VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

VYTVOŘENÍ BRÁNY PRO CHYTRÁ ZAŘÍZENÍ XIAOMI AQARA

GATEWAY FOR XIAOMI AQARA DEVICES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **PETR URBÁNEK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. JAN PLUSKAL**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Urbánek Petr**
Program: Informační technologie
Název: **Vytvoření brány pro chytrá zařízení Xiaomi Aqara**
A Gateway for Xiaomi Aqara Devices
Kategorie: Vestavěné systémy

Zadání:

1. Prostudujte chytrou bránu Xiaomi Aqara a způsob její komunikace s zařízeními.
2. Proveďte reverse-engineering komunikace mezi zařízením a bránou pro vybraná zařízení dle doporučení vedoucího.
3. Navrhněte funkční vzorek s využitím volně dostupných komponent schopný nahradit oficiální bránu Xiaomi Aqara.
4. Implementujte funkční vzorek a zjistěte za jakých podmínek je možné Vámi vyvinuté zařízení použít k nahrazení oficiální brány.
5. Integrujte implementovanou bránu s nástrojem Home Assistant a demonstrujte použití s vybranými Xiaomi Aqara senzory.

Literatura:

- Kinney, P. (2003, October). ZigBee Technology: Wireless Control that Simply Works. In *Communications design conference* (Vol. 2, pp. 1-7).
- Farahani, S. (2011). *ZigBee Wireless Networks and Transceivers*. Newnes.
- Gill, K., Yang, S. H., Yao, F., & Lu, X. (2009). A ZigBee-Based Home Automation System. *IEEE Transactions on consumer Electronics*, *55*(2), 422-430.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Pluskal Jan, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 26. října 2020

Abstrakt

Cílem této práce je vytvoření náhrady za bránu pro chytrá zařízení značky Aqara. Práce popisuje zařízení ekosystému Xiaomi Aqara, která jsou použita pro testování i reverse-engineering komunikace. Brána je vytvořena na platformě ESP32 a čipem cc2530. Zhotovená brána tvoří most mezi ZigBee a MQTT protokolem, čímž umožňuje propojení Aqara senzorů a aktorů se službou Home Assistant. Díky Home Assistant jsou veškerá data o domácnosti v bezpečí a vytváření automatizované domácnosti jednodušší.

Abstract

This paper deals with creating gateway for smart devices of brand Aqara. There is discussed ecosystem of Aqara devices, which are used for testing and reverse-engineering of communication. The Gateway is implemented on platform ESP32, which is using cc2530 system-on-chip. Newly made gateway makes a bridge between ZigBee and MQTT protocol, which provides connectivity betweeen sensors and actors from Aqara and Home Assistant. Thanks to the Home Assistant are all data about home safe and creating new home automatizations is a lot easier.

Klíčová slova

Aqara, XiaoMi GateWay, ZigBee, ESP32, CC2530

Keywords

Aqara, XiaoMi GateWay, ZigBee, ESP32, CC2530

Citace

URBÁNEK, Petr. *Vytvoření brány pro chytrá zařízení Xiaomi Aqara.* Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Pluskal

Vytvoření brány pro chytrá zařízení Xiaomi Aqara

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jana Pluskala. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Petr Urbánek
11. května 2021

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce, Ing. Janu Pluskalovi, za jeho cenné připomínky a postřehy a vedení až ke zdárnému konci projektu. Dále bych rád poděkoval všem, kteří se mnou v době psaní této práce měli trpělivost a podporovali mě.

Obsah

Kapitola 1

Úvod

V dnešní době se automatizace domácností těší stále větší oblibě. Sepnutí světel při vstupu do místnosti, automatické vypnutí vody při vytopení pračky a nebo jen zapnutí topení při poklesu teploty pod určitou mez. Existuje velké množství společností zabývající se touto problematikou jako jsou Google, Apple, Xiaomi, a ještě více chytrých zařízení, které se nám snaží ulehčit každodenní činnosti.

Obecně potřebujeme senzory na snímání dané veličiny, zařízení pro ovlivnění dané veličiny a bránu, která zajistí konektivitu. Uživatelé vybírají zařízení do chytrých domácností na základě několika faktorů. Zařízení nesmí být přemrštěně drahé a musí dobře vypadat. Instalace domácnosti nesmí být příliš složitá a časově náročná. V Neposlední řadě každý uživatel myslí i na bezpečnost. Osobní údaje a jejich ochrana je dnes velmi řešené téma. Bohužel, ne vždy má uživatel kontrolu nad tím, jaká se data odesílají a kam. Na základě těchto podmínek můžeme rozdělit uživatele na dvě skupiny, skupinu nakupující zařízení od čínských společností a skupinu, která chce záruku bezpečnosti svých osobních údajů.

Existují případy, kdy jsou data odesílány na čínské servery, které v dnešní době nemusí dodržovat nic jako GDPR, a proto je u nich únik dat pravděpodobnější. To se dá například řešit výběrem společnosti, která data posílá na evropské servery, nebo integrací do open-source frameworků. To však většinou brány od výrobce neumožňuje, a proto začaly vznikat projekty na vytvoření vlastní brány a propojení s open-source frameworky [16].

Tato práce se zaměřuje na vytvoření brány na ESP32 a cc2530 platformách pro Aqara zařízení komunikující pomocí standardu ZigBee propojené se službou Home Assistant [3]. Komerční řešení často vyžadují odesílání osobních údajů, což může vést k jejich zneužití. Také jsou v porovnání s platformami, jako např. Arduino nebo ESP, drahé.

Tato práce je rozdělena do šesti částí. První část je teoretická, je v ní popsána chytrá domácnost a její části. Dále zde jsou popsány protokoly, které chytrá domácnost využívá a spolu s riziky, které s sebou přináší. Druhá část je zaměřena na chytrá zařízení rodiny Xiaomi Aqara, popisuje jejich funkcionalita a technické parametry. Ve třetí části se rozebírá komunikace jednotlivých zařízení a způsoby odposlechu ZigBee sítí. Další kapitolou je návrh nové brány, kde je detailněji zmíněno, jakým způsobem bude nová brána vytvořena, z jakých HW částí bude složena a jak bude komunikovat v chytré domácnosti. V kapitole 6 je popsána samotná implementace brány. Jsou zde zmíněny důležité SW části a implementační nástroje. Ke konci práce se nachází kapitola testování popisující, jakým způsobem a jakými nástroji byla aplikace testována, v jakém testovacím prostředí probíhalo a jak ho duplikovat, a na konec výsledky daného testování.

Kapitola 2

Internet věcí

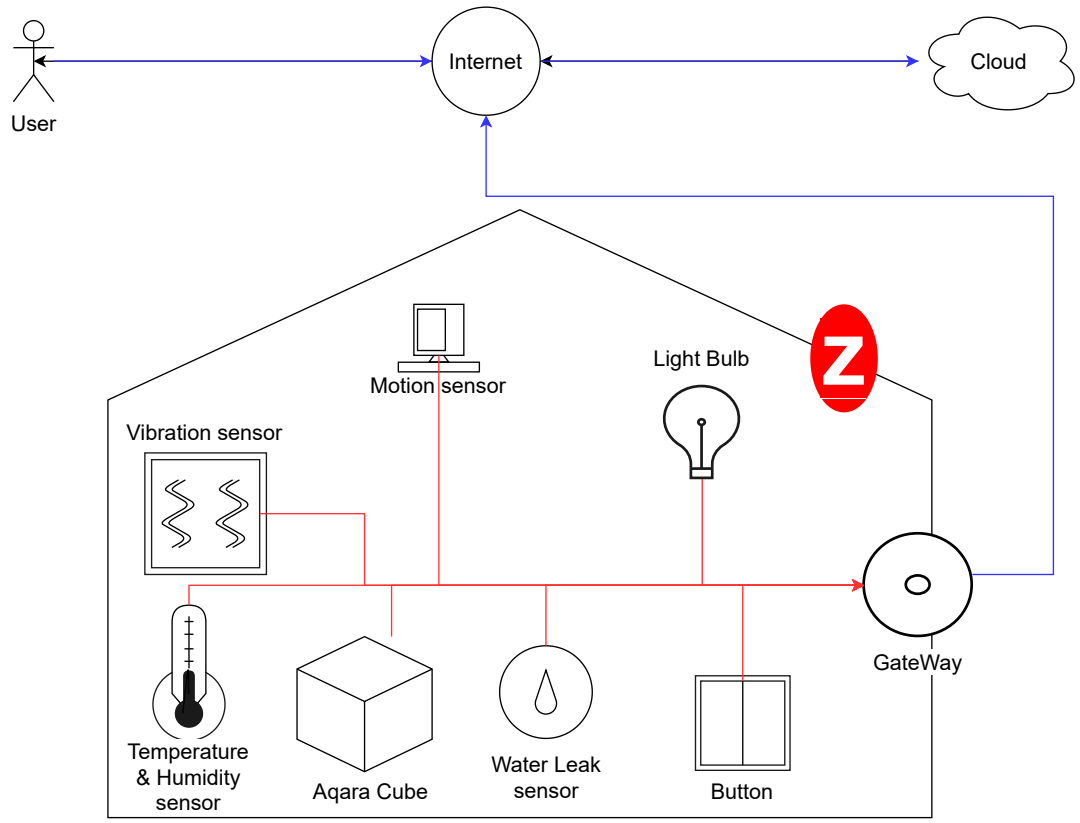
Internet of Things(IoT) neboli internet věcí představuje propojení internetu a věcí každodenní potřeby. IoT rozšiřuje všudypřítomnost internetu skrze spojení věcí s vestavěnými systémy pro snazší ovládání. To vede k velké rozvinuté síti zařízení komunikujících s člověkem. Hlavní cíl IoT je zautomatizovat rutinní činnosti a zvýšit kvalitu života [29].

Internet věcí je dnes vedoucí trend s množstvím výhod, ale také nebezpečí. S rostoucím rozvojem technologií se stává stále více oblíbeným, má více funkcionalit a jeho integrace je snazší. Paralelně s vývojem v oblasti IoT ale vznikají významné výzvy v ochraně soukromí a osobních dat [16].

2.1 Chytrá domácnost

Chytrá domácnost nebo chytrá budova je obvykle objekt se speciálně propojenými zařízeními, dovolující obyvatelům vzdáleně ovládat nebo automatizovat elektrické zařízení v objektu. Chytrá domácnost přináší benefity v podobě menší energetické náročnosti a snazšího ovládání domácnosti. Zároveň je možné domácnost ovládat odkudkoliv s přístupem k internetu [23].

Existují čtyři základní části chytré domácnosti. Vzdálení uživatelé, kteří přistupují k systému pomocí internetu. Příkazy, které jsou odesílany uživately do brány automatizované domácnosti pomocí wifi sítě. Brána 2.1.1 propojuje všechny IoT zařízení a poskytuje vzdálený přístup k domácnosti. Virtuální domácnost 2.1.2, která zodpovídá a kontroluje příkazy z pohledu zabezpečení. Po zkontrolování je příkaz odeslán skrze ZigBee koordinátor 2.1.3 určenému koncovému zařízení. To může být například žárovka nebo zámek dveří [15].



Obrázek 2.1: Diagram chytré domácnosti

2.1.1 Brána

Brána neboli gateway, je bod přístupu mezi lokální sítí a internetem. V našem případě budeme mluvit o bráně v kontextu automatizace domácnosti, tzv. Home Gateway. Ta zaručuje chod celého automatizovaného systému. Všechny zprávy odeslané senzory jsou zde přijímány a odesílají se zde zprávy kontrolérům. Jeden z nejdůležitějších účelů je přidávání a odebíraní zařízení ze sítě [15].

V rámci brány může být obsažena virtuální domácnost. V tomto případě nemusí komunikovat s cloudem. Častěji je však uložena na serveru a informace o stavu domácnosti jí jsou předávány [15].

2.1.2 Virtuální domácnost

Virtuální domácnost slouží ke kontrole zabezpečení v reálném čase. Bývá implementována na serveru. Je to virtuální prostředí automatizované domácnosti, kde se kontroluje každý příkaz. Zkontroluje se, zdali jsou odesílatelé autentizovaní, vnitřní stavba odesílaných zpráv není narušena a všechny příkazy jsou proveditelné v rámci vnitřní stability systému, možností koncových zařízení a bezpečnostních omezení. Dále se zde šifrují zprávy pro další zabezpečení komunikace v rámci lokální sítě [15].

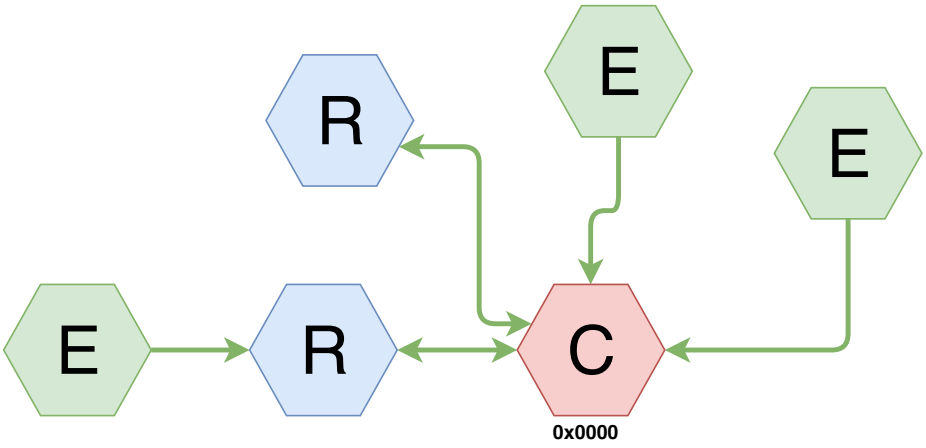
2.1.3 ZigBee koordinátor

Koordinátor zodpovídá za nastartování sítě. Ve fázi inicializace skenuje volné rádiové kanály a využije ten nejvhodnější, což je většinou ten s nejmenší aktivitou z důvodu nejmenší pravděpodobnosti potencionálního rušení. V této fázi je mu přiřazena adresa 0x0000 a přechází do stavu, kdy očekává požadavky od ZigBee zařízení k zahrnutí do sítě. Tyto požadavky se mohou odesílat přímo do koordinátoru, nebo přes router, se kterým má zařízení nejlepší signál. Při přijetí signálu zkontroluje, zdali se může zařízení přidat do sítě [15].

2.2 ZigBee

ZigBee je standard zaměřující se hlavně na systémy, které jsou napájeny baterií, kde jsou hlavními požadavky dlouhá životnost baterie, nízká cena a nízký přenos dat. V ZigBee aplikacích zařízení aktivně komunikují pouze limitovaný čas nutný pro předání dat. Když zrovna žádná data nepředávají, nachází se v power-saving módu, v režimu spánku, díky čemuž baterie v zařízení vydrží podstatně déle [17].

V ZigBee sítích se může nacházet libovolný počet zařízení [17]. Podporuje star, tree i mesh topologie sítě. V oblasti chytrých domácností se používá výhradně mesh topologie 2.2.



Obrázek 2.2: ZigBee Network Model

Obecné řešení sítě ZigBee automatizované domácnosti používá ZigBee koordinátor, routery a koncová zařízení. Každé zařízení v síti má přiřazenou neměnnou 64-bitovou MAC adresu a navíc krátkou 16-bitovou, která se nastavuje při každém přiřazení do sítě (fáze inicializace). Koncové zařízení se mohou přidávat do sítě jen po uživatelem definovaný čas, poté bude žádost zamítnuta. Pokud je zařízení autentizováno a přijato, je přidáno do databáze zařízení v koordinátoru. Všechny zprávy odeslané koncovými zařízeními směřují přes bránu chytrých zařízení do virtuální domácnosti [15].

2.3 Protokol MQTT

MQTT je protokol pro komunikaci uvnitř sítě internetu věcí. Je vyvíjen skupinou OASIS Open. Protokol je navržen jako nenáročný publish/subscribe klient/server přenos

zpráv, ideální pro vzdálené zařízení s jednoduchou implementací a minimální šířkou pásma sítě [21].

Protokol funguje na základě TCP/IP, nebo na základě jiných protokolů poskytujících bezeztrátový přenos v obou směrech. Vzor publish/subscribe umožňuje distribuci a rozdělení zpráv jednotlivým aplikacím. Zprávy jsou nezávislé na vlastním payloadu(obsahu). Rozděluje kvalitu distribuce zpráv na 3 úrovně. `Maximálně jednou`, kdy se zprávy dodávají na základě největšího snažení provozního prostředí. Může nastat ztráta dat. Používá se např. u senzorů okolního prostředí, u kterých nevadí, pokud se jedna ze zpráv ztratí, neboť se brzy odešle nová. U `alespoň jednou` úrovně víme jistě, že zpráva dojde, ale je možné, že se nám budou duplikovat. Možnost `přesně jednou` se ujišťuje, že zpráva přijde přesně jednou. Používá se např. u fakturačních aplikací, kdy duplikace nebo ztráta dat může vést k nesprávné kalkulaci [21].

Má malou režii na přenos a minimální výměnu protokolů, aby provoz po síti byl co možná nejmenší. Dále obsahuje mechanismus k upozornění zaujatých stran, pokud dojde k neplánovanému odpojení [21].

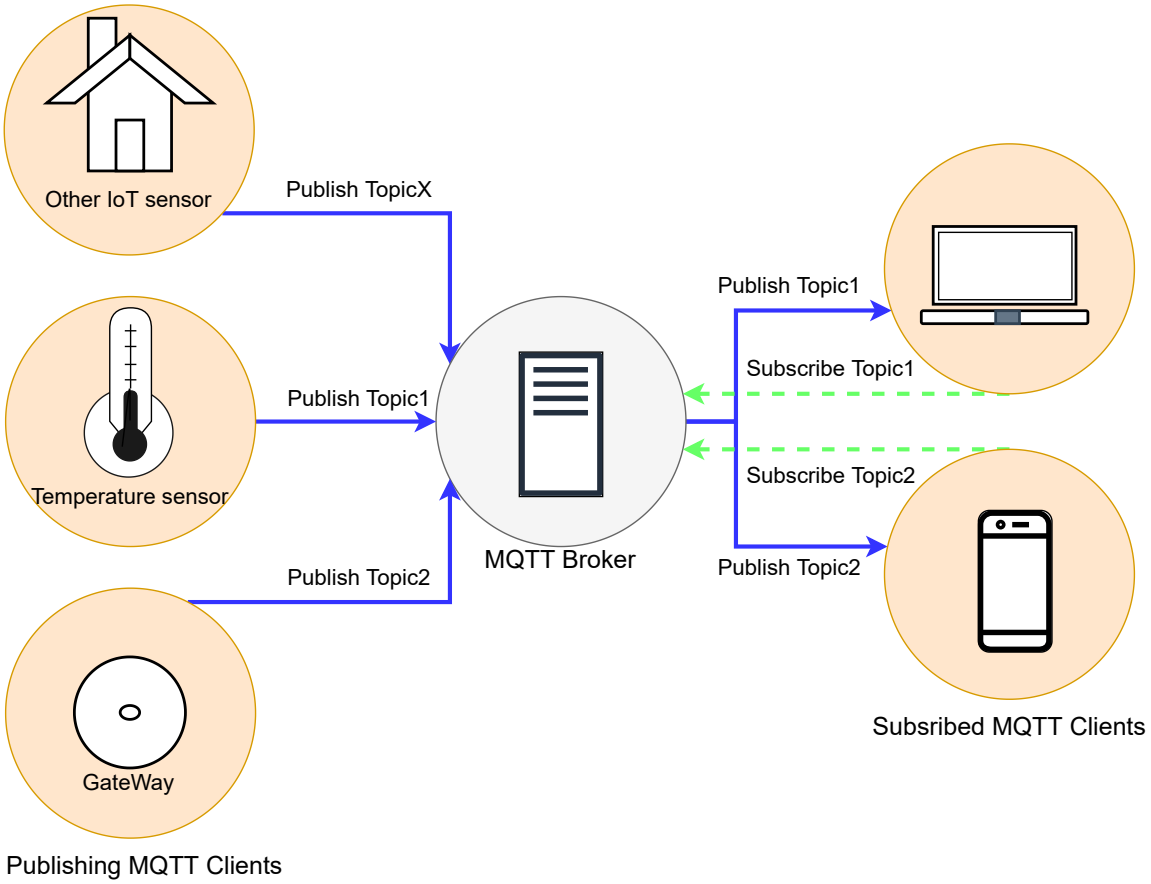
MQTT klient

MQTT klient, dále jen klient, může být v roli publisher i subscriber. Publisher představuje klienta, který právě vysílá zprávy. Subscriber je klient, který zprávy přijímá. Klient je jakékoliv zařízení s implementovanou MQTT knihovnou a připojené k nějakému MQTT brokeru 2.3 přes síť [2].

Implementace MQTT protokolu na klientské straně je přímočará a efektivní. Klientské knihovny jsou dostupné pro široké spektrum programovacích jazyků jako např. Python, C, C++ nebo Java [2].

MQTT broker

MQTT broker, dále jen broker, je protějšek MQTT klienta. Broker je zodpovědný za přijímání, filtrování a distribuci všech zpráv správným klientům. Dále drží data daného spojení všech připojených klientů včetně všech subscription a zmeškaných zpráv. Také zodpovídá za autentizaci a autorizaci klientů [2].



Obrázek 2.3: MQTT klient a MQTT broker

2.3.1 ESPHome

ESPHome je řešení k vytváření vlastního firmaware, dále jen FW, pro ESP8266 a ESP32. Autorem tohoto projektu je Otto Winter, který projekt založil v roce 2018, za účelem programování ESP zařízení s co nejmenší námahou. V ESPHome uživatel sepíše jednoduchý konfigurační soubor YAML, ve kterém je popsáno, které piny zařízení jsou využívány danými komponentami. Jak projekt rostl, stal se nejjednodušším způsobem pro integraci zařízení do Home Assistant. Stal se natolik důležitým, že skupina vlastnící práva k Home Assistant, Nabu Casa, odkoupila práva od zakladatele ESPHome, aby se ujistila, že tento projekt zůstane nadále prosperujícím [7].

ESPHome je open-source projekt, kde každý může přispět k vývoji. Díky tomu má velkou komunitu a mnoho komponent, se kterými je kompatibilní. Pokud není hledaná komponenta podporovaná, je jednoduché si vytvořit novou a přidat ji do ESPHome pro další možné zájemce. Nachází se zde návody a kuchařky jak docílit vlastní komponenty. Obsahuje také nástroje ke snadnému nahrávání, monitorování FW nebo vytváření konfiguračních souborů. Z konfiguračního souboru YAML se vytvoří vlastní FW [5].

2.3.2 Home Assistant

Home Assistant poskytuje platformu pro ovládání a automatizaci chytré domácnosti 2.4. Skládá se ze 3 částí. Operační systém Home Assistant, založený na linuxu, umožňuje funk-

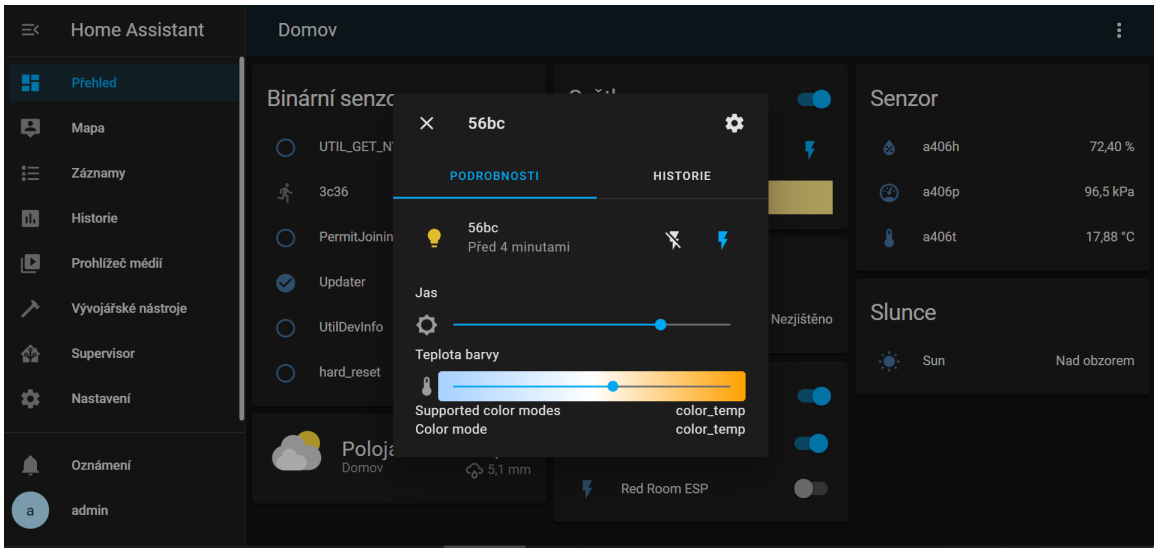
cionalitu zbylých dvou částí. Supervisor zodpovídá za přidávání doplňků, vytváření a obnovování záloh a aktualizování systému. Home Assistant Core zajišťuje správné naslouchání událostem a reakce na ně [3].

Je možné Home Assistant rozšířit integracemi. Každá integrace je zodpovědná za určitou oblast. Ze základu ji můžeme brát jako komponentu, která může naslouchat událostem nebo je spouštět. Nabízí nové služby a udržuje si svůj aktuální stav [3]. Pro komunikaci pomocí MQTT protokolu 2.3 existuje integrace Mosquitto MQTT broker.

V rámci komunikace MQTT protokolem a přidávání zařízení do sítě vznikla metoda zvaná `MQTT Discovery`. Ta umožňuje přidání zařízení do MQTT brokeru s minimální námahou na straně Home Assistant. Konfigurace probíhá způsobem, že při přidání nového zařízení se odešle konfigurační zpráva s tématem následujícího formátu.

```
homeassistant/<typ komponenty>/<unikátní ID>/config
```

Obsah této zprávy se odvíjí od typu komponenty. Určuje se v něm jméno nového zařízení, unikátní identifikátor, témata, kterým naslouchá a kterými se ovládá. Unikátní identifikátor je nutný, aby nevznikalo více stejných zařízení např. při znovupřipojení [4].



Obrázek 2.4: Home Assistant

2.3.3 Z-Stack

Z-Stack je programový balíček od Texas Instruments, který slouží k vývoji produktů založených standardu ZigBee 3.0 a deskách od tohoto výrobce. Pro vytvoření návrhu byla použita verze Z-Stack 3.0.2 zaměřující se na vývoj čipů cc253x [8].

Obsahuje integraci ZigBee Cluster Library [9], příklady samostatných aplikací pro automatizaci domácnosti, zdrojové kódy firmware a samotné předpřipravené FW pro vybrané desky. Dále zde najdeme užitečné nástroje pro testování, flashování, odposlech provozu na síti a dokumentaci k těmto nástrojům a standardům, které využívají. Existuje možnost si pomocí zdrojových kódů vytvořit vlastní FW, k tomu výrobce doporučuje využít IAR nebo CCS kompilátor [8]. FW dodávané v rámci balíčku jsou vytvořeny pomocí IAR kompilátoru

Specifikace Z-Stack ZNP

Z-Stack ZNP, ZigBee Network Processor, je cenově efektivní, nízko energetické řešení, které poskytuje plnou ZigBee funkcionalitu s minimálním vývojovým úsilím. Je obsaženo ve FW čipu, v tomto případě na desce cc2530, a obstarává všechny úlohy ZigBee protokolu. Ponechává ale zdroje aplikace volné k manipulaci mikrokontrolérem. Díky tomu je pro uživatele velice snadné přidat ZigBee novým nebo již existujícím produktům, protože poskytuje širokou volbu ve výběru mikrokontroléru. Z-Stack ZNP je propojeno s mikrokontrolérem pomocí sériového rozhraní SPI, UART, nebo USB [25].

Při komunikaci mezi mikrokontrolérem (ESP32) a Z-Stack ZNP (cc2530) se používají příkazy podle specifikace [26]. Pro správné fungování spojení je nutné využít transportního protokolu, který vytváří rámec zprávy v paketech pro správné přijetí/odeslání a zaručuje integritu celé zprávy. Je nutné, aby po sériové lince byl v jednu chvíli posílán pouze jeden příkaz a buď se počkalo na očekávanou odezvu, nebo časový limit. Fyzický přenos používá 8 datových bitů, nulovou paritu a jeden stop bit pro každý bajt. Pole, která jsou vícebajtová, jsou posílány jako LSB, tedy nejméně podstatný bajt jako první, a neexistuje žádné opatření pro opětovný přenos ztracených paketů [26].

Základní rámec každého příkazu se skládá minimálně z pěti bajtů, začátek rámce, velikost dat, dvou-bajtový identifikátor příkazu, pole bajtů dat a nakonec kontrolní sekvence rámce počítaná jako XOR všech bajtů od LEN po konec DAT [26]. Příklad rámce pro příkaz SYS_PING vypadá podle následující tabulky 2.1.

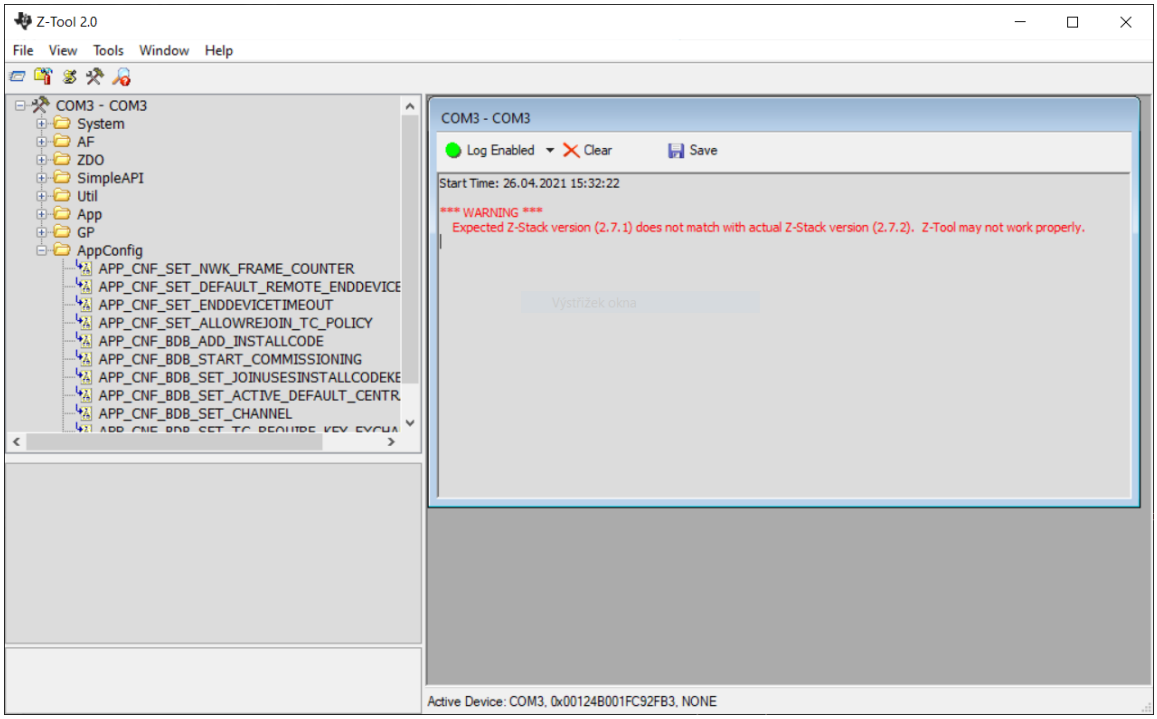
SOF 1	LEN 1	CMD0 1	CMD1 1	DATA 0	FCS 1
0xFE	0x00	0x21	0x01	N/A	0x20

Tabulka 2.1: Příklad příkazu SYS_PING

Z-Tool

Z-Tool je aplikace, kterou je možné využít pro komunikaci mezi TI ZigBee zařízeními. Je určena pouze pro Windows platformu a ke komunikaci s odposlouchávacím zařízení využívá standardní RS-232 sériový port [24].

Program slouží k debugování, testování a vývoji aplikace. Používá příkazy podle specifikace Z-Stack ZNP interface specification 2.3.3, které slouží k bližší specifikaci FW desky [24]. Pomocí nich se dá nastavit např. síťový klíč, vysílací kanál, povolení připojování nových zařízení a další možnosti sítě.



Obrázek 2.5: Z-Tool

2.4 Úniky dat

Žádná komunikace však není dokonalá a být stále online přináší i hrozby. Nahrávání a odesílání většího množství osobních údajů, než by člověk chtěl, do cloudů, o kterých vlastně nic neví, je proto na denním pořádku. Takový skrytý špion může být například zámek dveří, nebo i bezpečnostní kamera, odesílající, kdy přesně člověk pobývá doma [22].

Jedna taková kamera od společnosti Xiaomi zajistila zamezení integrace zařízením od této společnosti s Google Nest Hub. To se stalo poté, co uživatel z Nizozemska objevil fotky na Google Home Hub z neznámých oblastí údajně nahraných jeho bezpečnostní kamerou. To způsobilo, že společnost Google toto zařízení zakázala ve svém ekosystému na dobu, dokud se tato chyba stoprocentně nevyřeší [14].

Kapitola 3

Ekosystém Xiaomi Aqara

Xiaomi je čínská společnost založena v dubnu 2010, která se zabývá vývinem a výrobou spotřební elektroniky, jako jsou telefony, notebooky nebo Internet of Things zařízení, a softwaru k ní [30].

Aqara je označení nové generace bezdrátových IoT zařízení zaštítěné právě touto společností. Generace Aqara, se svým moderním vzhledem a širokou kompatibilitou, byla vytvořena především pro automatizaci domácností. Zaměřuje se na širokou škálu uživatelů, od studentů po korporáty. Pro větší bezpečnost jsou všechny schránky zařízení vytvořeny z anti-UV a ohni-odolných materiálů. Standardní implementace chytré domácnosti počítá s napojením na aplikaci Mi Home, ale zařízení jsou kompatibilní i s ostatními frameworky jako jsou Google Home, Home Kit nebo Home Assistant [18].

Hlavní výhodou této generace je dlouhá výdrž baterie, která zajištěna používáním ZigBee standardu. Ten umožňuje tzv. režim spánku zařízení, tzn. že zařízení neposílá data konstantně, ale pouze když si brána zažádá. Samotné zařízení můžeme použít ke sledování určité veličiny jako např. teploty nebo snímání pohybu, a dále reagovat na její změnu pomocí dalších zařízení.

Zařízení se rozdělují podle následující tabulky 3.1. Jejich rozdělení podle typů vzniklo na základě stránek produktů Xiaomi Aqara [18].

Hub	brána, camera hub
senzory	pohybové čidla, snímače teploty, otřesů
ovladače	tlačítka, aqara cube, chytrá zásuvka
světla	Aqara LED Light

Tabulka 3.1: Typy zařízení Aqara

3.1 Hub

Hub tvoří bránu automatizované domácnosti. Dále se v textu referuje jako GW. Její hlavní úloha je koordinace ostatních připojených zařízení. Přijímá zprávy od senzorů a odesílá zprávy kontrolérům. Tyto informace dále přeposílá na server virtuální domácnosti [15].

3.1.1 Aqara Hub EU version

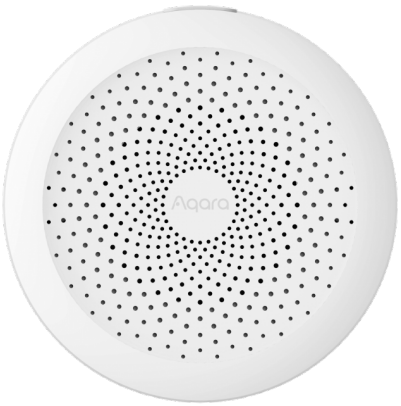
Aqara hub obsahuje noční světlo s nastavitelnou barvou (RGB) a jasem. Dále je v ní zakomponován reproduktor, sloužící například jako budík nebo alarm. Je možné ji propojit s hlasovým asistentem Siri od Apple. Maximální počet připojený chytrých zařízení je 32 [18].

Hub potřebuje konstantní napájení z elektrické sítě, a to 100 - 240 V AC o frekvenci 50/60 Hz. Funkcionalita je zaručena pouze v teplotách od $-10°C$ do $+40°C$ a nekondenzující vlhkosti v rozmezí 0 – 95 % RH. Skládá se z těchto hlavních částí: reproduktor, 18 světel led, Wi-Fi a ZigBee modul a vlastní CPU [18].

3.1.2 Aqara Hub M1S

Tento hub je vylepšená verze klasické Aqara hub. Nyní je možné hub propojit s hlasovými asistenty Alexa, Google Assistant, a dalšími. Maximálně je možné připojit až 128 zařízení [18].

Hub potřebuje konstantní napájení z elektrické sítě, a to 100 - 240 V o frekvenci 50/60 Hz. Funkcionalita je zaručena v teplotách od $-10°C$ do $+40°C$ a nekondenzující vlhkosti v rozmezí 0 – 95 % RH. Zabudovaná RAM má velikost 64/128 MB[18].



Obrázek 3.1: Aqara Hub M1S [11]

3.1.3 Aqara Hub M2

M2 hub má podobné vlastnosti jako M1S, ztratil však použití jako noční světlo, protože komponenta RGB světla byla odebrána. K hubu je možné připojit hlasový asistent z nabídky. Maximální počet připojených zařízení je 128. Narozdíl od předchozích generací je možné hub zapojit do internetové sítě i pomocí ethernet kabelu. Dále má v sobě zabudovaný 360° infra-červený ovladač pomocí něhož se dají ovládat i IR zařízení. Je možné připojit zařízení také pomocí BLE technologie [18].

Hub M2 se napájí pomocí Micro-USB portu. Potřebuje pouze zdroj o napětí 5 V a proudu 1 A, nebo 2 A, jako je např. mobilní nabíječka nebo powerbanka. Funkcionalita je zaručena v teplotách od −5°C do +50°C a nekondenzující vlhkosti v rozmezí 0 – 95 % RH.[18].

3.1.4 Aqara Camera Hub G2H

Camera hub je full HD 140° kamera, která má zároveň funkcionalitu hubu chytré domácnosti. Spojuje tak vyšší bezpečnost a automatizaci domácnosti. Rozpoznává obličeje pomocí umělé inteligence, má noční infra-červené vidění a na základě obrazu rozlišuje zóny aktivity. Video ukládá na SD kartu a po dobu deseti dní i do iCloud uložiště. Hub podporuje až 64 připojených zařízení, zároveň umožňuje oboustrannou zvukovou komunikaci. Funkcionalita je zaručena pouze s Apple HomeKit a aplikací Aqara Home [18].

Kameru je možné připevnit ke stolu, stěně i stropu. Musí být napájena stálým napětím 5 V a proudem 1 A. Funkcionalita je zaručena v teplotách od −10°C do +50°C. Maximální velikost SD karty do lokálního uložiště je 32 GB [18].

Obrázek 3.2: Aqara Camera Hub G2H [18]

3.2 Senzory

Senzory jsou zařízení, které detekují externí informace a překládají je ostatním zařízením do rozpoznatelné formy. Senzory detekují a měří fyzické údaje jako jsou tlak nebo teplota. Tyto informace odesílají určitým protokolem na základě typu komunikace. Mohou komunikovat bezdrátově pomocí standardů bluetooth, ZigBee a dalších, nebo po sériové lince, např. USART. Technické údaje o daných senzorech jsou shrnuty v tabulce 3.2.

3.2.1 Teplotní a Vlhkostní senzor

Měří teplotu, tlak a vlhkost v místnosti. Při změně teploty o určitou hranici, odešle nová data bráně. Teplotu měří s přesností $\pm 0,3\ °C$, tlak $\pm 0,12\ kPa$ a vlhkost s maximální odchylkou 3 %. Senzor se přidělává pomocí samolepky na libovolné místo. Pokud nastane abnormální změna teploty, tlaku nebo vlhkosti odešle varovnou zprávu [18].

3.2.2 Pohybový senzor

Pomocí senzoru snímajícího infra-červené světlo prohledává místnost v maximálním detekčním úhlu 170° na vzdálenost až 7 metrů. Pokud narazí na pohyb, odešle zprávu do brány. Doporučená výška instalace zařízení je mezi 1,2 – 2,1 m. V případě, že zařízení nebude připevněno v daném rozmezí, dojde ke snížení efektivní oblasti detekce. Zařízení je jednoduše připevnitelné pomocí samolepky a stojánek je rotovatelný o 360° pro snadné nastavení detekčního úhlu [18].

3.2.3 Dveřní a Okenní senzor

Tento senzor se skládá ze dvou částí. Na principu magnetického pole zjišťuje, zdali jsou součásti u sebe, nebo se oddálily. Při jakékoli změně se odešle notifikace bráně. Připevňují se pomocí samolepek a při instalaci by od sebe neměly být dále než 22 mm a měly by být zarovnány podle linky na bocích [18].

3.2.4 Senzor úniku vody

Tento senzor splňuje normu IP67, je tedy vodě a prachu odolný. Při zaplavení senzoru vodou až nad 0,5 mm se odešle zpráva do brány, která na to může zareagovat, například zavřením uzávěru vody. Při instalaci na kovové povrchy může dojít ke snížení dosahu signálu [18].

3.2.5 Vibrační senzor

Na principu akcelerátoru, pomocí tíhové síly monitoruje pohyb senzoru ve třech osách. Při otřesu, náklonu, nebo jakémkoliv jiném pohybu odešle notifikaci bráně. Instaluje se na jakýkoli povrch pomocí samolepek, ale na kovovém povrchu se může snížit dosah signálu [18].

3.3 Ovladače

Ovladače jsou zařízení, které vysílají data na základě určité akce. Tato akce může být např. stisknutí tlačítka, otočení o 90° nebo zatřesení. Technické údaje o vybraných ovladačích jsou shrnuty v tabulce 3.3.

Typ senzoru	Baterie	Funkční teplota	Funkční vlhkost	Životnost
Teplotní a vlhkostní	CR2032	−20°C - +50°C	0 – 95 % RH	2 roky
Pohybový	CR2450	−10°C - +40°C	0 – 95 % RH	2 roky
dveřní a okenní	CR1632	−10°C - +45°C	0 – 95 % RH	2 roky
úniku vody	CR2032	−10°C - +55°C	0 – 100 % RH	2 roky
Vibrační	CR2032	−10°C - +50°C	0 – 95 % RH	2 roky

Tabulka 3.2: Technické parametry senzorů

Mají mnoho možných variací použití, jako vypnutí, zapnutí světla, zamykání dveří nebo sepnutí alarmu při nebezpečí. Stejně jako u senzorů mohou využívat různých protokolů pro komunikaci s bránou.

3.3.1 Aqara kostka

Aqara Cube pracuje na principu akcelerátoru a gyroskopu, obsahuje 6 jednoduchých programovatelných gest: tlačení, zatřesení, rotaci, poklepání, převrácení o 90°, převrácení o 180°. Gesta se programují v aplikaci Mi Home. Na základě těchto akcí odesílá zprávy do brány. Brána se potom rozhodne na základě gesta, co vykoná za akci [18].

3.3.2 Chytrá zásuvka

Smart Plug umožňuje dálkové zapínání a vypínání přísunu elektřiny. To může nastat na základě kliknutí v aplikaci, nebo časovače, díky čemuž je možné plánovat zapnutí nebo vypnutí spotřebičů na základě času a data. Pokud zásuvka zjistí, že je teplota nebo úroveň elektrického proudu nad limit, automaticky vypne přísun elektřiny pro ochranu před případným nebezpečím [18].

3.3.3 Bezdrátový dálkový spínač

Wireless Remote Switch obsahuje 3, nebo 4 gesta, záleží na počtu tlačítek ovladače, buď jedno-tlačítkový, nebo dvou-tlačítkový. Klik, dvojklik, dlouhé podržení, (krátké stisknutí obou tlačítek) odešle notifikaci do brány. Připevňuje se pomocí samolepek na jakýkoli povrch, ale je určeno pouze pro použití ve vnitřních prostorách. Na tlačítku je garantováno 50 000 kliknutí [18].

Typ ovladače	Baterie	Funkční teplota	Funkční vlhkost	Životnost
Aqara kostka	CR2450	−10°C - +50°C	0 – 95 % RH	2 roky
Chytrá zásuvka	ze sítě	0°C - +35°C	0 – 950 % RH	neuvedeno
bezdrátový dálkový spínač	CR2032	−10°C - +50°C	0 – 95 % RH	2 roky

Tabulka 3.3: Technické parametry ovladačů

3.4 Světla

Bezdrátové ovládání osvětlení je jedno z nejpraktičtějších použití IoT. Připojením chytrých světel do virtuální domácnosti se mohou vzdáleně ovládat a nastavovat. Jeden z plusů používání chytrých světel je energetická efektivnost a redukce výdajů za elektřinu [12].

S použitím dalších zařízení jako jsou pohybové senzory, se domácnost může stát plně automatizovanou. Chytré zářivky už podporují více, než jenom zapnutí a vypnutí. Nyní se dají integrovat světla se změnou jasu, teploty světla, nebo i změnou barvy [12].

3.4.1 Žárovka s nastavitelným bílým světlem

White Tunable Led Light funguje jako jakákoliv jiná žárovka. Ale navíc umí ještě přizpůsobit jas a teplotu barvy pro pohodlnost uživatele. Je na výběr z několika předdefinovaných módů světla, ale je možné si i manuálně jas a tón nastavit. Přijímá zprávy od brány a na základě nich se přizpůsobuje její nastavení nebo sepnutí. Žárovku je možné zapnout/-vypnout, i když nefunguje internet, nebo je vypojená gateway, pouze není nastavitelná. Pro nastavování světla je nutné mít zapojenou gateway [18].

Žárovka má svítivost 806 lm a vydrží až 25 000 hodin provozu. Je nutné ji mít zapojenou v elektrické síti o parametrech 220–240 V a 50/60 Hz. Její spotřeba je 9 W a teplota světla je v rozmezí 2700 – 6500 K [18].

Obrázek 3.3: Aqara LED Light [1]

Kapitola 4

Reverse-engineering komunikace

Zařízení Xiaomi Aqara komunikují s bránou bezdrátově, pomocí nízko-rychlostního protokolu ZigBee. Ten byl vytvořen společností ZigBee Alliance v listopadu roku 2004. Samotný standard ZigBee vychází ze standardu IEEE 802.15.4 [13].

Komunikace mezi zařízeními používá ZigBee Cluster Library (ZCL) [9]. Tato knihovna definuje několik standardních profilů 4.1. Clustery definují hlavičku a payload uvnitř PDU. Dále definuje typy atributů, příkazy 4.2 a vestavěné odpovědi [19].

Zařízení dělíme na 3 druhy: Koordinátor, routery a koncová zařízení. Koordinátor je nejdůležitější komponenta IoT sítě. Koordinuje ostatní zařízení, umožňuje přidávání nových zařízení do sítě, přijímá všechny právy a odesílá je virtuální domácnosti na zpracování. Dále přijímá akce od virtuální domácnosti a zpětně posílá příkazy koncovým zařízením, s tím co mají udělat. Routery jsou nejčastěji zároveň koncová zařízení, která navíc rozšiřují působnost sítě tím, že se stanou mezičlánkem mezi koordinátorem a koncovými zařízeními. Díky tomu se mohou zprávy posílat na větší vzdálenost, vzniká tím ale nový problém s duplikací zpráv, kdy jedna zpráva přijde od koncového zařízení a druhá od routeru. Koncová zařízení odesílají data koordinátoru [20]. Data mohou obsahovat např. informace o stisknutí tlačítka, změně teploty, otevření dveří, nebo úniku vody.

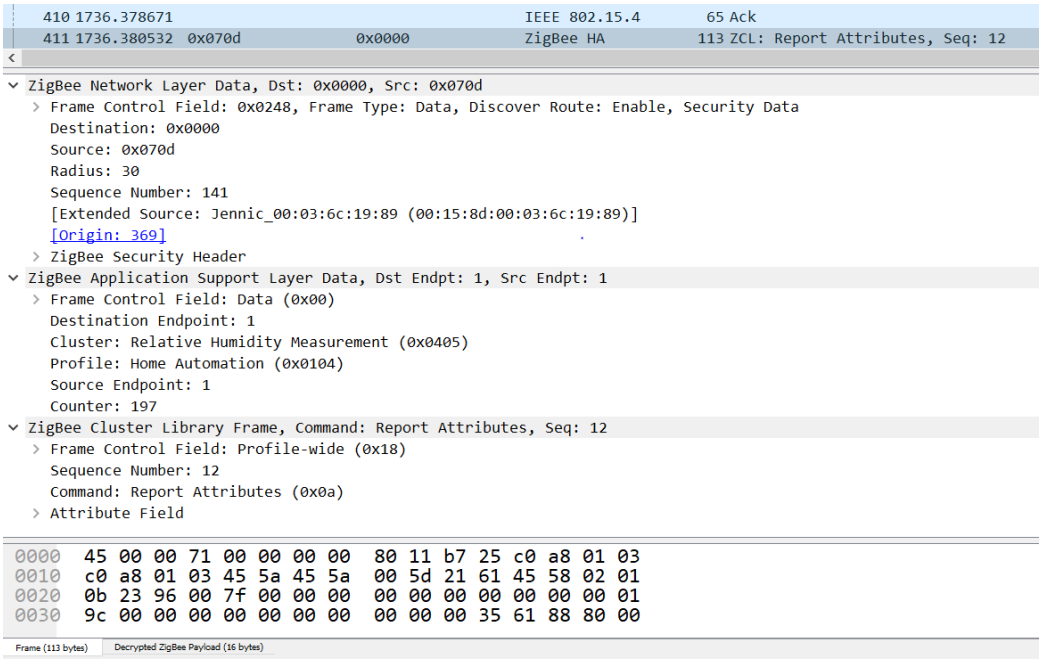
4.1 Odposlech komunikace

Pro zjištění reálně odesílaných dat jednotlivými zařízeními bylo nutné odposlechnout komunikaci v rámci Aqara ekosystému a aplikace Mi Home 3. Z důvodu, že ZigBee komunikace je rozdílná od klasické wifi nebo bluetooth, je nezbytné použít speciální odposlouchávací zařízení.

Nejdůležitější pakety u odposlechu jsou ty, které mají protokol ZigBee, ostatní nejsou podstatné a mohou se ignorovat. Všechny tyto pakety obsahují ZigBee Network a Application Support vrstvu, dále jen APS. V ZigBee Network vrstvě se vyskytuje zdrojová a cílová adresa a maximální počet skoků, tzn. kolikrát může paket projít přes router, aby byl stále validní. V APS se nachází pole Frame Control určující, jestli se jedná o data, nebo příkaz. Dále se zde na základě typu paketu může objevovat ZigBee Security Header a Command Frame, nebo identifikátor clusteru, identifikátor profilu aplikace a zdrojový a cílový koncový bod 4.1.

General Cluster	Cluster ID
Basic	0x0000
Power Configuration	0x0001
Identify	0x0003
Groups	0x0004
Scenes	0x0005
On/Off	0x0006
On/Off Switch Configuration	0x0007
Level Control	0x0008
Alarms	0x0009
Time	0x000A
Binary Input (Basic)	0x000F
Commissioning	0x0015
Door Lock	0x0101
Thermostat	0x0201
Colour Control	0x0300
Illuminance Measurement	0x0400
Illuminance Level Sensing	0x0401
Temperature Measurement	0x0402
Relative Humidity Measurement	0x0405
Occupancy Sensing	0x0406
IAS Zone	0x0500
IAS ACE (Ancillary Control Equipment)	0x0501
IAS WD (Warning Device)	0x0502

Tabulka 4.1: Identifikátory clusterů [19]



Obrázek 4.1: Vrstvy ZigBee paketu rozdělené ve wiresharku

Command	Description
Read Attributes	čtení atributů ze vzdáleného zařízení
Read Attributes Response	vygenerovaná odpověď na příkaz Read Attributes
Write Attributes	změna atributů na vzdáleném zařízení
Write Attributes Response	odeslaná odpověď na příkaz Write Attributes
Configure Reporting	konfigurace automatického odesílání hodnot atributů vzdáleného zařízení
Report Attributes	odeslání hodnot atributů
Discover Attributes	žádost o seznam atributů
Discover Attributes Response	odeslaná odpověď na příkaz Discover Attributes

Tabulka 4.2: Základní příkazy [19]

Při odposlechu komunikace jsou všechny pakety zašifrované. Musíme je rozšifrovat pomocí Transport Key. Ten odešle brána při registraci nového zařízení do sítě v APS vrstvě. Pokud APS obsahuje pole Frame Control indikující příkaz a v poli Command Frame se nachází identifikátor příkazu roven 0x05, tak v poli Key je chtěný transportní klíč. Ten je ale také zašifrovaný a pro každou síť se vytváří jiný. Aqara chytré zařízení používají pro zašifrování transportního klíče globální `Trust Center` klíč. Ten má hodnotu 5A:69:67:42:65:65:41:6C:6C:69:61:6E:63:65:30:39 [6]. Pomocí něj jsme schopni rozšifrovat transport key a pomocí transport key rozšifrujeme všechny ostatní pakety v komunikaci.

Pro automatizaci domácnosti jsou nejzajimavější pakety obsahující vrstvu ZigBee Cluster Library, dále jen ZCL 4.1. V jejich APS vrstvě se nachází identifikátor clusteru, profilu a zdrojový a cílový koncový bod. V rámci ZCL se vyskytuje identifikátor příkazu a pole atributů. Pakety od senzorů a kontrolerů budou nejčastěji obsahovat příkaz Report Attributes(0x0a). Dále zde je pole atributů. Každý atribut má vlastní identifikátor, datový typ a hodnotu 4.3.

Attribute	Data Type	Value
0x0055	0x21	0x0001
Present Value	16-Bit Unsigned Integer	1

Tabulka 4.3: Příklad atributu při stisku tlačítka

V případě, že identifikátor clusteru je roven 0x0013, profilu 0x0000 a hodnota obou koncových bodů 0, jedná se o tzv. Device Announcement. Ten poskytuje informace o možnostech nově připojeného zařízení jako jsou např. možnost nahradit koordinátor a alokace krátké adresy [9].

4.1.1 Způsoby odposlechu

K odposloucháváni ZigBee komunikace potřebujeme anténu, která daný signál bude přijímat, a software, který výstup z antény bude překládat v lépe čitelné formě. Jako anténu pro testování je použita cc2531 USB-stick od Texas Instruments. Pozdějije použita pro návrh brány deska cc2530.

Pro otestování funkčnosti, nebo v případě, že bychom neměli cc2531 USB-stick, je možné použít desku cc2530 i pro odposlech. Potřebujeme však k tomu programátor cc-debugger od Texas Instruments. Ten připojíme k počítači pomocí USB, a desku k němu podle ta-

bulky 4.4. Při správném zapojení by měl programátor svítit zeleně, v opačném případě je nutné stisknout tlačítko `Reset`. Pokud ani to nepomůže, nejspíš je chyba v zapojení nebo v nahraném firmware cc2530 [27].

cc-debugger	cc2530
GND	GND
DC	P2.2
DD	P2.1
Csn	P1.4
SCLK	P1.5
RESETn	RESETn
MOSI	P1.6
3v3	VCC
MISO	P1.7

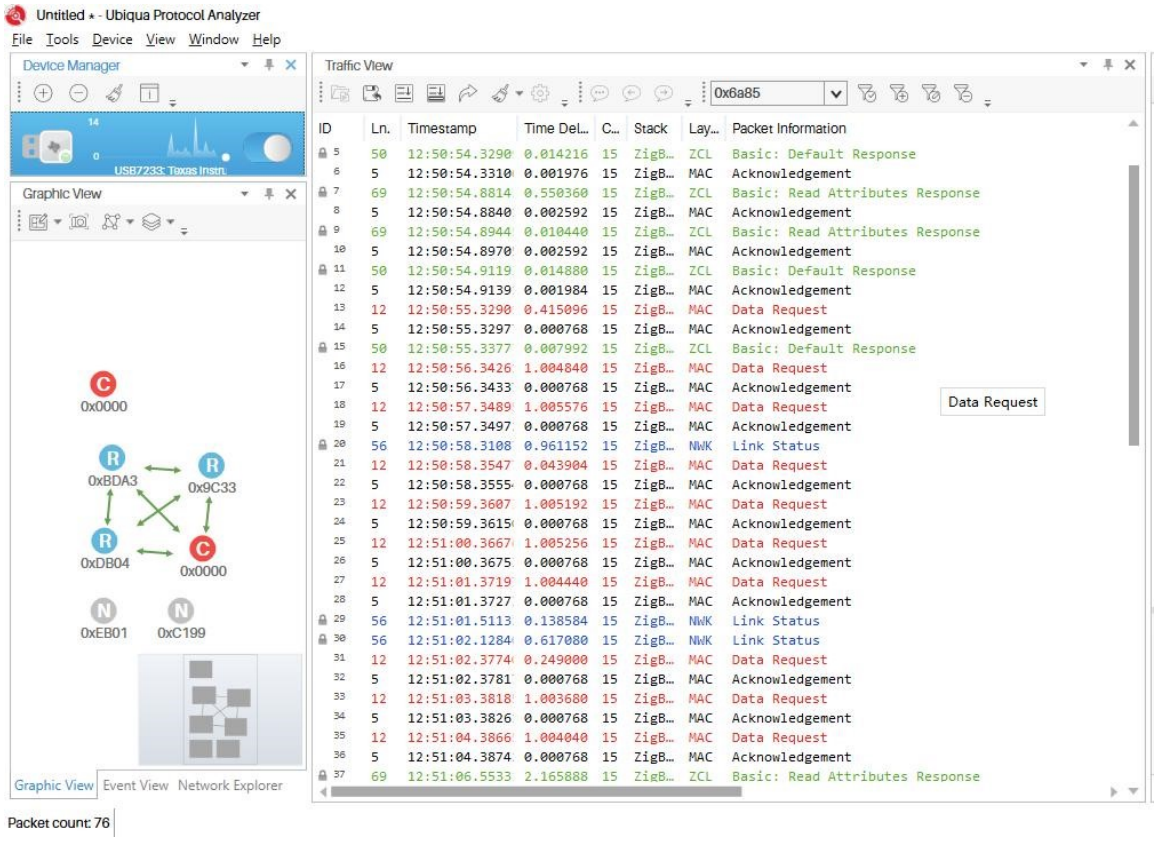
Tabulka 4.4: způsob zapojení návrhu

Existuje více možností, jaký software zvolit k odposlechu komunikace. Nejjednodušší na zprovoznění je využít program doporučovaný přímo výrobcem, Ubiqua Protocol Analyzer 4.1.1. Dalšími možnostmi je použít open-source software jako je například KillerBee, ty jsou ale většinou pouze pro Linux. Nejtěžší možností na zprovoznění je odposlech komunikace ve wiresharku 4.1.1.

Ubiqua Protocol Analyzer

Ubiqua protocol analyzer je nástroj pro pozorování provozu na IoT síti. Je vyvíjen společností Ubilogix. Její hlavní výhodou oproti wiresharku je automatická dešifrace zašifrovaných paketů na síti, jelikož bezpečnostní klíče jsou automaticky získávány za běhu odposlechu. Dále podporuje grafické vyobrazení sítě na základě příchozích paketů a naslouchání na více kanálech použitím více zařízení najednou [10].

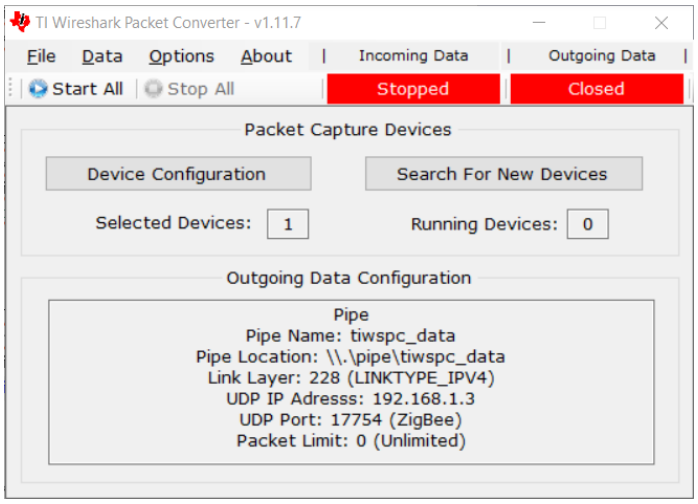
Pro očekávanou funkcionalitu je nutné internetové připojení a podporované odposlouchávací zařízení. Informace o podporovaných zařízeních se nachází v dokumentaci aplikace. Ubiqua protocol analyzer je komerční řešení. Předplatné stojí 65 $ měsíčně, nebo 650 $ ročně. Při vytvoření nového účtu je nabízeno týdenní předplatné zdarma na vyzkoušení [10].



Obrázek 4.2: Ubiqua Protocol Analyzer

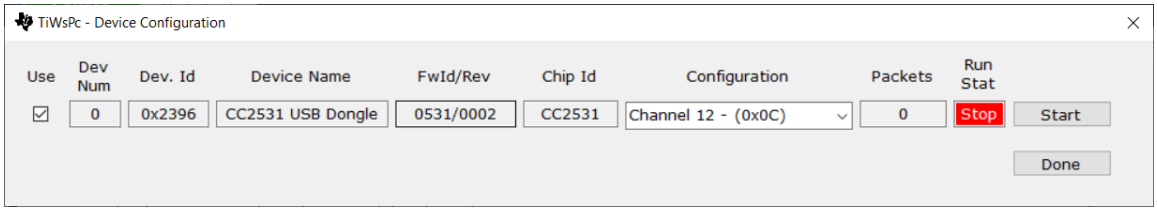
Wireshark

Abychom viděli odchycenou komunikaci ve wiresharku, musíme nejprve stáhnout a nainstalovat program TiWsPc, který konvertuje pakety do formátu využívaným wiresharkem 4.3. V menu `Data` je nutné zaškrtnout možnost `Use Pipe(Vista or higher)`.



Obrázek 4.3: Aplikace Wireshark Packet Converter po spuštění

Otevřením menu `Device Configuration` se zobrazí nové okno s připojenými zařízeními 4.4. V tomto menu zaškrtneme možnost `Use` pro zařízení, které chceme použít a vybereme kanál, kterému chceme naslouchat. Nakonec začneme odposlouchávat pomocí tlačítka `Start` [28].



Obrázek 4.4: Menu Device Configuration se zapojenou USB Stick cc2531

V tuto chvíli odposloucháváme komunikaci, ale odposlechnuté pakety nikde nezobrazujeme. Proto musíme vytvořit pipe ve wiresharku. Nejdříve vytvoříme nového zástupce pro program wireshark a modifikujeme ho tím, že v kolonce `Target` přidáme na konec následující text.

```
-i\\.pipe\tiwspc_data -k
```

V tuto chvíli by měly být otevřeny oba konce pipeline a odchycené pakety by se měly zobrazit ve wiresharku. Tyto pakety jsou ale zašifrovány, pro odšifrování je potřeba ve wiresharku v menu `edit-preferences-protocols-zigbee` přidat `Pre-configured keys`. Jeden z klíčů `Trust Center` klíč a druhý je klíč sítě. Ten se dá zjistit při přidávání nového zařízení v paketu označeném jako `Transport Key`.

4.2 Komunikace jednotlivých zařízení

Komunikace mezi zařízeními probíhá over-the-air pomocí protokolu ZigBee. Samotné zařízení mezi sebou přímo nekomunikují. Zprávy posílají přes prostředníka, GW. Ta je posílá virtuální domácnosti ke schválení. V reakci na zprávy od virtuální domácnosti GW odesílá příkazy koncovým zařízením [15].

4.2.1 Zařízení IoT sítě

Domácí automatizace je navržena flexibilně, dovoluje přidat různá zařízení od různých prodejců. Každé zařízení má vlastní FW, který zpracovává příchozí data, a odesílá zprávy koordinátoru. Dále může každé zařízení obsahovat vlastní bezpečnostní akce, aby nemohlo dojít k nechtěnému výsledku [15].

Různé typy zařízení posílají zprávy s profilem ZigBee Home Automation lišící se clusterem 4.1, protože každý typ určuje jinou veličinu. Zprávy reagující na nějakou změnu budou většinou obsahovat příkaz 0x0A, který značí `Report Attributes`. V payload poté najdeme určitý nenulový počet atributů. Každý atribut má svůj identifikátor, datový typ a vlastní hodnotu. `Report Attributes` atributy mají identifikátor s určitým významem, který se zároveň odvíjí i od daného clusteru.

4.2.2 Hub

Brána odesílá většinou odpovědi o tom, že přečetl poslané atributy, nebo odesílá příkazy daným kontrolérům, aby provedly naprogramované reakce. Mezi nejčastěji odeslané příkazy

řadíme `Report Attributes Repsponse` reagující na command `Report Attributes`. Aqara Hub zároveň komunikuje s Xiaomi Cloudem pomocí Wi-Fi.

Teplotní a Vlhkostní senzor

Senzor při změně teploty, tlaku, nebo vlhkosti odešle 3 zprávy na nejbližší router, který je přepošle bráně. První zpráva má cluster Temperature Measurement a odesílá jeden atribut identifikovaný jako Measured Value (0x0000) s datovým typem Signed 16-bit Integer. Druhá má cluster Relative Humidity Measurement a také odesílá jen jeden Measured Value atribut s datovým Unsigned 16-bit Integer. Třetí cluster je Pressure Measurement. Ten odesílá 3 atributy, první s naměřenou hodnotou a datovým typem Signed 16-bit Integer , druhý určující Scale s datovým typem Signed-8bit Integer a třetí se Scaled value a datovým typem Signed-16bit Integer.

Senzor úniku vody

Senzor při zaplavení nebo odplavení odešle zprávu s clusterem Security and Safety: IAS Zone na nejbližší router, který ji přepošle bráně. V Clusteru najdeme command Zone Status Change Notification (0x00). V payloadu najdeme Zone Status, Extended Status s doplňujícími informacemi o zóně, Zone ID identifikující konkrétní zónu a Delay definující zpoždění mezi změnou a přenosem informace ve čvrt-sekundách.

Pohybový senzor

Senzor při zaznamenání pohybu odešle zprávu nejbližšímu routeru, který ji přepošle bráně. Zpráva má cluster Occupancy Sensing. V payloadu najdeme atribut Occupancy(0x00) s datovým typem 8-bit Bitmap a vlastní hodnotou určující pohyb.

Dveřní a Okenní senzor

Senzor při zaznamenání oddálení svých částí odešle zprávu s General clusterem On/Off. V payloadu najdeme atribut On/Off (0x00) s datovým typem boolean a vlastní hodnotou určující oddálení a přiblížení částí senzorů. Když se senzory oddálí odešle se hodnota ON (0x01) a při přiblížení OFF (0x00).

Vibrační senzor

Senzor při zaznamenání vibrací o síle dle nastavení citlivosti (low, medium, high) odešle zprávu nejbližšímu routeru, který ji poté přepošle bráně. Použije se zde Door Lock cluster obsahující 1 atribut s hodnotou závisející na síle zatřesení.

Aqara kostka

Při jedné z akcí zaklepání, otočení o 90°, otočení o 180°, nebo zahrkání se odešle zpráva bráně. Zpráva má cluster typu Multistate Input a příkaz Report Attributes. V payloadu poté najdeme atribut identifikující Present Value, datovým typem Unsigned 16-bit Integer a vlastní hodnotou určující danou akci. Hodnota 3 představuje poklepání, 93 otočení o 90°, 130 otočení o 180° a zahrkání má hodnotu 0.

Bezdrátový dálkový spínač

Při zaznamenaní nějaké akce odešle zprávu nejbližšímu routeru, nebo bráně s clusterem Multistate Input a příkazem Report Attributes. Atribut odeslaný v příkaze má Id 0x0055 určující Present Value a datový typ Unsigned-16bit Integer a různou hodnotou. Hodnota 1 značí single-click, hodnota 2 double-click a hodnota 0 long-press

Žárovka s nastavitelným bílým světlem

Při vypnutí a zapnutí zářivky se odesílá zpráva z brány s clusterem On/Off nejbližší routeru, který ji přepošle dané zářivce. Ta ji příjme a odpoví na ní clusterem On/Off a příkazem Report Attributes atributy On/Off datovým typem boolean a hodnoutou sepnutí. Dále zářivce může být zaslán příkaz o změně jasu světla s clusterem Level Control a příkazem Move To Level. Ten udává v payloadu na jaký level jasu by se zářivka měla nastavit. Jako poslední druh zpráv je možné z brány poslat zprávu s clusterem Color Control a příkazem Move to Color Temeperature. V payloadu najdeme Color Temperature s datovým typem 16-bit Unsigned Integer a Transition time s datovým typem 8-bit Unsigned Integer a hodnotou určující čas ke změně teploty barvy v jednotkách x * 1/10 sekundy. Na obě zprávy odešle zářivka odpověď příkazem Report Attributes potvrzující konkrétní hodnotu úrovně jasu, nebo teploty světla.

Chytrá zásuvka

Při zapnutí nebo vypnutí chytré zásuvky se odešle zpráva bráně, nebo nejbližšímu routeru, který ji bráně doručí. Zásuvka odesílá cluster typu Multistate Input s příkazem Report Attributes. V atributech odešle typ atributu Present Value, datový typ Unsigned 16-bit Integer a hodnotu určující zapnutí/vypnutí.

Kapitola 5

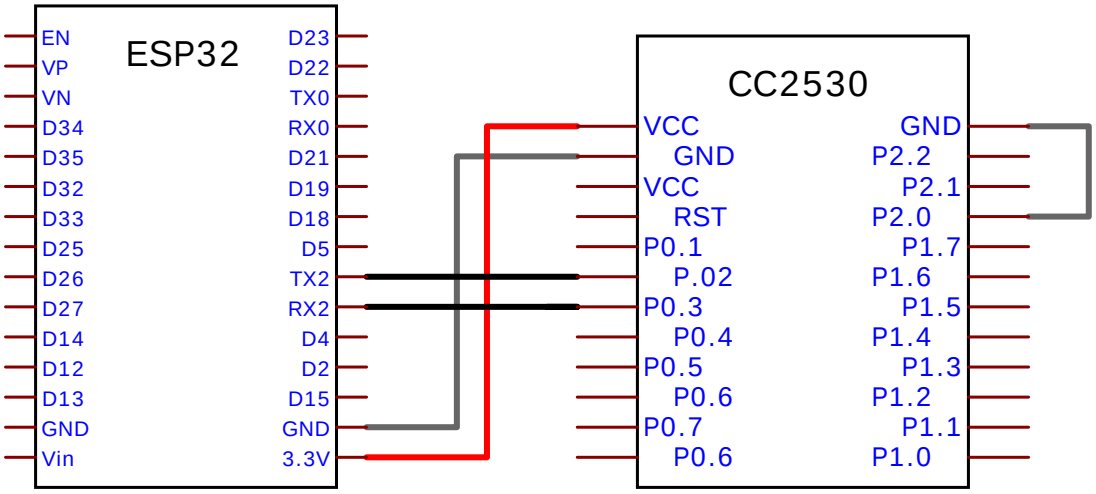
Návrh brány

Návrh vlastní brány chytrých zařízení se pokouší nahradit bránu, která se dosavadně používá v rámci ekosystému Xiaomi Aqara 3.1.1, použitím HW platforem ESP32 a cc2530. Návrh je zaměřen na integraci zařízení ekosystému Xiaomi Aqara 3 a zařízení od ostatních výrobců v této práci nejsou řešeny. Výsledkem by měla být integrovatelná brána do systému Home Assistant 2.3.2.

Brána od Xiaomi Aqara komunikuje pomocí ZigBee protokolu. Jsou ji posílány zprávy od senzorů a přijímá i příkazy od uživatelů. Sama odesílá informace o stavu sítě do cloudu a příkazové zprávy kontrolérům, o tom, co mají dělat. Z důvodu konstantního čekání na přijímání zpráv je vhodné, aby byla stále zapojena v elektrické síti, protože tato akce je energeticky náročná a baterie by se musely často měnit.

Pro nahrání FW byl použit nástroj přímo od výrobce desky, Texas Instruments, dále jen TI, SmartRF Flash Programmer. FW pro cc2530 byl použit defaultní, dodávaný v rámci Z-Stack 3.0.2 [8]. Ten se nahrál pomocí CC-Debuggeru 5.2, nástroje od TI určenému k nahrávání FW do desky, a zmíněného Flash Programmeru. Dále pomocí Z-Stack API se posílají sériové příkazy. Pro testování příkazů se použil další z nástrojů, Z-Tool 2.5. Podle FW na desce a verzi Z-Tool se zobrazuje jednoduché API sériových příkazů a umožňuje je odeslat na danou desku.

Reálný produkt se skládá z desky ESP32-devKitV1 připojené k cc2530(anténa) dle následující tabulky 5.1. Pin CFG1 musí být připojen ke GND, aby bylo možné s defaultním FW používat USART rozhraní [25]. Produkt je napájen klasickou mobilní nabíječkou přes micro USB konektor v ESP32. Pomocí frameworku ESPHome byla vytvořena vlastní komponentu, která obstarává sériovou komunikaci s deskou cc2530 a zároveň komunikuje také s MQTT brokerem v Home Assistant. Bezpečnost komunikace je zaručena pomocí FW na cc2530, který pakety šifruje. Bráně je možné v rámci ESPHome přiřadit další funkcionalitu přidáním nových komponent.



Obrázek 5.1: Návrh gateway

5.1 Komunikace

Prototyp se musí umět dorozumívat s jendotlivými zařízeními i s virtuální domácností v podobě Home Assistant. Pro komunikaci mezi těmito uzly se používají různé protokoly a technologie. Z tohoto důvodu se deska cc2530 stará o komunikaci mezi koncovými zařízeními a GW a ESP32 obstarává komunikaci s virtuální domácností. Zároveň musí mikrokontrolér ESP32 udržovat spojení s cc2530, aby skrze ní pomocí Z-Stack API 2.3.3 mohl komunikovat.

5.1.1 Komunikace mezi ESP32 a cc2530

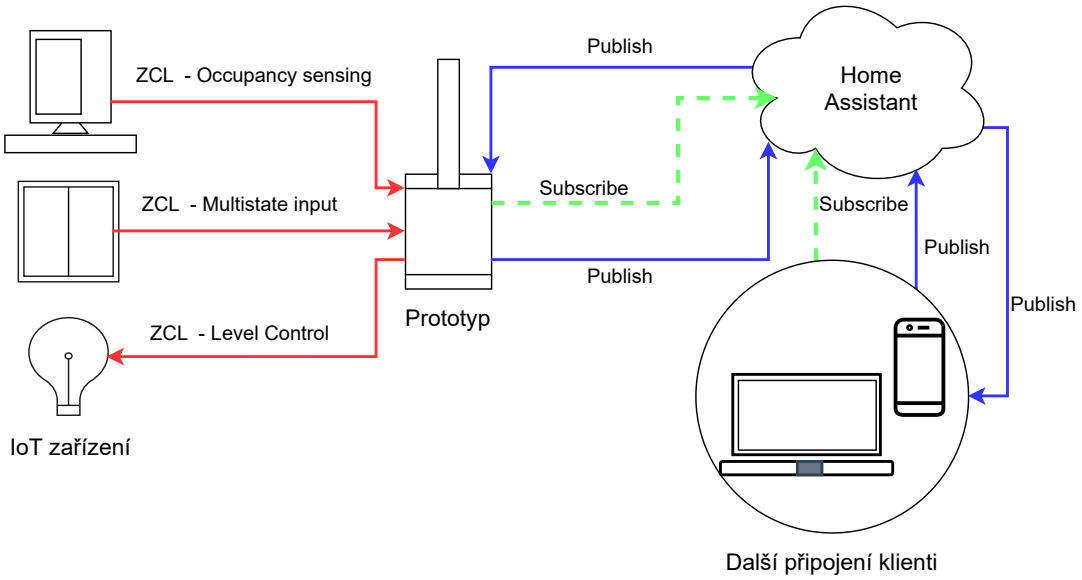
ESP32(mikrokontrolér) a cc2530(anténa) jsou propojeny pomocí sériového rozhrání UART, Rx a Tx pinů. Při připojení do elektrické sítě je nutné odeslat sekvenci příkazů, které připraví desku cc2530 k provozu. Dále je potřeba stále naslouchat na UART rozhraní, pokud nepřišel nějaký příkaz.

5.1.2 Komunikace mezi GW a koncovými zařízeními

GW komunikuje s koncovými zařízeními pomocí ZigBee protokolu. Pro posílání správných paketů využívá specifikaci Z-Stack ZNP 2.3.3. Mikrokontrolér posílá po sériové lince desce cc2530 příkazy a na jejich základě vytvoří FW na cc2530 ZigBee pakety a odešle je koncovému zařízení, nebo broadcastem.

5.1.3 Komunikace mezi GW a Home Assistantem

Spojení mezi GW a Home Assistantem probíhá přes protokol MQTT. Ve FW ESP32, vytvořeným v ESPHome, je definovaný MQTT Broker. Ten je napojený na Home Assistant integraci, Mosquitto MQTT Broker. Při přidání nového zařízení se vytvoří nový subscribe topic s komponentou a unikátním ID, které odpovídá přiřazené 16 bitové adrese nově připojeného zařízení. Poté při každé akci zařízení se odešle publish zpráva s tímto tématem.



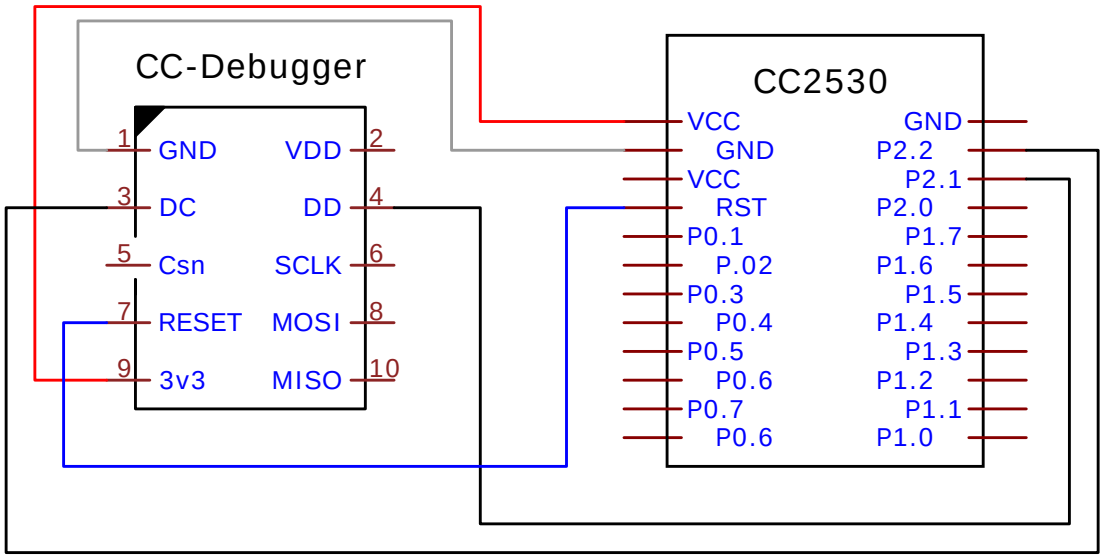
Obrázek 5.2: Diagram komunikace brány

5.2 Nahrání firmware

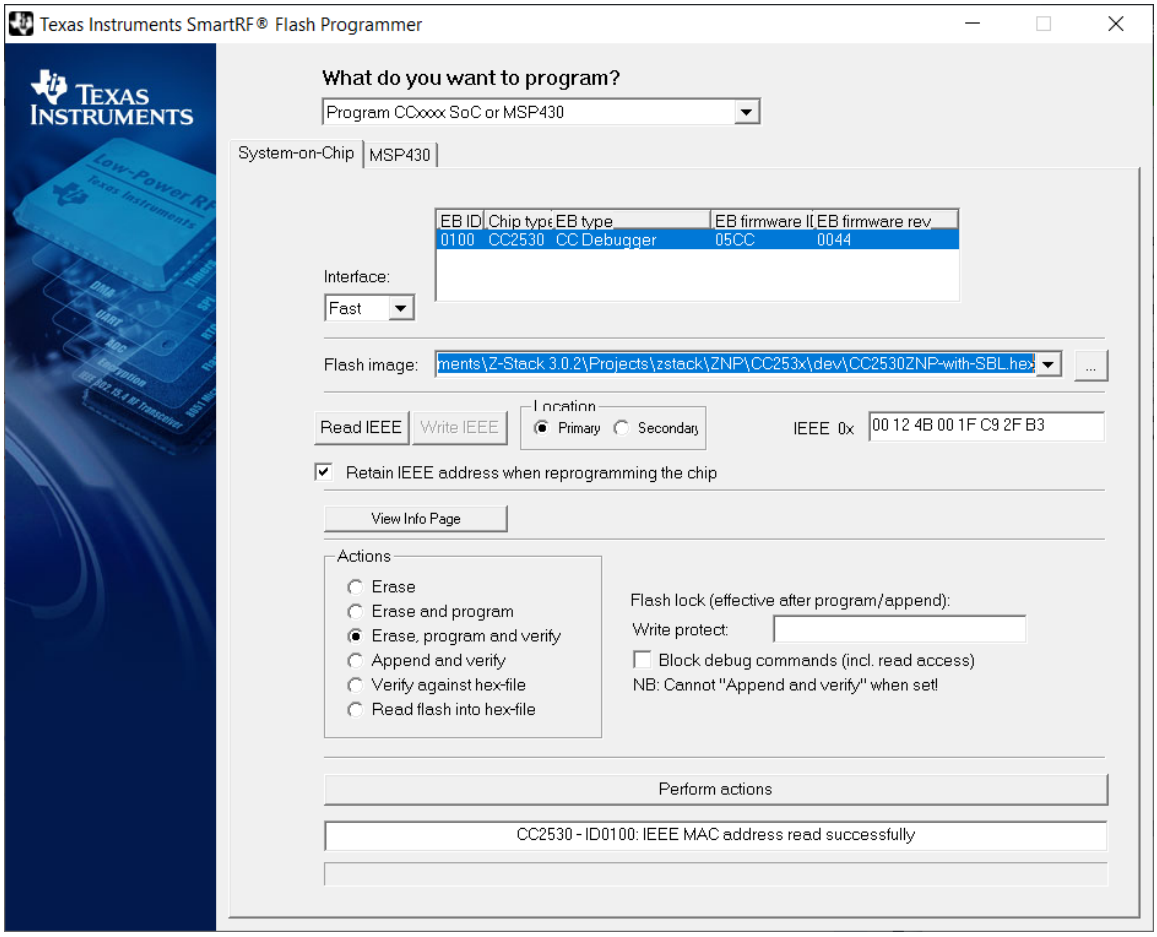
K nahrávání FW do desky cc2530 se používá převážně nástroj CC Debugger, navržený společností TI. Přestože je primárně určený k nahrávání FW, je možné ho použít i pro debugování aplikace na čipech CCxxxx založených na architektuře 8051. Dále je možné ho použít jako most pro odposlech sítě pomocí cc253x čipů [27].

Pro nahrávání FW je doporučeno použít software SmartRF Flash Programmer od TI, a pro debugování IAR Embedded Workbench od společnosti IAR Systems. FW je možné nahrát i jinými způsoby, např přes desku ESP, nebo Raspberry Pi.

Minimální zapojení pinů CC Debugeru a cc2530 odpovídá následujícímu schématu vyobrazeném na obrázku 5.3. V aplikaci Flash Programmer se vybere zařízení, do kterého se bude nahrávat. Požadovaný FW se zvolí v kolonce `Flash image`. Poté stačí kliknout na tlačítko `Perform actions` a počkat na nahrátí FW 5.4.



Obrázek 5.3: Minimální zapojení nutné pro nahrání FW



Obrázek 5.4: Aplikace SmartRF Flash Programmer

Kapitola 6

Popis implementace

Vestavěný systém brány chytrých zařízení je implementován jako vlastní komponenta v rámci ESPHome, který se umí připojit k Home Assistant, kde se vytváří automatizace domácnosti. Komponenta obstarává komunikaci mezi serverem Home Assitant a ZigBee zařízeními připojenými k síti.

Aby bylo možné bránu používat, je nutné ji mít připojenou k wifi a v elektrické síti. Aby se brána mohla připojit k wifi, musí se definovat ssid a heslo v konfiguračním souboru ESPHome 6.2.4. ESP32 je propojena s deskou cc2530 pomocí rozhraní UART, je proto nutné v mít v konfiguračním souboru ESPHome 6.2.4 správně definované piny a přenosovou rychlost použitou pro propojení. Pro komunikaci s Home Assistant 2.3.2 pomocí MQTT je nutné přidat integraci pro MQTT do Home Assistant. Ta bude představovat MQTT broker 2.3 a je potřeba ji definovat v konfiguračním souboru ESPHome 6.2.4.

Hlavní soubor má název zcl_mqtt_bridge.cpp a obsahuje třídu představující komponentu brány ZClMqttBridge. Hlavičkový soubor obsahuje prstředky pro jednodušší práci v kódu, jako jsou třídy pro ZNP příkazy 2.3.3 nebo ZCL clustery 4. Nakonec je v této kapitole rozebrán konfigurační soubor ESPHome a jeho tagy mqtt a wifi.

6.1 Implementační nástroje

Zdrojový kód je psán v jazyce C++ a skládá se ze 2 souborů, `zcl_mqtt_bridge.cpp` a jeho hlavičkového souboru `zcl_mqtt_bridge.h`. Využívá frameworku ESPHome a několika jeho komponent jako bázových tříd. V rámci vytváření FW pro ESP používá konfigurační soubor s příponou `.yaml`. V něm jsou definované další komponenty nutné pro komunikaci s Home Assistant. ESPHome je také použit pro nahrávání FW a monitorování komunikace brány.

6.2 Důležité části(třídy) programu

V této sekci se nachází klíčové prvky SW pro fungování brány chytrých zařízení. Jsou zde vysvětleny inicializační a spouštěcí sekvence pro desku cc2530, třídy pro práci s pakety a metody pro komunikaci s Home Assistant a koncovými zařízeními. Nakonec je zde popsán konfigurační soubor ESPHome.

6.2.1 Třída ZCluster

Třída ZCluster zjednodušuje práci při přijímání a odesílání ZigBee paketů. Skládá se z identifikátoru clusteru definovaného výčtovým typem(enum) v ZCLHelper 6.2.2, ze stringu zdrojové adresy, a seznamů atributů a příkazů. Příkazy se zatím nepoužívají, jsou zde pro úplnost a možné budoucí použití. Atributy se skládají z identifikátoru, typu a řetězcové(string) hodnoty atibutu. Identifikátor a typ jsou definovány enumem v ZCLHelper 6.2.2.

6.2.2 Třída ZCLHelper

Třída ZCLHelper obsahuje 4 druhy enumů. Enum pro datové typy, identifikátory clusterů, atributů a příkazů. Tyto enumy nejsou ještě plně definovány a při přidávání nových zařízení a funkcionalit se budou postupně doplňovat. Následně zde najdeme funkci *get_component_name*, která podle názvu modelu zařízení vybere o jaký druh a třídu komponenty se jedná. Je implementována jako jednoduchý `if else`, který se bude doplňovat při přidávání podpory novým zařízením. Funkci *n2hexstr* vytvářející string s hexadecimální hodnotou čísla, předaného jí parametrem. Funkce je vytvořena se zaměřením na nejmenší možnou velikost a nevyužívá proto žádné další knihovny, než standardní a string. Poslední implementovaná funkce v této třídě je *count_fcs*, která se stará o spočítání kontrolního bajtu pro bajtová pole definovaná Z-Stack ZNP 2.3.3. Z vektoru bajtů spočítá od LEN bajtu po poslední DATA bajt operací XOR kontrolní součet, který nakonec vrací.

6.2.3 Třída ZClMqttBridge

Třída ZClMqttBridge představuje komponentu brány chytrých zařízení a využívá některých tříd v rámci ESPHome pro jednodušší integraci. Ze třídy *Component* dědí metody *setup* a *loop* a je možné ji integrovat jako vlastní komponentu do ESPHome. Díky třídě *CustomMQTTDevice* je schopná použít metody *subscribe* a *publish* na základě definovaného MQTT brokeru v konfiguračním souboru. A nakonec třída *UARTDevice* předává metody a atributy nutné pro komunikaci po sériové lince UART.

setup

Metoda setup je zavolána při připojení ESP32 do elektrické sítě. V první řadě se inicializuje počítadlo Transakcí. To slouží ke správnému párování odeslaných zpráv z desky cc2530. Následně se 5 sekund čeká na inicializaci FW desky cc2530. Poté voláme metodu pro inicializaci cc2530, *zcl_mqtt_bridge_init*, v té zkontrolujeme podle síťového klíče, jestli musíme inicializovat FW celý znovu, nebo ho stačí pouze spustit.

Pokud klíče stejné nejsou, spustí se inicializační sekvence příkazů odeslaných přes UART desce cc2530 6.1. Konfigurační příkazy zapisují do nevolatilní paměti, dále jen NV. V inicializaci se nastaví síťový klíč stejný jako je `Trust Center` klíč. Pokud jsou klíče stejné, inicializační sekvence se přeskočí a pouze se spustí funkce *boot*.

Spouštěcí sekvence 6.2 se skládá z povolení připojení zařízení bez `Trust Center` klíče. Tento příkaz se zde vyskytuje, protože s výměnou TC klíče nefungovalo připojení routovacích zařízení jako jsou zářivka nebo chytrá zásuvka. Poté povolíme bráně začít fungovat jako koordinátor a zakážeme připojování zařízení, abychom měli pod 100% kontrolou, kdy budeme zařízení připojovat. Nakonec registrujeme koncové body. To umožňuje naslouchat zprávám od routerů a koncových zařízení.

Příkaz	vysvětlení
Soft Reset	vyresetuje zařízení
Config StartUp Option	Nastaví do NV výchozí stav při startu
Config Logical Type	Nastaví do NV typ zařízení na koordinátor
Config Direct CB	Nastaví do NV, získání odpovědí bez registrace
Config PAN ID	Nastaví do NV ID ZigBee sítě
Config EXT PAN ID	Nastaví do NV rozšířené ID ZigBee sítě
Config Channel	Nastaví do NV kanál, na kterém operuje
Config NWK Key	Nastaví do NV sítový klíč
BDB Set Channel	Nastaví kanál, na kterém zařízení naslocuhá
BDB Start Commisioning	Uvede koordinátor do provozu

Tabulka 6.1: `Inicializační sekvence`

Příkaz	vysvětlení
Soft Reset	vyresetuje zařízení
BDB Require TC Key	Při přidání nového zařízení se nekontroluje TC klíč
Start Up From APP	Uvede zařízení do provozu dle nastavení v NV
Permit Join	Zakáže přidávání nových zařízení do sítě
AF Register	Zaregistruje koncové body pro příjem zpráv

Tabulka 6.2: Spouštěcí sekvence

loop

Funkce *loop* umožňuje programu reagovat na vnější vlivy. Nachází se v ní metoda *receive*. Ta kontroluje, zdali po rozhraní UART přichází nějaký bajt. Příchozí bajty skládá do vektoru, pomocí funkce *n2hexstr* definované v třídě ZCLHelper 6.2.2 se vytvoří string pro vytvoření logu příchozí zprávy pod hlavičkou `CC2530 - received`.

Následně se zkontroluje typ příchozí zprávy. Pokud je příchozí zpráva na základě Z-Stack ZNP specifikace typu *ZDO_END_DEVICE_ANNCE* 2.3.3, znamená to, že bylo přidáno nové zařízení do sítě. Pokud je příchozí zpráva typu *ZDO_LEAVE_IND* 2.3.3, znamená to, že zařízení opustilo síť. V tomto případě se odešlou publish zprávy, které zajistí, aby zmizelo i z Home Assistant. Nakonec proběhne kontrola, zdali je zpráva typu *AF_INCOMING_MSG* 2.3.3. Pokud ano, rozebere se funkcí *check_ZCL_cmd* na základě identifikátoru(ID) clusteru, typu příkazu a typu atributů do objektu *ZCluster*. Tento objekt zastřešuje práci se ZCL a slouží pro zpřehlednění kódu 6.2.1.

Pokud cluster nemá žádné atributy, je tato zpráva považována za nevalidní a dále se ignoruje. V opačném případě se zkontroluje ID clusteru, ID a hodnota atributu. Od nich se odvíjí téma a obsah zprávy publish.

Pokud je cluster typu *basic* a atribut je typu *model_identifier*, tak v hodnotě atributu najdeme název modelu připojeného zařízení. Tato zpráva bývá poslána jako 1. po připojení zařízení do sítě. Když tato zpráva přijde, tak pomocí MQTT Discovery se vytvoří konfigurační zpráva MQTT. Podle názvu modelu určíme o jaký typ zařízení se jedná a podle zdrojové adresy clusteru vytvoříme unikátní identifikátor zařízení. Tyto informace se předají do tématu zprávy. Ta se následně odešle virtuální domácnosti, aby věděla o nově přidaném zařízení. Nakonec pokud zařízení očekává, že bude přijímat zprávy od GW, odešle požadavek na subscribe pod daným tématem.

Příchozí zprávy MQTT

Při odeslání příkazu z Home Assistant zařízení připojenému k bráně, Home Assitant vytvoří MQTT zprávu a odešle ji bráně. Při příjmu této zprávy se zavolá funkce *on_message*, nebo *on_json_message*. Ty mají v parametrech téma a obsah zprávy. Na základě tématu se určí jakému zařízení se odešle příkaz. Obsah určuje typ příkazu k odeslání. Po určení typu a obsahu zprávy se volá metoda *send_AF_Data_Request*, která poskládá zprávu typu *AF_DATA_REQUEST* 2.3.3 se správným clusterem a akcí. Ta se poté odešle po sériové lince desce cc2530, která z ní vytvoří ZigBee paket a distribuuje ho koncovému zařízení.

Odchozí zprávy pro cc2530

Pro odesílání zpráv po sériové lince desce cc2530 se používá
funkce *send_cmd_and_wait_for_response*. Ta odešle příkaz a zaloguje co odeslala s hlavičkou `CC2530 - write`. Poté 500 ms počká a zkusí přečíst odpověď. Tato funkce je připravena na kontrolu očekáváných odpovědí pro příkazy, ale zatím tato funkcionalita chybí.

6.2.4 Konfigurační soubor ESPHome

ESPHome využívá konfigurační soubor k vytvoření vlastního FW pro ESP32. Lze ho vytvořit pomocí ESPHome wizardu, nebo ručně. Je to YAML soubor obsahující různé tagy představující komponenty a jejich definice.

Konfigurační soubor návrhu brány se jmenuje `black.yaml`. Aby byla brána funkční je potřeba mít definované určité tagy. Hlavní tag `esphome`, ve kterém je definovaná paltforma ESP32 a typ desky, která je použita (u návrhu je využita esp32doit-devkit-v1). Dále je nutné definovat `wifi`. K té se bude deska připojovat po připojení do elektrické sítě. Pro připojení MQTT, v tagu `mqtt`, je definován odkaz na broker jako IP adresa, port(standardně 1883), uživatelské jméno(defaultně homeassistant), a heslo pro broker. Tag `uart` umožňuje komunikaci po sériové lince přes UART rozhraní. Je zapotřebí definovat Tx a Rx pin, přenosovou rychlost na 115 200 a unikátní identifikátor komponenty. Nakonec je potřeba vytvořit vlastní komponentu pod tagem `custom_component`, kde v lambda funkci v programovacím jazyce C++ vytvoříme instanci třídy *zcl_mqtt_bridge*. V konstruktoru ji je předán odkaz na uart, na kterém komunikuje s cc2530. Následně ji zaregistrujeme funkcí *App.register_component()*.

Bráně je možné přidat funkcionalitu jednoduchým přidáním tagu a definice. Tímto způsobem je přidáno tlačítko pro povolení přidávání zařízení do sítě. Je definováno tagem *switch* a při přepnutí do zapnuté fáze povolí na 29 sekund přidávat zařízení. Přepnutím do vypnuté fáze, je možné kdykoli přidávání ukončit.

Kapitola 7

Testování

Pro ladění funkcionální logiky vytvářené brány bylo možné využít virtuálního stroje, kde se dá simulovat server Home Assistant, desek ESP32 a cc2530 pro FW vlastní aplikace a nakonec nástroj Z-Tool 2.5 pro prvotní testování UART rozhraní. Dále byly využity zařízení z ekosystému Aqara 3 jako koncové a routovací zařízení sítě.

V momentě, kdy funkcionalita cc2530 a její zapojení byly otestovány, přesunulo se testování na propojení a komunikaci mezi cc2530 a ESP32. Zde se testovala spouštěcí a inicializační sekvence návrhu, přidávání a odebírání zařízení a v neposlední řadě příjem a odchod zpráv. Poté následovalo otestování propojení ESP32 a MQTT brokeru v Home Assistant.

7.1 Příprava testovacího prostředí

Pro vytvoření následujícího návrhu brány pro chytrá zařízení, je nutné mít desky plošných spojů cc2530, ESP32, CC Debugger a alespoň 5 propojovacích vodičů. Dále je nutné mít bezdrátové připojení k internetu, aby se ESP32 mohlo připojit k Wi-Fi.

Pokud ještě neexistuje instance Home Assistant, ke které by bylo možné se připojit, je nutné vytvořit vlastní 7.1.1. Následně se musí přidat integrace MQTT Mosquitto broker do Home Assistant.

Poté je nutné nahrát FW do desek cc2530 5.2 a ESP32. FW pro cc2530 použijeme stejný jako bylo popisováno dříve. Pro ESP32 je ale nezbytné konfigurační soubor ESPHome upravit. Důležité jsou pouze dva tagy, `wifi` a `mqtt`. U wifi tagu je zapotřebí upravit `ssid` a `password`. Ty se nastaví podle jména a hesla wifi, ke které se zařízení bude připojovat. V tagu `mqtt` se do brokeru vyplní IP adresa Home Assistantu, dále se musí změnit heslo. Heslo se nachází v nastavení integrace `Mosquitto broker` pod možností `RE-CONFIGURE MQTT`. V případě, že se změnilo jméno, uživatelské jméno nebo port MQTT integrace, musí se změnit i v konfiguračním souboru. Po provedení těchto změn, je možné nahrát FW do ESP32. Toho se docílí použitím příkazu *ESPHome black.yaml run*, kde soubor `black.yaml` je konfigurační soubor.

Následně stačí propojit desky ESP32 a cc2530 podle schématu 5.1 a v Home Assistant použít tlačítko *PermitJoiningReq*, které umožní na 29 sekund přidávat zařízení. Je možné ho kdykoli vypnout přepnutím zpět. Nakonec stačí jen přidat zařízení a tvořit vlastní automatizovanou domácnost.

7.1.1 Vytvoření instance Home Assistant

Jedna z možností umístění Home Assistant je v rámci virtuálního stroje. Proto je nutné mít program pro virtualizaci operačních systémů, jako je např. Oracle VM VirtualBox. Poté se stáhne operační systém Home Assistant z jejich stránek a připraví se virtualizačním prostředí.

Virtuální stroj Home Assistant je nutné nastavit. Při spuštění systému po zobrazení řádku „Welcome to HassOS“ se zadá příkaz *login* s uživatelským jménem „root“. Následně se příkazem *nmcli con edit "HassOS default"* spustí editace konfigurace. Zde se nastaví unikátní IP adresa zařízení příkazem *set ipv4.addresses x.x.x.x/24*, kde písmena `x` naznačují části IPv4 adresy. Poté nastavíme adresu `dns` a `gateway` stejnou jako má router, ke kterému se má zařízení připojit. To se provede příkazy *set ipv4.dns y.y.y.y* a *set ipv4.gateway y.y.y.y*, kde písmena `y` představují adresu routeru. Nakonec pomocí příkazů *save* a *quit* nastavení uložíme. Nyní je nutné Home Assistant restartovat, a poté bude nastaven a připraven k použití.

7.2 Alfa testování

Hlavní část testování byla prováděna pouze mnou, neboť jsem nestihl dodat komponentu do ESPHome s dostatečným předstihem. V budoucnu předpokládám zpětnou vazbu od komunity Home Assistant a její zakomponování do projektu.

Testování se skládalo ze sestavení brány, připojení do elektrické sítě a následného monitorování událostí na rozhraní UART a provozu na ZigBee síti. Poté jsem připojoval jednotlivé zařízení k síti, nejprve jsem se pokusil přidat zařízení bez externího povolení přidávání, což se, i přestože nemělo, povedlo. Dále jsem testoval přidávání a vytváření automatizací jednotlivých zařízení ekosystému Aqara 3. Nakonec jsem otestoval manuální provádění akcí skrze Home Assistant, např. změnu jasu chytré žárovky, vypnutí a zapnutí.

7.3 Výsledky testování

Zpětná vazba od komunity v době odevzdání práce bohužel chybí. Integrace brány do nové automatizované domácnosti sice není problematická, ale integrace již do zavedené sítě, přepojování všech zařízení a automatizací je zdlouhavé. Z tohoto důvodu očekávám zpětnou vazbu od komunity Home Assistant později.

Z vlastního testování jsem objevil některé chyby týkající se inicializační sekvence, kdy se neměnil klíč sítě podle nastavení nebo se v jakékoliv chvíli daly přidat zařízení. Tyto chyby mi při opravě poskytly náměty na vylepšení FW návrhu, např. přidání tlačítka pro povolení přidávání zařízení.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo vytvořit náhradu za bránu chytrých zařízení Xiaomi Aqara a následně ji integrovat do Home Assistant. Nezbytnou součástí bylo nastudovat problematiku týkající se internetu věcí, ZigBee sítí, frameworku ESPHome a Z-Stack ZNP příkazů. Dále bylo důležité zjistit, jaké ZigBee clustery chytrá zařízení odesílají a přijímají.

Výsledný produkt částečně nahrazuje bránu Xiaomi Aqara, ale pouze pro vybraná zařízení. Je však možné a chtěné ji v rámci open-source projektu ESPHome doplňovat o nová zařízení. Tím, že se jedná o komponentu ESPHome je velmi snadné ji integrovat do Home Assitant. Také je vytvořena z poměrně levných součástí, desek ESP32, cc2530 a micro USB kabelu od mobilní nabíječky. Bohužel se mi zatím nepovedlo, poskytnout resubscribe po vypojení a opětovném zapojení brány do elektrické sítě. Proto se zatím nesmí produkt odpojovat, neboť by se musela celá síť přidávat znovu.

Zdrojové kódy pro vytvoření komponenty jsou dodány na DVD vloženém v práci. Řešení je dostupné ve formě custom_component na veřejném github repozitáři https://github.com/HyperReap/zcl_mqtt_bridge a v rozpracované formě klíčové komponenty je v rámci `pull request` na https://github.com/esphome/esphome. Dále video prokazující funkcionalitu brány se nachází na youtube kanále https://www.youtube.com/channel/UCD6JE69X3Es0jRdojX95Baw v playlistu `ESPHome Components`.

Do budoucna doufám, že náhrada brány v rámci ESPHome nezůstane pouze vlastní komponentou, ale stane se klíčovou komponentou s množstvím dalších nastavení přes konfigurační soubor. Také by bylo dobré, aby přibyly zařízení od dalších výrobců, než jen Lumi United Technology, popř. vzniklo více variací této komponenty pro ostatní značky chytrých zařízení.

Literatura

[1] BATNA Sp. z o.o. sp. k.. *Xiaomi Aqara LED Light Bulb*. 2021 [cit. 2021-05-04]. Dostupné z: https://www.batna24.com/cz/p/xiaomi-aqara-led-light-bulb-inteligentni-zarovka-rmmpp.

[2] HiveMQ GmbH. *MQTT Essentials*. 2021 [cit. 2021-04-29]. Dostupné z: https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/.

[3] Home Assistant, Inc. . *Home Assistant Developer Docs*. 2021 [cit. 2021-04-26]. Dostupné z: https://developers.home-assistant.io/.

[4] Home Assistant, Inc. . *MQTT Discovery*. 2021 [cit. 2021-04-26]. Dostupné z: https://www.home-assistant.io/docs/mqtt/discovery.

[5] Jesse Hills. *Getting Started with ESPHome and Home Assistant*. 2021 [cit. 2021-04-27]. Dostupné z: https://esphome.io/guides/getting_started_hassio.html.

[6] Juan Carlos Pacheco . *Decrypting ZigBee packets with Wireshark*. Červen 2017 [cit. 2020-07-19]. Dostupné z: https://community.nxp.com/thread/331972.

[7] Paulus Shoutsen . *Nabu Casa has acquired ESPHome*. Home Assistant, Inc. , březen 2021 [cit. 2021-05-04]. Dostupné z: https://www.home-assistant.io/blog/2021/03/18/nabu-casa-has-acquired-esphome//.

[8] Texas Instruments Incorporated . *Z-STACK*. 2021 [cit. 2021-05-07]. Dostupné z: https://www.ti.com/tool/Z-STACK#related-design-resources.

[9] The Zigbee Alliance. *ZigBee Cluster Library Specification*. 075123. 2016 [cit. 2021-05-08]. Revision 6. Dostupné z: https://zigbeealliance.org/wp-content/uploads/2019/12/07-5123-06-zigbee-cluster-library-specification.pdf.

[10] Ubilogix International, Inc. . *Ubiqua*. 2021 [cit. 2021-04-26]. Dostupné z: https://www.ubilogix.com/ubiqua/.

[11] Zigbee Alliance. *Aqara Hub M1S*. 2020 [cit. 2021-05-04]. Dostupné z: https://zigbeealliance.org/zigbee_products/aqara-hub-m1s/.

[12] Zigbee Alliance. *How does Alliance Technology help my lightbulbs*. 2020 [cit. 2021-05-04]. Dostupné z: https://zigbeealliance.org/market-uses/lighting/.

[13] BARTEK, L. *Protokol ZigBee pro bezdrátové senzorové sítě*. Brno, CZ, 2014. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: http://hdl.handle.net/11012/55567.

[14] British Broadcasting Corporation. *Google denies Xiaomi access over security bug.* Leden 2020 [cit. 2020-01-10]. Dostupné z: https://www.bbc.com/news/technology-50981993.

[15] Gill, K., Yang, S.-H., Yao, F. a Lu, X. A ZigBee-Based Home Automation System. *Consumer Electronics, IEEE Transactions on.* Červen 2009, sv. 55, s. 422 – 430, [cit. 2020-01-08]. DOI: 10.1109/TCE.2009.5174403.

[16] Hadzovic, S., Mrdovic, S. a Radonjic, M. Identification of IoT Actors. *Sensors.* 2021, sv. 21, č. 6. DOI: 10.3390/s21062093. ISSN 1424-8220. Dostupné z: https://www.mdpi.com/1424-8220/21/6/2093.

[17] Kinney, P. ZigBee Technology: Wireless Control that Simply Works. In: *In Communications design conference.* říjen 2003, sv. 2, č. 2, s. 1–7 [cit. 2020-01-08].

[18] Lumi United Technology Co.. *Google denies Xiaomi access over security bug.* Leden 2020 [cit. 2020-01-11]. Dostupné z: https://www.aqara.com/.

[19] NXP Laboratories UK . *ZigBee Cluster Library User Guide.* JN-UG-3077. NXP Semiconductors, únor 2015 [cit. 2020-01-15]. Revision 2.1. Dostupné z: https://www.nxp.com/docs/en/user-guide/JN-UG-3077.pdf.

[20] NXP Laboratories UK . *ZigBee Home Automation User Guide.* JN-UG-3114. NXP Semiconductors, srpen 2016 [cit. 2021-05-7]. Revision 1.4. Dostupné z: https://www.nxp.com/docs/en/user-guide/JN-UG-3114.pdf.

[21] OASIS Open. *MQTT Version 5.0.* Oasis Open, 2019 [cit. 2021-04-29]. Dostupné z: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf.

[22] Owaida, A. *Google disables Xiaomi smart home integration after camera bug* [online]. Eset WeLiveSecurity, leden 2020 [cit. 2020-01-08]. Dostupné z: https://www.welivesecurity.com/2020/01/03/google-disables-xiaomi-smart-home-integration/.

[23] Rosslin, J., Robles, R. a Kim, T.-H. A Review on Security in Smart Home Development. *International Journal of Advanced Science and Technology.* Březen 2010, sv. 15.

[24] Texas Isntruments Inc.. *CC-Debugger User's Guide.* Texas Instruments, 2007 [cit. 2021-04-26]. Dostupné z: https://documentation.help/Z-Tool/Welcome.htm.

[25] Texas Isntruments, Inc. *ZigBee PRO Network Processor.* SWRA442. Texas Instruments, 2015 [cit. 2021-04-29].

[26] Texas Isntruments, Inc. *Z-Stack Monitor and Test API.* SWRA198. Texas Instruments, 2020 [cit. 2021-04-30]. Revision 1.178.

[27] Texas Isntruments Incorporated. *CC-Debugger User's Guide.* SWRU197H. Texas Instruments, srpen 2010 [cit. 2021-04-26]. Revised April 2014. Dostupné z: https://www.ti.com/lit/ug/swru197h/swru197h.pdf.

[28] Texas Isntruments Incorporated. *CC13x0 SimpleLink™ TI 15.4-Stack 2.x.x Embedded.* SWRU489A. Texas Instruments, srpen 2016 [cit. 2021-04-30]. 95-102 s. Revision 1.14. Dostupné z: https://www.ti.com/lit/ug/swru489a/swru489a.pdf.

[29] Xia, F., Yang, L. T., Wang, L. a Vinel, A. Internet of Things. *INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS*. 2012, sv. 25, č. 9, s. 1101–1102, [cit. 2020-01-18]. DOI: 10.1002/dac.2417.

[30] XiaoMi Corporation. *About Us*. Duben 2010 [cit. 2020-01-11]. Dostupné z: https://www.mi.com/global/about.