorů na českém trhu 37
4.3 Přehled kouřových senzorů na českém trhu 37
4.4 Přehled měřících senzorů na českém trhu 38
4.5 Přehled akčních členů na českém trhu 38

Úvod

Tato diplomová práce se věnuje oblasti IoT (Internet of Things). Konkrétně si na základě poskytnuté dokumentace bere za cíl vytvoření domácí brány umožňující komunikaci v rámci Z-Wave infrastruktury[1].

Aliance zaštitující vývoj Z-Wave platformy funguje již od roku 2005 a snaží se o vytvoření otevřeného protokolu pro automatické řízení chytré domácnosti. Výhodou tak je možnost využití protokolu jakýmkoliv výrobcem. Důsledkem je to, že se tímto způsobem dostávají na trh výrobky zaměřené jak na kvalitu tak i na cenu. O certifikaci výrobku se stará Z-Wave Plus.

V rámci teoretické části byla popsána komunikační platforma Z-Wave, knihovny a zařízení dostupné na českém trhu. Z-Wave síť využívá speciální frekvence a potřebuje tak alespoň jeden řídící člen sítě. V současné době si každý výrobce vyvíjí svoji aplikaci pro řízení Z-Wave sítě. Ty jsou kompatibilní i s jinými Z-Wave výrobky. Rovněž existují i projekty s otevřeným kódem, který se snaží o spojení různých platforem do jednoho řídícího centra (OpenHAB, Home Assistant, ...). Knihovna OpenZWave se pak snaží o podporu co nejvíce zařízení tak, aby si jakýkoliv uživatel mohl jednoduše vytvořit svojí aplikaci podporující Z-Wave protokol.

V praktické části této práce byl nahrán firmware do modulu ZM5304 od Silicon Labs. Tento modul byl poté pomocí UART (Universal Synchronous / Asynchronous Receiver and Transmitter) připojen k Raspberry Pi 3. Knihovnou OpenZWave byl modul ZM5304 nastaven jako řídící kontroler a spojen s Multisenzorem 6. Z něj byla vyčtena data o teplotě, vhlkosti, vibracích, světelnosti, ultrafialovém záření a pohybu. Připojen byl také akční prvek Wall Plug od firmy Fibaro, který slouží jako jednoduchá zástrčka, kterou lze dálkově vypínat a zapínat. Vytvořený systém umožňuje i využití šifrované komunikace mezi zařízeními.

K Raspberry Pi 3 byla připojena dotyková obrazovka a v programu Eagle navržena a vyrobena deska plošných spojů pro modul ZM5304. Tento komplet byl umístěn do těla brány, navrženého v programu SolidWorks a vytištěného na 3D tiskárně. Celý systém byl připojen k baterii tak, aby vzniklo jednoduché dotykové ovládací zařízení pro domácí bránu.

V diplomové práci bylo využito Python obalu pro OpenZWave[2] a prostředí Kivy[3] pro naprogramování jednoduchého grafického prostředí pro ovládání této domácí brány a připojených Z-Wave zařízení. V prostředí bylo také využito SQLite databáze pro časový sběr dat jednotlivých uzlů a možnost následného zobrazení v grafech. To může uživateli posloužit například pro lepší přehled o teplotě v průběhu dne.

[1] https://www.z-wave.com/

[2] https://github.com/OpenZWave/python-openzwave/

[3] https://kivy.org/

1 Internet věcí

Myšlenkou IoT (Internet of Things) je vzájemné propojení elektronických zařízení a získání dalších možností jejich využití. Aplikační oblasti IoT jsou například v dopravě, průmyslových firmách, zdravotnictví nebo chytré domácnosti. V chytré domácnosti obvykle vše funguje na bázi lokální drátové či bezdrátové sítě s centrálním (řídícím) modemem. Přístroje pro chytré domácnosti vyvíjí a vyrábí celá řada společností včetně Applu, Googlu, Samsungu nebo Amazonu. Dále existují aliance, které se snaží o vytvoření univerzálního ekosystému pro všechny ostatní výrobce. Jednou z takových je i Z-Wave Aliance, které se věnuje tato práce. Její podrobný popis začíná od kapitoly 2.

Příkladem konkurečního řešením je platforma KNX (Konnex). Formálně se jedná o sdružení tří systémů pro domácí automatizaci (European Installation Bus, European Home Systems Protocol, BatiBus). Pro přenos dat může využívat kroucené dvojlinky s rychlostí 9600 bit/s, PLC (PowerLine Communication) s centrální nosnou 110 kHz a rychlostí 1200 kbit/s, RF (Radio Frequency) na pásmu 868,3 MHz, infračervené záření nebo IP (Internet Protocol) médií jako ethernet, bluetooth, WiFi nebo FireWire. Je nezávislá na síťové topologii (hvězdicová, stromová, kruhová,...) [1].

Velice známá je i platforma Zigbee. O vývoj se stará Zigbee aliance založená v roce 2002. Za vývojem stojí firmy jako Huawei, Texas Instruments či Silicon Labs. Zigbee je kompletní IoT řešení založené na smíšené topologii. Snaží se o vytvoření univerzálního standardizovaného jazyka tak, aby všechna chytrá zařízení mohla spolupracovat. Nejnovější Zigbee 3.0 je vybudované na Zigbee PRO, který rozšiřuje standard IEEE 802.15.4 (Institute of Electrical and Electronics Engineers) přidáním síťových a aplikačních vrstev. Pracuje v 2,4 GHz ISM (Industrial, Scientific and Medical) pásmu. Využívá 16 kanálů s šířkou pásma 2 MHz. Umožňuje také provoz v pásmu 915 MHz a 868 MHz. Využívá bezpečnostní schéma AES-128 (Advanced Encryption Standard) jak na síťové, tak na aplikační úrovni. Zigbee síť může mít až 240 koncových zařízení[43].

Bezdrátová technologie EnOcean pod hlavičkou EnOcean aliancí vyvíjí kompaktní rádiové moduly, které garantují spolehlivost a bezdrátový přenos bez nutnosti užití baterie nebo externího zdroje. Moduly využívají bezdrátový standard EnOcean (ISO/IEC 14543-3-1X), s dosahem až 30 metrů. V evropě běží na frekvenci 868 MHz s rychlostí přenosu 125 kbit/s. Zařízení získávají energii z nepatrných změn pohybu, ze světla nebo z teploty[44].

Další své platformy si vyvíjejí přední světové firmy. Rozhodně nelze opomenout Apple se svým HomeKit, Google Assistant nebo Amazon Alexa.

2 Technologie Z-Wave

Z-Wave je kompletní řešení pro Internet věcí, která obsahuje veškeré definice protokolů pro síťovou, transportní a aplikační vrstvu. Pomocí platformy lze vytvořit senzory, ovladače, řídící moduly směrovače a brány, které jsou schopny pracovat v síti Z-Wave a poskytnout tak službu pro chytrou domácnost nebo kancelář[6].

Technologie Z-Wave je postavena na smíšené síťové topologii a k určení nejrychlejší trasy pro doručené zprávy využívá zdrojové směrování. Zdrojové směrování funguje na linkové vrstvě. Odesílatel nejdříve rozpošle záplavovou (broadcast) zprávu ke zjištění cesty k cíli. Pokud je příjemce dosažitelný, vrátí se odesílateli zpráva ve formě cesty. V Z-Wave síti je každé zařízení opatřeno identifikačním kódem. Při připojení do sítě kontroler rozpozná kód, určí jeho polohu a tím cestu pro zprávy.

Protože by tento způsob komunikace mohl zařízením zabírat spoustu paměti, Z-Wave vyvinula hierarchii mezi zařízeními, kdy některá fungují jako iniciátoři zpráv a další jako podřízená zařízení, která mohou jen reagovat na zaslané zprávy [5].

Dodnes neexistuje open-source implementace směrování Z-Wave. Oproti tomu existuje OpenZWave (kapitola 3.4), která je open source implementací ovladače Z-Wave pro PC (Personal Computer). Vlastní logika směřování se pak nachází přímo ve firmwarech jednotlivých Z-Wave vysílačů[6].

2.1 Z-Wave Protokol

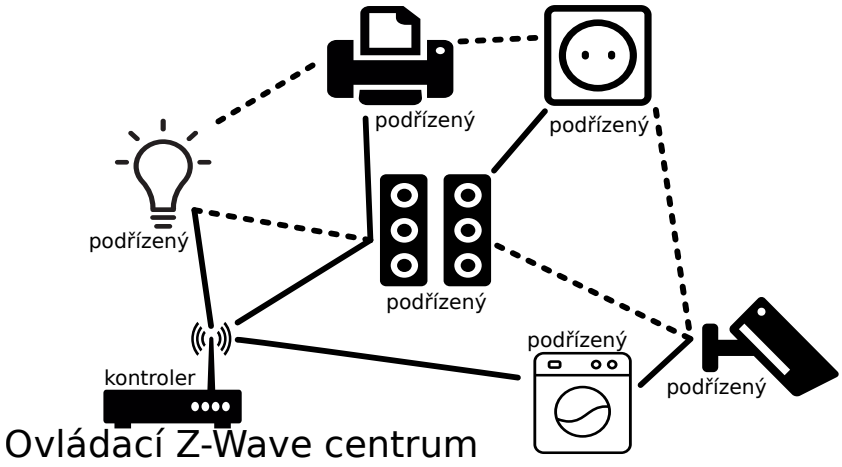
Z-Wave je spolehlivá bezdrátová technologie navržená a vyvýjená speciálně pro domácí automatizaci. Na rozdíl od jiných standardů, které se spoléhají na silně přetížené pásmo 2,4GHz a 5GHz, kde se nachází WLAN (Wireless Local Area Network), Z-Wave zařízení používá Z-Wave pásmo pod 1GHz, a tím je šance na rušení mnohem nižší.

2.1.1 Z-Wave síť

Jak bylo zmíněno, protokol využívá smíšenou síťovou topologii (viz obr. 2.1), každé zařízení v síti je tak známé jako uzel. Jedna Z-Wave síť může obsahovat maximálně 232 uzlů. Pokud je zapotřebí více uzlů, je zapotřebí připojit další kontroler, který se musí aplikačně propojit s původním kontrolerem.

Uzly jsou dvojího typu: kontrolery a podřízené uzly. Řídící jednotky jsou obvykle ve formě ručních dálkových ovladačů nebo rozhraní v počítačích. Přepínače, stmívače, snímače pohybu apod. jsou všechno podřízené uzly.

Iniciátor (zdrojové zařízení, které chce vysílat) hledá pro svoji cestu vždy nejkratší aktivní cestu a snaží se obcházet překážky. Tato cesta může být znatelně delší,



Obr. 2.1: Smíšená Z-Wave topologie

než přímá cesta iniciátor-příjemce. Síť je dynamická entita, to znamená, že zařízení mohou být přidána nebo odstraněna kdykoliv. Následně se pak využije systému zpětného upozornění, který informuje veškerá zařízení o změně sítě[18].

2.1.2 Identifikační kód

Identifikační kód zařízení se dá rozdělit do dvou částí. První část je čtyřbajtový domovský identifikátor. Tento kód mají totožná všechna zařízení konkrétní sítě. Kód tak definuje síť samotnou. Druhá část je identifikátor uzlu. Ten je představován hodnotou bajtu, kterou kontroler přiřadí jednotlivému uzlu při párování zařízení. Zde platí, že kontroler má vždy nejnižší číslo 1 a další zařízení čísla následující[6].

2.1.3 Třídy příkazů

Sítě Z-Wave se skládají z kontrolerů a podřízených uzlů, které mohou být buď ve formě senzorů nebo akčních členů. Pro zajištění kompatibility je protokol aplikační vrstvy přesně definován[6].
Třídy příkazů jsou rozděleny do čtyř kategorií:

- Aplikační.
- Řídící.
- Transportně-zapouzdřovací.
- Vrstvy síťového protokolu[11].

Není potřeba, aby každé zařízení umělo všechny příkazy aplikační vrstvy, protože například žárovka nemá implementováno rozhraní pro odečet teploty. Proto je aplikační vrstva rozdělena do tříd příkazů. Třída pak má podmnožinu funkčních příkazů. Dané zařízení patří do určitě třídy (či více tříd) aplikační vrstvy, které se pak kompilují do firmwaru zařízení. Seznam definovaných tříd příkazů je dostupný na stránkách

Silicon Labs[1]. Během párování pak zařízení oznámí svoji podporovanou sadu tříd. Každá třída je jedinečná hodnota bitu. Například spínač světel může oznámit, že používá třídu příkazů `Binary Switch`, která poskytuje příkazy pro získání, nastavení a hlášení stavu přepínače. Příkazy odeslané přepínači světla z nepodporovaných tříd příkazů jsou ignorovány[6].

Zařízení typu podřízený většinu času spí, aby šetřila baterie. Příkazy ke změně mohou příjmat jen v situaci, kdy jsou vzhůru a tak tyto příkazy nemusí proběhnout okamžitě[18].

2.1.4 Správa energie

Pro zařízení, která jsou napájena pomocí baterií a je u nich potřeba docílit malé spotřeby a dlouhé životnosti, vyvinula Z-Wave Aliance speciální řízení spotřeby energie. Spotřebu přitom určuje jak hardware tak software zařízení. Většinu času jsou přitom senzory v režimu spánku a je důležité, aby v tomto stavu spotřebovávali minimum energie. Běžná zařízení platformy Z-Wave spotřebovávají v režimu spánku méně než 1 µA za hodinu. Pro účely spotřeby se přitom rozdělují senzory na dva typy. Prvním typem jsou senzory, které jsou typicky aktivovány pohybem (například okení/dveřní senzory). U těchto senzorů je potřeba, aby se uměly rychle probudit a znovu uspat. Druhým typem senzorů jsou takové, které jsou spouštěny aktivitou na síti (například motor garážových vrat). Tyto senzory potřebují stále naslouchat aktivitě v síti, umět do jedné vteřiny reagovat, a přitom vydržet fungovat na baterii i několik let. O to se stará speciálně vyvinutá technologie FLiRS (Frequently Listening Routing Slave). Ta se snaží o to, aby tyto senzory odebírali méně než 40 µA za hodinu. [32].

FLiRS

FLiRS je Z-Wave standard a pro vývojáře byl uvolněný v červenci roku 2013. Jejím implementováním do podřízených zařízení, která naslouchají v intervalu jedné vteřiny, dojde ke snížení spotřeby v průměru na 50 µA. To je jednou ze silných stránek využití Z-Wave oproti WiFi, a to zejména u zařízení jako jsou termostaty či bezpečnostní zámky. Aby technologie WiFi měla stejně malou spotřebu, musela by prodloužit interval naslouchání na desítky minut. To je například právě u bezpečnostních zámků nepřijatelné.

Zařízení s implementací FLiRS střídá dva režimy: režim spánku a režim částečného probuzení. Tyto dva intervaly střídá v rozmezí od jedné do čtyř vteřin (podle

[1] `https://www.silabs.com/documents/login/miscellaneous/SDS13548-List-of-defined-Z-Wave-Comman xlsx`

návrhu vývojáře). V režimu částečného probuzení naslouchá signálu k úplnému probuzení. Pokud takový signál od některého uzlu sítě dostane, ihned se úplně probudí a komunikuje s daným uzlem standardními Z-Wave příkazy. Po delší nečinosti pak znovu přejde do režimu spánku[33].

2.1.5 Bezpečnostní vrstva

Tato vrstva patří mezi volitelné. Bezpečnostní vrstva poskytuje pomocí AES-128 (Advanced Encryption Standard) důvěrnost, autentizaci a odolnost proti útokům. Kontroluje se integrita zpráv (přijímač si musí být jistý tím, že zpráva byla vyslána bezpečným uzlem a nebyla změněna), důvěrnost zpráv (přečíst zprávu mohou jen ověřené uzly sítě) a čerstvost dat (zpráva byla odeslána nedávno). Zabezpečení funguje na principu uzel-uzel, kdy si uzly nejprve navzájem vymění symetrický klíč AES-128 a poté komunikují pomocí této šifry. Nezabezpečené uzly pak mohou fungovat jako opakovače pro tyto bezpečné uzly. Bezpečnostní řešení se skládá ze dvou úrovní:

- Aplikační úroveň poskytovaná výrobcem OEM.
- Úroveň aplikací, kterou poskytuje společnost Silicon Labs.

Každá z těchto částí zahrnuje možnost prevence, detekce a reakce na řadu možných útoku:

- Prevenci představují - to jsou technologie a systémy, které mají lidem zabránit v získávání informací.
- Detekce se snaží zjistit, zda někdo získal přístup a ohrozil důležité informace nebo procesy.
- Reakce umožňuje identifikaci „nebezpečných uzlů“ a snaží se jejich činost zastavit[8].

Bezpečnostní třída příkazů definuje řadu příkazů pro usnadnění a zpracování šifrovaných rámců v síti Z-Wave. Příkazy se zabývají třemi hlavními oblastmi:

- Zapouzdření.
- Správa příkazových tříd.
- Správa síťových klíčů.

Přímo autor konkrétní aplikace určuje, zda a které příkazy jsou podporovány pomocí zapouzdřené komunikace[9],

2.2 Komunikace

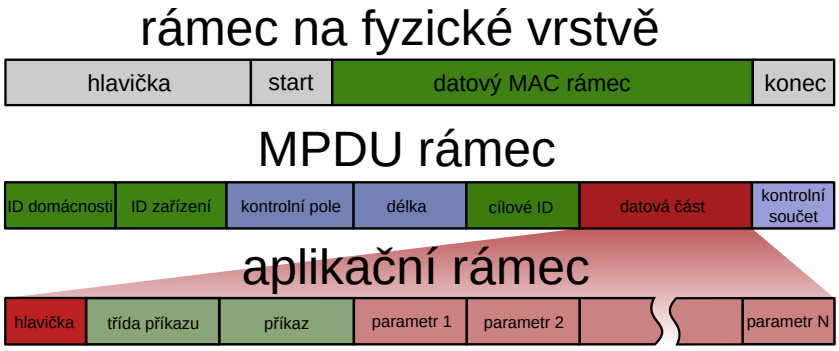
Komunikace protokol Z-Wave využívá celou řadu mechanizmů zajišťujících co možná největší spolehlivost sítě. Je navržený tak, aby se vždy pokousil o doručení rámce adresátovi, a to i když dojde ke změně sítě nebo přestanou být některé uzly dostupné[10].

2.2.1 Párování zařízení

Způsob zařazování přístrojů je podobný párování bluetooth zařízení. Uživatel nejdříve nastaví ovladač a nové zařízení do režimu párování, což se obvykle docílí pomocí vypínacího tlačítka nebo vynulováním zařízení. Vybrat pak lze konkrétní zařízení, které chce uživatel do sítě přidat, případně přidat všechna zařízení v dosahu[6].

2.2.2 Výměna dat Z-Wave

Na fyzické vrstvě se zprávy asynchroně vymění přes RF medium jako rámce MPDU (MAC Protocol Data Unit). MPDU obsahuje hlavičku s identifikačním a kontrolním polem, dále pak vrstvu aplikačních příkazů, zpráv a dotazů. Mezi hlavičkou MPDU a aplikační vrstvou mohou být další volitelné vrstvy, včetně vrstvy sítě a zabezpečení (viz obr. 2.2)[6].



Obr. 2.2: Z-Wave formát rámců

Pro bezdrátový přenos využívá Z-Wave bezlicenční frekvenční pásmo ISM (Industrial, Scientific and Medical) s frekvencí mezi 868.4 MHz a 956 MHz. Vždy záleží na konkrétním regionu, přičemž v rámci Evropské unie jsou dány dva kanály na frekvenci 868.4 MHz s rychlostí 9.6/40 kbit/s a šířkou pásma 300 kHz a na frekvenci 869.85 MHz s rychlostí 100 kbit/s a šířkou pásma 400 kHz. To vše podléhá standardu EN 300 220 [7].

2.3 Z-Wave Aliance

Aliance byla založena roku 2005 a skládá se z předních světových výrobců, kteří se zabývají vývojem technologie Z-Wave pro inteligentní domácnost a podnikové aplikace[2].

Prvotní technologii vytvořila společnost Zensys. Tato společnost v roce 2005 spojila síly se společnostmi Intermatic, Leviton, Wayne Dalton, Danfoss a Universal

Electronics a vytvořily Z-Wave Alianci. Postupně se do společnosti připojilo víc než 700 dalších členů[4].

Jednotlivé členství má celou řadu výhod,ať už marketingových, technologických, certifikačních, vzdělávacích nebo podpory[3].

Existují tři úrovně členství v alianci s různými podmínkami:

2.3.1 Plné členství

- Platí roční příspěvky ve výši $4 000.
- Vyrábí, vyvíjí nebo uvádí na trh výrobky založené na technologii Z-Wave.
- Veřejně publikuje obchodní zájem o Z-Wave standard pro chytrou domácnost.

2.3.2 Přidružené členství

- Platí roční příspěvky ve výši 400$.
- Veřejně publikuje obchodní zájem o Z-Wave standard pro chytrou domácnost.

2.3.3 Integrované členství

- Platí roční příspěvky ve výši 250$.
- Instaluje výrobky Z-Wave.
- Veřejně publikuje obchodní zájem o Z-Wave standard pro chytrou domácnost.

2.4 Z-Wave Plus

Z-Wave Plus je nový certifikační program, který pomáhá zákazníkům s identifikací produktů, které jsou z nové řady 500, což je pátá generace Z-Wave. U certifikovaných řešení je tak garantovaná vzájemná kompatibilita, kdy zařízení dokáží reagovat na přesně definované příkazy. Další výhodou nové generace je i zvýšení dosahu, prodloužení životnosti baterie, možnosti aktualizace OTA (Over The Air) pomocí bezdrátové sítě a přidání dalších RF kanálů. Z-Wave Plus přináší výhody především pro koncové zákazníky, kteří si mohou na webových stránkách ověřit, zda-li je zařízení certifikováno[2][42].

2.4.1 Série 700

V současné době se chystá do výroby nová série 700, která byla představena na CES 2018 (Consumer Electronics Show). Tato zařízení budou mít certifikaci Z-Wave Plus V2. Výhody oproti předchozí verzi jsou následující:

[2] `https://products.z-wavealliance.org/`

- Zvýšení účinosti zařízení napájených pomocí baterií.
- Nový systém probouzení a uspávání.
- Prodloužení dosahu zařízení.
- Navýšený výkon zařízení.
- Rychlejší a méně náročné šifrování.
- Předpřipravena řešení pro jednotlivé typy zařízení.
- Zpřísnění certifikačního procesu s ohledem na výkon.

Tato série si dala za cíl přetvoření chytré domácnosti na inteligentní domácnost. Toho by se mělo docílit především větší spoluprací mezi jednotlivými senzory a akčními prvky. Součástí sítě by měla být i umělá inteligence spolupracující na zefektivnění sítě. Pro běh umělé inteligence je potřeba značně zvýšit výkon zařízení. To bylo dosaženo využitím ARM® Cortex-M4 procesoru. Ten umožňuje běh robustních aplikací bez nutnosti externího procesoru. Proto došlo i ke zpřísnění certifikačního procesu. Nové špičkové zabezpečení S2 chrání a zařízení i cloud i proti hackerským útokům.

Druhým podstatným vylepšením nové série je práce s energií. To se ukazuje především u zařízení napájených bateriemi. Ta nyní slibují mnohem nižší spotřebu a díky tomu až 10 letou životnost na článkových bateriích. Díky tomu je umožněno zabudování senzorů i do sádrokartonových stěn či nábytku. Nadále však zaručují rychlé probouzení při neaktivitě.

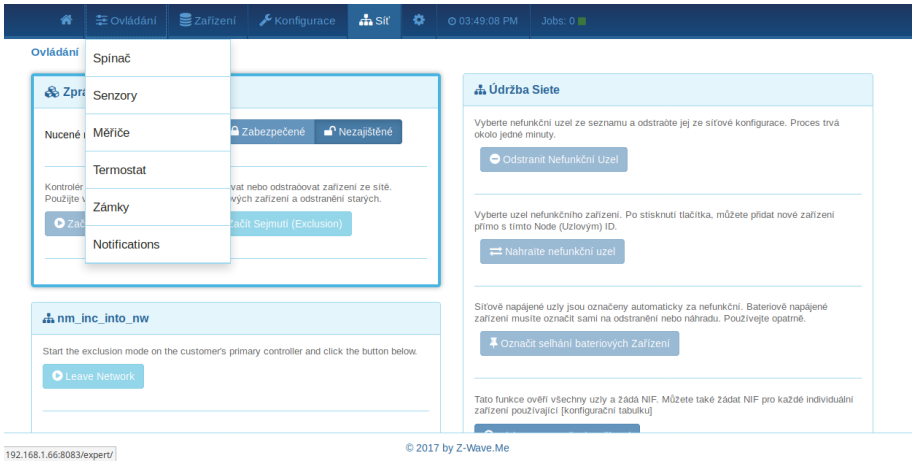
Třetí podstatnou změnou je i zvětšení dosahu zařízení. Nyní je při přímé viditelnosti možná komunikace až na vzdálenost 100 metrů, což je o polovinu víc než v předchozí generaci. Vzdálenost celé smíšené sítě pak byla rozšířena až na 400 metrů. Toho bylo docíleno zvýšením vysílacího výkonu a to bez nutnosti SAW filtru (Surface Acoustic Wave). [12].

3 Dostupné knihovny

Na internetu existuje celá řada knihoven podporující platformu Z-Wave. Ty jsou buď vyvíjené výrobcem zařízení anebo komunitou. Některé kromě protokolu Z-Wave podporují i další protokoly jako EnOcean, Hue, LIFX, ZigBee či KNX. Mezi nejznámější knihovny patří HomeSeer, Home Assistant, OpenZWave, SmartThings, Vera, Zipabox, Z-Wave>Me Indigo 7, Helix a OpenHAB(Open Home Automation Bus).

3.1 Z-Wave>Me

Z-Wave>Me je kompletní open-source knihovna napsaná v jazyce JavaScipt určená pro řídící kontroler, podporující jak Z-Wave, tak technologie jako EnOcean, WI-FI a zařízení využívající 433Mhz frekvenci. Podporována je široká škála hardwaru a systémů jako Raspberry Pi, Windows, Ubuntu, Debian, atd. Pro vytvoření sítě Z-Wave je potřeba Z-Wave>Me hardwaru jako UZB ((Universal Z-Wave Bus)) nebo RaZberry (Z-Wave karta připojitelná na Raspberry GPIO). Její uživatelské rozhraní je vidět na obr. 3.1.



Obr. 3.1: Uživatelské rozhraní Z-Wave>Me

Knihovnu lze rozdělit do jednotlivých vrstev, které jdou libovolně spojovat:

- Knihovna Z-Way - jádro Z-Wave knihovny.
- Knihovna EnOcean - implementace EnOcean ovladačů.
- Automatizační stroj - tato vrstva umožňuje využívat drátové a bezdrátové sítě třetích stran.
- Zkušební rozhraní Z-Wave Expert - toto je referenční uživatelské rozhraní pro funkce Z-Wave

- Uživatelské rozhraní Chytré Domácnosti - umožňuje úplné ovládání všech funkcí zařízení. Je napsáno v jazyce AJAX (Asynchronous JavaScript and XML).
- Cloud - umožňuje přístup k domácí síti na cestách. Nativní aplikace pro iOS a Android.

Hlavní síla Z-Wave>Me je v její otevřenosti [17].

3.2 Home Assistant

Home Assistant je domácí automatická platforma s otevřeným kódem napsaná v jazyce Python 3. Snaží se o sledování a kontrolu všech zařízení v chytré domácnosti. Jedním z podporovaných protokolů je i protokol Z-Wave. Pro připojení Z-Wave zařízení aplikace využívá otevřené knihovny OpenZWave (viz kapitola 3.4).

3.2.1 Hass.io

Hass.io je systém, který přemění Raspberry (nebo obdobné zařízení) na automatickou domácí stanici běžící na Home Assistant. Pomocí systému je možné se zaměřit na integraci zařízení a psaní podrobné automatizace.

3.3 OpenHAB

Je platforma s otevřeným zdrojovým kódem, která vytváří centrum chytré domácnosti. OpenHAB si ve své druhé verzi bere za cíl integrovat různé automatizační systémy, technologie a zařízení do jednoho řešení. Poskytuje tak společné uživatelské rozhraní bez ohledu na počet použitých systémů.

OpenHAB je napsaný v jazyce Java verze 8 a je založen na prostředí Eclipse SmartHome[1]. Platformu lze rozšířit pomocí mnoha doplňků, které poskytují široké spektrum možností od změny uživatelského rozhraní až po možnost komunikovat s velkým počtem fyzických zařízení[40].

3.3.1 Z-Wave vazba

OpenHAB pomocí doplňku podporuje rozhraní Z-Wave pro domácí automatizaci. Nerozlišuje jednotlivé výrobce ani vývojové série. Rozšíření podporuje velké množství zařízení[2] (v současnosti 802 zařízení od 114 výroců). Pro fungování je třeba připojit Z-Wave kontroler a zadat sériový port. Přidání nových zařízení do sítě se

[1] `https://www.eclipse.org/smarthome/`

[2] `https://www.openhab.org/addons/bindings/zwave/doc/things.html`

provádí ve dvou krocích. Nejdříve je zařízení přidáno do doporučené pošty jako neznámé a umožňuje tak uživateli okamžitou zpětnou vazbu, že zařízení bylo nalezeno. V druhém kroku se pak systém snaží rozpoznat přímo výrobce a typ zařízení[41].

3.3.2 OpenHABian

OpenHABian je speciálně nakonfigurovaný linuxový systém založený na Debianu, který je připravený pro všechny specifické potřeby openHABu. Obraz SD karty je připravený pro použití na Raspberry Pi a Pine A64. Slouží jako nejjednodušší způsob instalace automatizační platformy OpenHAB pro domácnost[39].

3.4 OpenZWave

OpenZWave je multiplatformní knihovna s otevřeným kódem[3], vytvořená jako podpora pro fungování domácích Z-Wave zařízení bez nutnosti znalosti protokolu Z-Wave. OpenZWave je přidružený člen Z-Wave aliance.
Hlavní část protokolu Z-Wave byla uvolněna v srpnu 2016. Během září téhož roku společnost OpenZWave vytvořila knihovnu umožňující vytváření aplikací spolupracujících se zařízeními Z-Wave.
Cílem projektu je vytvoření knihovny pro komunitu Z-Wave tak, aby podporovala co nejvíce zařízení a aby si tak každý mohl jednoduše přidat do své aplikace podporu Z-Wave protokolu.
Klíčové vlastnosti OpenZWave knihovny:

- Napsaná v jazyce C++.
- Má DotNet vrstvu pro Windows.
- Je kompatibilní s většinou systémů Windows, Mac OsX a Linux.
- Pracuje s více kontroléry.
- Pracuje s většinou spínačů, stmívačů, termostatů, energetických monitorů, snímačů pohybu, modulů spotřebičů, klíčů, senzorů dveří apod.[4].
- Má plně zdokumentovaný kód.
- Obsahuje i jednoduché ukázky kódu[18].

3.4.1 Knihovna

Veškeré funkce Z-Wave jsou v knihovně přístupné prostřednictvím manažerské třídy. Komunikace mezi počítačem a zařízeními probíhá asynchroně[18].

[3] `https://github.com/openzwave/`
[4] `http://www.openzwave.com/device-database`

3.4.2 Aplikace

Při tvorbě aplikace pomocí OpenZWave by měl vývojář myslet na specifikace Z-Wave platformy:

- Komunikace je asynchroní.
- Není zaručeno přijetí odpovědi na žádost.
- Síť Z-Wave se může kdykoliv změnit a je tak třeba, aby fungovala zpráva o zpětném volání.
- Někteří uživatelé mohou mít více jak jeden kontroler a je tak potřeba vyřešit vzájemnou komukaci.
- Není potřeba ukládat stav sítě. Z-Wave to dělá automaticky bez nutnosti zásahu aplikace[18].

3.4.3 Nejnovější verze 1.6

Během psaní této diplomové práce byla téměř po dvou letech (3.5.2019) vydána nová stabilní verze knihovny OpenZWave označená jako 1.6. Tato nová verze přináší vývojářům řadu vylepšení. Bylo celkem upraveno 1 117 souborů, přidáno 79 127 řádku kódu a 32 483 naopak smazáno. Významná změna proběhla například ve změně konfiguračních souborů. S tím souvisí i plán pro brzkou podporu dalších cca 1300 zařízení. Mezi hlavní změny a nové funkce patří:

- Nová třída ValueID, interního časovače a uživatelského kódu.
- Opravy chyb v S0 šifrování hlavně u multiinstantních zařízení.
- Přepínání zvuků.
- Centrální scéna.
- Konfigurace čísel a aktualizací souborů.
- Aktualizace oznámení.
- Lokalizace.
- Aktualizace třídy příkazů: binární, multisenzoru a termostatu.
- Aktualizace konfiguračního formátu zařízení, který nyní obsahuje informace o revizích a metadatech.
- Úplně nový dotMorten Wrapper (náhrada za starý dotNet).
- OZW automaticky ukládá mezipaměť bez použití Manager::WriteConfig.
- Možnost přiřadit instancím popisky[20].

Změny pro vývojáře

Maximální hodnota indexu pro hodnoty (ValueID) byla navýšena z 255 na 65535 (z uint8_t na uint16_t). Existuje nová hodnota BitSet ValueID reprezentující konfigurační proměnné, kde jednotlivé bity mají svůj určitý význam. OZW nyní podporuje

lokalizaci všech uživatelsky viditelných řetězců, například štítků, nápovědy a položek seznamů. To umožňuje aplikaci aby tyto informace zobrazila v nativním jazyce. Tato lokalizace je nyní uložena v `/config/Localization.xml`. Upozornění třídy příkazů `Alarm` byl přepsán tak, aby využíval nové funkce ze Z-Wave specifikace. OZW tak nyní umožňuje dynamicky specifikovat typ oznámení v novém konfiguračním souboru NotificationTypes.xml. Novější zařízení nyní podporují oznámení o "vyčištění"notifikací zpět do stavu nečinnosti po předem určeném časovém intervalu. Pokud je zařízení starší a určení intervalu nepodporuje, tak OZW tento interval nastaví automaticky na 5s. Nová Z-Wave specifikace objasnila, že všechny třídy příkazů uživatelského kódu musí být v ASCII (American Standard Code for Information Interchange) a tak ta nyní exportuje všechny tyto třídy jako hodnotový řetězec znaků. Třída příkazů pro hlavní scénu byla vytvořena pro pomoc vývojářům k zpětné analýza. Každá scéna je přitom exportovaná prostřednictvím jedinečné hodnoty a je podporováno maximálně 255 scén. U mnohoúrovňových přepínačů byl vyřešen problém s vracením neurčité hodnoty pomocí podpory nové 4. verze třídy příkazů. Knihovna rovněž nyní podporuje obnovení hodnot pro zařízení na bateriích, které se uspávají a probouzejí. Jednotlivým instancím lze také přiřadit popisky v konfiguračních souborech zařízení. Databáze těchto souborů obsahuje nyní parametr revize pro přehled o aktuální verzi. Soubor mezipaměti `zwcfg_*.xml` je nyní přejmenovaný na `ozwcache_*.xml`[20].

3.4.4 Licence

OpenZWave je software, který je zdarma a každý ho může upravovat a rozšiřovat za dodržení podmínek GNU Lesser General Public License[5] ve verzi tři nebo novější, které zveřejňuje Free Software Foundation[18].

3.4.5 Python obal pro OpenZWave

Python-openzwave je obal pro vývoj aplikací v jazyce python, s využitím OpenZWave knihovny napsané v programovacím jazyce C++. Obsahuje:

- Plnou implementaci obslužného manageru.
- API pro mapování Z-Wave sítě pomocí Python objektů.
- Správce příkazového řádku pro ladění aplikací.
- Příklady.
- Dokumentaci.

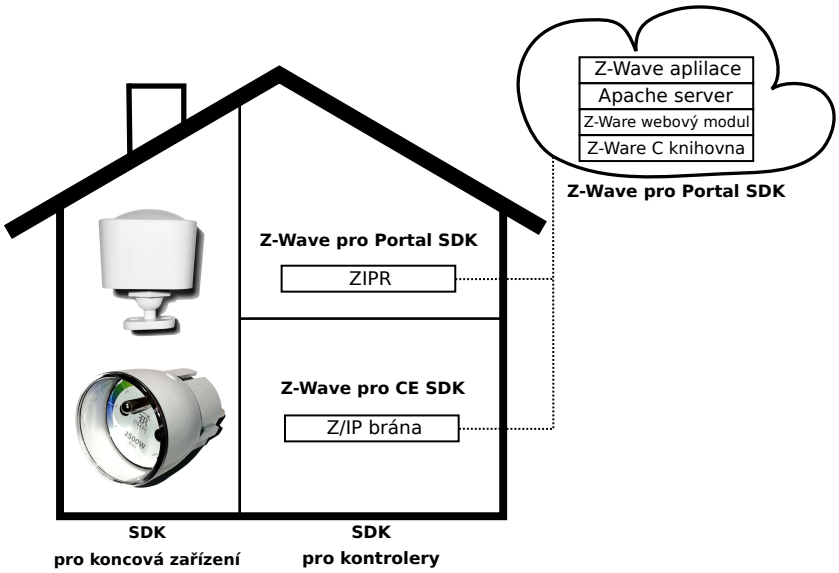
[5] `http://www.gnu.org/licenses/`

3.5 Z-Wave software

Pro platformu Z-Wave je volně dostupná celá řada aplikací od společnosti Silicon Labs[6] pro vývoj vlastních Z-Wave zařízení. Dále jsou také dostupná SDK (Software Development Kit) pro podřízená zařízení, i pro kontrolery.

3.5.1 SDK pro kontrolery

Vývojová sada nabízí prostředky k vývoji ovládacích prvků s podporou funkce IoT. SDK pro kontrolery tak nabízí prostředky pro vývoj vlastních aplikací s podporou Z-Wave protokolu. Obsahuje jak bránu Z/IP tak Z-Ware (Middleware pro Z-Wave kontroler běžící přes Z/IP), tak kompletní zdrojové kódy pro použití nebo upravení kontroleru. Z-Wave pro CE SDK obsahuje pracovní prototyp brány pro IoT. Spolu s branou Z/IP je k dispozici knihovna Z-Ware C, která připojený kontroler reprezentuje jako softwarové objekty jazyka C. Z-Wave pro Portal SDK obsahuje certifikovaný vyhodnocovací server Z-Wave Plus, který běží na operačním systému Ubuntu (viz obr. 3.2)[29].



Obr. 3.2: SDK pro kontrolery

Inteligentní domácnost

Z-Wave SDK pro CE obsahuje všechny nezbytné části pro návrh a tvorbu chytré domácnosti. Chrání vývojáře před nutností znalosti Z-Wave protokolu a transformuje všechna zařízení v síti do softwarových objektů, se kterými lze jednoduše pracovat.

[6] `www.silabs.com`

Součástí připraveného operačního systému Ubunutu pro zařízení TI BeagleBone Black je i brána Z/IP, která se připojuje k řídícímu Z-Wave kontroleru, ze série 500 nebo novější. Knihovna Z-Ware C pak slouží jako snadno srozumitelné API mezi programovacím jazykem C a Z-Wave[29].

Inteligentní cloud

Pro operační systém Ubuntu je dostupné Portal SDK, které obsahuje vyhodnocovací server, který kombinuje Z-Ware aplikace a standardní technologie, jako je Apache server[7] nebo OpenLDAP[8]. Slouží tak pro vývoj cloudových řešení. Všechny části jsou dostupné jako zdrojový kód s prostředím pro sestavení na operačním systému Ubuntu[29].

3.5.2 SDK pro koncová zařízení

SDK je určeno pro vývojáře, kteří vytvářejí zařízení kompatibilní se Z-Wave Plus nákladově efektivním způsobem. Software se skládá z knihoven Z-Wave pro podřízené uzly a pro kontrolery a z aplikačního rámce Z-Wave Plus, jakož i z kódu pro širokou škálu aplikací pro chytrou domácnost[30].

Knihovny

Z-Wave zařízení potřebují přesně definované hlavičkové soubory a knihovny. Ty jsou v SDK uspořádány do jednotlivých adresářů používaných pro vývoj kontrolerů a podřízených zařízení:

- Z-Wave\include - Hlavičky potřebné pro propojení protokolu Z-Wave s hardwarem.
- Z-Wave\IO_defines - Soubory s definicemi hardwaru potřebých pro tvorbu aplikace.
- lib\bootloader_ZW050x - Zavaděč užívaný s OTA aktualizací firmwaru.
- lib\controller_bridge_ZW050x - Obsahuje všechny soubory potřebné pro vytvoření aplikace Z-Wave bridge ovladače.
- lib\controller_portable_ZW050x - soubory potřebné pro vytvoření aplikace Z-Wave ovladače.
- lib\controller_static_ZW050x - soubory potřebné pro vytvoření aplikace statického Z-Wave ovladače.
- lib\controller_static_norep_ZW050x - soubory potřebné pro vytvoření aplikace statického Z-Wave ovladače bez funkce opakovače.

[7] `https://httpd.apache.org/`
[8] `https://www.openldap.org/`

- lib\controller_static_single_ZW050x - soubory potřebné pro vytvoření aplikace samostatného statického Z-Wave ovladače.
- lib\slave_enhanced_232_ZW050x - soubory potřebné pro vytvoření aplikace k síti větší jak 232 uzlů.
- lib\slave_prodtest_dut_ZW050x - soubory k sestavení testovací DUT (Device Under Test) aplikace na Z-Wave modulu.
- lib\slave_prodtest_gen_ZW050x - soubory k sestavení generátorové testovací aplikace na Z-Wave modulu.
- lib\slave_routing_ZW050x - soubory potřebné pro vytvoření routovací podřízené aplikace na Z-Wave modulu.
- lib\ext_nvm - externí ovladače pro trvalou paměť (NVM).
- lib\init_vars - soubor objektu init_vars.obj, který nahrazuje standardní inicializační proceduru Keil.
- lib\rf_freq - Různé soubory pro RF[28].

Aplikační rámec

Aplikační rámec usnadňuje implementaci robustních produktů v souladu s požadavky Z-Wave Plus. Stará se především o:

- Zjednodušení vývoje firmwaru pro nová zařízení Z-Wave.
- Poskytování robustní modulární knihovny a ukázkových kódů mnoha běžných zařízení.
- Pomoc při eliminaci chyb a zjednodušení programování[28].

Firmware

Součástí SDK je i řada připravených firmwarů pro jednotlivé moduly (SD3503 a ZM5304) či pro určité typy zařízení, od jednoduchého vypínače až k řídící jednotce. Jsou zde i zdrojové kódy připravené pro úpravu a překlad. Přeložené firmwary ve formě `hex` souborů jsou v několika variantách:

- Controller Bridge
- Controller Portable
- Controller Static
- Controller Static Single
- Slave Enhanced 232
- Slave Routing

3.5.3 Pomocné vývojové aplikace a knihovny

Silicon Labs také vytvořil několik nástrojů pomáhajících s vývojem a kontrolou Z-Wave sítě a jejich zařízení.

Z/IP Gateway

Z/IP Gateway je aplikace, která zpracovává veškerou komunikaci mezi Z-Wave a IP protokolem. Zpracovává tak všechny funkce Z-Wave jako je správa sítě, zabezpečení, notifikace a podpora více kanálů. Využívá DTLS (Datagram Transport Layer Security) pro zabezpečení lokální sítě a pro přístup ke vzdálenému serveru využívá tunelu se zabezpečením transportní sítě[29].

Z-Ware C Library

Knihovna, která se může připojit k Z/IP bráně a detekovat tak veškerá Z-Wave zařízení a převést je do softwarocých objektů programovacího jazyka C. Ty pak mohou být použité jako základ pro návrh inteligentní domácí brány[29].

Z-Ware Apache Web API Plugin

Tento modul slouží ke kombinaci knihovny Z-Ware C a aplikací Z-Ware se standardními technologiemi jako je například OpenLDAP a Apache Server a tvoří základ pro vývojáře k tvorbě cloudových Z-Wave řešení[29].

Z-Ware Apps

Z-Ware obsahuje sadu vzorových aplikací pro chytré televizory, počítače, tablety a chytré telefony. Z-Ware aplikace zachovávají jednotné uživatelské prostředí napříč různými platformami a zařízeními a umožňují koncovému zákazníkovi provádět komplexní úkoly pomocí stiknutí jednoho tlačítka. Z-Ware aplikace lze snadno upravovat tak, aby zahrnovaly například názvy a loga společností. Je možné také upravit barevný či celkový grafický vzhled aplikace tak, aby odpovídala grafice výrobce[29].

Z-Wave for CE SDK

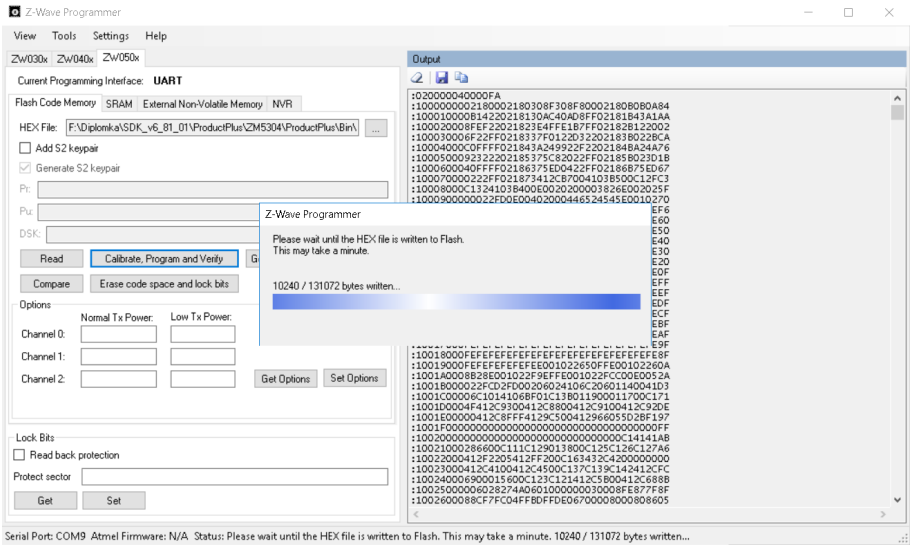
Obsahuje veškeré softwarové části pro návrh chytré domácnosti. Slouží jako řešení pro vývojáře chytrých Z-Wave sítí, bez nutnosti, aby vývojář znal Z-Wave protokol. Všechny Z-Wave zařízení převádí do softwarových objektů[29].

Zniffer

Aplikace Zniffer je vývojářský nástroj pro zaznamenávání Z-Wave RF komunikace a prezentaci rámců s grafickým uživatelským rozhraní na PC (Personal Computer)[30].

PC Programmer

Z-Wave programovací nástroj může být použit na programování jednoduchých Z-Wave čipů řady 200/300/400/500 během vývoje softwaru a hardwaru pro malou sérii výroby[30].
Tento program běžící na operačním systému Windows je v této práci využit při nahrání firmwaru do modulu ZM5304 (viz obr. 3.3)



Obr. 3.3: Z-Wave Programmer

PC Controller 5.30

Aplikace pro komunikaci se Z-Wave uzly jako jsou přepínače, senzory, pomocí statického ovladače. Tato aplikace vyžaduje modul Z-Wave naprogramovaný firmwarem Static Controller Serial API a připojený k příslušnému USB nebo sériovému portu[30].

USB ovladače

Ovladače pro spojení čipů se systémem Windows.

Z-Wave SmartStart QR-Tool

Nástroj Z-Wave SmartStart QR-Tool se snaží přesunout úkoly, které provádí koncové zařízení v síti Z-Wave k uživatelsky příjemnějšímu rozhraní brány. Odstraňuje

potřebu inicializace koncového zařízení k přidání do sítě. Automaticky při zapnutí zahájí připojování do sítě. To opakuje v dynamických intervalech tak dlouho, dokud se zařízení nepřipojí[31].

Micro RF LinkX

Pomáhá ověřit RF výkon zařízení bez ohledu na režijní náklady Z-Wave protokolu[30].

Editor pro XML (eXtensible Markup Language)

Slouží pro definici Z-Wave příkaz aplikační vrstvy. XML dokument lze použít například v nástroji Z-Wave Zniffer pro interpretaci tříd příkazů a zařízení. Kromě toho je pomocí editoru možné vytvořit základní soubor pro vývoj aplikace v programovacím jazyce C# a záhlaví pro jazyk C[30].

Z-Wave DLL (Dynamic-Link Library)

Z-Wave DLL je aplikační rámec, který zjednodušuje vývoj aplikací pro Windows Forms nebo na konzoly podporující Z-Wave pro platformu Microsoft .NET [30].

3.6 Výběr knihovny

Pro další pokračování v této práci byla zvolena knihovna OpenZWave. Mezi její hlavní výhody patří její otevřenost a podrobná dokumentace. Užití a vlastní úpravy nám dovoluje GNU licence. Oproti OpanHAB, Z-Wave>Me i Home Assistant se OpenZWave stará jen o platformu Z-Wave a proto neobsahuje pro účely práce zbytečně velké množství kódu.

3.7 Další použité knihovny

Pro programování vlastní aplikace bylo využito Python wrapperu pro OpenZWave. Dále byla využita knihovna Kivy pro grafické rozhraní a SQLite pro databázový záznam jednotlivých hodnot v uzlech.

3.7.1 Python-OpenZWave

Python-OpenZWave je wrapper pro knihovnu OpenZWave na převod C++ knihovny do Pythonu. Wrapper obsahuje:

- Plnohodnotného správce implementující nastavení.
- API pro převedení Z-Wave sítě do Python objektů.

- Správce příkazového řádku pro správu a ladění Z-Wave sítě.
- Webový příklad.
- Sadu testů sítě.
- Další příklady pro použití.
- Kompletní dokumentaci[21].

3.7.2 Kivy

Kivy je softwarová knihovna s otevřeným zdrojovým kódem pro rychlý vývoj aplikací s uživatelským rozhraním podporující multidotyková gesta. Kivy je momentálně k dispozici pod licencí MIT (Massachusetts Institute of Technology), starší verze pak existují pod licencí LGPL verze 3 (Lesser General Public License). Poslední verze knihovny umožňuje její využití pro aplikace s uzavřeným i otevřeným kódem, kdy úpravu knihovny není nutné zpětně publikovat[22].

Filosofií pro vývoj Kivy knihovny je:

- Čerstvost.
- Rychlost.
- Flexibilita.
- Zeměřenost.
- Financovanost.
- Bezplatnost.

3.7.3 SQLite

SQLite je databázový SQL (Structured Query Language) nástroj, napsaný jako knihovna programovacího jazyka C. Je zabudovaný do všech dnešních mobilních telefonů a je dodáván v mnoha běžných aplikacích. Formát souboru je stabilní, multiplatformní a zpětně kompatibilní. Zdrojový kód je OSS (Open-Source) a je zdarma dostupný pro jakékoliv účel použití[37].

Syntaxe

SQLite je klasický SQL databázový stroj, který se ovládá pomocí SQL příkazů. Rozumí většině standardních SQL příkazů a přidává některé vlastní. Seznam klíčových slov je k dispozici na officiálních stránkách[9][37].

Základními příkazy, které jsou využité v této práci jsou:

- CREATE TABLE - vytvoření tabulky.
- DROP TABLE - vymazání tabulky.
- SELECT - získání dat.

[9] `https://www.sqlite.org/lang_keywords.html`

- SUM, MIN, MAX, ... - agragační funkce při výběru.
- INSERT - vkládání hodnot.
- UPDATE - modifikace existujících dat.
- PRIMARY KEY - definice primárního klíče v tabulce.

4 Z-Wave zařízení na českém trhu

Již nyní je Z-Wave platforma značně rozšířená i na českém trhu. Výrobky podporující Z-Wave nabízejí všichni hlavní prodejci elektroniky včetně Alzy, CZC, Datartu a Electro Worldu.

4.1 Brány

Na českém trhu je celá řada ovládacích zařízení a systémů pro Z-Wave sítě. Ty se liší především svojí cenou, technickým a softwarovým zpracováním. Stručný přehled je v tabulce 4.1.

Tab. 4.1: Přehled bran na českém trhu

Výrobce	Název	Cena	Z-Wave Plus
Fibaro	Home Center Lite	7 100 Kč	ano
Fibaro	Home Center 2	15 500 Kč	ano
Fifthplay	Smartbox	2 900 Kč	ano
Zipato	Zipatile gateway	10 000 Kč	ano
Vera Control	Edge	4 100 Kč	ano
Vera Control	Plus	6 000 Kč	ano
Devolo	Home Control Central Unit	2 700 Kč	ano
Popp	Hub	6 000 Kč	ano

4.1.1 Fibaro

Fibaro je polská značka fungující v IoT a je jedním z nejrychleji se rozvíjejícím výrobcem automatické domácnosti v Evropě. Centrála a hlavní výrobní část se nachází v Poznani a oddělení výzkumu a vývoje v Zielona Gora. Společnost nyní zaměstnává cca 400 zaměstnanců.

Home Center Lite

Jedna z nejprodávanějších centrálních bran na českém trhu. Zařízení využívá procesor Cortex A8. Podporuje platformu Z-Wave. Součástí je i Wi-Fi, kterou používá pro připojení a ovládání uživatelem. To probíhá pomocí chytrého mobilního telefonu se systémem Android či iOS. Brána dokáže spravovat až 230 zařízení, ale doporučuje se propojit maximálně 150. Cena v ČR se pohybuje kolem 7 100Kč s DPH.

Home Center 2

Funguje stejně jako Lite verze s rozdílem, že centrální jednotka s je poháněna procesorem Intel Atom 1.6GHz a má 4Gb paměti na SSD disku. Oproti Lite verzi dokáže spravovat plnohodnotně 230 Z-Wave zařízení. Součástí je i hlasové ovládání Lili. Je vybaven Lua (procedurální skriptovací jazyk) interpretem. Lze také využít jako nadřízený kontroler pro další jednotky. Cena v ČR se pohybuje kolem 15 500Kč s DPH.

4.1.2 Fifthplay

Firma fifthplay je belgická high-tech společnost a dceřiná společnost skupiny Niko. Snaží se vstupovat na trh prostřednictvím dodavatelů energie, provozovatelů telekomunikačních sítí, čemuž přizpůsobuje i návrh svých zařízení.

Smartbox

Domácí brána kompatibilní s operačním systémem Android a iOS a využívá spojení pomocí Wi-Fi. Se zařízeními v chytré domácnosti komunikuje pomocí Z-Wave platformy ve spojení s platformou ČSOB NaDoma[1]. Bránu lze pořídit za 2 900Kč.

4.1.3 Zipato

Firma Tri plus grupa d.o.o. je technologická firma se zaměřením na IoT. Od roku 2010 firma vyvýjí vlastní platformu Zipato. Zakladatelem firmy je Sebastian Popovic.

Zipatile gateway

Centrální jednotka s dotykovým displejem, která dovoluje ovládat chytrou domácnost s použitím mobilního telefonu. Kromě protokolu Z-Wave podporuje také protokoly Bluetooth 4.0, Wi-Fi a ZigBee. V jednotce je také integrovaný teplotní, vlhkostní, pohybový, světelný, hlukový a vibrační senzor. Pořizovací cena se pohybuje kolem 10 000Kč.

4.1.4 Vera Control

Společnost Vera Control se snaží udělat technologie inteligentních domácností pro zákazníky cenově dostupnější. Produkty společnosti Vera Control jsou postavené na platformě MiOS[2] (Media Inteface Operating System).

[1] `https://nadoma.csob.cz/`
[2] `http://www.mios.com/`

Edge

Domácí Z-Wave brána s 128 MB RAM, 600MHz MIPS (Microprocessor without Interlocked Pipeline Stages) SoC (System on a Chip) procesorem s WiFi, LAN a USB. Napájena je pomocí 12V zdroje a umožňuje připojit až 220 zařízení. Obsahuje uživatelské rozhraní UI7 a stojí 4 100Kč.

Plus

Domácí Z-Wave brána s 128 MB NAND Flash a 256 MB DDR3 pamětí, s procesorem 800GHz. Připojit se lze pomocí WiFi, GLAN a USB. Dovoluje tvorbu scén, časovačů, pokojů, událostí a notifikací. Stojí 6 000Kč.

4.1.5 Devolo

Devolo je německá high-tech firma specializující se na chytré domácí sítě. Společnost dbá na energetickou nenáročnost sítě.

Home Control Central Unit

Ovládání chytré domácnosti probíhá pomocí webového prohlížeče. Zabezpenčení přenosu probíhá přes německý certifikovaný systém. Funguje jako jednoduchá elektrická zástrčka s RJ-45 a USB konektorem. Jeho cena v českých obchodech je 2 700Kč.

4.1.6 Popp

Německá elektrotechnická firma fungující více jak 80 let. Momentálně se zabývající především poskytováním vysoce kvalitních bezdrátových komponentů pro chytré domácnosti.

Hub

První vlastní brána této firmy, která byla uvedena na trh roku 2016. Obsahuje procesor MT7620A taktovaný na 580MHz, 128MB DDR2 paměť, 16MB flash + 4GB paměti na interní SD kartě. Je to plně certifikovaný Z-Wave Plus kontroler řady 500. Obsahuje čtyři USB porty a jeho cena je v České republice 6 000Kč s DPH.

4.2 Senzory

Senzory jsou podřízená zařízení, která dokáží získávat data z okolí a umí je předat řídící jednotce. Na českém trhu jsou k dispozici především pohybové senzory, měřiče

teploty, vlhkosti, detektory kouře a zaplavení. Koupit se dají i multisenzory, které sdružují více senzorů dohormady.

4.2.1 Pohybové senzory

Pohybové senzory se využívají především pro hlídání domácnosti před nedovoleným vniknutím cizího člověka. Přehled dostupných pohybových senzorů je v tabulce 4.2.1.

Tab. 4.2: Přehled pohybových senzorů na českém trhu

Zařízení	Výrobce	Kód produktu	Cena	Z-Wave Plus
Okenní/ dveřní senzor	VISION	ZD2102EU-5	890 Kč	ano
Okenní/ dveřní senzor	Devolo	MT02648	1 073 Kč	ano
Okenní/ dveřní senzor	Fifthplay	ZD2102EU-5	829 Kč	ano
Okenní/ dveřní senzor	Fibaro	FGDW-002	1 262 Kč	ano
Detektor pohybu	Fibaro	FGMS-001	1 569 Kč	ano
Detektor pohybu	D-Link	DCH-Z120	990 Kč	ano
Detektor zaplavení	Devolo	MT02756	1 190 Kč	ano
Detektor zaplavení	Fifthplay	ZF5201EU-5	829 Kč	ano
Detektor zaplavení	Fibaro	FGFS-101	1 528 Kč	ano

4.2.2 Senzory kouře

O bezpečnost v domácnosti se starají také detektory kouře, které hlídají především případný vznik požáru. Přehled dostupných Z-Wave detektoru je v tabulce 4.2.2.

Tab. 4.3: Přehled kouřových senzorů na českém trhu

Zařízení	Výrobce	Kód produktu	Cena	Z-Wave Plus
Detektor kouře	Devolo	MT:2651	1 305 Kč	ano
Detektor kouře	Fifthplay	ZS6101EU-5	1 399 Kč	ano
Detektor kouře	Popp	POPE004001	2 831 Kč	ano
Detektor kouře	D-Link	DCH-Z310	1 799 Kč	ano
Detektor kouře	Neo Coolcam	DCH-Z310	990 Kč	ne
CO detektor	Fifthplay	FGSD-002	1 899 Kč	ano
CO detektor	Fibaro	FGSD-002	2 530 Kč	ano

4.2.3 Ostatní senzory

Senzory měřící teplotu, vlhkost, světlo a další bývají obvykle v kombinaci jako tzv. multisenzor. Přehled těhto senzorů dostupných na českém trhu je zpracován v tabulce 4.2.3.

Tab. 4.4: Přehled měřících senzorů na českém trhu

Měřené veličiny	Výrobce	Kód produktu	Cena	Z-Wave Plus
Teplota a vlhkost	Fibaro Philio	PAT02-1B	1 499 Kč	ano
Teplota, vlhkost, světelnost a pohyb	Fifthplay	ZP3111EU-5	949 Kč	ano
Elektrický průtok a proud	Fibaro	AEO-HEM1-3	4 299 Kč	ne

4.3 Akční členy

Akční člen je podřízené zařízení, které převádí informační část na mechanickou a naopak. V chytré domácnosti se jedná nejčastěji o spínače, zásuvky, ovládače rolet apod. Přehled dostupných zařízení na českém trhu je v tabulce 4.3.

Tab. 4.5: Přehled akčních členů na českém trhu

Zařízení	Výrobce	Kód produktu	Cena	Z-Wave Plus
Vypínač	Devolo	MT2652	1 390 Kč	ano
Tlačítko	Fibaro	FGPB-101	1 190 Kč	ano
Chytrý zámek	Danalock	V3-BTZE	1 399 Kč	ano
Dveřní stanice	Fibaro	DB-D101S	8 990 Kč	ne
Žaluziový modul	Fibaro	FGR-222	1 490 Kč	ne
Spínaná zásuvka s měřením	Fibaro	FIB-FGR-222	1 689 Kč	ano
Spínací modul	Fibaro	FGS-213	1 528 Kč	ano
Stmívač	Fibaro	FGD-212	1 519 Kč	ano
Řízení RGBW LED pásků	Fibaro	FGRGBWM-441	1 599 Kč	ne
Rádiový termostat	Fibaro	FGT-001	2 049 Kč	ano
Heatit Termostat	Fibaro	TF021	3 490 Kč	ano
Ovladač motorů	Aeotec	ZW141-C	1 450 Kč	ano

4.4 Shrnutí

I na českém trhu je již dnes k dostání mnoho různých bran, senzorů i akčních členů. Největší počet výrobků má polská firma Fibaro. Nejzákladnější domácí bránu lze dnes pořídit za 2 700 Kč. Jen jedna z bran přitom obsahuje vlastní displej. Všechny se spoléhají na ovládání pomocí připojeného chytrého telefonu. Základní deska Raspberry Pi 3 se dá přitom pořídit do 1 000 Kč a užitý modul ZM5304 za $10.

To vede k myšlence tvorby domácí brány pomocí Raspberry Pi 3 a modulu ZM5304, doplněné o jednoduchý dotykový displej tak, aby uživatelské prostředí ovládacího systému bylo jak bezpečné, tak uživatelsky co nejpřívětivější.

5 Zařízení využitá k vytvoření Z-Wave sítě

Cílem této části je navrhnout základ domácí brány Z-Wave sítě tak, aby uměl získávat od Z-Wave zařízení informace (například o teplotě vzduchu) a aby také uměla Z-Wave přístroj ovládat (např. chytrou žárovku). Pro ukázkové řešení byla přitom vybrána jako vhodná testovací zařízení Raspberry Pi 3, ZM5304, Z-Wave Static Controller, CP2102-MINI, Aeotec MultiSensor 6 a Fibaro Wall Plug.

5.1 Raspberry Pi 3

Model třetí generace Raspberry Pi, který byl uveden na trh v únoru 2016. Hlavní novinkou této verze je zabudovaná Wi-Fi 802.11 b/g/n a Bluetooth 4.1 LE (Low Energy). Má o třetinu vyšší frekvenci procesoru i grafického jádra VideoCore než starší verze. Nová verze rovněž podporuje 64-bitovou architekturu. Hlavní specifikace modelu jsou:

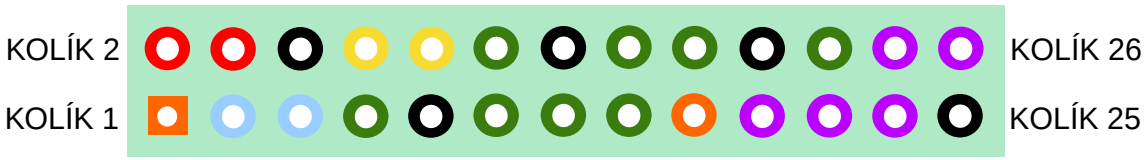
- Čtyřjádrový procesor Broadcom BCM2837 s frekvencí 1.2GHz podporující 64-bitovou architekturu.
- 1 GB RAM (Random-Access-Memory).
- Síťová karta BCM43438 podporující Wi-Fi, bluetooth a FM (Frequency Modulation) přijímač.
- 100 Mbps Base Ethernet.
- 40-pinové rozšíření GPIO (General Purpose Input/Output).
- 4xUSB 2.0 port.
- HDMI (High-Definition Multimedia Interface) standardní velikosti.
- Stereo audio výstup a kompozitní video port.
- Port fotoaparátu CSI (Camera Serial Interface) pro připojení Raspberry Pi kamery.
- Port DSI (Display Serial Interface) pro připojení dotykového Raspberry Pi displeje.
- Micro SD port pro operační systém.
- Micro USB port pro napájení[24].

5.1.1 Externí 40 PIN GPIO

GPIO kolíky lze konfigurovat jako univerzální vstup/výstup (viz obr. 5.1). Všechny Raspberry Pi, která jsou na trhu, podporují připojení k napětí 3.3V. Větší napětí může GPIO zničit[35].

Kolíky potřebné pro tuto práci jsou:

- Napájení +3V3 [1 a 17]



Obr. 5.1: 40 kolíkové GPIO

- Zem [6, 9, 14, 20 a 25]
- UART (Universal Synchronous / Asynchronous Receiver and Transmitter) TxD0 [8]
- UART RxD0 [10]

5.1.2 UART

Na čipu Raspberry Pi jsou dva vestavěné výstupy pro UART (PrimeCell UART(PL011)) a mini UART. Jsou implementovány pomocí různých hardwarových bloků, takže mají mírně odlišné vlastnosti. Všechna rozhraní ovšem podporují napětí 3.3V, takže připojení jiného systému, který využívá jinou úroveň napětí, musí být věnována zvláštní pozornost. Ve výchozím nastavení je PL011 u Raspberry napojen k bluetooth modulu, zatímco mini UART je použit pro Linuxový konzolový výstup.
V systému Linux se pak mini UART připojuje jako `/dev/ttyS0/` a PL011 jako `/dev/ttyAMA0/`. Primární UART závisí na modelu Raspberry Pi a je přístupný přes `/dev/serial0`.
Ve výchozím nastavení jsou kolíky přenosu a příjmu UART na GPIO 14 a 15, respektive jako 8 a 10 kolík na GPIO[34].

5.2 Z-Wave Static Controller (ACC-UZB2-E)

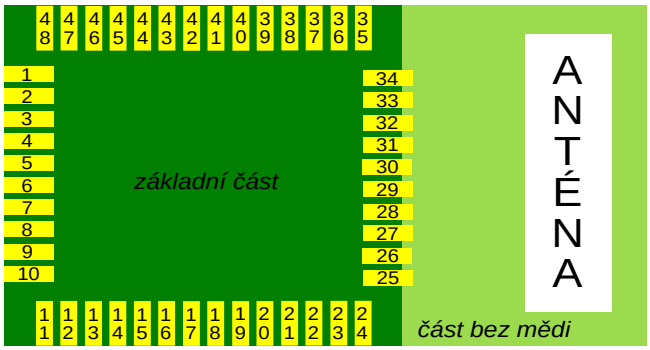
UZB (Universal Z-Wave Bus) je založen na sériovém rozhraní SD3503 SoC a obsahuje krystalové pasivní RF komponenty a anténu s regionálním SAW filtrem. Zařízení má dobře zdokumentované rozhraní Z-Wave Serial API přes rozhraní USB, takže jej lze aktualizovat bez nutnosti výměny hardwaru. Kontroler je napájen pomocí USB a má zabudovanou spirálovou anténu speciálně pro Z-Wave. Pro výrobu zařízení s využitím UZB je nutné vyžádat schválení FCC (Federal Communications Commission) a CE (Conformité Européenne)[27].

5.3 Z-Wave modul ZM5304 s anténou

Modul ZM5304 je plně funkční samostatný modul s vestavěnou anténou a s úplným schválením FCC a předběžným schválením CE. Je navržen tak, aby byl co nejsnadněji integrovatelný do zařízení jako jsou televize, set-top boxy a brány. Zařízení má dobře zdokumentované rozhraní Z-Wave Serial API buď přes rozhraní USB nebo UART (Universal Synchronous / Asynchronous Receiver and Transmitter), pomocí niž lze zařízení jednoduše používat i aktualizovat[25].

5.3.1 Konfigurace vývodů

Na základní desce najdeme celkem 48 vývodů (viz obr. 5.2). Přítomné jsou také vývody pro rozhrání USB a UART. Vývody mají obdelníkový tvar s rozměry 0.6 na 0.9mm[26].



Obr. 5.2: Schéma vývodů modulu ZM5304 (pohled z vrchu)

Jednotlivé očíslované vývody modulu jsou:

- V_{DD} [9] - Napájení modulu.
- GND [1, 8, 10, 11, 24, 25, 28, 30, 34, 35, 48] - Zem.
- NC [3, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 26, 27, 29, 31, 32, 33, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47] - Vývody pro mechanické uchycení.
- RESET_N [2] - Aktivní signál pro resetování modulu.
- UART0 RX [5] - Přijem dat z hostitelského portu.
- UART0 TX [4] - Vysílání dat do hostitelského portu.
- USB_DP [7] - USB 2.0 plnorychlostní positivní diferentní signál.
- USB_DM [6] - USB 2.0 plnorychlostní negativní diferentní signál.

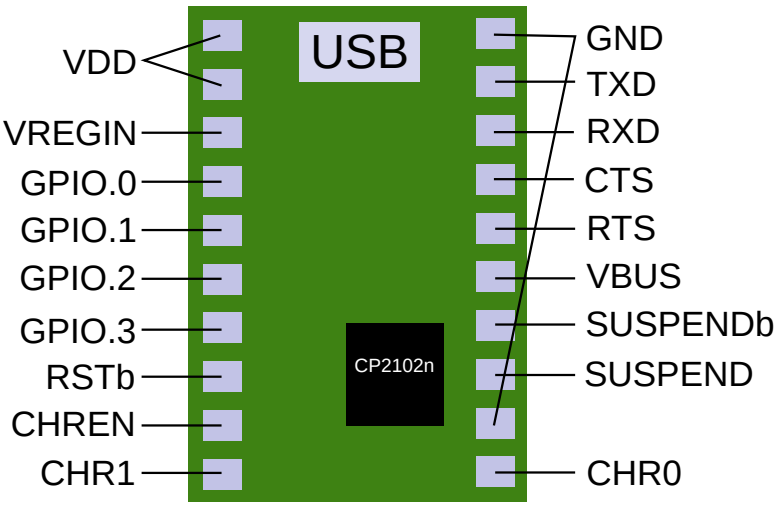
5.3.2 Programovací mód

Firmware je možné aktualizovat pomocí USB nebo UART rozhraní. Pro první aktualizaci musí být použité rozhraní UART, protože USB není ve výchozím stavu

povoleno. Do ISP (In-System Programming) programovacího módu dostaneme modul pomocí aktivního nízkého RESET_N signálu po dobu 5.2ms. Programovací jednotka pak čeká na sériový příkaz k povolení rozhraní před samotnou aktivací ISP přes rozhraní UART[26].

5.4 CP2102-MINI

CP2102-mini je UART-USB převodník od společnosti Silicon Lab, který pomocí volně dostupných ovladačů vytváří virtuální COM porty. Toto zařízení je kompatibilní s USB 2.0 a nabízí celkem 20 výstupů (viz obr. 5.3).



Obr. 5.3: Schéma CP2102n-mini - vrchní pohled

5.5 Aeotec MultiSensor 6 - ZW100

Multisenzor je Z-Wave zařízení 5. generace s certifikací Z-Wave Plus (viz obr. 5.4). Napájen je buď mini-USB kabelem anebo pomocí dvojice CR123A baterií. Při použití baterií vydržív zařízení pracovat až 2 roky. Senzor dokáže předávat informace ohledně pohybu, teploty, světla, vlhkosti, vibrací a UV (Ultraviolet) záření.

Senzor je založen na modulu ZM5101. Podporuje zabezpečený přenos pomocí kódování AES-128. Podporuje celkem 16 příkazových tříd:

- Informačně združená verze 1.
- Základní verze 1.
- Konfigurační verze 1.
- Firmwarově aktualizační verze 2.
- Notifikační verze 3.

Obr. 5.4: Multisenzor 6

- Bezpečnostní verze 1.
- Multiúrovňově senzorová verze 5.
- Budící verze 2.
- Asociační verze 2.
- Lokálně resetovací verze 1.
- Výrobně specifická verze 2.
- Úrovňově nabíjecí.
- Bateriová verze 1.
- Senzorově binární verze 1.
- Verze vydání 2.
- Z-Wave Plus informační verze 2.

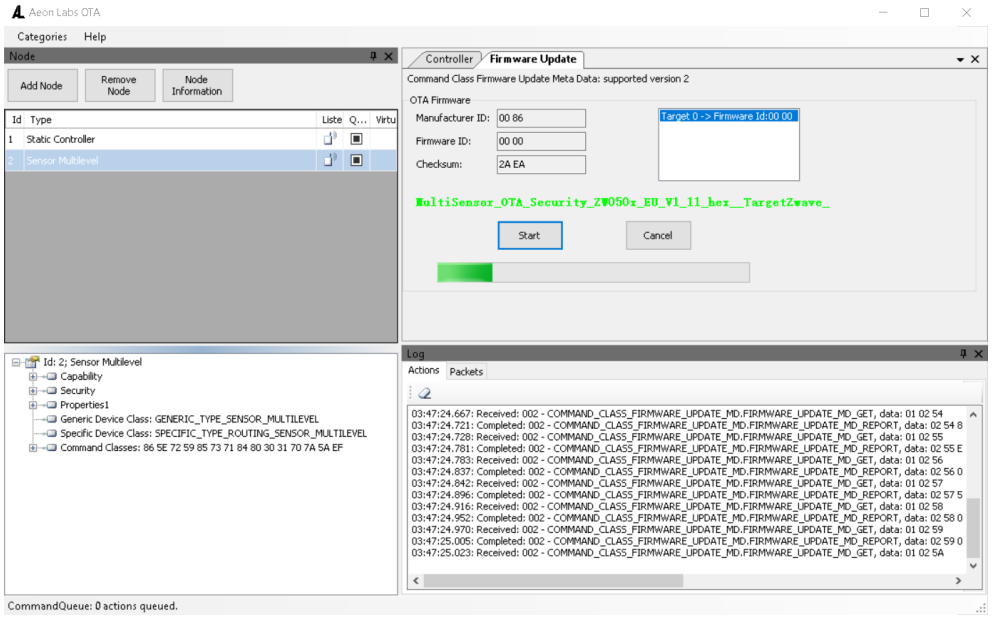
Senzor funguje jako podřízené zařízení a může přidat maximálně 5 přidružených uzlů. Při zjištění pohybu snímačem pošle do všech přidružených uzlů základní sadu (0xFF). Pokud není po určité době zjištěn žádný pohyb, pošle všem přidruženým uzlům základní sadu (0x00). Stejně tak po nakonfigurované době posílá hlášení o stavu teploty, vlhkosti, jasu a ultrafialového záření[13]

5.5.1 Aktualizace firmwaru

Pro aktualizaci firmwaru pro Multisenzor 6 od Aeotec je k dispozici na stránkách podpory od výrobce návod s programem a nejnovější verzí firmwaru rozděleným podle frekvence. Pro tento účel byla použita verze: European Union version frequency - version 1.11[1].

Stažený `.zip` archiv je nejdříve potřeba rozbalit. Uvnitř se nachází program Aeon Labs OTA. Po spuštění programu se v menu a nastavení objeví kontroler pro řízení sítě. Vybráním kontroleru se zobrazí uzly dané sítě. Pokud uzel v dané síti není, lze ho jednoduše přidat pomocí tlačítka `Add Node`. Zvolením senzorového uzlu lze

[1] `https://aeotec.freshdesk.com/helpdesk/attachments/6057343658`



Obr. 5.5: Aeon Labs OTA

již jednoduše vpravo zvolit záložku firmware update a tlačítkem `Start` aktualizovat (viz obr. 5.5) [15].

5.6 Fibaro Wall Plug - FGWPE-102

Fibaro Wall Plug je univerzální, dálkově ovladatelný adaptér do zásuvky s certifikací Z-Wave Plus (viz obr. 5.6). Toto zařízení může být umístěno kdekoliv, kde chce uživatel ovládat a kontrolovat napájení elektrických zařízení až do zatížení 2500W. Hlavními vlastnosti jsou:

- Kompatibilita se Z-Wave i Z-Wave Plus.
- Podpora ochraného módu (síťový bezpečnostní Z-Wave mód) s kódováním AES-128.
- Extrémně jednoduchá instalace zařízení.
- Z-Wave opakovač.
- Aktivní měřič výkonu a spotřeby elektrické energie.
- Indikace zátěže pomocí vícebarevného LED prstence.

Zástrčka je vybavena funkcí pro měření výkonu a energie. Pro signalizaci používá svítící LED (Light-Emitting Diode) prstenec, který mění barvu podle výkonu. Na zástrčce je ovládací tlačítko sloužící pro zapnutí/vypnutí a připojení/odpojení ze sítě Z-Wave [16].

Specifikace zařízení:

- Zdroj napájení - 230V AC, 50/60 Hz.

Obr. 5.6: Wall Plug

- Jmenovitý proud zátěže - 11A.
- Spotřeba energie do 1.6W.
- Výkon - do 2.5kW.
- Bezdrátový protokol - Z-Wave.
- Z-Wave série - 500.
- Bezdrátová frekvence - 868.4 nebo 869.8 MHz.
- Dosah - v budově 40 metrů, venku až 50 metrů.
- Provozní teplota - 0 až 40°C.
- Rozměry - 43 x 65 mm.

5.6.1 Ovládání

Pro připojení zástrčky do Z-Wave sítě je třeba nejdříve zapojit zařízení do elektrické zásuvky, potom na kontroleru vyvolat vyhledávání nového uzlu a to samé udělat i na Wall Plug zástrčce. To se provede rychlým zmáčknutím jediného tlačítka třikrát za sebou. Pokud je vyžadováno zabezpečené připojení, musí být připojení prováděno nejdále dva metry od Z-Wave kontroleru [16].

Do menu se lze dostat podržením jediného tlačítka. Přitom pak barvy LED prstence signalizují:

- Po třech vteřinách (bílá) vypnutí a zapnutí signalizace.
- Po šesti vteřinách (žlutá) vymazání paměti o elektrické spotřebě.
- Fialová barva značí test dosahu Z-Wave sítě.
- Žlutá barva dává možnost resetu zařízení.

5.6.2 Třídy

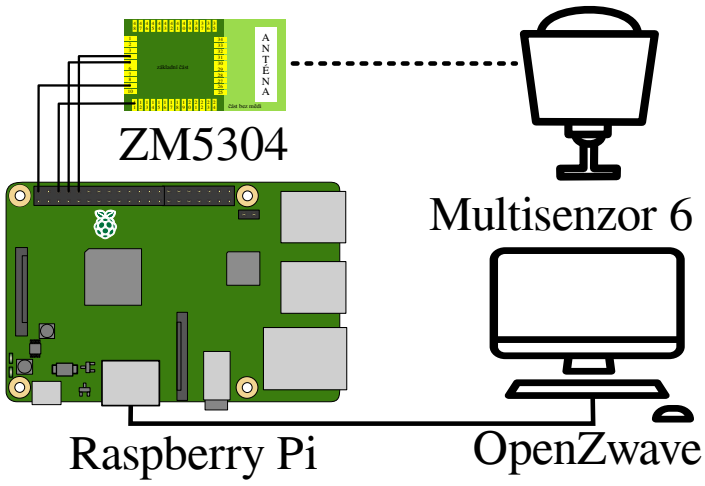
Zařízení je založeno na modulu ZM5202 a podporuje celkem 18 příkazových tříd[14]:

- Aplikační status.

- Asociační skupinové info.
- Asociační verze 2.
- Základní.
- Konfigurační.
- CRC16 zapouzdřovací.
- Lokálně resetovací.
- Firmware aktualizační verze 3.
- Výrobně specifická verze 2.
- Měřící verze 2.
- Multikanálová asociační verze 3.
- Notifikační verze 5.
- Výkonová.
- Bezpečnostní.
- Multiúrovňově senzorová verze 5.
- Senzorově binární.
- Verze vydání 2.
- Z-Wave+ informační verze 2.

6 Realizace Z-Wave sítě

Součástí práce je realizace Z-Wave sítě pomocí vybraných zařízení a knihoven (viz obr. 6.1). Jednotlivé kroky jsou popsány v této kapitole.



Obr. 6.1: Realizovaná Z-Wave síť

6.1 Instalace Raspian OS (Operating System)

Instalace Rasbian OS pro Raspberry Pi je možná dvěma způsoby. Buď stažením a rozbalením `.img` obrazu disku a nebo tzv. NOOBS (New Out Of Box Software) verzí. Pro instalaci je potřeba jiný počítač s SD (Secure Digital) čtečkou karet a mini SD kartu o velikosti alespoň 8Gb.

6.1.1 NOOBs

Při zvolení NOOBS verze se na stránkách `raspberrypi.org/downloads` stáhne aktuální verze systému. SD kartu lze pak naformátovat buď na OS Windows pomocí programu `SD Memory Card` Formatter[1] nebo na Linuxu pomocí `gparted` a `fdisk`. Na takto zformátovanou kartu se zkopíruje stažený a rozbalený `.zip` archiv a vloží se do Raspberry Pi. Po spuštění se postupuje dle instrukcí na obrazovce.

6.1.2 Pomocí obrazu disku

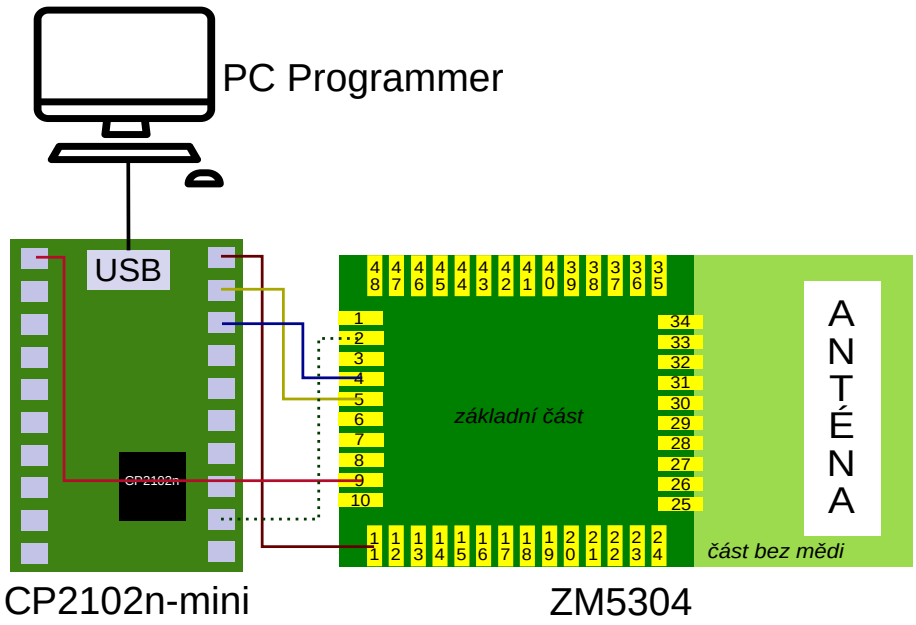
Při instalaci pomocí `.img` obrazu se stáhne aktuální verze a ta se např. pomocí nástroje `Etcher` rozbalí a zapíše na SD kartu. Program `Etcher`[2] je multiplatformní

[1] `https://www.sdcard.org/downloads/formatter_4/`
[2] `https://www.balena.io/etcher/`

a dokáže pracovat jak s `.img` tak se `.zip` soubory. Takto připravenou SD kartu lze vložit do Raspberry, kde se načte již nainstalovaný systém.

6.2 Nahrání firmwaru do ZM5304

Pro nahrání firmwaru do modulu ZM5304 je třeba využít počítač s operačním systémem Windows a aplikací PC programmer (viz kap. 3.5.3). Dále pak USB-UART převodník CP2102N-mini s doinstalovanými ovladači[3]. Potom je nutné USB kabelem propojit převodník s počítačem a spustit aplikaci. Je třeba k převodníku propojit vodiči i samotný modul ZM5304. Jeden vodič propojí napájení, druhý zem, třetí a čtvrtý UART konektory TXD (Transmit Data) a RXD (Receive Data) (viz obr. 6.2) Na kartě pro moduly řady 500 (ZW050x) lze detekovat Z-Wave zařízení (tools -> detect device -> Silicon Labs CP2102x USB to UART Bridge (COM x)) a vybrat požadovaný firmwarový `hex` soubor pro nahrání. V našem případě je to: `serialapi_controller_static_ZM5304_EU.hex`. Soubor lze nahrát pomocí tlačítka `Program and Verify`. Aplikace pak vyzve k zmáčknutí reset_n tlačítka (assert RESET_N). Toho je docíleno tak, že kolík reset_n modulu se spojí se zemí na převodníku. Po potvrzení se začne firmware nahrávat do modulu.

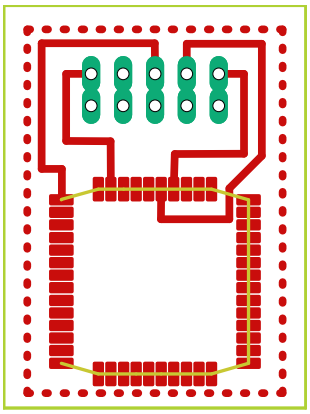


Obr. 6.2: Propojení ZM5304, CP2102n a PC

[3] `https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers`

6.3 Připojení ZM5304 k Raspberry Pi 3 pomocí UART

Připojení modulu k Raspberry Pi 3 je obdobné jako připojení k převodníku. Součástí Raspberry Pi 3 je GPIO (viz. kapitola 5.1.1), ke kterému je nutno připojit vodiče (napájecí, zem a UART) z modulu ZM5304. Pro jednodušší připojení a pevnější konstrukci byla s touto prací v programu Eagle[4] od firmy Autodesk (který je v zkladní verzi zdarma) navržena deska plošných spojů (viz obr. 6.3), která po spojení s modulem slouží jako jakási souprava pro Raspberry Pi 3.



Obr. 6.3: Deska plošných spojů pro ZM5304

Pro vlastní zprovoznění je potřeba ještě v systému Raspberry udělat několik nezbytných kroků. Nejdříve je třeba jako spávce upravit konfigurační soubor `/boot/config.txt`. Zde je nutné na konec přidat řádek:

```
enable_uart=1
```

a poté systém restartovat. Potom je třeba zadat do konsole následující příkazy:

```
sudo systemctl stop serial-getty@ttyS0.service
sudo systemctl disable serial-getty@ttyS0.service
```

a v souboru `/boot/cmdline.txt` vymazat:

```
console=serial0,115200
```

a v souboru `/boot/config.txt` přidat na konec řádek:

```
dtoverlay=pi3-miniuart-bt
```

Tímto bylo vypnuto bluetooth a bylo zapnut UART rozhraní.

[4] `https://www.autodesk.com/products/eagle/overview`

6.4 Instalace OpenZWave

Pro instalaci knihovny OpenZWave je nejlepší cestou stažení a překlad zdrojových kódů z github stránek[5]. Před samotným překladem je potřeba doinstalovat některé vývojářské nástroje pomocí příkazů:

```
apt-get update
apt-get install libgnutls28-dev libgnutlsxx28
```

Dále je také potřeba stáhnout a nakonfigurovat knihovnu libmicrohttpd:

```
cd
wget ftp://ftp.gnu.org/gnu/libmicrohttpd/libmicrohttpd
   -0.9.19.tar.gz
tar zxvf libmicrohttpd-0.9.19.tar.gz
mv libmicrohttpd-0.9.19 libmicrohttpd
cd libmicrohttpd
./configure
```

Teď je možné knihovnu přeložit a nainstalovat:

```
make
make install
```

Nyní je možné stáhnout a přeložit samotnou OpenZWave knihovnu. U té je možné dodat ještě parametr pro stažení opravdu poslední vývojové verze namísto té stabilní (–brench Dev):

```
cd
git clone --brench Dev https://github.com/OpenZWave/
   open-zwave.git
cd open-zwave
make
```

6.4.1 Control Panel

Po přeložení open-zwave knihovny lze doinstalovat webové GUI (Graphical User Interface) open-zwave-control-panel:

```
cd
git clone https://github.com/OpenZWave/open-zwave-
   control-panel.git
cd open-zwave-control-panel
```

[5] https://github.com/OpenZWave/open-zwave.git

V `Makefile` souboru zkontrolujeme odkaz na knihovny:

```
OPENZWAVE := ../open-zwave
LIBMICROHTTPD := /usr/local/lib/libmicrohttpd.a
```

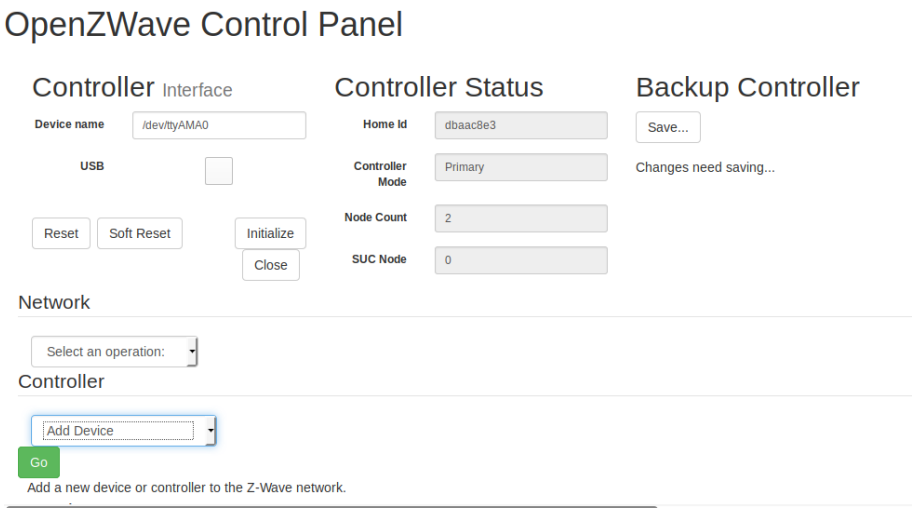
Teď už lze control-panel přeložit a spustit:

```
make
ln -sd ../open-zwave/config
./ozwcp -p 8888
```

Posledním příkazem příkazem se spustí control panel na lokální adrese na portu 8888.

6.4.2 Připojení modulu ZM5304 a Multisenzoru 6

Při spuštění webového prohlížeče na lokální adrese počítače a portu 8888 se zobrazí grafické rozhraní (viz obr. 6.4), kde je nejdříve potřebné zadat adresu modulu a inicializovat ho. Ten by měl být připojený jako `/dev/ttyAMA0`. V terminálu to lze ověřit příkazem: `ls -l /dev`. Pokud je modul inicializovaný, lze zadávat jednotlivé příkazy a vyčítat data z připojených senzorů. Pro přidání Multisenzoru 6 je ho nejprve zapotřebí přivést do párujícího módu. Toho lze docílit dvojím zmáčknutím tlačítka na senzoru (kontrolka se rozbliká zeleně). Ve webovém prohlížeči je třeba v sekci `Controller` vybrat možnost `Add Device` a stiknout zelené tlačítko `Go`. Poté se modul a senzor automaticky spárují.



Obr. 6.4: Open zwave control panel

6.4.3 Vyčtení dat z Multisenzoru

V Control Panelu se přidá řádek mezi spárovaná zařízení. Při kliku na tento řádek se objeví možnosti, které zařízení podporuje. V případě Multisenzoru 6 nabídne položku pro vypnutí/zapnutí senzoru, řádky s výčtem hodnot o teplotě, záření, vlhkosti, ultrafialového záření, typu a úrovni alarmu, ID zdrojového uzlu, vloupání a úrovně nabití. Tlačítkem `Refresh` pak lze požádat o aktuální hodnoty. Také lze nastavit, jak se budou hodnoty zobrazovat, například se teplota bude zobrazovat ve stupních Celsia nebo Fahrenheita.

6.5 Řešení pomocí jiných knihoven

V případě potřeby využití Z-Wave společně s dalšími platformami, je dobré využít některé z knihoven probíraných v kapitole 3.

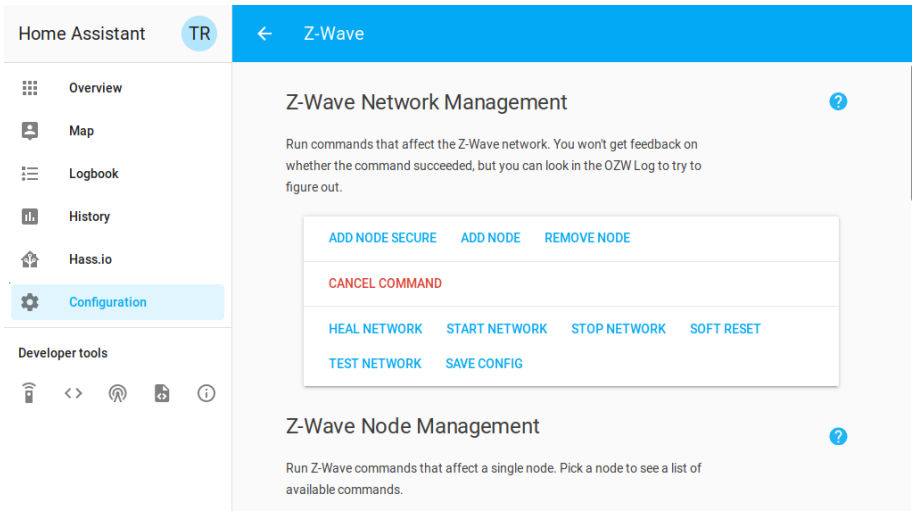
6.5.1 Instalace Home Assistant

V případě zprovoznění Z-Wave sítě pomocí Home Assistant (viz obr. 6.5) a Raspberry je nejjednoduší způsob využít systém Hass.io. Hass.io je operační systém pro Raspberry Pi s nainstalovanou a nakonfigurovanou aplikací Home Assistant. Instalační soubor pro danou verzi Raspberry je možné stáhnout na oficiálních stránkách[6]. Aplikace `Etcher` zmíněná v kapitole 6.1.2 pak nainstaluje systém na SD kartu. Je-li potřeba změnit či vynutit nastavení sítě, je třeba použít zformátovaný USB disk s názvem `CONFIG`, zde vytvořit složku s názvem `network` a v ní soubor `my-network`. Vzory tohoto souboru jsou dostupné na githubu[7]. Po vložení SD karty a USB disku do Raspberry probíhá první startování až 20 minut. Následně je možné se k platformě připojit na adrese: `http://hassio.local:8123`.

6.5.2 Instalace OpenHABian

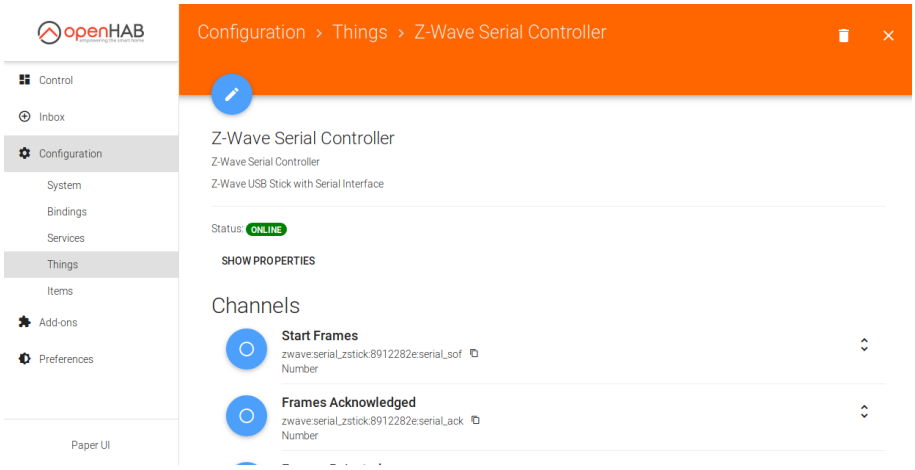
Podobně jako u zprovoznění Z-Wave sítě pomocí Home Assistant (viz kap. 6.5.1) je nejjednoduším způsobem pro zprovoznění využít přímo připraveného obrazu SD karty se systémem OpenHABian založeném na Rasbian Lite[8] (viz obr. 6.6). Motto tohoto systému je "nahraj, připoj, počkej a užívej". Nejdříve ze všeho je třeba obraz karty stáhnout z Githubu[9]. K rozbalení na SD kartu se použije aplikace `Etcher` zmíněná v kapitole 6.1.2. Takto připravená karta se vloží do Raspberry Pi. Po zapnutí

[6] `https://www.home-assistant.io/hassio/installation/`
[7] `https://github.com/home-assistant/hassos/blob/dev/Documentation/network.md`
[8] `https://www.raspberrypi.org/downloads/raspbian/`
[9] `https://github.com/openhab/openhabian/releases`



Obr. 6.5: Home Assistant

a připojení k internetu je třeba vyčkat 15-45 minut na doinstalování potřebných balíčků. Doba přitom záleží na rychlosti připojení a výkonu použitého hardwaru. K OpenHAB se dá pak připojit pomoci IP (Internet Protocol) adresy Raspberry a portu 8080 (`http://openhabianpi:8080`).



Obr. 6.6: OpenHAB

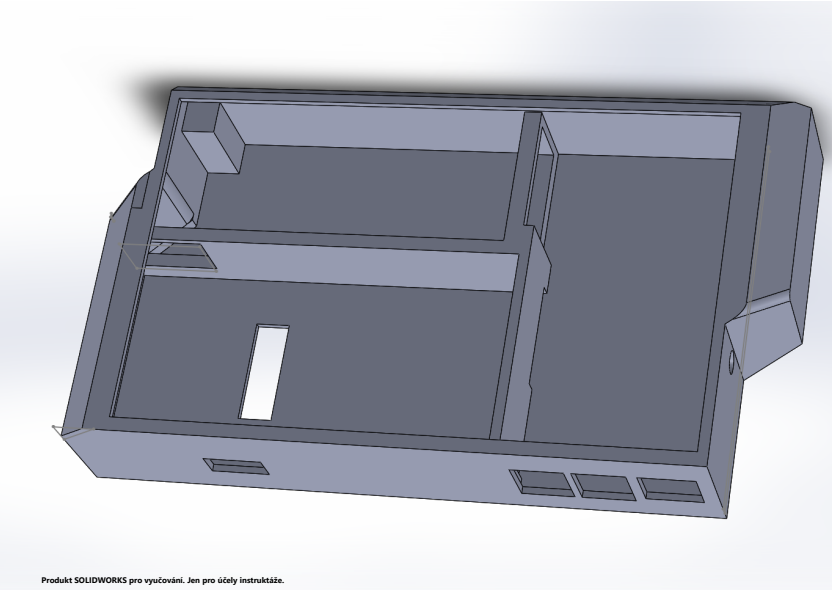
7 Vývoj vlastní brány

Jedním z cílů této práce bylo navrhnout a sestrojit vlastní bránu sloužící jako řídící centrum chytré sítě s platformou Z-Wave. Tato část práce staví na kapitolách 3 a 6 a bere si za cíl vytvoření celkového uživatelsky přívětivého řešení brány, které by mohlo sloužit jako alternativa ke komerčním řešením zmíněné v kapitole 4. K tomu byl využit zmíněný modul ZM5304, Raspberry Pi a vlastní program napsaný pomocí knihovny OpenZWave (viz. kap. 7.3). Pro lepší ovládání byl zakoupený dotykový displej pro Raspberry Pi. Pomocí programu SolidWorks, který je ve studentské verzi pro studenty VUT v Brně (Vysoké učení Technické) zdarma, bylo navrženo tělo přístroje a to bylo vytištěno na 3D tiskárně. Celková konstrukce pak funguje obdobně jako tablet.

Ke konstrukci jsou dále potřeba propojovací kabely a powerbanka sloužící k napájení Raspberry Pi i bez přímého připojení do sítě. Powerbanka musí mít výstupní napětí 5V a proud aspoň 2A. Pro tento konkrétní případ byla použita powerbanka Remax MyMax PowerBank CARBON 10000mAh, která stojí kolem 200Kč.

7.1 Návrh těla brány v programu SolidWorks

Jméno programu SolidWorks vzniklo složením dvou slov, slova *Solid*, které znamená prostorový objekt, a slova *Works*, které v překladu znamená práce. Je to tedy program pro práci s 3D objekty, pro jejich tvorbu a následné testování.



Obr. 7.1: Tělo navrženého tabletu (vrchní pohled)

Tvorba začíná vytvořením prvotní 2D skici v základní rovině. Pro tvorbu skici můžeme využít obdelníku, kruhu nebo jakéhokoliv jiného tvaru a pomocí inteligentních kód úsečkám přiřadit požadované rozměry. Po vytvoření základního tvaru se objekt převede do 3D pomocí vysunutí nebo rotace. Na jakoukoliv z jeho rovin nebo na rovinu základní se dá navázat tvorbou další skici, která může při převodu do 3D sloužit jak pro přidání tak i odebrání objektu. Dalším užitečným nástrojem je například zaoblení hran podle poloměru.

Tímto způsobem se postupně vytvoří díl (viz obr. 7.1). Z jednotlivých dílů se dají vytvořit sestavy. Výhodou je i to, že výsledné 3D objekty lze zpětně převést do dvourozměrných technických výkresů[36].

7.2 Konstrukce brány

Nejdříve je třeba odříznout u LCD (Liquid Crystal Display) obrazovky přebytečné výstupky pro uchycení. Takto upravený LCD displej umístíme do navrženého dvojdílného těla brány, které bylo vytištěného na 3D tiskárně. HDMI kabelem je propojen vstup na displeji s HDMI výstupem na Raspberry Pi 3. Pro funkčnost dotyku u obrazovky je rovněž nutné USB kabelem propojit Raspberry Pi 3 a LCD obrazovkou. Do těla je následně třeba vložit powerbanku a USB výstup, který garantuje proud 2A, propojit kabelem s Raspberry Pi 3. Poslední kabel je natažen mezi vstupem powerbanky a připraveným otvorem v těle brány (viz obr. 7.2).

Obr. 7.2: Konstrukce navržené brány (spodní pohled)

7.2.1 Instalace dotykové LCD obrazovky

Pro podporu dotykového ovládání LCD obrazovky je třeba stáhnout a nainstalovat ovladač z Githubu [1]:

```
git clone https://github.com/goodtft/LCD-show.git
chmod -R 755 LCD-show
cd LCD-show/
sudo ./LCD7B-show
```

7.3 Vývoj vlastní aplikace

Pro vytvoření vlastní aplikace pro domácí brány pomocí platformy Z-Wave bylo použito Python obalu pro OpenZWave, frameworku Kivy a vývojového prostředí PyCharm.

7.3.1 Python OpenZWave

Nová instalace OpenZWave knihovny se na OS Ubuntu provede pomocí balíčkového manageru Pythonu pip[2]. K tomu je třeba doinstalování několika dalších balíčků:

```
sudo apt-get install make libudev-dev g++ libyaml-dev pip
```

Dále je třeba vytvořit lokální virtuální kopii Pythonu a tu aktivovat:

```
virtualenv --python=python3 venv3
source venv3/bin/activate
```

V tomto prostředí lze pak jednoduše pomocí pip naistalovat python_openzwave a několik dalších nezbytných knihoven:

```
(venvX) pip install python_openzwave cython wheel six
   pyserial
```

Druhou možností je stažení a přeložení balíčků ze zdrojových kódů[3]. Správnost instalace je pak možné ověřit spuštěním testovacího souboru `pyozw_check`.

```
cd ~venv3/bin
pyozw_check -i -d /dev/ttyACM0
```

[1] https://github.com/goodtft/LCD-show

[2] https://pypi.org/project/pip/

[3] https://github.com/OpenZWave/python-openzwave

Překlad zdrojových kódů

Pro použití poslední vývojové verze Python-OpenZWave je třeba si stáhnout a přeložit zdrojové kódy. Ty jsou ke stažení na githubu a k stažení je nejlepší použít klonovací příkaz verzovacího systému Git.

```
git clone https://github.com/OpenZWave/python-openzwave/
cd python-openzwave/
```

Pro přeložení pak stačí zadat příkaz:

```
make build
```

Pokud se překlad nepovede (což je ve vývojových verzí častý problém), tak je třeba po opravě chyby před opětovným překladem vyčistit soubory po předchozím pokusu:

```
sudo make clean
```

Například ve verzi, která byla testována při psaní této práce je v `openzwave/cpp/src/Driver.cpp` chyba na 3222 - 3225 řádku, kde místo příkazu strncpy je třeba použít příkaz memcpy. Instalace přeloženého kódu je jednoduchá:

```
sudo make install
```

7.3.2 PyCharm

Pro rychlý a efektivní vývoj aplikace v programovacím jazyce Python je dobré využít vývojové prostředí. Jedním z takových je i PyCharm od firmy Jet Brains (viz obr. 7.3). To je možné stáhnout zdarma v komunitní verzi a v případě potřeby si lze zaplatit profesionální verzi. Na systému Ubuntu lze jednoduše nainstalovat snap[4] balíček pomocí příkazu:

```
sudo snap install pycharm-community --classic
```

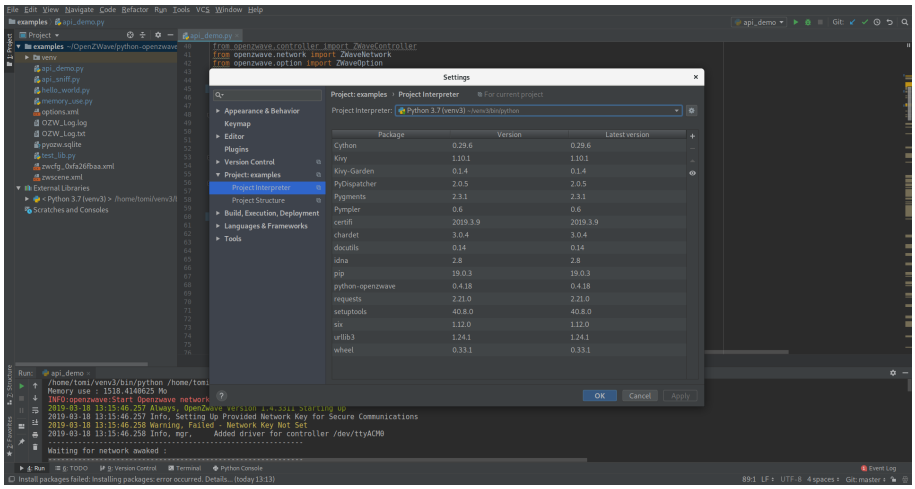
Pro správný překlad a spuštění kódu je zapotřebí v nastavení změnit cestu Python interpretru na lokální virtuální kopii.

7.3.3 Kivy

Pro vývoj dotykové aplikace bylo využito grafické prostředí Kivy. Instalace do lokální virtuální kopie je velice snadná:

```
source venv3/bin/activate
pip install cython
```

[4] `https://snapcraft.io/`



Obr. 7.3: PyCharm interpretr

```
pip install pygame
pip install kivy
```

Dále už stačí ve zdrojovém kódu importovat kivy aplikaci a vytvořit aplikační třídu (viz ukázkový kód):

```
import kivy
from kivy.app import App
from kivy.uix.button import Button

class TestApp(App):
  def build(self):
    return Button(text='Hello World')

TestApp().run()
```

7.3.4 Import knihoven

Na začátku Python programu je třeba importovat knihovny, které jsou v aplikaci využity. V tomto případě se jedná především o již zmíně knihovny OpenZWave a Kivy. Dále jsou potřeba standardní knihovny jako logging, sys, os, resource, math, time a knihovna pydispatch, která slouží pro zachytávání signálů ze sítě:

```
import kivy
from kivy.app import App
import openzwave
from openzwave.node import ZWaveNode
```

```
import time
from pydispatch import dispatcher
...
```

7.3.5 Grafické rozhraní v Kivy

Nejsnadnější a nejpřehlednější způsob tvorby grafického rozhraní s knihovnou Kivy je pomocí externího `.kv` souboru, který importujeme:

```
Builder.load_file('./soubor.kv')
```

V tomto souboru jsou jednotlivé objekty tvořeny deklarativním způsobem pomocí stromové architektury:

```
<SomeScreen>:
name: "some_screen"
  BoxLayout:
    id: box_layout
    orientation: "vertical"
    TextInput:
      id: tinput
    Button:
      on_release: root.confirm()
      id: confirm_button
      text: "Connect"
```

7.3.6 Nastavení Z-Wave komunikace

Před samotným připojení k Z-Wave síti, je třeba definovat několik základních nastavení:

```
options = ZWaveOption(device, config_path="../openzwave/
   config", user_path=".", cmd_line="")
options.set_log_file("OZW_Log.log")
options.set_append_log_file(False)
options.set_console_output(False)
options.set_save_log_level('Debug')
options.set_logging(True)
```

Aby nastavení fungovalo, tak je potřeba jej zamknout:

```
options.lock()
```

7.3.7 Připojení k síťi

Nyní již lze Z-Wave síť vytvořit a nastartovat:

```
network = ZWaveNetwork(options, autostart=False)
network.start()
```

Network objekt se nesmí použít dřív, než je síť připravena. To lze zařídit opakujícím se cyklem:

```
for i in range(0,90):
  if network.state>=network.STATE_READY:
    print("done")
    break
  else:
    time.sleep(1.0)
```

7.3.8 Zachytávání signálů

Z-Wave podporuje notifikace pomocí signálů. Ty lze v OpenZWave zachytávat pomocí `python-louie`. Celé to funguje tak, že `dispatcher` pošle signál po kanálu (o to se stará OpenZWave) a klient se zaregistruje k jeho zachytávání. Signálů je celá řada, a to síťových, ovládácích, notifikačních, chybových,... [19].

Nejdříve je tedy třeba navázat spojení s určitým kanálem. Například pro informaci o přidání nového uzlu do sítě provedeme připojení:

```
dispatcher.connect(self.node_added, ZWaveNetwork.
   SIGNAL_NODE_ADDED)
```

Dále je potřeba definovat, co se má v případě zachycení takového signálu vykonat:

```
def node_added(self, network, node):
  print('Added node: '+str(node.node_id))
```

7.3.9 Připojení uzlu do sítě

Připojení nešifrovaného uzlu do sítě je provedeno příkazem:

```
network.controller.add_node(False)
```

Zachycením signálu ze Z-Wave sítě pak lze přidání uzlu ověřit:

```
dispatcher.connect(self.node_added, ZWaveNetwork.
   SIGNAL_NODE_ADDED)
```

```
def node_added(self, network, node):
  print('Added node: '+str(node.node_id))
```

V případě připojení zabezpečeného uzlu do sítě se nejdříve musí nastavit 16-ti bytový bezpečnostní klíč. Ten se nachází v souboru `options.xml`:

```
<Option name="NetworkKey" value="0x06, 0x09, 0x03, 0x09,
   0x06, 0x03, 0x06, 0x09, 0x03, 0x09, 0x06, 0x03, 0x06,
   0x09, 0x03, 0x09" />
```

Šifrovanou komunikaci pak aktivujeme parametrem příkazu add_node:

```
network.controller.add_node(True)
```

Šifrovaná komunikace probíhá pomaleji, protože veškerá komunikace mezi zabezpečenými uzly se 3x opakuje a kontroluje. Po spojení zařízení napíše, které z tříd příkazů budou komunikovat zabezpečeně a které nezabezpečeně.

7.3.10 Vyčtení dat ze zařízení a ovládání zásuvky

Pro získání dat je potřeba projít jednotlivé třídy příkazů (cmd) a z nich vyčíst jednotlivé hodnoty (val):

```
for cmd in network.nodes[node_id].command_classes:
  for val in network.nodes[node_id].
     get_values_for_command_class(cmd):
```

Pro uživatele jsou důležité především hodnoty z kmene `Users`:

```
network.nodes[node_id].values[val].genre == "User"
```

Hodnoty, které jsou jen pro čtení a nedají se upravovat, je možné např. uložit do objektu `Label` a ten následně zobrazit:

```
if network.nodes[node_id].values[val].is_read_only:
  label = Label(text=value.label+value.data_as_string)
```

V případě hodnot, které je možné přepisovat (například ovládání chytré zásuvky), se při jejich zpracování lze řídit jejich typem, který zjistíme pomocí atributu type, např.:

```
elif network.nodes[node_id].values[val].type == "Bool":
```

Pokud je hodnota přepisovatelná a booleovského typu, je vhodné pro větvení použít konstrukci `Switch` (přepínač).

7.3.11 Lambda funkce

Lambda funkce je jednořádková funkce také známá jako anonymní funkce. Funkce je užitečná, pokud chceme jednu funkci použít vícekrát. Lambda funkce může mít libovolný počet argumentů, ale má pouze jeden výraz [45].

```
lambda argumenty: výraz
```

V programu psaném pro tuto práci se lze s lambda funkcí setkat především při tvorbě nových aktivních prvků (tlačítek), kterým přirazujeme některou z funkcí, na kterou je třeba se odkázat dříve než je tlačítko vytvořené.

7.3.12 SQLite a Phyton

Pro použití SQLite v jazyce Python jsou běžně používaná dvě rozhraní, a to PySQLite a APSW (Another Python SQLite wrapper).

PySQLite

PySQLite poskytuje standartizované rozhraní DBI (Database Interface) API pro Python, které je kompatibilní nejen s databází SQLite, ale také s dalšími SQL databázemi, jako je MySQL, Oracle či PostgreSQL. PySQLite je standartní součástí Python knihovny od verze 2.5[38].

APSW

Pokud aplikaci stačí pouze SQLite knihovna, je vhodnější použít APSW modul. Ten poskytuje nejtenčí možnou vrstvu nad SQLite. Je navržen tak, aby napodoboval nativní SQLite C API, a proto v něm lze dělat vše to, co v nativním SQLite C API. Kromě toho poskytuje mnoho nízkoúrovňových funkcí a umožňuje psaní implementace virtuální tabulky pomocí Pythonu[38].

Instalace APSW se nejsnadněji provede pomocí správce Python balíčku pip:

```
pip3 install https://github.com/rogerbinns/apsw/releases/
   download/3.27.2-r1/apsw-3.27.2-r1.zip --global-option=
   fetch --global-option=--version --global-option=3.27.2
    --global-option=--all --global-option=build --global-
   option=--enable-all-extensions
```

Vytvoření/připojení databáze a ukončení spojení

První, co je potřeba v programu využívajícím SQLite udělat, je vytvoření připojení k existující databázi pomocí funkce *connection()*. Ta otevře spojení s databází a

vrátí objekt, reprezentující databázi. Pokud soubor, ke kterému se snažíme připojit neexistuje, tak se automaticky sám vytvoří:

```
import apsw
database = apsw.Connection("values.db")
```

Při ukončení programu je třeba spojení ukončit:

```
database.close()
```

Vytvoření tabulky

Před samotným vytvořením tabulek, reprezentujících v tomto případě jednotlivé uzly v síti, je třeba vytvořit zavoláním metody *Cursor()* objektu vráceného metodou *Connection()*. Pomocí kurzoru a příkazu *execute* lze do datbáze zadat jakýkoliv SQL příkaz:

```
cursore.execute("sql příkaz")
```

V naší aplikaci se při každém připojení nového uzlu do sítě se vytvoří nová tabulka, do které se budou zapisovat data podle času. Protože je nežádoucí, aby se pro danou veličinu ve stejný čas zapsaly dvě hodnoty, je vhodné tabulkám definovat primárního klíče:

```
CREATE TABLE IF NOT EXISTS nodeX (label, time, data,
   PRIMARY KEY (label, time))
```

Záznam hodnot a vyčtení dat z tabulky

Pro záznam hodnot je využito zachytávání signálů (viz kap. 7.3.8) pro novou hodnotu a pro změnu hodnoty. Při každém takovém signálu se nové hodnoty zapíši SQL příkazem do příslušné tabulky:

```
INSERT INTO nodeX(label, time, data) VALUES("val-label",
   "actual_time", "val_data")
```

V případě potřeby vyčtení hodnot se využije příkaz:

```
SELECT time, data FROM nodeX WHERE label="val-label"
```

Aby se jednotlivé hodnoty z tabulky načetly do objektové proměnné Pythonu, je nutné spustit příkaz:

```
values = cursor.fetchall()
```

7.3.13 Zobrazení grafu v Kivy

K zobrazení grafu v aplikaci je třeba doinstalovat rozšiřující modul `Kivy-garden`:

```
pip install kivy-garden
garden install graph
```

Po importu se nejprve vytvoří pole grafu. Tomu se určí rozmezí zobrazovaných hodnot, označení os, barva a další potřebné atributy. Do proměnné plot se načtou jednotlivé dvojrozměrné hodnoty. Výsledný graf se vloží do pole grafu a tím se zobrazí.

7.4 Uživatelské rozhraní aplikace

Pro ovládání Z-Wave sítě byla pomocí Python wrapperu pro OpenZWave vyvinuta ukázková aplikace. Ta obsahuje čtyři základní obrazovky. V této verzi umí:

- Připojit se pomocí kontroleru k Z-Wave síti.
- Získat informace o jednotlivých uzlech sítě.
- Přidat a odebrat zařízení sítě v zabezpečeném i nezabezpečeném režimu.
- Zobrazit informace o uživatelských hodnotách uzlu.
- Ovládat akční prvky jako je například chytrá zásuvka.
- Automaticky zapisovat uživatelské hodnoty do SQLite databáze.
- Zobrazit graf vývoje určité uživatelské hodnoty ve zvoleném období.
- Resetovat síť do továrního nastavení, obnovit informace a vyléčit uzel.

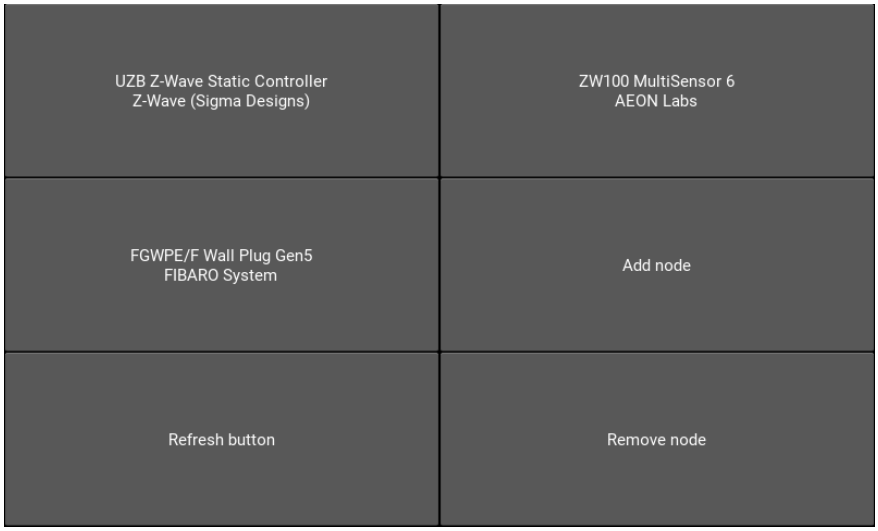
7.4.1 Obrazovka pro připojení k síti

První obrazovka, která se zobrazí po spuštění aplikace, slouží pro inicializaci kontroleru a připojení k síti (viz obr. 7.4). V dialogu si uživatel zvolí souborovou cestu ke kontroleru. Pro UZB Static Controller je cesta `/dev/ttyACM0`, pro modul připojený přes GPIO rozhraní Raspberry Pi 3 je cesta `/dev/ttyAMA0`. Tlačítko ve spodní části obrazovky pak slouží pro potvrzení cesty a připojení k Z-Wave síti.

7.4.2 Hlavní obrazovka

Po připojení k síti si aplikace načte všechny informace o síti a uzlech do objektové proměnné. Hlavní obrazovka zobrazí pro každý uzel v síti tlačítko (viz obr. 7.5). Při stisknutí tlačítka daného uzlu se zobrazí obrazovka informací (viz kap. 7.4.3). Dále jsou tu tlačítka pro přidání a odebrání zařízení. Při aktivaci tlačítka pro přidání nového uzlu se zobrazí vyskakovací tlačítko pro výběr mezi zabezpečeným a nezabezpečeným režimem. Poslední z tlačítek na hlavní obrazovce je pro obnovu a opětovné načtení hlavní obrazovky.

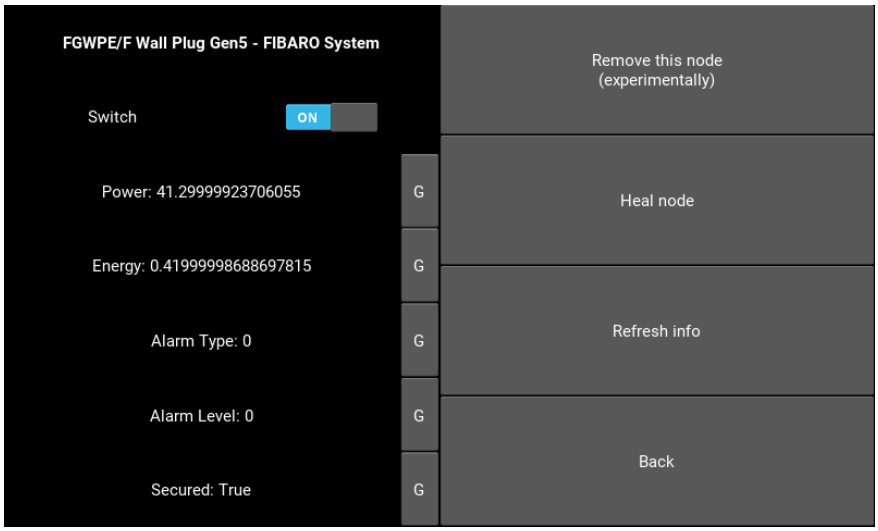
Obr. 7.4: Obrazovka pro připojení k síti



Obr. 7.5: Hlavní obrazovka

7.4.3 Obrazovka informací o uzlu

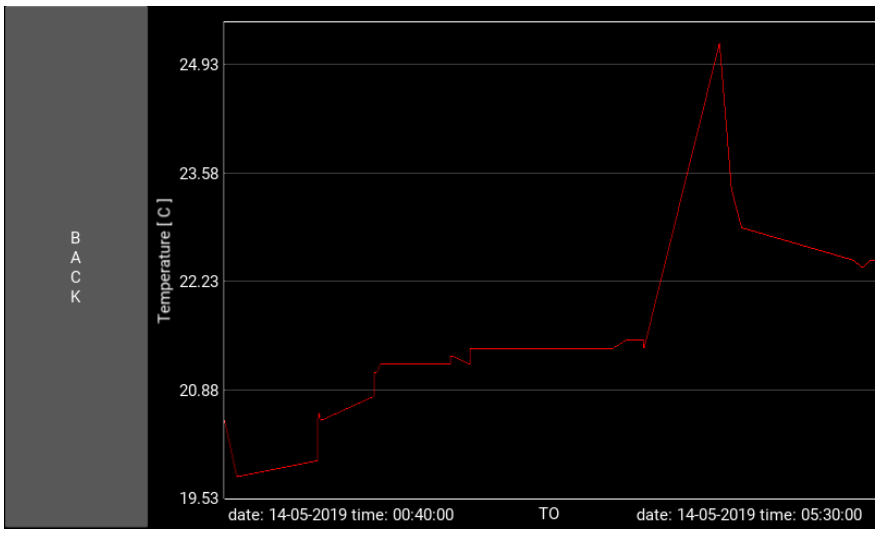
Při stisku tlačítka určitého uzlu se zobrazí informační obrazovka (viz obr. 7.6). Ta je rozdělena na dva sloupce. V pravém sloupci jsou tlačítka pro odstranění daného uzlu ze sítě, vyléčení uzlu, obnově informací a návratu na hlavní obrazovku. V levém sloupci se nacházejí jednotlivé aktuální uživatelské hodnoty vyčtené z daného uzlu, a to ve formátu `popis: data`. U akčního prvku, jako je například chytrá zásuvka, se zobrazí přepínač pro zapnutí a vypnutí. U každé uživatelské hodnoty je vpravo tlačítko `G`, které slouží pro zobrazení vývoje dané hodnoty v čase (viz kap. 7.4.4). V případě kontroleru je na levé spodní části ještě tlačítko pro resetování sítě do továrního nastavení a vymazání veškerých dočasných souborů a databází.



Obr. 7.6: Obrazovka informací o uzlu

7.4.4 Obrazovka grafu

Poslední obrazovka aplikace slouží pro zobrazení grafu. Při stlačení `G` tlačítka u dané hodnoty na informační obrazovce se zobrazí vyskakovací okno pro výběr rozmezí, které chce uživatel na grafu zobrazit. Buď může levým tlačítkem vybrat všechny údaje z databáze anebo rozmezí upraví a stiskne tlačítko `Select`. Obrazovka grafu má na levé straně jediné tlačítko pro návrat na předchozí obrazovku (viz obr. 7.7). Zbytek obrazovky tvoří graf, kde na ose x je datum a čas v kterém se data zobrazují, na ose y je pak rozmezí hodnot veličiny včetně jednotky.



Obr. 7.7: Obrazovka grafu vybrané hodnoty vybraného uzlu

8 Závěr

Cílem diplomové práce bylo vytvoření domácí brány komunikující pomocí technologie Z-Wave. Brána byla realizována pomocí Raspberry Pi 3 a modulu ZM5304 od Silicon Labs. Zařízení přitom je konfigurovatelné například pomocí webového prohlížeče. Platforma Z-Wave, vyvíjená Z-Wave Aliancí, je jedním z nejznámějších řešení chytré domácnosti, a to i v České republice. U předních prodejců elektroniky si může zákazník koupit brány, senzory i akční členy a to především od polské firmy Fibaro. O vzájemnou kompatibilitu se stará Z-Wave Aliance pomocí certifikace Z-Wave Plus. Z-Wave síť využívá smíšenou síťovou topologii, kde každé zařízení v síti funguje jako uzel, a každý tento uzel si pamatuje nastavení sítě do které patří. Přitom pro přenos dat se využívá bezlicenční ISM pásmo s frekvencí pod 1 GHz. Tímto způsobem se systém vyhýbá přeplněným pásmům 2,4 GHz a 5 GHz.

Během této práce bylo nejprve zapotřebí zdokumentovat celou Z-Wave platformu, a to včetně dostupných knihoven pro tvorbu domácí brány. Dále bylo nutné vyřešit nahrání firmwaru do modulu ZM5304 bez použití velice drahé desky ZDP03. To bylo uskutečněno pomocí dostupné aplikace Z-Wave Programer a převodníku CP2102n-mini. Po nahrání firmwaru bylo potřebné zprovoznit Raspberry Pi 3 s operačním systémem Raspbian a připojit k němu modul ZM5304. Pro připojení modulu přes UART bylo nejprve nutno odpojit od Raspberry Bluetooth modul. Pro ověření funkčnosti sítě byla jako vhodná zvolena knihovna OpenZWave a GUI rozhraní Control Panel, jejichž kód bylo potřeba přeložit a konfigurovat speciálně pro Raspberry. Nakonec byl k takto vytvořené síti připojen Multisenzor 6 a byla z něho vyčtena data o teplotě, světlu, vlhkosti, pohybu, UV záření a vibracích. Dále byla připojena zástrčka Wall Plug, kterou je možné pomocí systému vzdáleně ovládat. Control Panel také umožňuje načítat hodnoty senzoru a ovládat připojená zařízení vzdáleně přes webové rozhraní. Pro lepší připojení modulu ZM5304 k Raspberry Pi byla v programu Eagle navržena deska plošných spojů. Pro sestrojení kompletní uživatelsky přívětivé ovládací Z-Wave brány bylo také použito dotykové obrazovky, powerbanky a těla systému. To bylo navržené v programu SolidWorks a vytisknutého na 3D tiskárně. Do takto vytvořeného zařízení byla s využitím knihovny OpenZWave, Python wraperu, grafického rozhraní Kivy a SQLite naprogramovaná aplikace, pomocí níž se lze připojit k Z-Wave síti, přidat a odebrat zařízení (zabezpečeně i nezabezpečeně), vyčíst z nich data a ovládat WallPlug zásuvku. Navíc je možno sbírat data v čase a následně je zobrazovat v grafech.

V diplomové práci tak byly dosaženy všechny podstatné stanovené cíle. Sestrojená brána je univerzální a kód aplikace je možné dál rozšiřovat. Jedním z rozšíření aplikace by mohlo být využití scén, které jsou přímo definovány Z-Wave platformou a které jsou podporovány OpenZWave knihovnou.

Literatura

[1] KNX *KNX System Specifications* [cit. 2018-12-5]. Dostupné z: http://www.sti.uniurb.it/romanell/Domotica_e_Edifici_Intelligenti/110504-Lez10a-KNX-Architecture%20v3.0.pdf

[2] Z-WAVE ALIANCE *About* [online] [cit. 11. 10. 2018]. Dostupné z: <https://z-wavealliance.org/z-wave-alliance-overview/>.

[3] Z-WAVE ALIANCE *join* [online] [cit. 11. 10. 2018]. Dostupné z: <https://z-wavealliance.org/join/>.

[4] Z-WAVE ALIANCE *history* [online] [cit. 11. 10. 2018]. Dostupné z: <https://z-wavealliance.org/z-wave_alliance_history/>.

[5] ROBLES, Rosslin John a Kim TAI-HOON *Applications, Systems and Method s in Smart Home Technology.* International Journal of Advanced Science and Technology. 2010(15), 39. ISSN 2005-4238.

[6] BADENHOP, Christopher W., Scott R. GRAHAM, Benjamin W. RAMSEY, Barry E. MULLINS a Logan O. MAILLOUX. *The Z-Wave routing protocol and its security implications.* Computers & Security. 2017, 68(68), 112-129. DOI: 10.1016/j.cose.2017.04.004. ISSN 01674048. Dostupné také z: https://linkinghub.elsevier.com/retrieve/pii/S0167404817300792

[7] Z-WAVE ALIANCE *Z-Wave Frequency Coverage* [online]. 2012 [cit. 2018-10-21]. Dostupné z: http://store.zwavecenter.com/image/catalog/PDF/Z-Wave_Frequency_Coverage.pdf

[8] Software Design Specification *Z-Wave Application Security Layer* [cit. 2018-11-21]. Dostupné z: https://www.silabs.com/documents/login/reference-manuals/SDS10865-Z-Wave-Application-Security-Layer-S0.pdf

[9] Silicon Labs *Z-Wave Security Framework* [cit. 2018-11-21]. Dostupné z: https://www.silabs.com/products/wireless/mesh-networking/z-wave/specification/security

[10] Silicon Labs *Instruction Z-Wave Network Installation and maintenance Procedures User Guide* [cit. 2018-11-21]. Dostupné z: https://www.silabs.com/documents/login/user-guides/INS12712-Z-Wave-Network-Installation-maintenance-Procedures-User-Guide.pdf

[11] Silicon Labs *Z-Wave Management Command Class Specification* [cit. 2018-12-4]. Dostupné z:
https://www.silabs.com/documents/login/miscellaneous/SDS13782-Z-Wave-Management-Command-Class-Specification.pdf

[12] Silicon Labs *What is Z-Wave 700* [cit. 2019-13-3]. Dostupné z:
https://www.silabs.com/products/wireless/mesh-networking/z-wave/700-platform/what-is-z-wave-700

[13] Z-Wave Alliance *MultiSensor 6 - Z-Wave Protocol Implementation Conformance Statement* [cit. 2018-12-4]. Dostupné z:
https://products.z-wavealliance.org/products/2511/pics

[14] Z-Wave Alliance *FIBARO Wall Plug E/F - Z-Wave Protocol Implementation Conformance Statement* [cit. 2019-4-9]. Dostupné z:
https://products.z-wavealliance.org/products/2818

[15] Aeotec Group *Update Multisensor 6 firmware via Z-Stick or Z-Stick Gen5* [cit. 2019-5-10]. Dostupné z:
https://aeotec.freshdesk.com/support/solutions/articles/6000036562-how-to-update-multisensor-6-z-wave-firmware-

[16] Fibaro system *Fibaro Wall Plug - Operating manual* [cit. 2019-5-8]. Dostupné z:
https://manuals.fibaro.com/content/manuals/en/FGWPEF-102/FGWPEF-102-EN-A-v2.1.pdf

[17] Z-WAVE>ME *about* [online] [cit. 11. 10. 2018]. Dostupné z: <https://z-wave.me/z-way/>.

[18] OpenZWave *introduction* [online] [cit. 26. 10. 2018]. Dostupné z: <http://www.openzwave.com/dev/>.

[19] OpenZWave *Hello World* [online] [cit. 22. 4. 2019]. Dostupné z: <https://github.com/OpenZWave/python-openzwave/blob/master/docs/hello_world.rst>.

[20] OpenZWave *1.6 Release Notes* [online] [cit. 22. 4. 2019]. Dostupné z: <https://github.com/OpenZWave/open-zwave/wiki/OpenZWave-1.6-Release-Notes>.

[21] Python-OpenZWave *Documentation* [online] [cit. 1. 5. 2019]. Dostupné z: <http://openzwave.github.io/python-openzwave/README.html>.

[22] Kivy *Welcome to Kivy* [online] [cit. 2. 5. 2019]. Dostupné z: <https://kivy.org/doc/stable/>.

[23] Kivy *User's Guide » Philosophy* [online] [cit. 2. 5. 2019]. Dostupné z: <https://kivy.org/doc/stable/philosophy.html>.

[24] Raspberry *Raspberry Pi 3* [online] [cit. 27. 10. 2018]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.

[25] Silicon Labs *ZM5304 Modul* [online] [cit. 27. 10. 2018]. Dostupné z: <https://www.silabs.com/products/wireless/mesh-networking/z-wave/modules/zm5304-module>.

[26] ZM5304 Modul *DataSheet* [online] [cit. 27. 10. 2018]. Dostupné z: <https://www.silabs.com/documents/login/data-sheets/DSH12461-14.pdf>.

[27] Silicon Labs *Z-Wave Static Controller* [online] [cit. 27. 10. 2018]. Dostupné z: <https://www.silabs.com/products/development-tools/wireless/mesh-networking/z-wave/usb-z-wave-static-controller-reference-design>.

[28] Silicon Labs *Embedded SDK* [online] [cit. 12. 11. 2018]. Dostupné z: <https://www.silabs.com/products/development-tools/software/z-wave/embedded-sdk>.

[29] Silicon Labs *Controller SDK* [online] [cit. 12. 11. 2018]. Dostupné z: <https://www.silabs.com/products/development-tools/software/z-wave/controller-sdk>.

[30] Silicon Labs *Z-Wave Software and Tools* [online] [cit. 2. 11. 2018]. Dostupné z: <https://www.silabs.com/products/development-tools/software/z-wave>.

[31] Silicon Labs *Introduction to Z-Wave SmartStart* [online] [cit. 4. 11. 2018]. Dostupné z: <https://www.silabs.com/documents/login/white-papers/introduction_to_z-wave_smartstart_091317.pdf>.

[32] Silicon Labs *Z-Wave for Energy Management* [online] [cit. 11. 12. 2018]. Dostupné z: <https://www.silabs.com/products/wireless/mesh-networking/z-wave/benefits/technology>.

[33] Silicon Labs *FLiRS* [online] [cit. 11. 12. 2018]. Dostupné z: <https://www.silabs.com/documents/login/white-papers/Z-Wave-FLiRS.pdf>.

[34] Raspberry Pi *UARTs* [online] [cit. 5. 11. 2018]. Dostupné z: <https://www.raspberrypi.org/documentation/configuration/uart.md>.

[35] Raspberry Pi *GPIO* [online] [cit. 5. 11. 2018]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/README.md>.

[36] SolidWorks *Učební text* [online] [cit. 1. 5. 2019]. Dostupné z: <http://sos-vyskov.cz/strojtech/download/ucebnice-solidworks.pdf>.

[37] SQLite *What Is SQLite* [online] [cit. 1. 5. 2019]. Dostupné z: <https://www.sqlite.org/index.html>.

[38] SQLite Python *Tutorial* [online] [cit. 1. 5. 2019]. Dostupné z: <http://www.sqlitetutorial.net/sqlite-python/>.

[39] OpenHAB *OpenHABian* [online] [cit. 20. 11. 2018]. Dostupné z: <https://www.openhab.org/docs/installation/openhabian.html>.

[40] OpenHAB *Welcome* [online] [cit. 20. 11. 2018]. Dostupné z: <https://www.openhab.org/docs/>.

[41] OpenHAB *ZWave Binding* [online] [cit. 21. 11. 2018]. Dostupné z: <https://www.openhab.org/addons/bindings/zwave/>.

[42] Z-WAVE ALIANCE *Z-Plus* [online] [cit. 11. 11. 2018]. Dostupné z: <https://z-wavealliance.org/z-wave_plus_certification/>.

[43] Zigbee alliance *Zigbee 3.0* [online] [cit. 6. 12. 2018]. Dostupné z: <https://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>.

[44] EnOcean alliance *What is EnOcean?* [online] [cit. 6. 12. 2018]. Dostupné z: <https://www.enocean-alliance.org/what-is-enocean/>.

[45] Python Tips *Lambdas* [online] [cit. 28. 4. 2019]. Dostupné z: <http://book.pythontips.com/en/latest/lambdas.html>.

Seznam symbolů, veličin a zkratek

AC Alternating Current
AES Advanced Encryption Standard
AJAX Asynchronous JavaScript and XML
API Application Programming Interface
APSW Another Python SQLite wrapper
ASCII American Standard Code for Information Interchange
CE Conformité Européenne
CES Consumer Electronics Show
COM Communication port
CPU Central Processing Unit
CSI Camera Serial Interface
DBI Database Interface
DLL Dynamic-Link Library
DSI Display Serial Interface
DTLS Datagram Transport Layer Security
DUT Device Under Test
GPIO General Purpose Input/Output
HDMI High-Definition Multimedia Interface
Hz Hertz
IEEE Institute of Electrical and Electronics Engineers
ISM Industrial, Scientific and Medical
ISP In-System Programming
IoT Internet of Things
IP Internet Protocol
LCD Liquid Crystal Display
LED Light-Emitting Diode
LGPL Lesser General Public License
GUI Graphical User Interface
FCC Federal Communications Commission
FLiRS Frequently Listening Routing Slave
FM Frequency Modulation
MAC Medium Access Control
MIT Massachusetts Institute of Technology
MiOS Media Inteface Operating System
MIPS Microprocessor without Interlocked Pipeline Stages
MPDU MAC Protocol Data Unit
NOOBS New Out Of Box Software

OpenHAB Open Home Automation Bus
OTA Over The Air
OS Operating System
OSS Open-Source
PC Personal Computer
PLC PowerLine Communication
RAM Random-Access-Memory
RF Radio Frequency
RXD Receive Data
SoC System on a Chip
SAW Surface Acoustic Wave
SD Secure Digital
SDK Software Development Kit
SQL Structured Query Language
TXD Transmit Data
UART Universal Synchronous / Asynchronous Receiver and Transmitter
USB Universal Serial Bus
UV Ultraviolet
UZB Universal Z-Wave Bus
VUT Vysoké učení Technické
WLAN Wireless Local Area Network
XML eXtensible Markup Language