České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra měření
Program: Otevřená informatika
Specializace: Internet věcí

Využití levných zařízení od různých výrobců k ovládání domácího okolí pro lidi s omezenou schopností pohybu

Use of cheap devices from various manufacturers to control the home environment for people with limited mobility

BAKALÁŘSKÁ PRÁCE

Vypracoval: Ondrej Nečas
Vedúci práce: Ing. Petr Novák, Ph.D.
Rok: 2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Nečas** Jméno: **Ondrej** Osobní číslo: **492311**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra měření**

Studijní program: **Otevřená informatika**

Specializace: **Internet věcí**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Využití levných zařízení od různých výrobců k ovládání domácího okolí pro lidi s omezenou schopností pohybu

Název bakalářské práce anglicky:

Use of cheap devices from various manufacturers to control the home environment for people with limited mobility

Pokyny pro vypracování:

Lidé s omezenou schopností pohybu (např. na vozíku) využívají nejčastěji levné komponenty pro chytré domácnosti (případně IoT), a to zejména pro ovládání jednoduchých domácích zařízení. Největší překážkou je však nekompatibilita různých výrobců, a hlavně složitost ovládacích aplikací, tedy často nepoužitelných pro tento typ cílových uživatelů.
1) Prostudujte levná zařízení pro řízení (chytré) domácnosti určená zejména pro běžné použití (světla, zásuvky, topení, okna, žaluzie, ...) a jejich ovládací aplikace. Zaměřte se na současné využití levných zařízení od různých výrobců a rovněž na vhodnost jejich ovládacích aplikací pro zmíněnou cílovou skupinu lidí.
2) Navrhněte / vytvořte jednoduchou domácí centrálu / hub pro vybraná zařízení různých výrobců za účelem sjednocení jejich ovládání zejména pro uživatele s omezením pohybu.
3) Vytvořte komunikační bod této centrály / hubu umožňující ovládání prvků domácnosti z aplikace vytvářené speciálně pro již zmíněné uživatele (např. tablet s aplikací na míru umístěný na invalidním vozíku). Současně pamatujte na i na možnost vzdáleného řízení dohlížející osobou.
4) Navrhněte a ověřte rovněž možnost připojení zcela vlastního zařízení (například simulace ovladače TV) do navrhované centrály / hubu.

Seznam doporučené literatury:

[1] WWW stránky nalezených / obdobných projektů, manuály a další relevantní informace
[2] Price Mark, C# 8.0 and .NET Core 3.0 - Modern Cross-Platform Development, Packt, 2019
[3] MacDonald Matthew, Pro WPF 4.5 in C#, APress, 2012
[4] Noviello Carmine, Mastering STM32, LeanPub, 2017
[5] další potřebné materiály poskytne vedoucí práce

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Petr Novák, Ph.D. oddělení kognitivních systémů a neurovědy, Český institut informatiky, robotiky a kybernetiky

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **14.01.2022** Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce:
do konce letního semestru 2022/2023

Ing. Petr Novák, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. _____
Datum převzetí zadání

_____
Podpis studenta

Prehlásenie

Prehlasujem, že som svoju bakalársku prácu vypracoval samostatne a použil som iba podklady (literatúru, projekty, SW atd.) uvedené v priloženom zozname.

V Praze dne .....

.....
Ondrej Nečas

Poďakovanie

Ďakujem vedúcemu práce Ing. Petru Novákovi, Ph.D. za neoceniteľné rady a pomoc pri tvorbe bakalárskej práce.

Ondrej Nečas

Obsah

Abstrakt

Práca sa zaoberá preskúmaním dostupných zariadení a aplikácií Internetu vecí (IoT), vhodnosťou ich ovládacích prvkov a návrhom konceptu jednotného ovládacie bodu pre domácu automatizáciu u osôb s obmedzenou schopnosťou celkového pohybu alebo zníženou zručnosťou pri horných končatinách. Teoretická časť obsahuje súhrn niektorých existujúcich softvérových a hardvérových systémov. Posudzuje ich vhodnosť a dostupnosť pre nasadenie u ľudí so špeciálnymi potrebami. Motivácia práce spočíva v predstavení jedného z možných riešení narastajúcej komplexnosti domácej automatizácie a uľahčení jej nasadenia práve u netradičných skupín spotrebiteľov. Na základe zistených nedostatkov navrhuje koncept "centrály" (IoT HUB), ktorá by zjednotila prístupové brány pre integráciu rozličných IoT protokolov, poskytovala štandardizovanú dátovú štruktúru na komunikáciu s týmito protokolmi a zjednodušila vývoj špeciálnych riadiacich aplikácií. Tento koncept následne prakticky overuje s použitím protokolu Zigbee, technológie Zigbee2MQTT a virtuálneho COM port zariadenia. Zigbee predstavuje zástupcu komerčne dostupných IoT protokolov a COM port osobitne vytvorené riešenie domácej automatizácie. Avšak dizajn centrály je navrhnutý univerzálne, aby bolo možné integrovať aj ďalšie IoT protokoly. Výstupom je prototyp riešenia tejto domácej centrály, vytvorený za účelom overenia realizovateľnosti zvoleného prístupu.

Kľúčové slová: IoT, Zigbee, MQTT, Zigbee2MQTT, HUB

Abstract

The thesis explores the available devices and applications of the Internet of Things (IoT), the suitability of their control components and the design of a single control point for home automation suitable for people with limited overall movement capability or reduced capability of their upper limbs. The theoretical section summarizes some existing software and hardware systems. It assesses their suitability and availability for deployment to people with special needs. The motivation of the work lies in presenting one of many possible solutions to the increasing complexity of home automation and making it easier to deploy with non-traditional consumer groups. Based on the shortcomings identified, the thesis proposes the concept of IoT HUB, which would unify the access gates for the integration of different IoT communication protocols, provide a standard data structure for communication with these protocols and facilitate the development of special control applications. Subsequently, this concept is practically verified using Zigbee protocol, Zigbee2MQTT technology, and a virtual COM port device. Zigbee represents a commercially available IoT protocol and a virtual COM port represents a specially designed home automation solution. However, the design of the HUB is universal so that other IoT protocols can be integrated as well. The output of the thesis is a prototype of the IoT HUB solution, designed to verify the feasibility of the chosen approach.

Key words: IoT, Zigbee, MQTT, Zigbee2MQTT, HUB

Zoznam obrázkov

Úvod

Pre bežného spotrebiteľa predstavujú IoT zariadenia zvýšenie nielen osobného komfortu, ale napríklad aj bezpečnosti v každodennom živote. Avšak pri použití v domácnostiach ľudí s hendikepom alebo u dôchodcov, môže nabrať ich použitie omnoho väčšieho rozmeru.

Domáca automatizácia predstavuje u týchto skupín zlepšenie kvality každodenného života, ako aj napomáha k nezávislému spôsobu života. Do týchto kategórií spadajú napríklad osoby na invalidnom vozíku, ľudia s poruchami audio-vizuálnymi, pohybovými (najmä ruky a nohy), kognitívnymi a podobne. Hlavným účelom je pre nich zjednodušenie ovládania bežných domácich zariadení alebo spotrebičov, ktorých použitie môže pre znevýhodneného alebo staršieho človeka predstavovať veľkú výzvu. Spomedzi jednoduchých zariadení spomeňme napríklad osvetlenie v domácnosti, spínače, aktuátory (otváranie okien, dverí, žalúzie) alebo rôzne senzory (vlhkosti, teploty, tlaku alebo dotyku) a iné. Spektrum výrobkov sa teda pohybuje od jednoduchých monitorovacích senzorov až po pomerne zložité ovládanie rôznych aktuátorov, spínacích relé a celých komplexných spotrebičov. Dostupnosť a možnosti chytrých výrobkov sa z roka na rok zvyšujú a tento trend do budúcna určite nebude spomaľovať [1].

Najväčšie prekážky vo využití týchto výrobkov u osôb s obmedzenou schopnosťou pohybu predstavuje ich cena, zložitosť implementácie v domácnosti, nevyhovujúca vzájomná kompatibilita a zložité ovládanie. Tieto faktory nie len odrádzajú od zakúpenia IoT systému, ale taktiež značne komplikujú jeho nasadenie. Zmienené osoby zväčša nedisponujú financiami potrebnými pre rozsiahle alebo komplexné použitie IoT zariadení. Najatraktívnejšie sa z toho dôvodu javia cenovo dostupné systémy, zložené z jednoduchých a lacných komponent od rôznych výrobcov. Napriek rozsiahlej dostupnosti sú však primárne určené pre bežného spotrebiteľa a častokrát nespĺňajú požadované potreby na použitie u zmienenej skupiny používateľov. Založené sú v prvom rade na bezdrôtových technológiach komunikácie medzi zariadeniami. Rozsah dostupných technológií a protokolov je široký. Tento prístup predstavuje veľkú výhodu v rýchlosti a jednoduchosti nasadenia. Nové systémy sa zavádzajú do už prítomnej infraštruktúry, čím odpadá núdza o inštaláciu dodatočných zdrojov. Na druhej strane má táto forma aj mnoho nevýhod. V lokalite sa môže vyskytovať nadmerné rušenie komunikačného prostredia. Toto rušenie spôsobené zväčša inými IoT zariadeniami spôsobuje, že sa prenos komunikácie stáva obtiažnym. Taktiež smerovanie komunikácie medzi zariadeniami v spomenutých podmienkach predstavuje výzvu. Ich nedostatočná vzájomná integrácia vytvára núdzu o prítomnosť aplikačného riešenia u každého protokolu samostatne [2].

Nedostatočná interoperabilita medzi rôznymi, ale súčasne použitými technológiami je jedným z hlavných nedostatkov vo využití IoT. Vertikálna architektúra používaných protokolov spolu s heterogenitou dát predstavuje hlavolam v nasadení IoT. Tento hlavolam je o to väčší, pokiaľ ide o nasadanie u ľudí, ktorý sa neradia medzi bežných spotrebiteľov. Dokonca aj v doméne jedného protokolu sa môžu vyskytnúť problémy s kombináciou zariadení od rôznych výrobcov. Z dostupných zdrojov je zreteľné, že významná časť pozornosti je smerovaná na túto problematiku [3]. Práve na tento aspekt sa táto práca bližšie zameriava.

Jednými z dostupných riešení by mohli byť rôzne softvérové projekty na domácu automatizáciu. Pomocou aplikačného rozhrania v telefóne alebo webového prehliadača je možné monitorovať a regulovať domácnosť tvorenú kombináciou rôznych protokolov. Avšak zložitosť takýchto ovládacích riešení je pre našu skupinu používateľov (najmä s obmedzením pohybu celkového alebo horných končatín) prekážkou. Súčasne narastajúca komplexnosť riadiacich prvkov v rozhraniach týchto aplikácií činí ich nasadenie nevhodným.

Pre vyššie zmienené nedostatky v IoT systémoch sa táto práca zaoberá návrhom architektúry pre horizontálnu integráciu dostupných IoT zariadení. Táto architektúra by umožňovala prepojenie IoT technológií a uľahčila ich nasadenie u ľudí s telesným obmedzením. Podpora na mieru tvorených aplikácií a systémov dohľadu je zaistená predstavením jednotného prístupového bodu. Takýto bod podporuje použitie mnohých protokolov, čím redukuje nároky na zaobstaranie osobitnej infraštruktúry. Prístup a jeho vhodnosť je následne overená čiastočnou implementáciou zmienenej architektúry vo forme "centrály" (HUB). Centrála je tvorená hardvérovou stránkou, ktorá slúži na komunikáciu medzi jednotlivými protokolmi a softvérom, ktorý ponúka komunikačné rozhranie riadiacim aplikáciam. Na overenie konceptu je implementovaný protokol Zigbee, ako zástupca komerčných systémov a rozhranie pre COM port (prípadne LAN), ako osobitne vytvorené riešenie domácej automatizácie.

Štruktúra práce je organizovaná do siedmich kapitol. Ciele sú bližšie priblížené v kapitole 2. Kapitola 3 predstavuje súčasnú situáciu IoT systémov a komunikačných protokolov. Rovnako rozoberá koncept domácej automatizácie. Kapitola 4 sa zaoberá návrhom architektúry centrály . Rozoberá jej použitú logickú štruktúru a jednotlivé prvky. Následne je v Kapitole 5 zahrnutá implementácia použitá na overenie predstaveného konceptu. Popísané sú použité technológie a postupy. V Kapitole 6 je demonštračná činnosť centrály spoločne so stručným návodom na jej použitie. Kapitola 7 tvorí záver tejto práce. Pojednáva o dosiahnutých výsledkoch, navrhuje možné zlepšenia a uvádza témy, ktorými by sa bolo možné ďalej zaoberať.

Kapitola 1

Ciele zadania

Každý cieľ zadania je rozvedený do samostatnej podkapitoly. Príslušná podkapitola obsahuje podrobnejšie popísané témy a bližšie vymedzuje rozsah príslušnej časti práce. Na záver sú zaradené dve dodatočné sekcie. Prvá je text, ktorý vymedzuje rozsah a oblasti, ktorým sa práca nevenuje a nie sú v nasledujúcich kapitolách rozoberané. Druhá popisuje želané výsledky, ktorými sa práca primárne zaoberá.

1.1 Prostudujte levná zařízení pro řízení (chytré) domácnosti

Prostudujte levná zařízení pro řízení (chytré) domácnosti určená zejména pro běžné použití (světla, zásuvky, topení, okna, žaluzie, ...) a jejich ovládací aplikace. Zaměřte se na současné využití levných zařízení od různých výrobců a rovněž na vhodnost jejich ovládacích aplikací pro zmíněnou cílovou skupinu lidí.

Tento bod práce je zameraný primárne na rešerš dostupných systémových riešení z prostredia IoT. Súčasne bližšie špecifikuje cieľovú skupinu ľudí s obmedzenou schopnosťou pohybu. Bod bol už z časti spracovaný v úvode textu. Detailnejšie rozdelenie existujúcich riešení bude rozdelené viac do hĺbky v nasledujúcej kapitole spolu s vymedzením cieľovej skupiny.

1.2 Navrhněte / vytvořte jednoduchou domácí centrálu / hub

Navrhněte / vytvořte jednoduchou domácí centrálu / hub pro vybraná zařízení různých výrobců za účelem sjednocení jejich ovládání zejména pro uživatele s omezením pohybu.

Množina výrobcov chytrých zariadení a ich dostupných výrobkov je veľmi rozsiahla, rovnako tak počet technológií a protokolov, na ktorých sú založené. Momentálne je nutné pre každý komunikačný protokol mať vlastnú aplikáciu, ktorou je protokol ovládaný. U výrobkov na báze rovnakého protokolu je možné pripojiť zariadenie jedného výrobcu do aplikačného prostredia druhého, avšak nie vždy je na to ovládacia aplikácia potrebne prispôsobená. Pri vytvorení domácej centrály sme sa primárne zamerali na štandardizáciu ovládania nielen medzi zariadeniami na báze

rovnakého komunikačného protokolu, ale aj medzi protokolmi samotnými. Týmto prístupom by sme chceli dosiahnuť jednotné ovládacie prostredie z pohľadu užívateľa. Zjednotenie pripojenia rozličných protokolov na úrovni hardvéru v centrále a následne štandardizácia na úrovni softvéru. V takto definovanom prostredí poskytovanom centrálou by bolo výrazne jednoduchšie tvoriť ovládacie aplikácie, špeciálne prispôsobené pre potreby ľudí nielen s obmedzenou schopnosťou pohybu. Pre účely overenia funkčnosti tejto centrály sme vybrali zariadenia, ktoré sa často vyskytujú v domácnostiach ľudí práve s obmedzenou možnosťou pohybu.

1.3 Vytvořte komunikační bod této centrály / hubu

Vytvořte komunikační bod této centrály / hubu umožňující ovládání prvků domácnosti z aplikace vytvářené speciálně pro již zmíněné uživatele (např. tablet s aplikací na míru umístěný na invalidním vozíku). Současně pamatujte na i na možnost vzdáleného řízení dohlížející osobou.

V súčasnej dobe je pripojenie k internetu v domácnostiach väčšinou zabezpečené prípojkou poskytovateľa s dostupnou prenosovou infraštruktúrou v danej oblasti. Tá je napojená do lokálneho smerovača, ktorý pre obydlie vytvára samostatnú sieť (LAN) podporujúcu technológie WiFi a Ethernet. Centrála (HUB) samozrejme využíva túto existujúcu infraštruktúru. Hlavný prístup do centrály zo strany aplikácie bude teda cez túto sieť. Na nižších vrstvách bude komunikácia jednotne prebiehať formou TCP/IP. Takýmto spôsobom je zaistené pripojenie pre lokálnu aplikáciu (napríklad na tablete osoby s znevýhodnením) a súčasne aplikáciu vzdialeného dozoru alebo pomoci.

1.4 Navrhněte a ověřte rovněž možnost připojení zcela vlastního zařízení

Navrhněte a ověřte rovněž možnost připojení zcela vlastního zařízení (například simulace ovladače TV) do navrhované centrály / hubu.

Nakoľko sa potreby už spomenutej skupiny ľudí môžu často líšiť v prostriedkoch, ktoré chceme ovládať, je viac než vítané zakomponovanie možnosti, ako pripojiť pre nich na mieru vytvorené zariadenie. Na fyzickej úrovni sa zariadenie pripojí do centrály a tá ho následne transformuje a začlení medzi ostatné, už pripojené zariadenia na iných protokoloch. Týmto spôsobom by bolo možné obsluhovať spotrebiče nepodporované použitím dostupných komerčných IoT výrobkov. Ak nie je pre danú osobu možné využívať už existujúce ovládacie prvky zabezpečené výrobcom spotrebiča alebo je nutné špeciálne ovládanie, špecifické pre potreby tej ktorej osoby, je tento prístup jedinou možnosťou na integráciu do domácej automatizácie. Ako príklad by sme uviedli ovládač od televízie. Po pripojení vlastného IoT zariadenia (simulujúce bežný ovládač TV), je televíziu možné obsluhovať priamo z aplikácie na tablete, pomocou jednoduchých dotykov obrazovky. Problém u osôb, ktoré nie sú schopné používať bežný ovládač od televízie, by mohol byť napríklad v neschopnosti ovládač pevne uchopiť a stlačiť potrebné tlačidlá pre prepnutie programu alebo zmenu hlasitosti.

1.5 Čomu sa práca nevenuje

Práca sa nevenuje bezpečnosti komunikácie či už medzi zariadeniami samotnými, zariadeniami a centrálou alebo centrálou a riadiacou aplikáciou. Taktiež sa nevenuje bezpečnosti a dostupnosti dát, ktoré sú uchovávané v centrále alebo pridružených aplikáciach. Ďalej sa nezaoberá výškou spoľahlivosti použitých technológií a protokolov. Spoľahlivosť komunikácie nebola nijakým spôsobom meraná, ani sa ňou žiadna časť práce nezaoberá. Výstupom práce nie je dokumentácia konečného výrobku alebo aplikácie určenej pre výrobu. Centrála spoločne s použitou aplikáciou momentálne neboli testované, ani iným spôsobom neprišli do kontaktu s ľuďmi s obmedzenou schopnosťou pohybu, pre ktorých sú určené.

1.6 Čomu sa práca venuje

Hlavným cieľom je overenie konceptu architektúry jednotnej centrály a špeciálnej aplikácie, ako centrálneho komunikačného bodu pre rozličné technológie v IoT domácnosti a jednako posúdenie vhodnosti nasadenia takéhoto prístupu v domácnostiach pre ľudí s obmedzenou schopnosťou pohybu. Posúdenie výhod, nevýhod alebo možných prekážok, ktoré tento prístup oproti konvenčnému prístupu prináša.

Kapitola 2

Súčasný stav

Ovládanie domáceho okolia ("smart home" alebo aj inteligentná domácnosť) je paradigma z prostredia IoT. Ide o prístup, kedy je fyzická obsluha bežných domácich spotrebičov alebo zariadení delegovaná na IoT zariadenia. Tie sa ovládajú formou aplikácie alebo inteligentného panelu. Takýto prístup vytvára domácnosť, ktorá je z logického hľadiska jeden ucelený systém a môžme naň nazerať viac ako na komunikačnú sieť, než ako na jednotlivé spotrebiče.

V kapitole bude najprv uvedený koncept IoT. Jeho význam a definícia. Nakoľko ide o širokú tému, je vhodné sa pozrieť na jeho taxonómiu. V nej sa prejde od abstraktnej logickej klasifikácie po jednoduché koncové prvky. Predstavené budú odvetvia architektúry dostupných riešení, dátové protokoly, technológie, infraštruktúra, výrobky a taktiež aj ovládanie. Z nej prejdeme na oblasť domácej automatizácie, presne povedané riadenie okolia, ktoré je primárnym zameraním tejto práce. Na to bude hlbšie špecifikovaná skupina ľudí, pre ktorú je plánom rozšírenie využitia domácej automatizácie. Popíšu sa jej potreby na ovládanie domácnosti a možnosti, ako by bolo možné takéhoto ovládania docieliť.

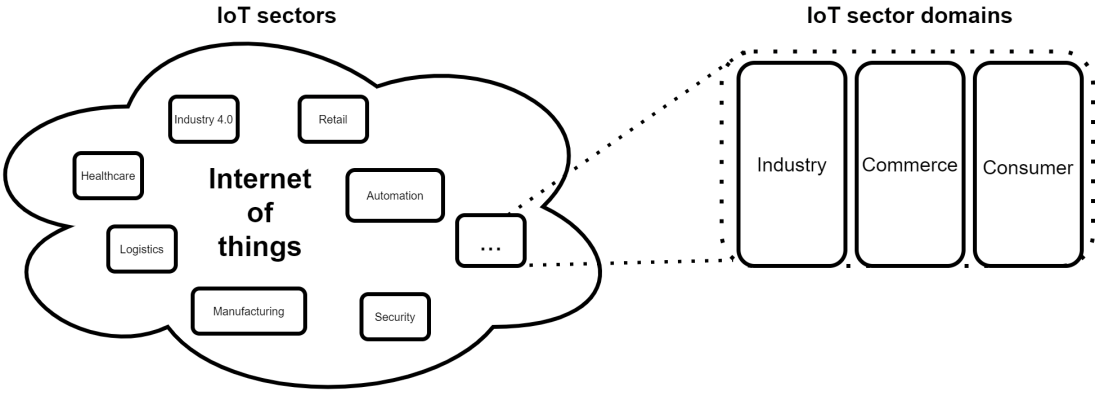
2.1 Internet of Things

Je ťažké úplne presne definovať termín "Internet of Things" (IoT). Neexistuje jedna všeobecne zaužívaná definícia, ktorá by bola kolektívna či už pre akademickú, priemyselnú alebo komerčnú sféru.

Množstvo článkov a prác spoločne odkazuje, že termín IoT pravdepodobne vznikol v roku 1999. Vtedy bol prvýkrát spojený s myšlienkou zachytávania dát z reálneho sveta. Keby počítače zberom dát (bez pomoci človeka) vedeli o predmetoch všetky informácie, ktoré sa dajú vedieť, tak by sme boli schopní výrazným spôsobom redukovať odpad, straty alebo náklady v našich každodenných životoch [4]. V širšom zmysle slova ide o systém prepojených zariadení. Ide o zariadenia výpočtové, mechanické alebo digitálne. Objekty, zvieratá alebo ľudí s jedinečnými identifikátormi a schopnosťou transportu dát cez komunikačnú sieť bez nutnosti interakcie človeka s človekom alebo človeka s počítačom [5]. Systém, kde fyzické predmety navzájom komunikujú medzi sebou či už vo forme "machine-to-machine" (M2M) alebo "person-to-computer" (P2C) [6]. Súhrn infraštruktúr spájajúcich komunikačné objekty, poskytujúci ich správu, zabezpečujúci transport, uloženie, spracovanie a prí-

stup k dátam, ktoré sú generované užívateľmi alebo inými systémami [7].

Od roku 1999 sa použitie mnohonásobne rozšírilo do všetkých oblastí našej spoločnosti. Každým rokom sú predstavené nové odvetvia, kde využívame služby a riešenia založené na IoT. Za tieto roky bolo v koncepte zahrnutých množstvo nových technológií a postupov, ktoré vytvorili veľmi široké spektrum použitia. Na základe tejto obšírnosti je veľmi ťažké určiť celkové logické rozdelenie IoT, jeho infraštruktúry, protokolov, zariadení, systémov a služieb. Ako už bolo spomenuté, tak dostupné zdroje sa v presnej definícii, čo je IoT často rozchádzajú. Neexistuje teda bohužiaľ všeobecne zaužívaná taxonómia. V nasledujúcich podkapitolách preto uvádzam vlastnú taxonómiu, ktorá ponúka dostatočný prehľad do problematiky, ale detailnejšie nezachádza do odvetiev a technológií, ktoré nesúvisia s oblasťou zamerania tejto práce.



Obr. 2.1: Internet of Things taxonómia [8].

2.1.1 Odvetvia využitia IoT

Využitie je možné nájsť vo všetkých odvetviach ľudskej spoločnosti. Z bežných použití by som uviedol monitorovanie výrobných procesov, logistika, doprava, zdravotníctvo, podpora domácnosti a pracoviska, "smart" mestá, bezpečnosť, komunikácie alebo systémový dohľad [9]. Uplatnenie sa nachádza spomedzi iných aj v exotickejších sférach ako produkcia potravín, poľnohospodárstvo, odpadové hospodárstvo alebo boj s požiarom. Taktiež je silný záujem o aplikovanie IoT v sociálnych sieťach, väčšie zapojenie v "cloudových" platformách, integrácia s umelou inteligenciou, uplatnenie vo vývoji environmentálne priateľských systémov. Spracovanie a zber dát čoraz väčšieho množstva dát (Big Data), preventívna údržba, inteligentné riadenie výroby, správa miest a množstvo iných [10], [11].

2.1.2 Domény IoT riešení

Dostupné IoT riešenia sa dajú rozdeliť do troch hlavných domén. Do domény priemyselnej (Industrial Internet of Things - IIoT), komerčnej a spotrebiteľskej. Toto členenie je založené na rozdielnych skupinách koncových užívateľov, komplexnosti riešení v danej doméne, veľkosti nasadenia, doby životnosti, určení koncových zariadení, ceny systémov, presnosti meraní, spoľahlivosti prenosu, type zbieraných dát

apod. Avšak rozdelenie nemá presne stanovené hranice. Aplikácie IoT sa môžu prelínať medzi doménami vo svojom rozsahu využitia. Primárnym dôvodom tohto členenia je vniesť štruktúru do chaotického systému z hľadiska toho, ako komunikáciu medzi zariadeniami využívame, a o ktorých objektoch komunikujeme.

Priemyselné IoT

"Industrial Internet of Things" (IIoT) je zameraný primárne na existujúce rozsiahle automatizované systémy v priemysle, logistike, poľnohospodárstve apod. Hlavnou úlohou je dramatické zvýšenie produktivity a efektívnosti vo výrobe. Najčastejšie použitie je vo veľkých fabrikách alebo zložitých výrobných procesoch, kde hrá rolu výkonnosť, presnosť, výrobné náklady alebo spoľahlivosť [12]. Použitie IoT v tejto kombinácii prináša napríklad bezpečnosť pri manipulácii s komplexnou mašinériou alebo dôkladné monitorovanie výrobného postupu. Prináša nahradenie jednoduchých opakujúcich sa úkonov v administratíve, montáži, kontrole kvality alebo plánovaní pomocou veľkého počtu senzorov a aktuátorov rôzneho druhu. Vďaka tomu sa náklady zmenšujú a kvalita spolu s objemom produkcie zvyšuje. Takýto vývoj výrobných procesov sa označuje pod pojmom "Priemysel 4.0" (Industry 4.0) [13].

Jeden z príkladov nových riešení v oblasti IIoT je takzvaný "Predictive monitoring". Skladá sa z hromadenia veľkého množstva dát ako vibrácie, teplota, vlhkosť, hustota, prúd, napätie apod. Následne sa s pomocou algoritmov strojového učenia (Machine learning) predvída zlyhanie výrobnej techniky. Týmto spôsobom sa v predstihu zamedzí možnému poškodeniu napríklad kritickej infraštruktúry u výrobnej linky. Vďaka tomu ostáva efektivita procesov na rovnakej úrovni a minimalizujú sa náklady, ktoré by mohli vzniknúť v dôsledku prerušenia výroby [14].

Komerčné IoT

Naproti zameraniu na stroje u IIoT sa "Commercial IoT" orientuje smerom k ľuďom a firmám. Cieľom je prostredie našej spoločnosti mimo domova. Poskytuje služby, ktoré nám zvyšujú komfort, pomáhajú s monitorovaním a riadením okolia, zlepšujú organizáciu, vytvárajú pozitívnu skúsenosť u zákazníkov [12]. Do tejto domény sa dajú zaradiť odvetvia ako inteligentné mestá, doprava, zdravotníctvo, kancelárie, obchody, budovy, marketing a iné. Dáta sú v týchto sférach zbierané predovšetkým o ľuďoch alebo o prostredí, v ktorom sa pohybujú. Zo špecifických aplikácií sa dajú vybrať inteligentné lôžka v nemocniciach, komunikácia medzi vozidlami v doprave, zaznamenávanie odpadového hospodárstva [15], zníženie energetickej spotreby budov, riadenie prístupových systémov do kancelárií, odpočet spotreby vody a pod.

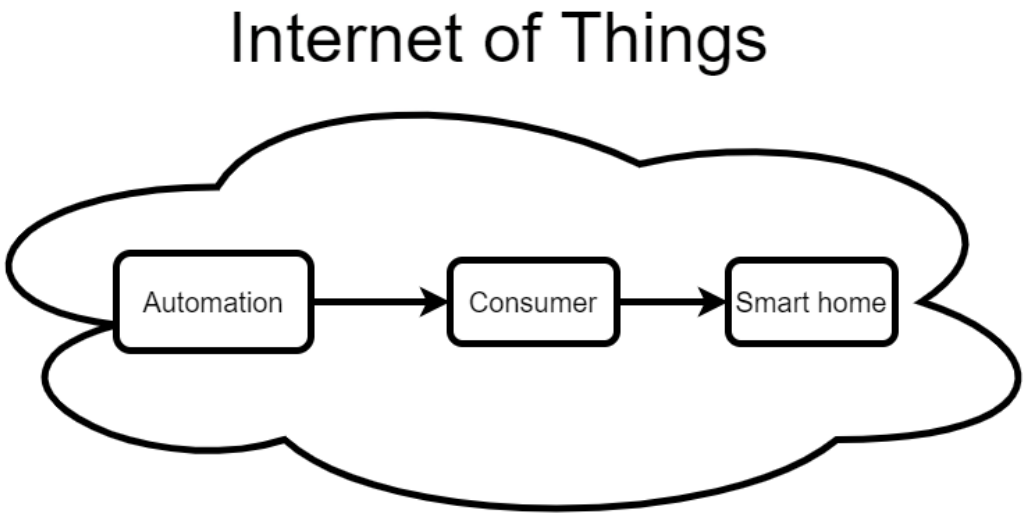
Spotrebiteľské IoT

"Consumer IoT" je o individuálnych užívateľoch, ich rodinách a domácnosti. Takéto zariadenia zvyšujú náš osobný komfort, uľahčujú každodenné úkony, pomáhajú s riadením domácnosti, poskytujú pohodlie alebo zábavu. Najčastejšie je táto doména spájaná s monitorovaním alebo automatizáciou domácnosti, ale aj napríklad s výrobkami, ktoré môžeme nosiť na našom tele. Z klasických príkladov sa

dajú uviesť inteligentné hodinky, ovládanie svetiel, kontrola zámkov, riadenie televízie alebo rádia, zber informácií o domácom prostredí, spravovanie spotrebičov, dohľad nad osobami a pod. Práve na spotrebiteľské prostredie, presnejšie na správu a monitorovanie domácnosti je zameraná táto práca.

2.1.3 Inteligentná domácnosť

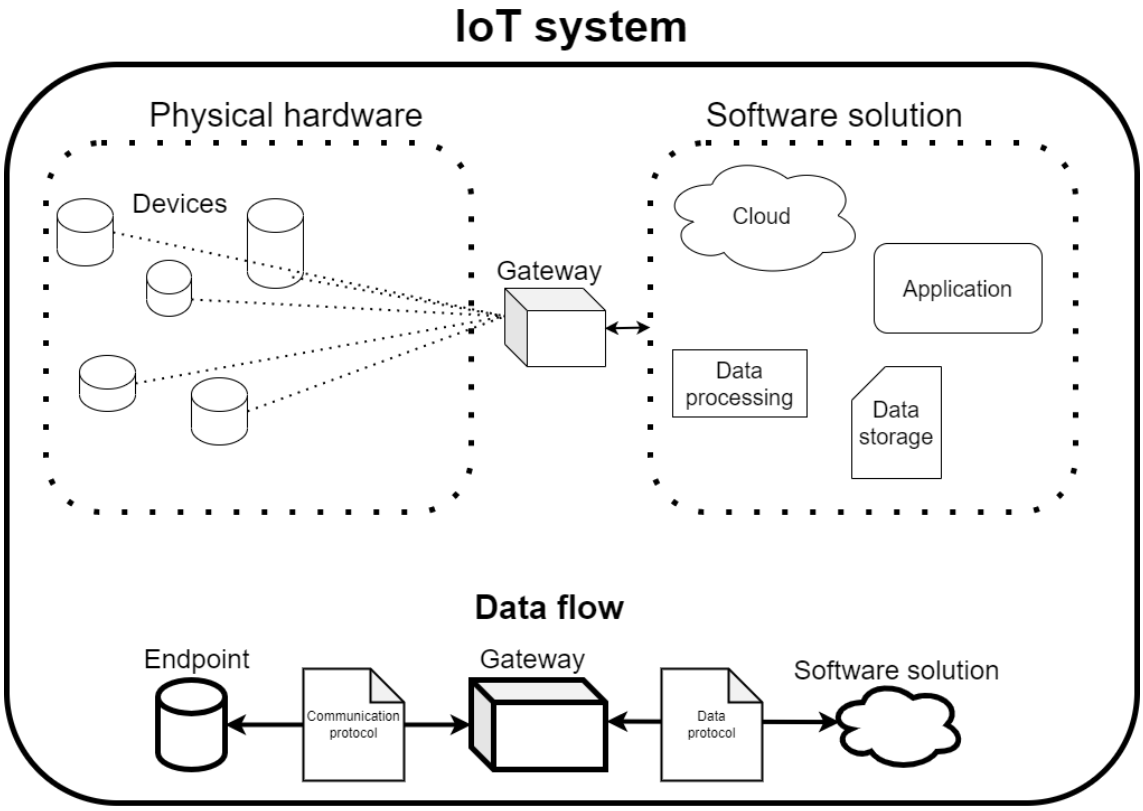
Inteligentná domácnosť môže byť opísaná množstvom spôsobov. Jeden z konceptov je domácnosť s automatizovaným systémom, ktorý obsahuje senzory, aktuátory a inteligentné riadenie. Tento princíp ponúka pohodlný, jednoduchý a bezpečný systém, ktorý zlepšuje kvalitu života a tvorí domáce spotrebiče viac ovládateľnými pre seniorov a ľudí s postihnutím [16]. Obsahuje v sebe spotrebiče a zariadenia, ktoré medzi sebou komunikujú alebo je možné komunikovať s nimi. Opakujúce sa alebo zložité úkony za nás vykonávajú stroje, čo šetrí čas a zvyšuje osobné pohodlie. Inteligentné senzory zbierajú dáta o sebe a svojom okolí. Detailné informácie o chode obydlia sú nám potom poskytnuté pre spracovanie. Týmto spôsobom máme prehľad napríklad o spotrebe celej domácnosti, stave našich zariadení alebo pohybe osôb v jej priestoroch. S pomocou inteligentných aktuátorov celé prostredie následne ovládame na diaľku. Avšak s koncovými zariadeniami nekomunikujeme priamo. Na to sú potrebné inteligentné aplikácie napríklad v telefóne alebo tablete, ktoré pre nás vytvárajú most medzi našimi inštrukciami a koncovými bodmi.



Obr. 2.2: Zameranie práce z pohľadu IoT [8].

2.1.4 Prvky IoT systému

Štruktúra systému inteligentnej domácnosti v zásade nie je pevne daná. K dispozícií je množstvo riešení, ktoré v konečnom dôsledku vykonávajú rovnakú funkciu (zabezpečujú chod inteligentnej domácnosti), ale ich interné zloženie sa líši. V praxi si každý výrobca implementuje vlastný hardvér aj softvér a snaží sa udržovať akési "know how". V nasledujúcich podkapitolách by som predstavil stavebné bloky, ktoré sú z môjho pohľadu nevyhnutné na základný chod takéhoto systému v prostredí inteligentnej domácnosti.



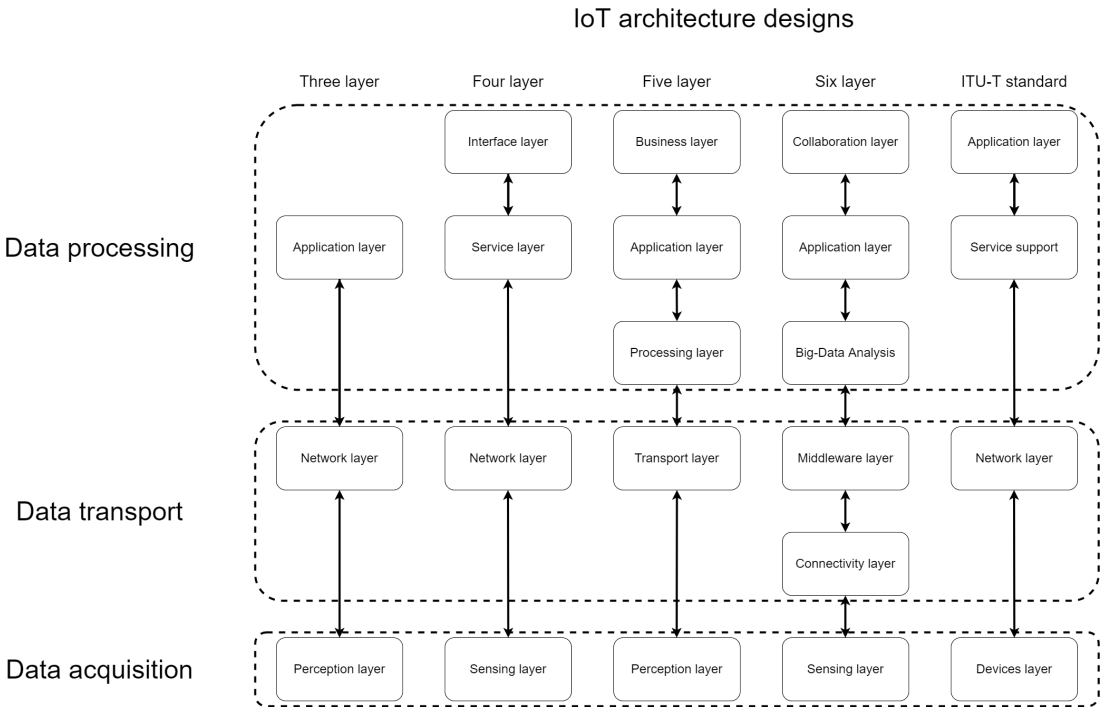
Obr. 2.3: Rozdelenie IoT systému [8].

Architektúra

Existujú rôzne návrhy akoby mala byť IoT architektúra rozdelená. Členenie na logickej úrovni môže vyzerať rozdielne. Napriek veľkému množstvu aplikácií a asociovaných technológií sa objekty dajú spojiť pomocou toku informácií. Základnou jednotkou sú dáta, ktoré musia byť získané, transportované, spracované a dostupné koncovej entite [7]. Na základe takéhoto pohľadu sa dajú vyčleniť tri hlavné procesy, ktoré sa s dátami dejú od ich vytvorenia až po použitie. Prvým je získanie (zber) alebo použitie dát (Data acquisition). Tie sa následne musia transportovať (Data transport) na spracovanie a využitie (Data processing).

Ako je na Obr. 2.4 viditeľné, v praxi je použitý rozdielny počet vrstiev pri návrhoch IoT systémov. Podľa [17] je všeobecná IoT architektúra v literatúre tvorená z troch základných vrstiev: “perception”, “network”, “application”. Vrstva vnímania (Perception layer) je fyzická vrstva, ktorá pozoruje a meria fyzické parametre použitím inteligentných zariadení. Na zber sú využité rôzne meracie technológie. Sieťová vrstva (Network layer) dohliada/kontroluje príjem dát zo zariadení a prenáša ich až k aplikačnej vrstve pomocou sieťových protokolov. Nakoniec aplikačná vrstva (Application layer) zaobstaráva služby pre spotrebiteľov.

V práci [10] je architektúra zhodnotená v štyroch vrstvách. Aplikačná vrstva bola rozdelená medzi rôzne služby (Service layer) a riadiace rozhranie (Interface layer) pre používateľov.



Obr. 2.4: Rozdelenie IoT architektúry na logické vrstvy [8].

Ďalšou populárnou architektúrou je päť vrstiev. Pribudla vrstva pre sprocesovanie získaných dát (Processing layer), aby boli pretvorené do potrebných štruktúr, roztriedené podľa obsahu, uložené, analyzované a pod. Až následne sú v rozumnej forme predané aplikáciam. Na vrchole stromu je zahrnutá ešte logika pre biznis (Business layer), zastrešujúca celý IoT systém, kde sa deje analýza nazývaná "big data", rozhodovanie biznis stratégie a plánuje sa budúce smerovanie.

Návrh [18] pre strojové učenie a umelú inteligenciu v IoT obsahuje šesť vrstiev. Vrstva pripojenia (Connectivity layer) zamedzuje vzniku chýb pri spojenom použití heterogénnych senzorov. Účelom "middleware" (Middleware layer) je zabezpečenie interoperability hardvéru. Toto zjednotenie uľahčenie prácu vývojárom softvéru, aby nemuseli rozmýšľať nad rozdielnymi ekosystémami, ktoré aplikácia riadi. Následne pribudlo zabudovanie "big data" (Big-data layer) analýzy na spracovanie pre inteligentné algoritmy a služby. Nakoniec vrstva kolaborácie (Collaboration layer) je použitá na zlepšenie služieb využitím strojového učenia a umelej inteligencie cez zber, distribúciu a ohodnotenie doteraz získaných dát.

Rôzne kritéria a snaha o štandardizáciu boli predstavené organizáciami ako napríklad "ITU Telecommunication Standardization" (ITU-T). Na Obr. 2.4 je znázornené poslednou schémou. Ako všeobecné členenie sa uvažuje architektúra so štyrmi vrstvami.

Ani jedno zo zmienených usporiadaní však nie je vhodné na využitie vo všetkých odvetviach IoT. V praxi si výrobcovia a poskytovatelia väčšinou implementujú svoju vlastnú architektúru, ktorá sedí ich špecifickému riešeniu a odvetviu. Väčšina by však mohla byť braná ako rozšírenie základného modelu s tromi vrstvami.

Koncové zariadenia

Z IoT infraštruktúry sú v domácnosti v každom prípade nutné dva typy fyzického hardvéru "gateway" (brána) a koncové body (senzory alebo aktuátory).

Brána predstavuje most medzi rozdielnymi komunikačnými technológiami. Pracuje ako mediátor na otvorenie spojenia medzi koncový bodmi a aplikačnou vrstvou IoT systému. S jej pomocou je možné zostaviť komunikáciu medzi zariadeniami samotnými alebo medzi zariadeniami a IoT aplikačnými službami. Brána je typicky hardvérová, ale môže byť implementovaná aj formou softvéru. Taktiež zabezpečuje úlohy prekladu prijímaného protokolu, agregácie dát, lokálneho "pre-procesovania", ukladania dát a iného. Nakoľko pracujú koncové body s malou spotrebou energie, je brána efektívny spôsob na komunikáciu s aplikačnou vrstvou. Následne je v bráne komunikácia konvertovaná do dátového protokolu a poslaná ďalej. V praxi je mnoho realizácií takejto brány, pričom štandardne každý výrobca implementuje vlastnú pre svoj IoT systém a protokol [19].

Koncové body sú priamo zariadenia prepojené s reálnym okolím. Najtypickejší pre IoT zariadenia je zber dát, ale na trhu sa čoraz viac objavujú produkty, ktoré disponujú vstupne-výstupnou funkcionalitou. Takéto zariadenia prijímajú pokyny a sú vybavené mechanikou nutnou k zabezpečenie fyzickej zmeny. Zo zástupcov sa dajú spomenúť inteligentné svetlá, zásuvky, vypínače, tlačidlá, termostaty, chladničky, práčky, žalúzie, ventilátory, zámky, dotykové senzory, hlásiče CO2 a mnoho ďalších.

Komunikačné technológie

Komunikácia medzi koncovými zariadeniami samotnými alebo koncovými zariadeniami a zvyškom IoT infraštruktúry môže prebieha v dvoch kategóriách. Po fyzickom drôtovom médiu alebo bezdrôtovou formou.

Drôtová komunikácia je prístup s využitím pevnej kabeláže, ktorý má nesporné výhody. Komunikácia je spoľahlivá. Prostredie v drôte sa nemení a je pevne dané. Pokiaľ je vedená z bodu do bodu, tak sa nemusí deliť o komunikačné prostriedky s inými entitami. Pokiaľ je zdieľaná, napríklad ako u zberníc, tak dochádza k minimálnemu počtu kolízií vďaka použitiu modelov ako "master-slave". Využitie je najmä v priemysle, kde sa kladie veľký dôraz na doručenie dát v správnom poradí a čase. Nevýhoda spočíva v nutnosti inštalácie novej infraštruktúry pri každom novom zariadení. To výrazným spôsobom obmedzuje škálovateľnosť a flexibilitu už vytvorených sietí. Z tohto dôvodu je takýto prístup nevhodný do domácej automatizácie, kde sa IoT inštaluje väčšinou dodatočne, do prostredia, ktoré sa po fyzickej stránke už nemení. Medzi zástupcov sa radia rôzne zbernice. Napríklad DALI alebo KNX. Sériová linka ako RS485 alebo RS232. Využívajú sa hlavne v priemyselnom riadení alebo pri výstavbe inteligentných budov. Dôležitým zástupcom je Ethernet. Jeho využitie je vďaka IP protokolu dostupné aj v bežných domácnostiach a domácej automatizácii, ale pre už spomenuté nevýhody v nutnosti dodatočnej kabeláže, takéto riešenie vo väčšine prípadov neprichádza v úvahu.

Bezdrôtová komunikácia naproti fyzickému médiu je omnoho flexibilnejšia. Do domácností sa zavádza bez nutnosti nových zdrojov. Nevýhodou je, že prístup k

bezdrôtovému médiu je zdieľaný všetkými zariadeniami, ktoré vysielajú na rovnakej frekvencii. Vzniká problém súťaženia o dostupné zdroje. Taktiež je problémom smerovanie tejto komunikácie. Zariadenia častokrát operujú z batérie a energetické nároky pre výrobcov predstavujú výzvu. Napriek tomu je inštalácia veľmi jednoduchá a nadmerná väčšina produktov do domácnosti operuje práve na tomto princípe. Medzi najznámejších z dostupných zástupcov patria WiFi, Bluetooth, Bluetooth Low Energy (BLE), Zigbee, Z-wave, LoRaWan, LoRa, Sigfox, NB-IoT, RFID, NFC. Rozdelenie je ďalej možné na ich využitie do krátkej (Short-distance), strednej (Medium-distance) alebo dlhej (Long-distance) vzdialenosti.

"Long-distance" (LoRaWan, LoRa, Sigfox, NB-IoT) - Takéto technológie dokážu prenášať komunikáciu na maximálnu vzdialenosť až niekoľkých kilometrov. Na druhú stranu je nutné využívať nízku prenosovú rýchlosť a taktiež malú veľkosť dát. Odozva sietí tiež dosahuje vysokých hodnôt. Komunikácia je skôr občasná pre vysoké energetické nároky na vysielanie. Taktiež k tomu sa prostredie domácej automatizácie pohybuje v ráde maximálne niekoľko desiatok metrov. Vyznačuje sa intenzívnou komunikáciou medzi zariadeniami a v prípadoch prenosu obrazu alebo zvuku ide o komunikáciu, ktorá môže dosahovať pomerne veľkých rozmerov. Z toho dôvodu sú tieto technológie do domácej automatizácie nevhodné a z hľadiska tejto práce nemá väčší význam sa nimi zaoberať.

"Medium-distance" (WiFi, Bluetooth, BLE, Zigbee, Z-wave) - Zástupcovia bežne využívaní v domácej automatizácii. Založený na IEEE 802.11 a IEEE 802.15. Ide o technológie v oblasti WLAN (Wireless Local Area Network) a WPAN (Wireless Personal Area Network) operujúce v spektre 2.4 GHz, 5 GHz, 868–921 MHz. Dostupné IoT systémy na spotrebiteľskom trhu sú založené práve na týchto riešeniach. Každý z protokolov má svoje výhody aj nedostatky.

WiFi podporuje vzdialenosti do 100 metrov. Je spoľahlivá a umožňuje pripojenie veľkého počtu zariadení. Avšak má vysoké nároky na energiu, čo môže u zariadení s batériou predstavovať problém. Vyžaduje veľký "bandwidth" a na frekvenciách 2.4 GHz a 5 GHz už operuje veľké množstvo výrobkov a miera rušenia je pomerne vysoká. Na druhú stranu v drvivej väčšine je už obsiahnutá v bežnej domácnosti, čo činí inštaláciu nových zariadení jednoduchou.

Bluetooth sa vyznačuje vzdialenosťou nižšou zhruba do 15 metrov. Má nižšie energetické nároky a dobre podporuje prenos zvuku. Podporuje bohužiaľ len pripojenie maximálne 7 zariadení v jednom momente, čo predstavuje prekážku pri plošnom nasadení.

Bluetooth Low Energy (BLE) bol vyvinutý pre posielanie malých správ a nízku spotrebu energie. Je určený skôr na nesúvislý prenos dát, ale posledné verzie protokolu sa aj tento problém snažia adresovať. Nachádza uplatnenie aj v priemysle. BLE aj Bluetooth sú dobre využiteľné ako nositeľné výrobky na telo.

Z-wave predstavuje kompromis v doteraz predstavených technológiach. Dosahuje vzdialeností do 100 metrov a má rozumné energetické nároky. Rýchlosť prenosu je nižšia ako u WiFi, ale na rozdiel od Bluetooth podporuje pripojenie veľkého množstva zariadení (okolo 230). Ďalšou výhodou je garancia interoperability zariadení od rôznych výrobcov, nakoľko je tento štandard vlastníctvom "Z-Wave Alliance", ktorá dohliada na certifikáciu nových výrobkov. To sa bohužiaľ odráža aj na cene týchto zariadení. Neverejnosť špecifikácie môže v určitých prípadoch predstavovať prekážku pri integrácii.

Zigbee sa na druhej strane javí ako riešenie predošlých nedostatkov. Vyznačuje sa podobnými parametrami ako Z-wave, ale štandard je verejne známy. Množstvo súčasne pripojených výrobkov môže byť až 65 000 na jednu sieť. Zariadenia sú energeticky nenáročné vďaka nízkej veľkosti správ a porovnateľnej rýchlosti prenosu ako u Z-wave. To je bohužiaľ stále obmedzujúce pri prenose zvuku alebo videa. Nevýhodu oproti Z-wave predstavuje nedostatočná podpora zariadení medzi výrobcami. Preto sa pri nasadení v domácnosti štandardne využíva len jeden výrobca, pre zabezpečenie dobrej vzájomnej kompatibility.

"Short-distance" (RFID, NFC) - Ako posledné sú technológie so vzdialenosťou prenosu do circa 20 centimetrov. Najčastejšie použité u inteligentných kariet alebo "tagov". Veľkou výhodou je nepotrebnosť batérie alebo externého napájania, nakoľko sa posiela jednorázovo len malé množstvo dát a energia na prenos sa generuje z vytvoreného el-mag. poľa. Osamotene v domácej automatizácii uplatnenie nemajú. Z toho dôvodu sa kombinujú s inými druhmi technológií, čo výrazným spôsobom zvyšuje možnosti ich použitia.

Prenosové protokoly

Dátové protokoly sa z pohľadu IoT systému špecializujú na formu, akou sú dáta prenášané medzi zariadeniami a po externej sieti. Z pohľadu OSI modelu by išlo o transportnú, sieťovú a aplikačnú vrstvu. V prostredí domácej automatizácie je špecifické použitie TCP/IP na potreby smerovania komunikácie (sieťová a transportná vrstva). Využitie napríklad mobilných sietí je takisto možné, ale v drvivej väčšine prípadov nepraktické. Výber protokolu na aplikačnej vrstve má väčšie možnosti výberu. Predstavené budú najrozšírenejšie z pohľadu použitia.

HTTP (HyperText Transfer Protocol) - je postavený na forme "request-response" pre "client-server" aplikácie. Jeden z najstarších protokolov a hlavný prenosový protokol webu. Nachádza všestranné využitie, avšak disponuje pomerne vysokou náročnosťou na prenosové zdroje. Správy dosahujú zbytočne rozsiahlych veľkostí pri prenose.

CoAP (Constrained Application Protocol) - taktiež pre aplikácie client-server. Vyvinutý ako následník HTTP. V praxi sa však veľmi neujal. Používaný najmä ako "one-to-one" komunikácia.

MQTT (Message Queue Telemetry Transport) - bol vyvinutý pre potreby IoT. Primárne "many-to-many" komunikácia prechádzajúca jednotným serverom (Broker). Založený na "publish-subscribe" modeli vhodnom pre zariadenia s obmedzenými zdrojmi. Uspôsobený pre prenos malých správ a jednoduchú komunikáciu medzi koncovými zariadeniami. Osvedčuje sa ako dobrá voľba pri tvorbe riešení z prostredia IoT.

Iné možnosti zahŕňajú Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), WebSocket, Modbus a iné.

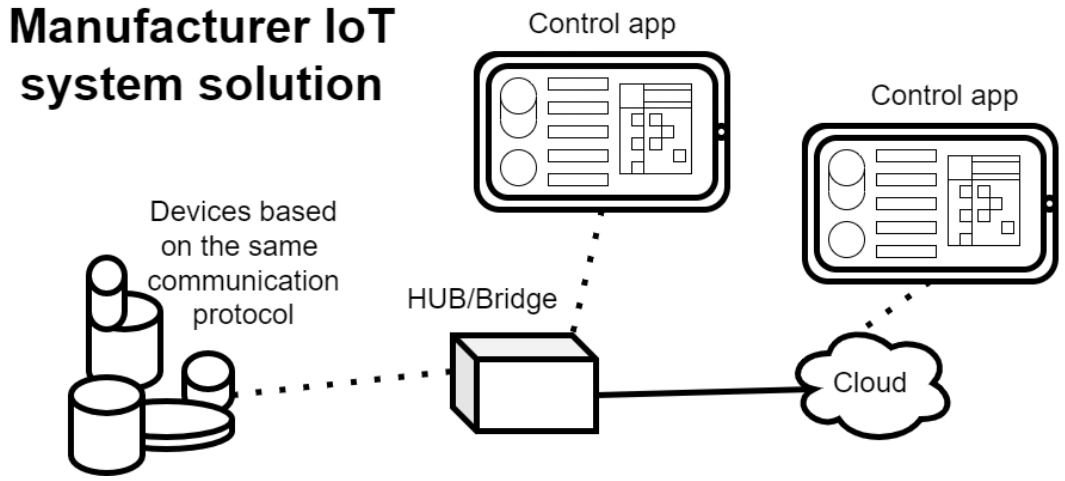
Aplikačné riešenia

Doteraz boli spomenuté časti systému, s ktorými väčšinou človek v každodennej prevádzke neprichádzal do kontaktu. Tie popisovali zber a presun dát. Posledným

dielom je samotná interná logika operujúca nad týmito dátami. V prípade inteligentnej domácnosti ide o ovládacie rozhranie medzi IoT systémom, jeho koncovými zariadeniami a človekom. Z tejto stránky sa dajú identifikovať štyri prístupy nasadzované v praxi. Usporiadané sú z pohľadu náročnosti ovládania. Prvým je riešenie s využitím inteligentnej aplikácie ponúkané výrobcami inteligentných zariadení. To sa čiastočne prelína s druhým, čím sú "cloudové riešenia" najmä od veľkých spoločností ako Amazon alebo Google. Ďalšie sú zmienené riešenia založené na koncepte jednotnej centrály (HUB), ktorá združuje mnohé z funkcionalít predchádzajúcich systémov. Nespokojnosť s predchádzajúcimi možnosťami a uzavretosť riešení dala podnet nadšencom a "open source" komunite, aby vytvorili vlastné platformy na adresáciu nedostatkov v komerčných systémoch. Z pohľadu užívateľa je aplikácia od výrobcu priamočiaré ovládanie pomocou predefinovaj aplikácie. Cloudové platformy sú tak isto relatívne jednoduché na integráciu, ale ponúkajú omnoho väčšie spektrum funkcionality. Pri použití domácej centrály (HUB) sa náročnosť na nastavenie a spojazdnenie systému začína komplikovať, najmä pri systémoch, kedy sú detaily ovládania prenechané na nastavenie užívateľom. Posledná možnosť vyhovuje viac odbornej verejnosti, domácim kutilom a nadšencom, ktorí sa do problematiky ovládania domáceho prostredia vyznajú alebo je to oblasť ich záujmu.

Aplikácia výrobcu

Prvým nápadom na vybavenie domácnosti inteligentnými výrobkami je navštívenie stránok bežných predajcov elektroniky (napríklad Alza, CZC, Datart a pod.). Tí ponúkajú pomerne širokú škálu výrobkov založených na rôznych komunikačných protokoloch.



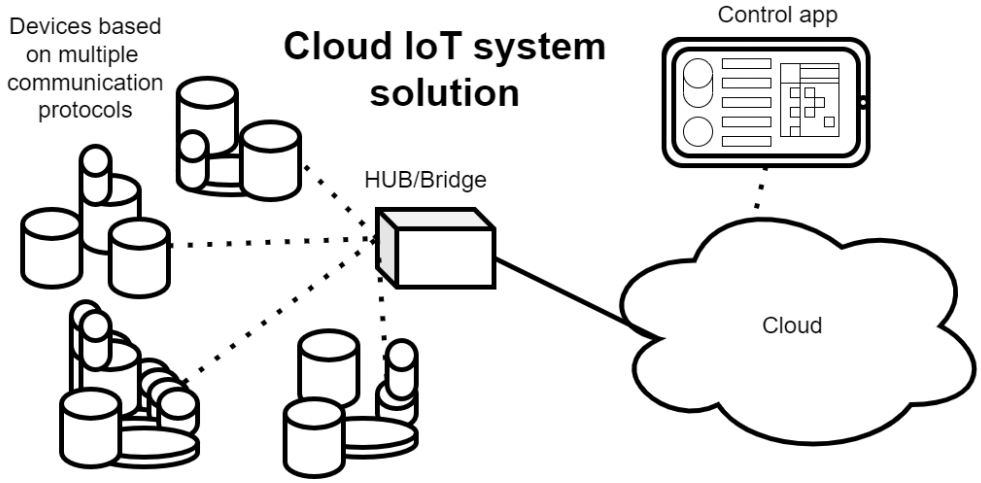
Obr. 2.5: Diagram IoT systému ovládaného cez aplikáciu výrobcu zariadení [8].

Záleží od zariadenia, ale je bežnou praxou, že výrobca poskytuje k vlastným výrobkom štandardne ovládaciu aplikáciu, dostupnú v obchodoch Androidu alebu iOS. S touto aplikáciou sa komunikuje cez HTTP protokol a je potrebný prístup k internetu. Z toho dôvodu je nutné ešte okrem zariadenia samotného kúpiť aj "bránu", ktorá je pripojená do domácej WiFi siete/smerovača. Brána prepája komunikačný protokol zariadenia s lokálnou sieťou. Následne záleží od postupu výrobcu. Brána sa bežne registruje na serveroch výrobcu. Z aplikácie sa pomocou užívateľského účtu

pripojí na tieto servery. Niektoré brány umožňujú pripojenie aj výrobkov od iných výrobcov v rovnakom komunikačnom protokole, ale táto možnosť nie je garantovaná.

Cloudové riešenia

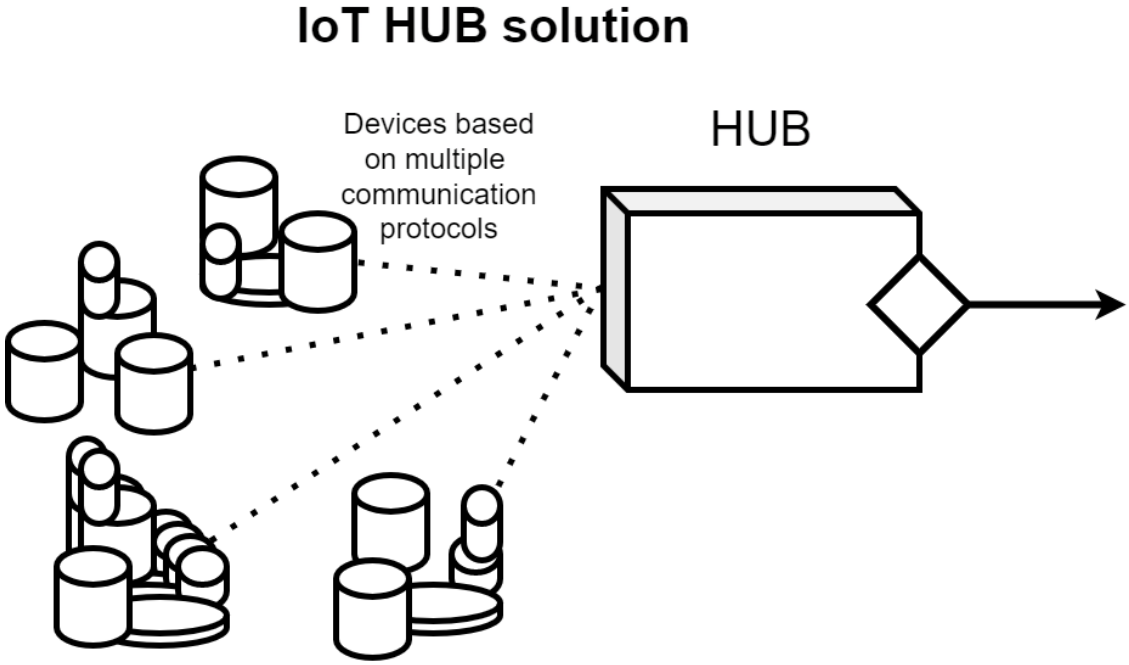
Ďalšia možnosť je využitie cloudovej platformy. Na inteligentnú domácnosť sa zamerali aj veľké spoločnosti ako Microsoft, Amazon, Google alebo Apple. Sústredili sa na softvérovú stránku systémov. Na rozdiel od homogénneho prístupu výrobcu na integráciu len jediného komunikačného protokolu, platformy podporujú heterogénnu integráciu mnohých protokolov. Je to možné predovšetkým vďaka veľkým vývojovým tímom, ktoré udržujú podporu projektov a komplexného softvéru ako Google home alebo Amazon Alexa. Výrobcovia zariadení túto integráciu prijali a vytvárajú svoje výrobky kompatibilné s týmito platformami. Výrobca zariadení vďaka tomu môže svoje zdroje zamerať viac na vývoj v oblasti hardvéru. V prípade cloudu sú možnosti samotnej aplikácie rozmanité. Po prepojení brány s internetom je pomocou aplikácie v telefóne alebo na počítači možné riadiť celú domácnosť. U riešení ako Amazon Alexa je dostupný aj samotný Amazon Alexa hub, ktorý slúži ako brána a samotné ovládanie pomocou hlasových povelov. Medzi dostupné platformy patria: Amazon Alexa, Google Home, AppleHomeKit a iné.



Obr. 2.6: Diagram IoT systému ovládaného cez cloudovú platformu [8].

Domáca centrála (HUB)

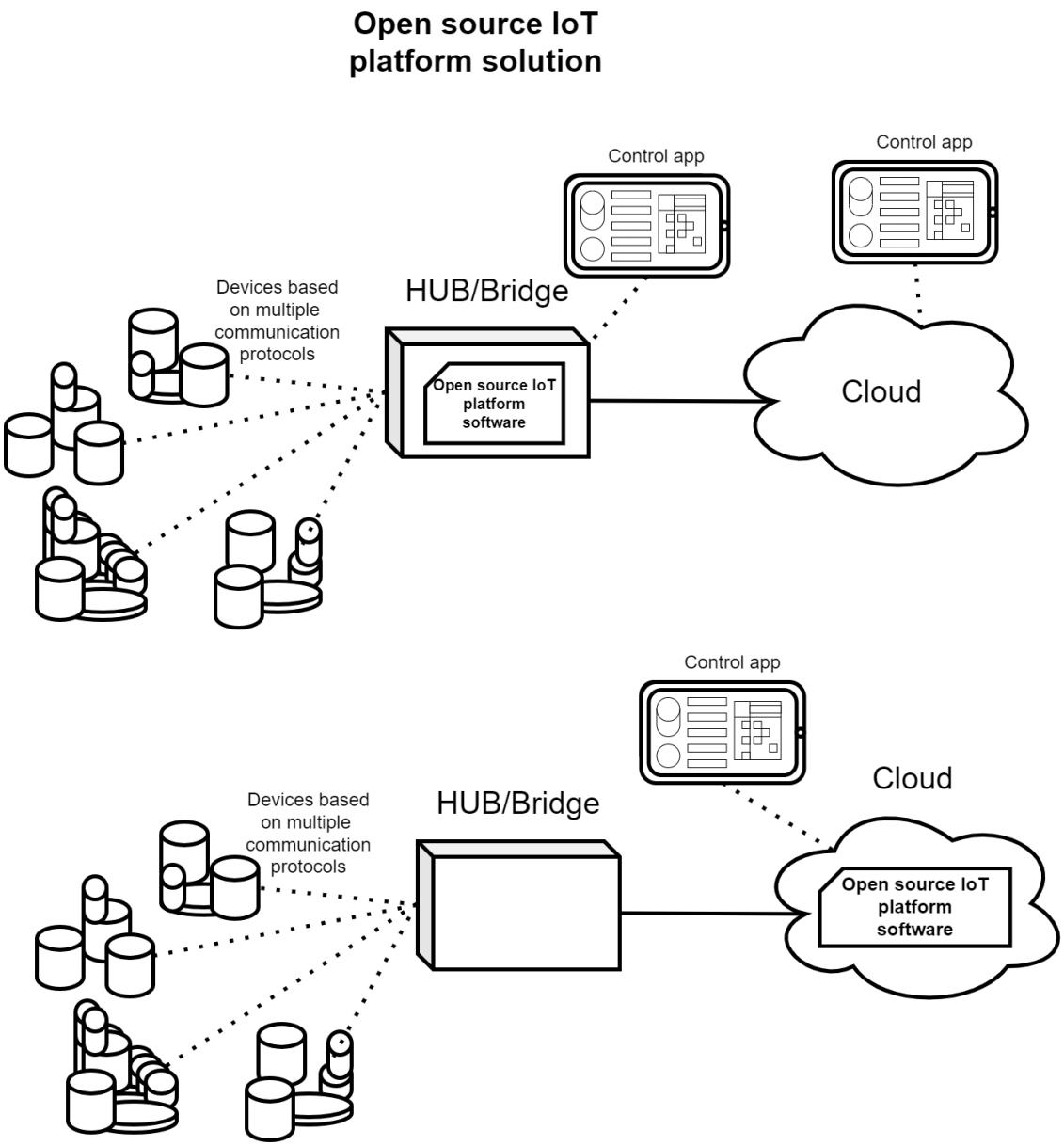
Práve na problematiku heterogenity sa zamerali spoločnosti, ktoré ponúkajú riešenie samostatnej domácej centrály (HUB). Tieto zariadenia dokážu integrovať množstvo komunikačných protokolov. Ponúkané sú možnosti, kedy je podporované pripojenie na cloudové platformy (ovládanie prebieha ako v predchádzajúcej možnosti) alebo sú dostupné aj centrály, ktoré nepotrebujú pripojenie na internet. V takom prípade je softvérová časť IoT systému obsiahnutá v centrále a ovládacie aplikácie s ňou interagujú z prostredia domácej siete. Bežne je týmto prístupom zabezpečená veľmi detailná dostupnosť dát a vysoká granularita ovládacích procesov. Pre ľudí, ktorí sú takpovediac laikmi v tejto oblasti, predstavuje takýto postup skôr príliš náročné nastavenie ovládania a zadovážia si radšej ľahšiu z vyššie uvedených možností.



Obr. 2.7: Diagram IoT systému ovládaného cez jednotnú centrálu [8].

Open source platformy

Posledná možnosť je smerovaná skôr na nadšencov a domácich kutilov. Ide o softvérové riešenie inteligentnej IoT domácnosti, ktoré ponúka vysokú granularitu ovládania. Na zostavenie a nasadenie takejto platformy je potrebný celkový prehľad v IoT technológiách. Častokrát si vyžaduje nasadenie rôznych mikrokontrolérov a znalostí programovania alebo písania ovládacích "scriptov". Samozrejme sú aj možnosti priaznivejšie pre laických užívateľov ako Home Automation, ale ovládacie prostredie môže byť náročné na obsluhu vďaka veľkému počtu ovládacích prvkov a zložitému systému nastavenia. Na druhú stranu je tento prístup najversatilnejší zo všetkých spomenutých. Medzi dostupné možnosti patrí OpenHAB, ThingsBoard, ThingsSpeak, Thinger, ZHA (Zigbee Home Automation), HA (Home Assistant), Domoticz a iné.



Obr. 2.8: Diagram IoT systému ovládaného cez jednotnú Open source platformu [8].

2.2 Ľudia s obmedzenou schopnosťou pohybu

Záujem práce je vytvoriť dostatočne univerzálne zariadenie, ktoré spája inteligentné prostredie a ľudí s obmedzením pohybu. Fyzicky obmedzení ľudia stále preferujú mať pocit osobnej nezávislosti a začlenenia medzi fyzicky zdatných. Toho je možné čiastočne docieliť integráciou IoT zariadení v ich domácnosti. Cieľom je preto poskytnúť most, ktorý by odstránil ťažkosti v ovládaní niektorých spotrebičov a zariadení pre znevýhodneného človeka.

Mnoho ľudí je postihnutých nejakým pohybovým obmedzením. U niektorých už od narodenia, napríklad formou vrodených vád a porúch. Bohužiaľ však narastá počet ľudí, ktorí sú v pohybe obmedzení po rôznych úrazoch, častokrát po havárii motorových vozidiel, mozgových príhodách, ale ešte častejšie pri športoch alebo zábavách ako sú lyžovanie, rôzne lety a zoskoky. Vtedy dochádza hlavne k nená-

vratnému poraneniu dolných (nôh) a horných (rúk) končatín. Takéto osoby nemajú fyzické schopnosti na vykonávanie aktivít nezávisle. Do tejto skupiny môžme zaradiť aj seniorov, kde sa pohybové obmedzenia rozvíjajú s pribúdajúcim vekom.

Práca je zameraná primárne na osoby s postihnutím horných končatín, jemnej motoriky a pohybového aparátu, ktorými zdraví ľudia bežne ovládajú domáce spotrebiče od pása vyššie.

2.2.1 Domácnosť

Domáce prostredie týchto osôb sa nemusí nijakým zásadným spôsobom líšiť od prostredia zdravých ľudí. Tí s poranením nôh zostávajú na invalidnom vozíku, ale sú schopní manipulovať s bežnými predmetmi (napríklad ovládač TV). Bohužiaľ, nie sú jednoducho schopní otvoriť vyvýšené vetracie okno alebo zapnúť/vypnúť vyššie položené zariadenie. Takýto ľudia však pomocou vhodne vytvorených mechanických pomôcok a prípadne bežných aplikácií môžu svoje okolie stále celkom dostatočne ovládať.

Ľudia s poranením rúk sú v omnoho väčšej nevýhode. Jemná motorika rúk je pre človeka veľmi dôležitá v každodennom živote. Väčšina bežne dostupných aplikácií pre ovládanie okolitého prostredia je vytvorená pre ľudí disponujúcich dostatočne jemnou motorikou rúk. Pre tento typ ľudí sú už (v podstate úplne) nepoužiteľné obvyklé zariadenia (napríklad ovládač TV) a ani komerčne dostupné aplikácie pre ovládanie okolia (domáce riadiace systémy). V tomto prípade je nutné veľmi často vytvárať úplne na mieru špecifické mechanické ovládače (veľké a robustné tlačidlá) alebo špeciálne tabletové aplikácie (veľké plošné tlačidlá eliminujúce rôzne tiky).

2.2.2 Potreby

Vzorovým príkladom môže byť človek na invalidnom vozíku s pevne prichyteným tabletom. Tento tablet by mal mať dostatočne veľký displej. Tablet je umiestnený veľmi blízko tela pre obmedzený dosah rúk, napríklad z dôvodu ich trvalého poranenia. Maximálna presnosť dotyku na obrazovke tabletu sa môže hodnotiť na celé centimetre. Taktiež hrá rolu ešte veľmi často opakovaný dotyk (pre tiky, záchvevy, ...). Na toto musí byť aplikácia dostatočne prispôsobená.

Ako druhý príklad možno uviesť osobu na invalidnom vozíku bez možnosti manipulácie rukou. V tomto prípade je tablet (pevne uchytený na vozíku) ovládaný pomocou ceruzky alebo ukazovadla držaného v ústach, prípadne pevne pripevneného k hlave osoby. Tu je presnosť dotyku na obrazovke tabletu ešte viac obmedzená. Takáto osoba môže svoje limitované okolie ovládať len, resp. iba pomocou špeciálne navrhnutej aplikácie. V tomto prípade je veľmi často potrebná ďalšia osoba zaisťujúca (väčšinou vzdialene) dozor. Tento dozor (napríklad vzdialene po telefónnom rozhovore s obyvateľom bytu) zabezpečí otvorenie vetracieho okna.

2.2.3 Nedostatky súčasných produktov

Pre takýchto ľudí sa žiadny zo štyroch spomenutých systémov nezdá vhodný. Aplikácie a systémy výrobcov sú ľahké na ovládanie alebo implementáciu. Avšak

sú uspôsobené pre zdravotne zdatných ľudí a nepodporujú dostatočnú modularitu pre úpravu na nasadenie v špeciálnych domácnostiach. Taktiež pri využití mnohých komunikačných protokolov je nutné si zadovážiť hardvér (bránu alebo most) pre každý protokol osobitne. Nehovoriac o tom, že každý výrobca odporúča použiť jeho vlastný hardvér.

Cloudové služby sa z tohoto pohľadu zdajú modulárnejšie. Nadbytočný hardvér sa dá častokrát nahradiť len jedinou bránou. Nevýhodou je však príliš komplexné ovládanie rozdielnych koncových zariadení spolu s komplexnými ovládacími prvkami v dostupných aplikáciach.

Domáce centrály od rôznych výrobcov sa zdajú ako stredná cesta medzi už spomenutým. Ponúkajú interoperabilitu, ale opäť narážajú na neprispôsobivé aplikácie. K tomu sa taktiež pripája trend, z ktorého sa zdá, že s príchodom cloudových služieb je pre výrobcu výhodnejšie zameranie na hardvér koncových zariadení a vývoj ovládacieho softvéru prenecháva veľkým cloudovým spoločnostiam.

Z hľadiska prístupu je posledná možnosť "open source" platformy najpriaznivejšia čo do použitého hardvéru, aj do prispôsobenia ovládania. Nevýhodu však tvorí potreba mať o systémoch prehľad a dostatočné znalosti v rôznych technológiach. Taktiež nie je vždy dokumentácia dostatočne podrobná, nakoľko ju tvoria zväčša nadšenci. Rovnako je otázna aj dlhodobá podpora takýchto služieb, pokiaľ ich nezastrešuje spoločnosť, ale komunita nadšencov.

Ako bolo už spomenuté, u človeka na vozíku s poškodením horných končatín, by mal potrebný systém zabezpečovať jednoduché prispôsobivé aplikácie rôzneho druhu. Takéto aplikácie by sa ľahko upravovali a nezáviseli by od hardvéru, na ktorom bežia alebo ktorý ovládajú. Práve týmto sa zaoberá návrh centrály predstavený v nadchádzajúcej časti. Centrála pôsobí ako abstrakcia nad koncovými entitami a poskytuje jednoduché ovládacie možnosti. Na týchto možnostiach sa dajú budovať jednoduché aplikácie, ľahko prispôsobiteľné pre špecifické potreby ľudí s obmedzeným pohybom.

Kapitola 3

Návrh

Na základe spomenutých nedostatkov v poslednom odseku predchádzajúcej kapitoly táto práca navrhuje trochu odlišné riešenie ovládania okolia pre ľudí s obmedzeným pohybom. Systém je teda navrhnutý s dôrazom na už vymenované prvky IoT systému. Presnejšie:

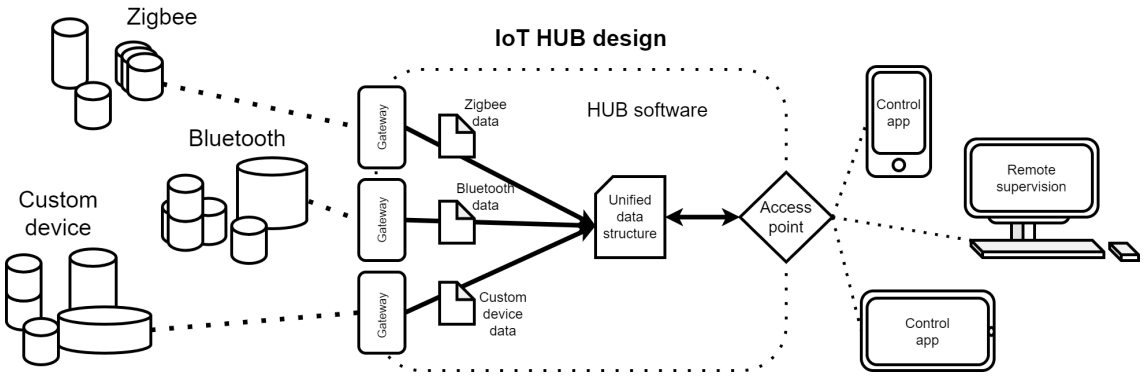
- Architektúra systému
- Koncové zariadenia
- Komunikačné technológie
- Prenosový protokol
- Aplikačné riešenie

Taktiež je z pohľadu prepojenia týchto prvkov nutné zahrnúť:

- Hardvér pre komunikáciu so zariadeniami a aplikáciami
- Hardvér pre výpočtové operácie ponúkaných služieb
- Softvér spravujúci internú logiku, spracovanie dát, poskytnutie služieb
- Dátové štruktúry predstavujúce zariadenia

3.1 Návrh centrály (HUB)

Centrála musí spĺňať niekoľko základných požiadaviek. Komunikuje s rôznymi koncovými zariadeniami pomocou rozdielnych komunikačných protokolov. Táto komunikácia prebieha cez hardvérovú bránu príslušnej technológie. Prijaté dáta závislé od komunikačnej technológie sú spracované do jednotnej dátovej štruktúry, ktorá predstavuje abstrakciu nad použitým protokolom. Spracovanie pracuje aj opačným smerom, kedy sa hodnoty z jednotnej dátovej štruktúry prevedú na potrebný protokol a odošlú cez bránu do cieľového zariadenia. Ďalej centrála poskytuje prístup k stavu a možnosť vykonávania riadiacich operácií nad týmito dátovými štruktúrami. Tento prístup je zabezpečený cez jednotné rozhranie. Ako bolo v úvode kapitoly zmienené, tieto schopnosti sú zabezpečené pomocou logických segmentov, ktorými je centrála tvorená. Návrhom jednotlivých segmentov sa zaoberá zvyšok tejto podkapitoly.



Obr. 3.1: Základná schéma centrály navrhnutej v tejto práci [8].

3.1.1 Architektúra systému

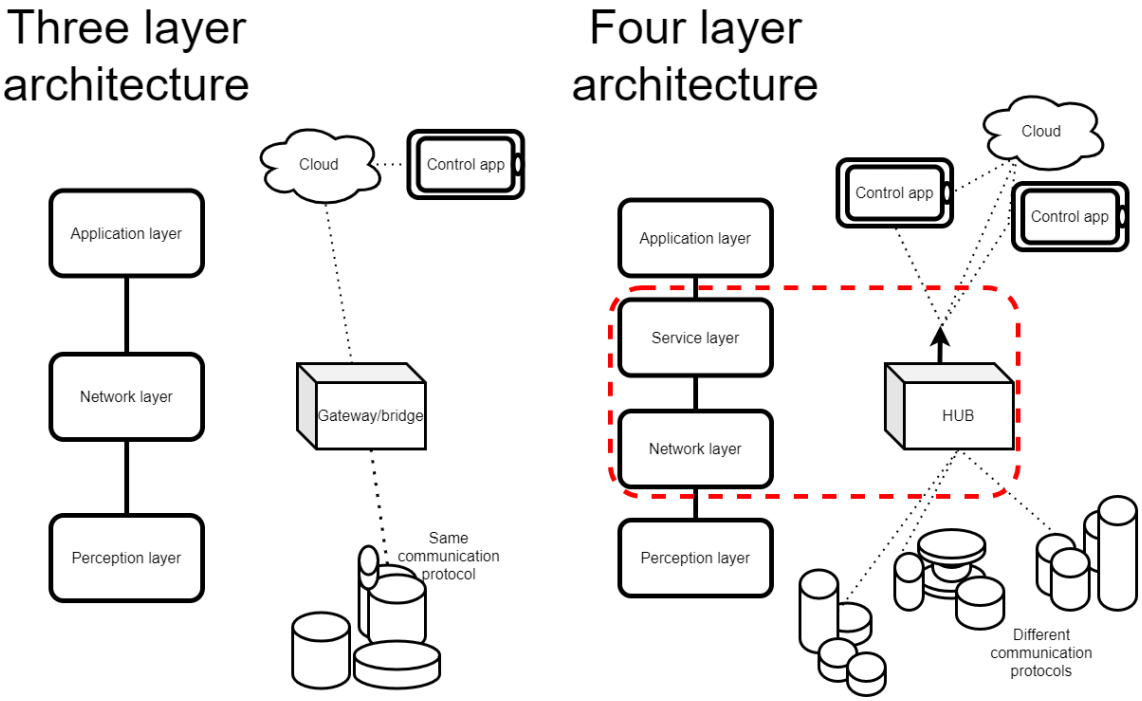
Použité ovládanie okolia sa bližšie nešpecifikuje na zahrnutie umelej inteligencie ani nie je základom pre budúce komerčné systémy určené širokej verejnosti. Systém by mal byť čo najjednoduchší, aby sa zamedzilo komplexnosti z už skôr predstavených riešení. Z toho dôvodu sa javia dobrými voľbami rozdelenia o troch alebo štyroch vrstvách. V prípade centrály sa ako najlepší spôsob javí použitie štyroch vrstiev. Tri sú vhodné pokiaľ by išlo len o použitie na báze brány (Gateway) alebo mostu (Bridge). Vtedy sa dáta zariadení nijak nespracujú a len sa posielajú do aplikačnej vrstvy, kde prichádza k spracovaniu. Na rozdiel od toho sa v našom prípade dáta primárne spracujú v centrále (HUB). Pod pojmom spracovanie rozumieme ich analýzu a transformáciu do interných logických štruktúr. V opačnom prípade (Gateway, Bridge) sa len prepošlú z jedného protokolu do druhého a nie sú s nimi vykonané operácie, ktoré by sa zaoberali nesenými informáciami. Interná logika stavu zariadení ostane uložená v centrále a aplikačnej vrstve je poskytnutý len ovládací bod. Cez tento bod sa následne dotazujú aplikácie na stav a riadenie prvkov. U troch vrstiev sa táto logika deje na aplikačnej vrstve. V prípade centrály to bude na takzvanej servisnej vrstve (Service layer).

Práca sa bude primárne špecializovať na návrh a realizáciu centrály. Z toho pohľadu najnižšia vrstva vnímania (Perception layer), na ktorej operujú IoT zariadenia, nebude ďalej rozoberaná. Rovnako najvyššia aplikačná vrstva (Application layer) pre riadiace aplikácie a vzdialený dohľad nebude implementovaná do hĺbky. Avšak bude využitá v testovacej aplikácií pre potreby overenia celkovej činnosti. Primárne zameranie bude na prostredné vrstvy sieťového spojenia (Network layer) a dostupnosti služieb (Service layer) cez vytvorený prístupový bod.

V sieťovej vrstve je okrem komunikačných protokolov zahrnutý aj softvér nutný na prijatie takýchto protokolov, ich dekódovanie a vyzdvihnutie dát. Tie sú následne posunuté vrstve služieb. Vrstva služieb primárne poskytuje jednotný prístupový bod pomocou prenosového protokolu na komunikáciu s aplikáciami. Tie operujú príkazmi nad dátovými štruktúrami. Softvér zabezpečuje transformáciu štruktúr na dáta pre jednotlivé komunikačné protokoly a ich odovzdanie sieťovej vrstve. Súčasťou centrály je taktiež aj hardvér, na ktorom sa potrebné procesy odohrávajú.

Práve na adresáciu heterogenity komunikačných protokolov by dátové štruktúry reprezentujúce jednotlivé zariadenia mali mať jednotnú formu. Na tejto abstrakcii

nad komunikačnou reprezentáciou sa ľahšie operuje aplikačným príkazom. S takýmto prístupom sú zariadenia rovnakej činnosti ovládané rovnakými príkazmi od koncového užívateľa a je úlohou centrály, aby zabezpečila transformáciu týchto príkazov a dát do potrebnej formy pre daný komunikačný protokol. Jednotná dátová štruktúra predstavuje súčasť softvéru vo vrstve služieb.



Obr. 3.2: Rozdiel v architektúre IoT systému s tromi a štyrmi vrstvami. Centrála (HUB), ktorou sa práca zaoberá, operuje na sieťovej vrstve (Network layer) a vrstve služieb (Service layer) [8].

3.1.2 Koncové zariadenia (Perception layer)

Pre ovládanie okolia v domácnosti a jej dohľad sú použité rôzne IoT výrobky. Takéto výrobky je možné zakúpiť v mnohých webových obchodoch alebo kamenných predajniach. Prvotným nápadom bolo informácie zariadení v jednom komunikačnom protokole transformovať vlastnou funkciou do spoločnej dátovej štruktúry. To sa ukázalo ako problematické. Ťažké je reprezentovať tieto zariadenia v jednotnej dátovej štruktúre, nakoľko po odchytení komunikácie zo Zigbee a Bluetooth sa zistilo, že dáta sa neodosielajú v jednotnej forme. Základná štruktúra je jednotná, podľa špecifikácie protokolu, ale výrobcovia si implementujú dátovú časť po svojom. Rovnako tak je rozdiel medzi technológiami. Tak isto pri pripojení vlastného zariadenia cez napríklad USB-HID nie je špecifikované, akú formu by odoslané dáta mali mať. Tento fakt bude musieť byť riešený podporou softvéru centrály pre rozličné zariadenia (rôzne komunikácie a rôzne protokoly). Aby zariadenie s centrálou naisto spolupracovalo, je nutné ho otestovať a špeciálne mu prispôsobiť softvér. Tento fakt je aj jedným zo základov, prečo doteraz vytvorené centrály nie sú z dlhodobého hľadiska schopné podporovať tak širokú škálu výrobkov IoT. Taktiež to vysvetľuje, prečo jediné riešenia, ktoré sú toho schopné (cloudové platformy veľkých spoločností, resp. “open source” platformy komunít nadšencov), obsahujú celé skupiny ľudí, zaoberajúcich sa vývojom a podporou platformového softvéru.

3.1.3 Komunikačné technológie (Network layer)

Centrála by mala podporovať pripojenie najpoužívanejších komunikačných protokolov. Ako bolo spomenuté v tretej kapitole, sú to predovšetkým bezdrôtové komunikačné protokoly na strednú vzdialenosť (WiFi, Bluetooth, BLE, Zigbee, Z-wave).

Drôtové nasadenie v domácnosti nie je úplne typické. Z pohľadu dostupných zariadení by však centrála mala obsahovať Ethernetovú prípojku (pre možnosť pripojenia do internetu). Tá býva v domácej infraštruktúre už častokrát zahrnutá a nie je nutná jej dodatočná inštalácia. Dobrým zástupcom vhodného príkladu amatérskeho využitia komponentov pre domáce riadenie je česká firma Papouch [20]. Ponúka vstupno-výstupné moduly pripojiteľné cez Ethernetovú linku. S nimi by bolo možné ovládať alebo spínať aj výkonové zariadenia a spotrebiče. Najzaujímavejší v tomto ohľade je ich rad zariadení Quido [21]. Tie poskytujú ovládanie dokonca nielen cez Ethernet , ale aj cez USB a sériové linky RS232 a RS485.

Vzhľadom k tomu, že sa pohybujeme v prostredí ľudí so špeciálnymi potrebami, by centrála mala podporovať aj pripojenie vlastných zariadení, vytvorených pre túto skupinu ľudí. Tie by sa mohli pripojiť rôznymi zbernicami (sériová linka, DALI, I2C, SPI a pod.). Nie je možné obsiahnuť všetky. Ako najvhodnejšie čo do možností a lokálnej vzdialenosti pripojenia sa javí použitie USB (Universal Serial Bus). Obsahuje ho takmer každé zariadenie (počítač). Býva nedeliteľnou súčasťou väčšiny mikrokontrolérov, ktoré by tvorili drvivú väčšinu špeciálnych zariadení na mieru a je všeobecne používaný vo výpočtovej technike.

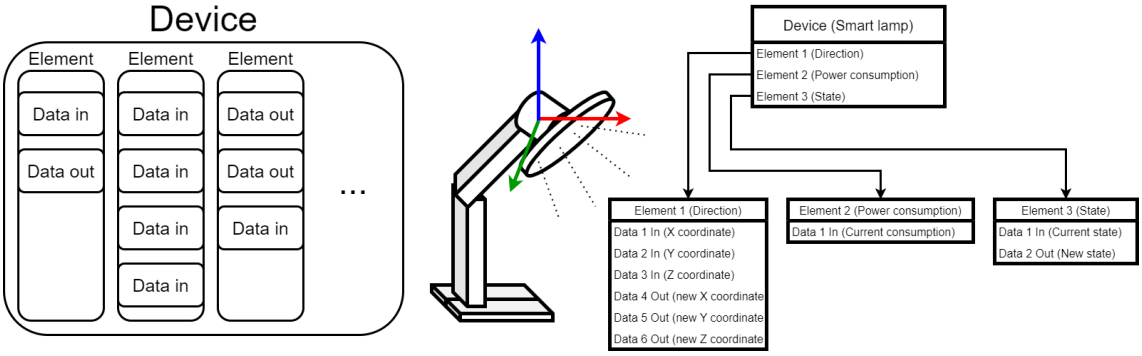
3.1.4 Hardvér a softvér sieťovej vrstvy (Network layer)

Aby centrála mohla komunikovať s komunikačnými protokolmi potrebuje mať pre každý jeden príslušné hardvérové rozhranie. Tieto takzvané brány (Gateway) sú najčastejšie antény, dekódovacie procesory, periférie prípojok ako USB a pod. K nim alebo v nich sú zabudované ovládacie programy (Driver), ktoré daný hardvér ovládajú a umožňujú odovzdávanie dát z formátu komunikačného protokolu (strana koncového zariadenia) do formátu zmysluplného pre spracovanie v centrále (spracovanie a ďalšie využitie dát). V prípade centrály navrhovanej v tejto práci sú to periférie, ktoré podporujú bezdrôtovú komunikáciu pre protokoly (WiFi, Bluetooth, BLE, Zigbee, Z-wave), USB periférie a periférie pre Ethernetovú sieťovú kartu s prevodníkom (LAN port). Po prijatí dát bránami sú z komunikačných protokolov informácie ďalej poskytnuté vrstve služieb na spracovanie. V opačnom prípade sú zase dáta od vrstvy služieb cez bránu vyslané koncovému zariadeniu po potrebnom komunikačnom protokole. Pri implementácii môže ísť o periférie už zabudované na doske mikrokontroléru (interné WiFi, BlueTooth, ..) alebo iného použitého počítača centrály. Rovnako môže ísť o rôzne adaptéry, USB dongle (hardvér, ktorý sa pripojí do iného zariadenia a zmení jeho funkcionalitu), prevodníky alebo samostatné mikrokontroléry.

3.1.5 Dátová štruktúra zariadení (Service layer)

Jednotná dátová štruktúra koncových zariadení predstavuje veľmi vhodnú abstrakciu nad samostatnými komunikačnými protokolmi. Mala by disponovať vše-

obecnou stavbou, aby do nej bolo možné zahrnúť širokú paletu zariadení. Jedným z dobrých príkladov je použitie DTDL (Digital Twins Definition Language) [22], ktorú implementuje spoločnosť Microsoft pri integrácii IoT výrobkov vo svojom "cloude". Tá zahŕňa podrobný popis každého zariadenia spolu s jeho možnosťami ovládania. Štruktúra predstavená v tejto práci nezachádza do tak hlbokej komplexnosti. Primárne sa skladá z troch častí. Každé IoT zariadenie predstavuje samostatnú entitu (Device). Táto entita v sebe obsahuje jeden alebo viac senzorov (Element). Senzory o sebe posielajú údaje alebo prijímajú príkazy. Preto sa ešte každý senzor skladá z dátových jednotiek (Data). Takáto jednotka môže buď slúžiť len na monitorovanie stavu nejakého parametra okolia, vtedy to je vstupná jednotka (Data In) alebo je ňou možné odosielať príkazy na vykonanie nejakej zmeny, vtedy to je výstupná jednotka (Data Out). Dátové jednotky sú buď len vstupné alebo výstupné, ale jeden senzor ich môže obsahovať viac.



Obr. 3.3: Príklad rozdelenia IoT zariadenia na logické časti u inteligentnej lampy [8].

Pre vysvetlenie delenia bude použitý príklad na Obr. 3.3 Inteligentná lampa má zdroj svetla natočený v smere súradníc x,y,z. O natočení si uchováva aktuálne súradnice a taktiež je jej možné nové súradnice natočenia poslať ako príkaz. Zaznamenáva svoju spotrebu energie vo wattoch a podporuje ovládanie pre zapnutie alebo vypnutie svetelného zdroja. Či je zapnutá sa uchováva ako parameter. Lampa je v spoločnej dátovej štruktúre reprezentovaná ako jedno zariadenie (Device). To má tri senzory (Elementy): natočenie (Direction), spotrebu energie (Power consumption) a stav či je zapnutá/vypnutá (State). Každý senzor má dátové jednotky (Data), ktoré je možné buď čítať alebo do nich zapisovať. V prípade smeru natočenia, sú evidované aktuálne koordináty natočenia v (Data 1 x, Data 2 y, Data 3 z) a je možné poslať nové koordináty natočenia v (Data 4 x, Data 5 y, Data 6 z). Spotrebu energie je možné len čítať zo senzoru (Element 2) a dátovej jednotky (Data 1). Taktiež sa uchováva, či je lampa zapnutá/vypnutá a zapísaním príkazu do (Data 2 out) v senzore (Element 3) je možné tento stav meniť.

Každá z logických častí (Device, Element, Data) okrem odkazov medzi sebou ešte obsahuje parametre potrebné pre zaistenie identifikácie zariadenia, spôsob komunikačného protokolu, samotné popísanie jednotlivých parametrov koncového bodu a ich dátové typy. Aké parametre zvoliť bude bližšie popísané v implementačnej časti práce, nakoľko sa to v návrhu nedá jednoznačne určiť. Operácie komunikácie nad dátovými jednotkami štruktúry sú dve. Buď je z dátovej jednotky (Data in) čítané alebo je do nej (Data out) zapisované. Pri pokuse o čítanie alebo zápis softvér centrály preloží žiadosť do komunikačného protokolu daného zariadenia a pošle

potrebnou bránou von. Následne prijaté potvrdenie alebo aktuálne dáta od zariadenia sa prepíšu naspäť do jednotnej štruktúry a aktualizácia sa pošle klientskym aplikáciam.

3.1.6 Hardvér a softvér vrstvy služieb (Service layer)

Samotné služby poskytované centrálou musia bežať na výpočtovej technike. Túto funkciu môže celkovo zastávať každý počítač s dostatočným výkonom. Všeobecne je v ovládaní okolia populárne použiť buď nejaký miniatúrny mikrokontrolér alebo malý (mikro, mini) počítač, nakoľko sú nároky na výpočtové zdroje relatívne nízke a takéto riešenia sú dostupné z finančnej stránky, aj zo stránky priaznivých rozmerov pre umiestnenie v domácnosti.

Mikrokontroléry ako STM, ATMEGA, ESP, Arduino a pod. sú cenovo dostupné možnosti. Problémom však býva zložité vytváranie sofvéru, aby bol efektívne spustiteľný na obmedzených zdrojoch, ktoré ponúkajú. Taktiež často chýba podpora vyšších ovládacích funkcií na riadenie samotného mikrokontroléru. Pre obmedzené schopnosti neobsahujú operačný systém a spravovanie zdrojov musí byť zabezpečené samotným vývojárom centrály.

Naproti tomu mikropočítač zastáva rolu základu pre hardvér a softvér centrály bez väčších problémov. Má dostatočný výkon a pamäť pre inštaláciu základného operačného systému, čo výrazným spôsobom urýchľuje vývoj. Tak isto sú už ponúkané s perifériami ako USB, WiFi, Ethernet alebo Bluetooth. Sú teda jasnou voľbou pre základ centrály na ovládanie okolia. K dispozícii existujú možnosti ako Asus Tinker Board, Raspberry PI, Banana PI a iné.

Samotný softvér už môže byť implementovaný rôznymi spôsobmi. Použiteľné sú jazyky ako C/C++, C#, Java, Python, Javascript a pod. Hlavným cieľom je spracovanie dát prijatých od koncových zariadení, ich uloženie a dostupnosť, správa klientskeho pripojenia a poskytovanie jednotného prístupového bodu.

3.1.7 Prenosový protokol (Service layer)

Táto časť centrály je najdôležitejšou súčasťou v tvorbe jednotného prístupového bodu pre riadiace a dohliadacie aplikácie. K dispozícií je veľké množstvo možností (HTTP, MQTT, CoAP a pod.). Pre jednotnú implementáciu je vhodné vybrať len jednu. Proces využitia by vyzeral asi takto: K centrále by sa pripojila riadiaca aplikácia na tablete (tablet môže byť pripevnený napríklad na invalidnom vozíčku) alebo aplikácia na telefóne. Z internetu by sa k centrále pripojila aj jedna alebo viac dohliadacích aplikácií. Každé spojenie je nutné dlhodobo udržiavať. Po týchto spojeniach by následne prúdila dátová komunikácia.

Z hľadiska podpory v oblasti IoT a internetu sú dve možnosti HTTP alebo MQTT. Ostatné protokoly sú využívané skôr v špecifických situáciach a nemajú tak rozsiahlu základňu používateľov. HTTP (Hypertext Transfer Protocol) je najpoužívanejší protokol internetu. Je dobre podporovaný a ľahko implementovateľný pomocou dostupných knižníc. Ďalšou voľbou by mohol byť novší MQTT (Message

Queueing Telemetry Transport). Vytvorený pre potreby IoT. HTTP pri poslaní jednej správy vyžaduje dodatočné poslanie nadmerného počtu dát, ktoré nenesú užitočnú informáciu z hľadiska aplikácie. U MQTT je vymenená veľká časť neužitočných dát pri vytváraní prvotného spojenia, ale následný prenos veľkého množstva dátových správ už nevyžaduje podstatné množstvo neužitočných dát oproti prenosu cez HTTP. Takýto prístup je najmä pre využitie pri IoT zariadeniach, ktoré posielajú malé správy v krátkych intervaloch pravidelne počas dňa.

V bežnom použití však je najvýhodnejší HTTP. Má široký rozsah využitia v mnohých odvetviach. Aplikácie, či už webové, mobilné alebo na počítačoch týmto protokolom už štandardne medzi sebou komunikujú. Je dlhodobo zaužívaný a má veľkú podporu medzi množstvom vývojárov a projektov. Preto sa javí ako prvá voľba, ktorá je z dlhodobého hľadiska dobre podporovaná.

3.1.8 Aplikačné riešenie (Application layer)

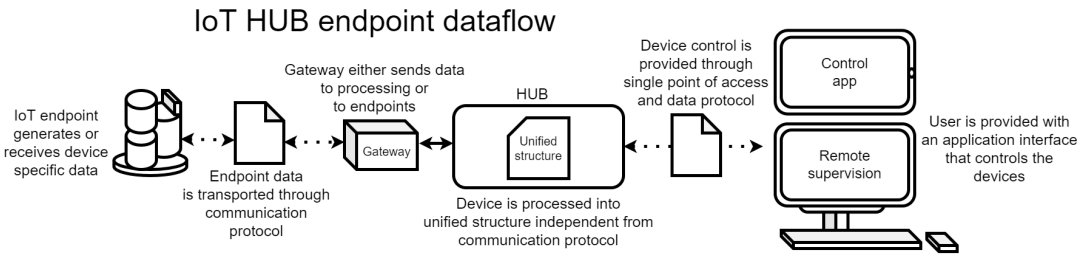
Ovládanie pre osoby s obmedzeným pohybom by malo byť predovšetkým jednoduché. Pri problémoch v oblasti rúk je dôležité, aby ovládacie prvky mali dostatočnú veľkosť. Preferujú sa veľké ovládacie plochy tabletov oproti mobilným telefónom. Taktiež by prvky nemali byť usporiadané príliš komplexným spôsobom, riadili sa intuitívne a priamočiaro.

Dohliadacia aplikácia (pre bežný dohliadací personál) by na druhej strane mala ponúkať širšiu funkcionalitu. Okrem možnosti riadenia by mala ešte obsahovať možnosť monitorovania prostredia. Tieto informácie z okolia môžu byť pre obmedzenú osobu v domácnosti zbytočnými, ale pre dohliadacieho člena rodiny alebo opatrovateľa sú však dôležité.

3.1.9 Zhrnutie návrhu centrály

Dobrým spôsobom, ako si predstaviť celkový návrh centrály pre ovládanie okolia, je pozrieť sa na tok dát z koncového zariadenia do ovládacej aplikácie. Na začiatku koncové zariadenie vygeneruje o sebe alebo svojom stave dáta. Tie následne komunikačným protokolom (napríklad bezdrôtovo cez Bluetooth) pošle do éteru. Túto správu zachytí Bluetooth brána (anténa hardvéru) umiestnená na centrále. Brána ich odovzdá softvéru centrály na spracovanie. Softvér ich podľa použitého komunikačného protokolu transformuje do jednotnej dátovej štruktúry. Tá je udržovaná a aktualizovaná v pamäti centrály. V prípade vyžiadania informácií o zariadení je dátová štruktúra použitá ako predstaviteľ zariadenia. Informácie o štruktúre sú z centrály vyslané pomocou dátového protokolu do príslušnej aplikácie.

Pokiaľ sa aplikácia rozhodne vyslať zariadeniu príkaz, tak pomocou dátového protokolu sa spojí s centrálou a odovzdá jej informácie o ovládanom zariadení. Softvér centrály správu spracuje. Vyhľadá si dátovú štruktúru, ktorá dané zariadenie predstavuje. Pomocou parametrov v štruktúre vytvorí správu špecifickú pre komunikačný protokol daného zariadenia a pošle ju potrebnej bráne. Brána túto správu následne vyšle (napríklad bezdrôtovo cez Bluetooth) do éteru. Adresované zariadenie správu zachytí a vykoná v nej zadaný príkaz.



Obr. 3.4: Návrh logických častí centrály z pohľadu prenosu dát [8].

3.2 Návrh vlastného zariadenia

Nakoľko veľa z ľudí, ktorí trpia obmedzením pohybu potrebuje špecifické zariadenia, by centrála mala podporovať aj pripojenie vlastného výrobku. Takýto výrobok môže byť na mieru vytvorený pre znevýhodneného človeka, na ovládanie alebo monitorovanie podľa jeho vlastných potrieb.

Ako najdôležitejšia časť sa javí komunikačný protokol, po ktorom by zariadenie s centrálou komunikovalo. Pri použití jednej z konvenčných metód ako WiFi, Zigbee, Bluetooth by integrácia takéhoto zariadenia nemala byť problémová. Avšak väčšina týchto výrobkov je takpovediac "vytvorená na kolene" z rôznych mikrokontrolérov, častokrát bez integrácie jednej z vyššie uvedených technológií. Jediná periféria, ktorú drvivá väčšina z nich obsahuje, je USB. Tá sa javí ako všestranný spôsob, akým by tieto zariadenia bolo možné k centrále pripojiť.

Najvhodnejšie protokoly, ktoré sa dajú využiť sú HID (Human Interface Device) a virtuálny COM port. Tie popisujú hlavne prenos dát. Samotná štruktúra odovzdávaných dát je otázkou. Závisí od implementácie samotnej centrály a koncových zariadení, s ktorými komunikuje.

HID (Human Interface Devices)

Ide o protokol pre triedu zariadení s generickým USB podporujúci klávesnice, myšky, herné ovládače a pod. Pred nástupom HID mohli zariadenia použiť len striktne definované protokoly pre myšky a klávesnice. Nové zariadenia vyžadovali buď špecializované ovládače (Driver) alebo "overloading" už existujúcich dát. Na druhú stranu HID predstavuje štandardizované a jednoducho programovateľné rozhranie pre takéto zariadenia, bez nutnosti vývoja nového ovládacieho softvéru [23]. Základom je existencia "reports" a "report descriptors". Reporty sú samotné dáta predávané medzi zariadením a klientom. Report descriptor popisuje formát a význam týchto dát.

Virtuálny COM port

COM port (alebo "Serial port") je vstupne-výstupné rozhranie podporujúce "serial" komunikáciu zariadenia a počítača. Moderné počítače perifériami COM port už nedisponujú, nakoľko sériovú linku vystriedalo USB. Komunikácia je zabezpečená v momente iba jedným "bitom", z toho dôvodu je sériová. Avšak používa sa "USB-Serial port" adaptér, ktorý dokáže zabezpečiť COM port aj na zariadeniach

bez fyzickej periférie. Adaptér má formu softvérového ovládača a vytvára takzvaný virtuálny COM port. Ten sa z logického pohľadu javí, akoby bola pripojená fyzická periféria sériového portu. USB oproti sériovej linke dokáže mať rýchlejší prenos, a preto ju nahradil [24].

Kapitola 4

Implementácia

V tejto kapitole bude predstavená implementácia centrály (IoT HUB) navrhovanej v Kapitole 4. Je dobrou otázkou, ako takúto implementáciu začať. Najprv sa má rozvíjať najzložitejší prvok celého systému alebo naopak najjednoduchší? Taktiež je možné začať prvkami uprostred a postupovať smerom von alebo opačne z okrajov do centra. Najrozumnejšími sa v tomto prípade zdali dve možnosti:

Prvá postupuje od najvyššieho logického prvku na najvyššej logickej vrstve smerom k najnižšiemu. V tomto prípade by to bolo implementáciou jednotného prístupového bodu pre aplikácie v internom softvére. Odtiaľ by sa postupovalo až po prístupové brány komunikačných protokolov.

Druhý postupuje analogicky opačným smerom. Začne sa od hardvéru prístupových brán, kedy rozvoj smeruje nahor cez softvér a končí u prístupového bodu aplikácií. V práci bol práve tento druhý postup zvolený. Hlavným dôvodom bolo obmedzenie z hľadiska dostupných zariadení, na ktorých bude predmetný koncept testovaný. K dispozícii boli tri zariadenia založené na komunikačnom protokole Zigbee a jedno špeciálne vytvorené na komunikáciu pomocou USB-VCP port (prípadne USB-HID).

Implementácia začne od použitej brány. Následne bude predstavený softvér centrály. Ten je založený na jazyku C# a platforme .NET Core. Tie boli zvolené vďaka svojej prenositeľnosti medzi zariadeniami pracujúcich na rôznych operačných systémoch. Sú dlhodobo podporované, vyvíjané a spravované spoločnosťou Microsoft. Softvér bude primárne vyvíjaný na počítači s operačným systémom Windows pre jednoduché ladenie a až následne sa prenesie na Raspberry PI (uvažované zariadenie umiestnené v domácom prostredí).

4.1 Použité koncové zariadenia

Pre zostrojenie centrály boli k dispozícii tri IoT zariadenia. Všetky založené na komunikačnom protokole Zigbee. Dokopy simulovali základné vstupno-výstupné funkcie. Vybrané boli tak, aby reprezentovali niektoré charakteristické typy periférií. Ideálne: sledovací kontakt (dvere, okná, skrinky a pod.), ovládacie zariadenie (zásuvka a pod.) a privolanie pomoci ("help" tlačidlo a pod.). Prvé vykonáva funkciu senzoru pre kontakt. Druhé je spínateľná zásuvka. Tretie pracuje ako (najčastejšie "help") tlačidlo, ktoré vysiela informácie o svojom stlačení.

BlitzWolf® BW-IS2 ZigBee Contact Sensor

Kontaktný senzor použiteľný medzi iným napríklad na dvere alebo okná. V tejto práci predstavuje typ vstupného zariadenia. Menšia časť obsahuje magnet. Väčšia časť predstavuje riadiacu/vysielaciu jednotku. Po priblížení alebo vzdialení magnetu, senzor hlási aktuálny stav kontaktu. Predajca uvádza, že zariadenie musí byť použité s ich Zigbee bránou BW-IS 1 Gateway. Následné ovládanie je možné pomocou ich aplikácie BlitzWolf app alebo Alexa Google assistant [25].

Obr. 4.1: Ilustračné foto BlitzWolf BW-IS2 ZigBee Contact sensor [26].

Immax Neo Smart Plug (07048L)

Druhé zariadenie je IoT zásuvka. Táto zásuvka umožňuje spínanie prúdu buď pomocou tlačidla na svojom okraji alebo pomocou príkazov odoslaných cez protokol Zigbee. V práci predstavuje typ vstupno-výstupného zariadenia. Je schopné hlásiť stav či je zapnutá/vypnutá alebo umožňuje zopnutie na diaľku. Bohužiaľ výrobca už tento model ďalej na trhu neponúka [27] a v súčasnej dobe je nahradený novou verziou. V dostupnom návode na použitie sa uvádza podpora pre aplikáciu výrobcu, Amazon Alexa, Google Home, AppleHomeKit [28]. Na svojich stránkach výrobca prepája Zigbee zariadenia s vlastným mostom Immax Neo Bridge [29].

Obr. 4.2: Ilustračné foto Immax Neo Smart Plug (07048L) [30].

Aqara Wireless Mini Switch

Posledné zariadenie predstavuje tlačidlo. Hlási správy či bolo stlačené "raz", "dvakrát", je "stlačené dlhodobo" a následne "bolo uvoľnené". Typom sa v práci radí k výstupným zariadeniam. Je možné ho použiť napríklad ako zvonček alebo núdzové tlačidlo. Ide teda o vstupné zariadenie. Výrobca tiež uvádza, že je potrebná Aqara HUB pre operáciu. Funkčné s Apple Home app a Aqara Home app [31].

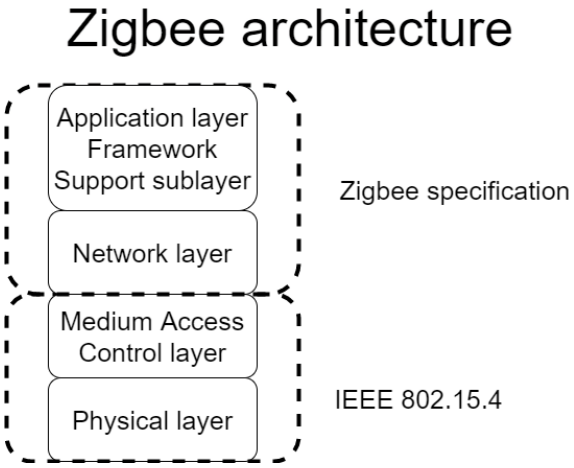
Obr. 4.3: Ilustračné foto Aqara Wireless Mini Switch [32].

4.2 Zigbee

Použité koncové zariadenia sú schopné komunikácie po protokole Zigbee. Ten je založený na IEEE 802.15.4 štandarde. Navrhnutý pre komunikáciu na krátku vzdialenosť (zhruba do 100 metrov) s dôrazom na nízku energetickú náročnosť. To si ale

vyžiadalo, že má pomerne nízku prenosovú rýchlosť. Operuje prevažne na 2.4 GHz, ale podporuje aj 915 MHz a 868 MHz. Rýchlosť prenosu je do 250 kbps u 2.4 GHz, čo predstavuje relatívne malé veľkosti správ. "The Zigbee Alliance" je zodpovedná za vývoj, certifikáciu a propagáciu protokolu. Aliancia sa skladá z veľkých spoločností zaoberajúcich sa elektronikou.

Architektúra pozostáva zo štyroch vrstiev: fyzickej (Physical layer), prístup k médiu (Medium Access Control layer - MAC), sieťovej (Network layer) a aplikačnej (Application layer).

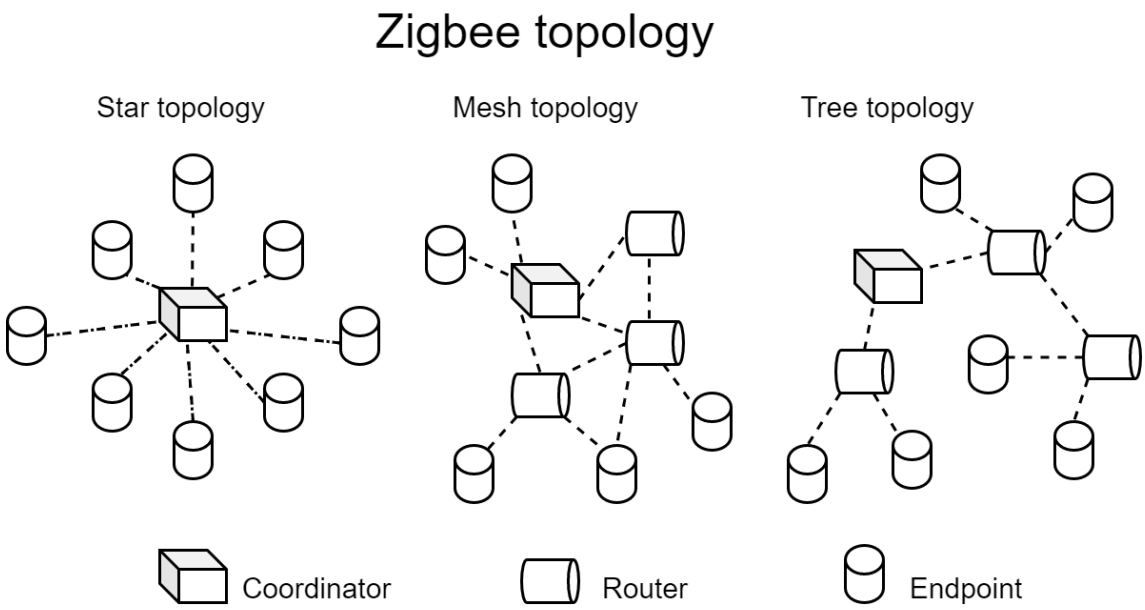


Obr. 4.4: Architektúra Zigbee [8].

Fyzická a MAC vrstva sú špecifikované v IEEE 802.15.4. Zaoberajú sa príjmom a posielaním dátových rámcov, kontrolou chýb dátových rámcov, hodnotením energie pri vysielaní, prístupom k bezdrôtovému médiu. Sieťová a aplikačná vrstva je definovaná v protokole Zigbee. Sieťová sa zaoberá vytvorením logickej siete, pridelením adries, pripájaním nových zariadení. Aplikačná poskytuje dve podvrstvy. Podporná (Support sublayer) udržuje tabuľky zariadení, ich adries a profilov, filtruje duplicitné rámce alebo rámce od neregistrovaných zariadení a profilov. Štruktúrna vrstva (Framework layer) predstavuje implementáciu koncových zariadení, ako sú "data requests" a "data confirmation". Túto vrstvu si definuje každý výrobca osobitne. Z pohľadu Zigbee zariadení existujú tri druhy: "Coordinator", "Router", "Endpoint". Koordinátor inicializuje, kontroluje a udržuje Zigbee PAN (Personal Area Network). Predstavuje bod, cez ktorý prúdi všetká dátová premávka.

Úlohou smerovača je preposielať správ z iných entít. Pripája sa na koordinátora alebo iný smerovač. Taktiež môže pôsobiť ako koncové zariadenie pre zber a prijímanie dát. Koncové zariadenie pracuje ako senzor alebo aktuátor. Pripája sa na koordinátor alebo smerovač. Zaoberá sa samotným zberom dát alebo vykonávaním príkazov.

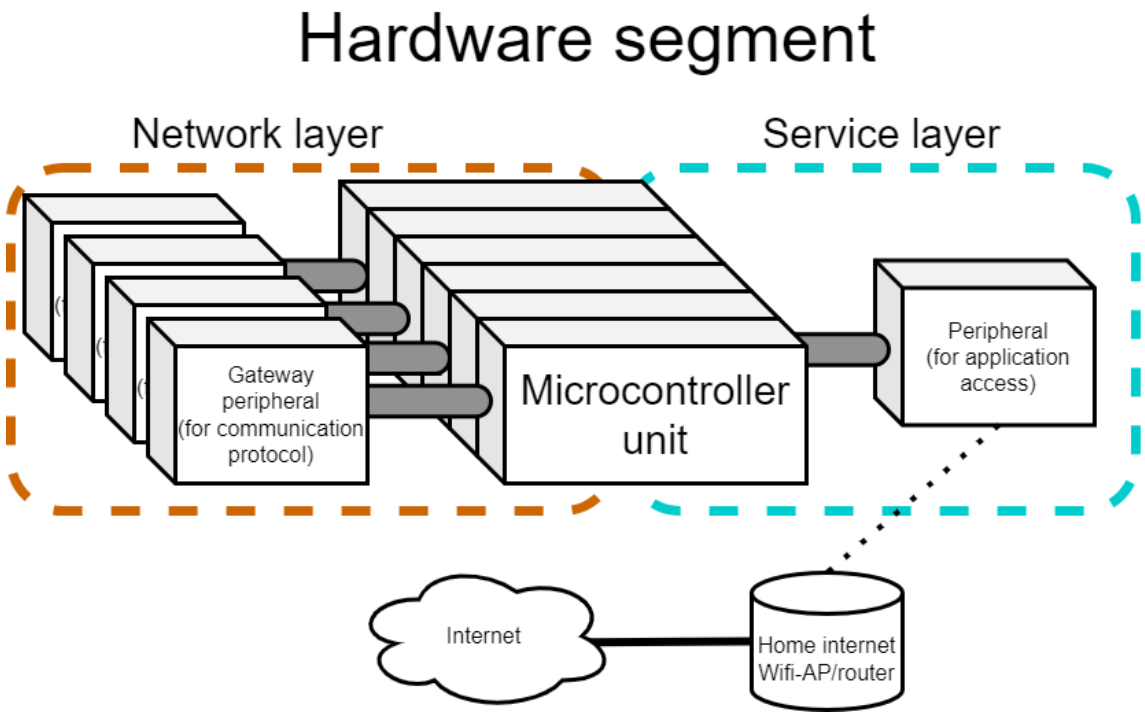
Z hľadiska sieťovej topológie podporuje Zigbee "star", "mesh" a "cluster tree". V hviezde sa všetky zariadenia pripoja na koordinátora. V mesh sa zariadenia môžu medzi sebou ľubovoľne prepojiť, pokiaľ im to vlastný typ dovoľuje. V takom prípade existuje mnoho smerovacích ciest. V cluster tree sa využijú hlavne smerovače k rozširovaniu stromu do menších podstromov [33], [34], [35].



Obr. 4.5: Rozdelenie sieťových Zigbee topológií [8].

4.3 Hardvér centrály

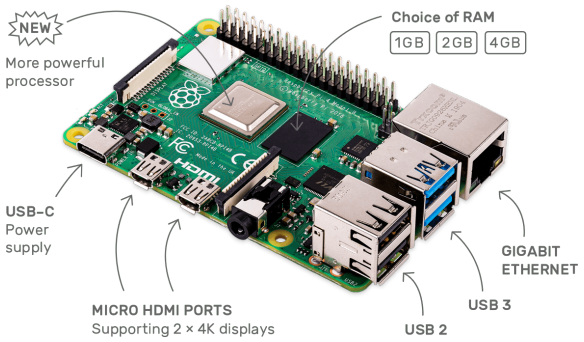
Hardvér vytváranej centrály sa v tejto práci primárne skladá z dvoch častí. Zigbee brána (anténa) na komunikáciu s koncovými zariadeniami a Raspberry Pi mikropočítač, na ktorom bude spustený ovládací softvér.



Obr. 4.6: Diagram hardvérovej časti centrály [8].

4.3.1 Raspberry Pi 4 Computer Model B

K vypracovaniu bol použitý mikropočítač Raspberry Pi 4. Model s pamäťou 4GB. Model obsahuje štandardnú WiFi a Ethernet periférium [36]. Ethernet bude použitý na pripojenie do domáceho smerovača pre prístup do lokálnej siete. Jednotka taktiež obsahuje USB periférie, na ktoré sa pripojí vlastné USB-VCP/USB-HID zariadenie a brána pre komunikačný protokol Zigbee.



Obr. 4.7: Raspberry Pi 4 Computer Model B ilustračné foto [37].

4.3.2 Zigbee koordinátor

Pre zachytávanie Zigbee komunikácie je potrebná príslušná anténa. K tomuto bolo využité zariadenie SONOFF ZigBee 3.0 USB Dongle Plus. Tento dongle je univerzálna Zigbee brána schopná zachytiť bezdrôtovú komunikáciu po Zigbee protokole. Založená je na čipe CC2652P od Texas Instruments. Dongle pracuje ako koordinátor Zigbee siete a je z výroby "pre-flashed" s "open source" softvérom, ktorý spravuje lokálneho Zigbee koordinátora [38]. Avšak je sním nutné použiť ďalší kus softvéru tretej strany nazvaný Zigbee2MQTT (prevod správ protokolu Zigbee do protokolu MQTT), ktorý taktiež musí byť spustený na Raspberry Pi.

Obr. 4.8: SONOFF ZigBee 3.0 USB Dongle Plus ilustračné foto [39].

Pôvodne sa uvažovalo o použití samotnej Zigbee antény, ale taký prístup vyžaduje komplexnú implementáciu koordinátora (celkovo nahradiť Zigbee2MQTT). Neskôr sa použil Sonoff Zigbee bridge [40], ktorý po nahratí softvéru tretej strany slúžil ako prevodník medzi protokolom Zigbee a WiFi. Avšak komunikácia bola častokrát nespoľahlivá, kedy koordinátor dostával chybné rámce od koncových zariadení. Pravdepodobne to bolo z dôvodu, že Zigbee aj WiFi antény operujú na rovnakej frekvencii 2.4 GHz a tak mohlo dochádzať k vzájomnému rušeniu komunikácie vplyvom ich veľmi blízkeho umiestnenia.

Pri použití zmieneného hardvéru sa ešte dodatočne používa protokol MQTT na prenos informácií o zariadeniach. Pôvodne sa uvažovalo o implementácii bez tohto protokolu. Avšak je potrebný pre softvér tretej strany Zigbee2MQTT, ktorý komunikáciu s koordinátorom na USB dongle zásadným spôsobom uľahčuje.

4.4 MQTT a Zigbee2MQTT (Zigbee brána)

Pri implementácii sa pôvodne uvažovalo, že by celé ovládanie centrály bolo napísané v jednom programovacom jazyku. Tento "monolith" kódu by ovládal aj koordinátora aj samotný softvér. Prvotný predpoklad bol, že by sa dáta mohli zo Zigbee čipu presúvať po sériovej linke (vytvorenej v rámci USB-VCP) priamo do Raspberry Pi, kde by boli spracované. Po hlbšom pohľade sa však táto predstava už nejavila dosiahnuteľnou. Takýto prístup je nadmerne zdĺhavý na implementáciu a predstavuje komplexnú výzvu. Ako príklad sa dá uviesť použitý Zigbee2MQTT, ktorý sa v skutočnosti skladá z troch samostatných projektov (ako bude ďalej opísané). Z toho dôvodu je efektívnejšie použiť už odskúšaný softvér a vyhnúť sa problémom, ktoré by mohli nastať pri implementácii vlastného softvéru.

Napríklad na Zigbee čipe CC2652P od Texas Instruments je pôvodne nahraný ich Z-stack (Zigbee stack). K nemu majú celkom rozsiahlu príručku pre vývojárov koncových Zigbee IoT zariadení Z-Stack 3.10 User's Guide [41]. Kód je písaný v C a vyžaduje pomerne rozsiahle znalosti do problematiky. Na druhej strane na USB Dongle použitom v práci je už nahraný modifikovaný Zigbee stack, s ktorým dokáže Zigbee2MQTT bez problémov komunikovať a udržovať si lokálne stav celej siete. To predstavuje veľkú výhodu pre vývoj softvéru, ktorý operuje nad samotnými dátami a nemusí riešiť ich špecifické spracovanie pre komunikačnú sieť alebo bezdrôtové médium.

4.4.1 Zigbee2MQTT

Je "open source" projekt rozvíjaný Koenom Kantersom. Súbory sú dostupné cez GitHub webový repozitár alebo oficiálnu webstránku [42]. Projekt ponúka prepojenie Zigbee a MQTT pre ľahšiu integráciu IoT v inteligentnej domácnosti. Účelom je najmä prepojiť rozdielnych predajcov s open-source platformami ako Home Assistant alebo openHAB. Dosiahnuté to je tým, že sa softvér Zigbee2MQTT pripája na dostupnú anténu zariadenia a tvorí jednotnú prístupovú bránu spolu so Zigbee sieťou. Projekt momentálne podporuje vyše 950 zariadení od viac ako 150 rôznych predajcov. V komunitách domácich kutilov alebo nadšencov do IoT je hojne využívaný.

Projekt sa skladá z troch modulov. Prvý operuje nad hardvérom (napríklad nad Texas Instruments) a zaisťuje hlavnú komunikáciu s čipom. Druhý mapuje individuálne zariadenia do takzvaných "Zigbee clusters" (nadstavba nad základným protokolom, definujúca komunikáciu medzi zariadeniami, aby komunikovali medzi sebou navzájom). Posledný je Zigbee2MQTT pôsobiaci ako most medzi prvými dvoma a MQTT. Taktiež si udržuje celkový stav systému v internej databáze, aby ostal nemenný aj po reštarte [43].

Zdrojové súbory sú vyvíjané v jazyku Typescript (odnož JavaScriptu) a spúšťajú sa nad JavaScriptovým behovým prostredím (runtime) "Node.js". To predstavuje asynchrónne prostredie, kde programátorovi odpadá starosť od riešenia niektorých konceptov, ako sú správa vlákien, blokovanie I/O, "deadlock" a iné [44]. Tieto výhody sú na druhej strane vyvážené náročnejšími požiadavkami na zdroje a nutnosťou inštalácie Node.js.

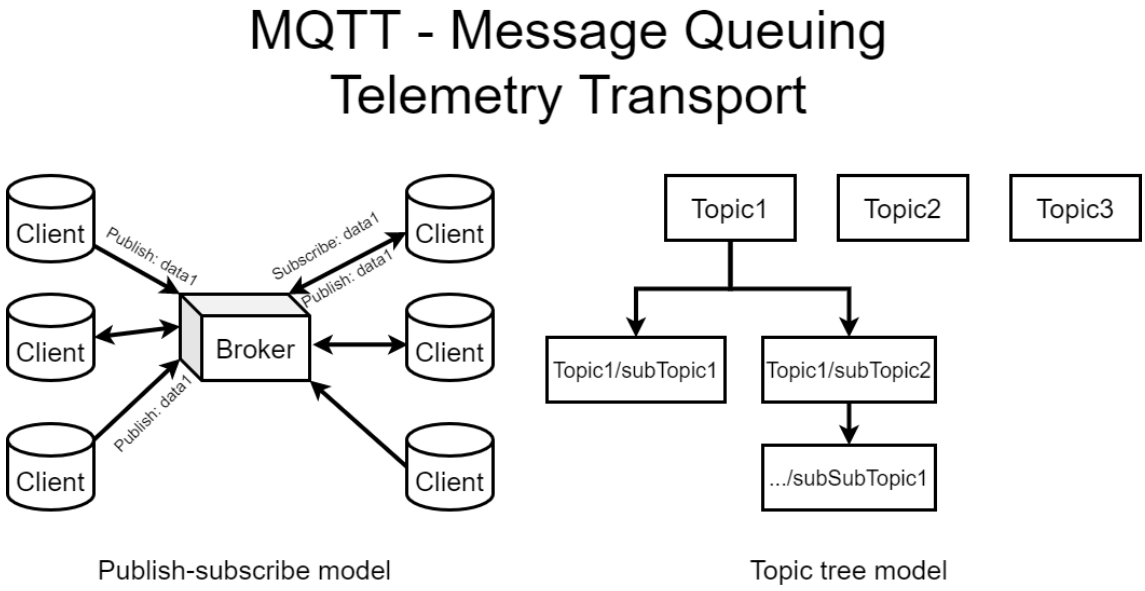
4.4.2 MQTT (Message Queuing Telemetry Transport)

Náročnosť veľkosti hlavičiek dátových rámcov a nadmernej komunikácie pri HTTP alebo CoAP dala podnet na vznik MQTT. Protokol ponúka jednoduchý spôsob transportu dát medzi zariadeniami. Väčšinou transportovaný cez TCP/IP, ale je možné ho posielať aj v iných komunikačných protokoloch. Je vhodný pre siete, kde je dôležitá malá veľkosť dát a nízka odozva. Založený na "publish-subscribe" architektúre s dvoma entitami "client" a "broker". K brokerovi sa klienti pripájajú a udržujú s ním spojenie.

Všetky správy medzi klientami alebo klientami a brokerom musia prejsť cez brokera samotného. Dáta sa posielajú do takzvaných "topic". Každý klient môže poslať správu na ľubovolný topic (publish). Pokiaľ chce klient dostávať správy z nejakej topic, musí ju však u brokera začať odoberať (subscribe). Broker následne prijaté správy rozosiela klientom, ktorí daný topic odoberajú. Každá správa obsahuje nemennú hlavičku o dvoch "bytoch" a dáta o veľkosti maximálne 256MB. Každé spojenie sa môže rozdeliť do štyroch častí: "connection", "authentication", "communication" a "termination". Broker štandardne operuje na porte 1883 a klienti s ním nadväzujú spojenie.

Topics sú štruktúrované v hierarchii podobnej stromovej štruktúre. Preto klient môže odoberať konkrétny topic alebo celý podstrom z odvodených topicov. Topic vytvárajú klienti u brokera. Ten si udržuje záznamy existujúcich topicov a subtopicov. Taktiež si udržuje informácie o klientoch a topicoch, ktoré odoberajú. Aj po strate spojenia s klientom môže broker správy pre odoberaný topic ukladať a po obnove spojenia ich klientovi dodatočne poslať. MQTT správa pozostáva vždy z topic, kam sa posiela a dátovej časti (Payload).

Rovnako je možné nastavenie "Quality of Service" (QoS) v troch úrovniach. V prvej je správa odoslaná a nie je zaručené jej doručenie. Druhá úroveň zaručí, že správa bude doručená aspoň raz. Posledná zaručuje, že správa dorazí práve raz [45], [46].



Obr. 4.9: MQTT model "publish-subscribe" a rozdelenie MQTT Topic [8].

4.4.3 Integrácia Zigbee2MQTT a MQTT broker

Integráciou Zigbee2MQTT do systému dostaneme Zigbee bránu, ktorá plní funkciu MQTT klienta a pripája sa na MQTT brokera, ktorého jej definujeme. Celý systém je spočiatku vyvíjaný na počítači s operačným systémom Windows v jazyku C# a platform .NET Core.

Prvotne je nutné nainštalovať prostredie Node.js, to je prístupné z oficiálnej stránky [44]. Následne sa stiahne a nakonfiguruje Zigbee2MQTT projekt podľa [47]. Hlavná konfigurácia je v súbore "data/configuration.yaml". Tam je pre beh celého softvéru nutné primárne zvoliť:

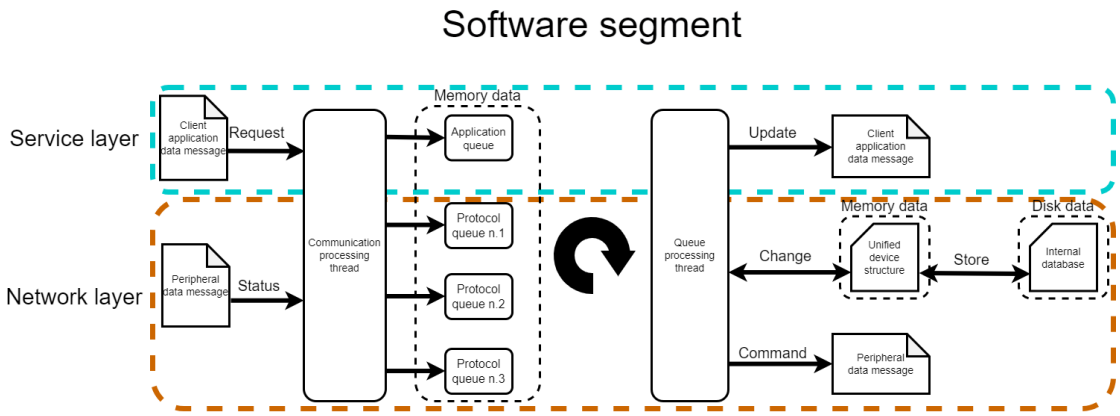
- na ktorom virtuálnom COM porte je pripojená Zigbee anténa.
- IP adresa MQTT brokera spolu s portom kam sa má Zigbee2MQTT klient pripojiť
- základný MQTT topic na publish-subscribe
- povolenie na pripojenie nových Zigbee koncových zariadení
- Aplikačné riešenie

To zahŕňa minimum pre spustenie Zigbee2MQTT. Softvér sa spustí konzolovým príkazom "npm start" (Pozor, pre správny beh je potrebné mať spusteného MQTT brokera. Bez nadviazaného MQTT spojenia Zigbee2MQTT len vytvorí Zigbee sieť, ale správy nepredá ďalej). Správy saypublishované na broker majú dvojaký tvar. Za základný topic sa zvolí "Zigbee". Potom Zigbee2MQTT "publishuje" správy na MQTT topic "Zigbee/bridge/<ďalšie subtopic>", kedy "publishuje" správy týkajúce sa koordinátora a svojej vlastnej funkcie. Druhý typ je, že publishuje správy na topic "Zigbee/<identifikátor koncového zariadenia>". V každej správe sú následne priradené dáta vo formáte JSON.

Možností ako implementovať MQTT broker je mnoho. Sú dostupné programy ako ActiveMQ, RabbitMQ, Apache Kafka, Mosquitto a pod. Avšak, aby centrála nemusela mať spustených niekoľko paralelných programov naraz, bol MQTT broker implementovaný priamo v kóde pomocou knižnice MQTTnet pre .NET Core framework. Tento prístup ponúkol veľkú flexibilitu, nakoľko implementácia vlastného brokera dovolila zachytiť každú prichádzajúcu správu a ďalej ju analyzovať.

4.5 Softvér centrály

SONOFF Dongle, Zigbee2MQTT a MQTT vytvárajú potrebnú komunikačnú bránu pre dáta o Zigbee zariadeniach. Ďalším krokom je samotný softvér centrály. Ten sa dá z hľadiska architektúry rozdeliť na vrstvu služieb a vrstvu siete. Tieto vrstvy sa v ňom prelínajú.



Obr. 4.10: Logická štruktúra softvéru centrály [8].

Z hľadiska logických blokov, ako je znázornené na Obr. 4.10 má niekoľko častí. Hlavné sú dva typy: dátové štruktúry, ktoré držia informácie a procesné vlákna, ktoré nad nimi vykonávajú operácie. Medzi dátové štruktúry patria:

"Queue" - Každý komunikačný protokol má vlastnú pre prijaté správy, kedy sú založené na modele FIFO (First In First Out). Taktiež aj správy od klientov sa spracovávajú týmto štýlom.

"Unified device structure" - Dáta o zariadeniach protokolov sa pretransformujú do jednotnej štruktúry predstavenej v Kapitole 4.

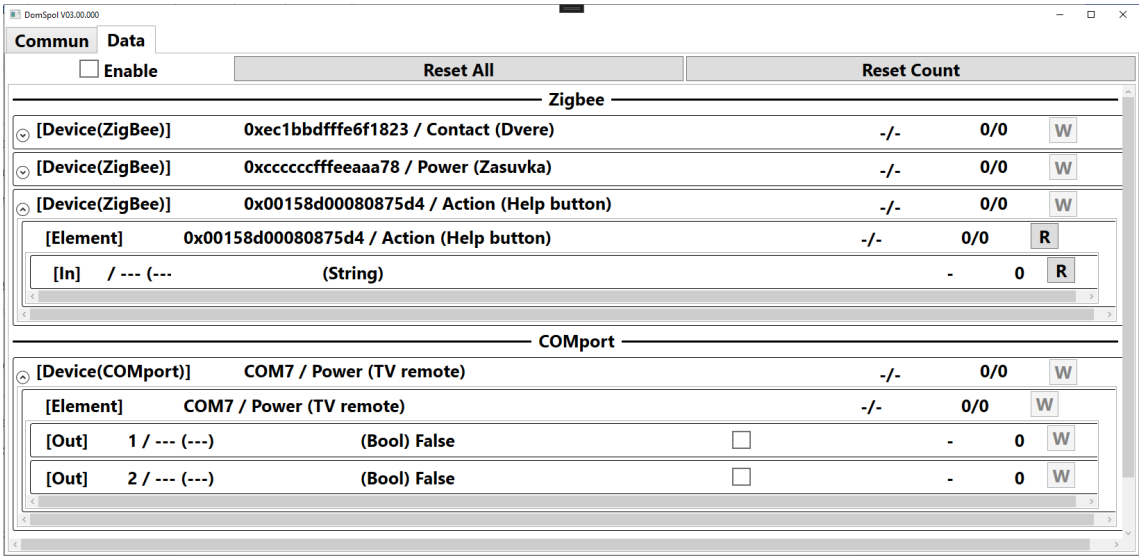
Dobrá otázka je či by sa mali queues vyprázdňovať sériovo jedným vláknom alebo by každá queue mala mať vlastné vlákno, ktoré by ju obsluhovalo (pri viac vláknových procesoroch výhodou). V práci je použité len jedno vlákno pre lepší prehľad a hlavne z dôvodu dobrého testovania. Taktiež musí každá queue operovať s vlastným "mutex" (mutual exclusion) zámkom, keďže je obsluhovaná dvoma vláknami. Vlákna sú:

"Communication processing" - V práci ide o hlavné vlákno, ktoré najprv vytvorí všetky zdroje, a potom asynchrónne prijíma komunikáciu, ktorou plní queues (vkladá správy do queue).

"Queue processing" - Druhé, vedľajšie vlákno, má na starosti spracovanie prijatých správ a ich následné rozkódovanie (vyberá správy z queue). Operuje nad

unified structures. Odosiela správy von z centrály buďto priamo na príslušný komunikačný protokol alebo samotným klientom.

Implementovaný softvér bol napísaný v editore Visual Studio 2022 Community. Ako programovací jazyk bol zvolený C# a platforma .NET Core. Tie poskytujú dobrú prenositeľnosť medzi zariadeniami a dlhodobú podporu zo strany Microsoftu. Základ softvéru bol poskytnutý vedúcim práce, nakoľko sa projekty v určitých konceptoch zhodujú. Tento základ bol následne upravený a rozšírený do terajšej podoby softvéru centrály. Testovanie softvéru prebiehalo formou WPF (Windows Presentation Foundation - UI framework) aplikácie, kedy bol objekt "unified structure" mapovaný na komponenty aplikácie.



Obr. 4.11: Ukážka testovacej WPF aplikácie [8].

4.5.1 Jednotná dátová štruktúra

Implementovaná štruktúra bola založená na návrhu z Kapitoly 4. Pre bližšie priblíženie je vhodné sa pozrieť na jej zápis vo formáte XML. Ten sa využíva pri načítaní tejto štruktúry do pamäte centrály.

Každé koncové zariadenie tvorí "device", ktorý si udržuje odkaz na svoj takzvaný "DeviceConfig<protokol>" ako je na Obr. 4.12 vidno v prípade protokolu Zigbee. Tam je dôležitý "communication type" (CmnType) označujúci ako by mali byť dáta spracované pri odosielaní. "Identification" (Ident) je dôležitá pre komunikačný protokol, nakoľko je v ňom podľa nej zariadenie identifikované. "Name" je názov zariadenia, ako by sa malo zobrazovať. "Desc" udáva jeho dodatočný popis.

Jeden device môže obsahovať viacero senzorov/aktuátorov (Elements). Každý element si udržuje odkaz na zase svoj "ElementConfig<protokol>". Parametre sa od device nelíšia.

Pri elemente potom rozlišujeme jednotlivé dátové jednotky (Datas). Opäť reprezentované svojou "DataConfig<protokol>". "DataType" označuje formát, ktorý

jednotka príjma alebo udržuje. “Direction” (Dir) popisuje či sa jedná o vstupnú alebo výstupnú jednotku. “JsonValue” špecifikuje, o aký povel pre zariadenie (vyzdvihovaný z JSON správy) ide.

```
<Devices>
 <DeviceConfigZigBee CmnType="ZigBee" Ident="0xccccccfffeeaaa78" Name="Power" Desc="Zasuvka">
  <Elements>
   <ElementConfigZigBee Ident="0xccccccfffeeaaa78" Name="Power" Desc="Zasuvka">
    <Datas>
     <DataConfigZigBee Ident="1" DataType="Bool" Dir="In" JsonValue="Power" />
     <DataConfigZigBee Ident="2" DataType="Bool" Dir="Out" JsonValue="Power" />
    </Datas>
   </ElementConfigZigBee>
  </Elements>
 </DeviceConfigZigBee>
</Devices>
```

Obr. 4.12: Zápis jednotnej dátovej štruktúry vo formáte XML [8].

Takýchto konfigurácií môže byť mnoho pre každý komunikačný protokol. Tie sa následne transformujú do objektov tried (Class) obsahujúcich odkazy na jednotlivé konfigurácie. Nad týmito triedami sa už následne operuje nezávisle od použitého komunikačného protokolu zariadenia. Ovládacia aplikácia vykonáva zmeny práve nad triedami a nemusí sa zaoberať, ako dáta prispôsobiť pre špecifický komunikačný protokol.

4.5.2 Jednotný prístupový bod

Tento bod ponúka aplikáciam rozhranie pre komunikáciu. Pôvodne mal byť podľa návrhu implementovaný pomocou protokolu HTTP (prípadne len TCP/IP). Nakoľko však už bol v programe zahrnutý protokol MQTT, tak je veľmi vhodné ho využiť a HTTP vôbec nepoužiť. V takom prípade sú riadiace aplikácie samotnými MQTT klientami a pripájajú sa na brokera implementovaného v centrále. Komunikácia je od nich následne spracovaná rovnakým spôsobom v samostatnej queue akoby prichádzala od komunikačného protokolu koncového zariadenia. Externe pripojená aplikácia sa v podstate z hľadiska centrály (zjednodušene povedané) tvári ako súbor vstupne/výstupných zariadení.

Rozdiel je hlavne vo využitých MQTT topicoch. Kým sa Zigbee hlási do topicu “Zigbee”, tak klienti sa analogicky hlásia do topicu “Client”. Pokiaľ chce ovládacia aplikácia nastaviť hodnotu, tak pošle správu na topic “Client/<identifikátor zariadenia>/set” spolu s hodnotou v dátovej časti správy. Pri žiadaní hodnôt sa subtopic “set” zmení na “get”. Identifikátory odkazujú na “Ident” použitý v jednotnej dátovej štruktúre.

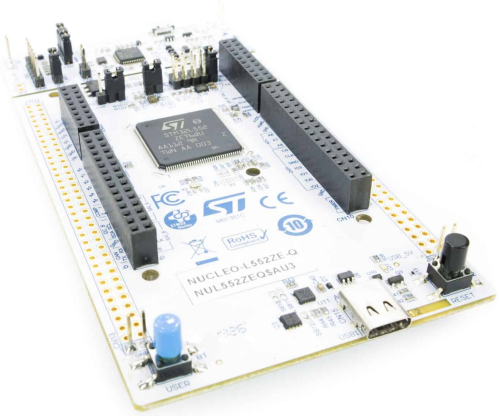
4.6 Vlastné zariadenie

Pre overenie možnosti pripojenia špecifického zariadenia bola implementovaná podpora pre komunikačný protokol založený na USB. Základom zariadenia je mikrokontrolér komunikujúci s centrálou pomocou virtuálneho COM portu vytvoreného

cez USB (USB-VCP). Pôvodne sa uvažovalo nad použitím protokolu HID, ale nepodarilo sa ho sprevádzkovať pre obojsmernú komunikáciu. Spracovanie správ je obdobné ako u Zigbee.

Zariadenie samo o sebe je jednoduchého charakteru. Hlavnou motiváciou je demonštrácia, že centrála podporuje aj výrobky, ktoré sú pre cieľovú skupinu ľudí vytvorené na mieru. V takom prípade je schopná ovládanie spotrebičov spôsobmi, aké nie sú dostupné na komerčnom trhu.

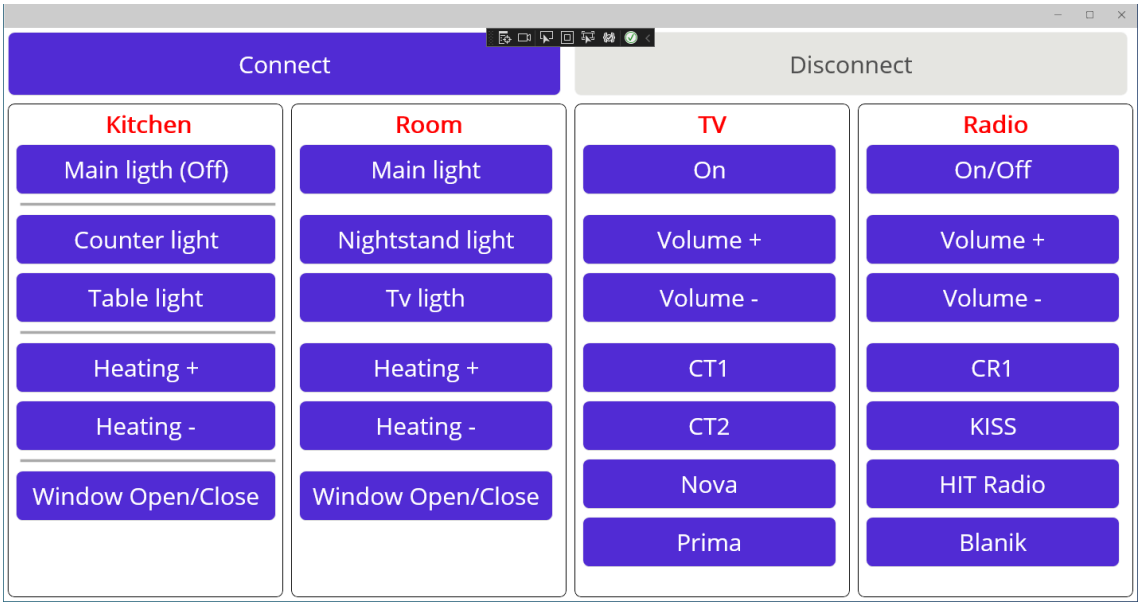
Hlavnou časťou je mikrokontrolér STM32L552ZET6Q, na ktorý je napojených päť LED diód. Pomocou príkazov nad štruktúrou zariadenia je tieto diódy možné ovládať (rozsvietiť/zhasnúť). Táto jednoduchá funkcionalita simuluje napríklad bežný IR (Infra red) ovládač od TV v domácnosti. Každá dióda predstavuje povely ako zapnúť/vypnúť, zvýš/zníž hlasitosť, kanál dopredu/dozadu a pod.



Obr. 4.13: STM32L552ZET6Q ilustračné foto [48].

4.7 Ovládacia aplikácia

Rozhranie jednoduchej aplikácie ovládajúcej koncové zariadenia je zobrazené na Obr. 4.14. Hlavnú funkciu, ktorú musí aplikácia implementovať je MQTT klient, schopný nadviazať spojenie s MQTT brokerom centrály. Celá komunikácia je založená na MQTT topicu “Client”, ktorý bol bližšie popísaný v paragrafe o prístupovom bode. Aplikácia je napísaná v .NET Core a využíva .NET MAUI (Multi-platform App UI). To znamená, že sa rovnaký kód dá preložiť na rozdielne platformy (Android telefón, Android tablet, Windows PC, iOS PC alebo tablet a pod.) Následne na týchto platformách je jej funkčnosť identická. Predstavuje dobrý príklad aplikácie na tablet pre vozíčkarov, ktorý ovláda pomocu špeciálnych spotrebičov (IR ovládač) napríklad TV, rádio alebo ďalšie zariadenia v byte.



Obr. 4.14: Ukážka užívateľského rozhrania pre pripojenie k centrále, ktoré bolo vytvorené v .NET MAUI [8]

Kapitola 5

Použitie

Táto kapitola predstavuje súhrn úkonov na použitie centrály v domácom prostredí. V prípade použitia mikropočítača ako Raspberry Pi, je nutné mu nahrať operačný systém a nastaviť behové prostredie. Následne je v samotnom programe potrebné nakonfigurovať zdrojové súbory pre správne základné nastavenie. Centrála bola sama o sebe testovaná na dvoch platformách.

Prvou je počítač s operačným systémom Windows, kde bol celý softvér práce spočiatku vyvíjaný. Pod Windows je softvér centrály vytvorený ako aplikácia WPF. Vďaka tomu je k dispozícii základné ladenie cez aplikačné okno. V takomto režime celý systém beží na lokálnej adrese "127.0.0.1". Neskôr na tomto počítači bola vytvorená aj vzorová ovládacia aplikácia cez .NET MAUI. Systém dokáže reagovať na zmeny stavov a povely pre zariadenia Zigbee alebo USB-VCP.

Druhou platformou bolo experimentálne odskúšanie na mikropočítači Raspberry Pi 4 Model B. Ten sa pripojil do smerovača domácej siete. Na adrese, ktorá mu bola pridelená, ponúkal svoje služby. Na mikropočítači však nie je dostupné WPF, takže prípadné ladenie je obtiažné. Tak isto kód musí byť "oklieštený" o súbory, ktoré sa nedokážu kompilovať. Aj v tomto prípade však systém reagoval na zmeny stavov a povely pre zariadenia Zigbee alebo USB-VCP. Taktiež ho bolo možné ovládať cez .NET MAUI na počítači v rovnakej sieti alebo cez rovnakú aplikáciu, ale na mobilnom telefóne v rovnakej sieti.

5.1 Nastavenie prostredia pre beh programu

Pre spustenie systému je nutné nastaviť prostredie pre kód centrály. Nasledujúce závislosti sa musia vyskytovať na platforme, kde je centrála spustená. Samozrejme platforma musí mať prístup do lokálnej siete.

Prostredie **Node.js -** prevodník Zigbee2MQTT pracuje v prostredí javascriptu. Spolu s Node.js je potrebný aj "npm" (Node Package Manager). Ten je dôležitý pre interakciu s repozitárom balíčkov a závislostí Node.js.

Prostredie **.NET Core -** samotné zdrojové súbory centrály sú vyvíjané v jazyku C# na platforme .NET Core.

Koordinátor **USB Zigbee Dongle -** bez donglu pripojeného k zariadeniu nie je možné spustiť Zigbee2MQTT.

Súbory **Zigbee2MQTT -** pre ovládanie Zigbee koordinátora.

Pre vývoj sú nutné:
Editor kódu **Visual Studio 2022 -** posledná verzia bola použitá pri vývoji a ladení softvéru. Pre editáciu .NET MAUI aplikácie je nutné mať nainštalované aj **Visual Studio 2022 preview**.

Editor kódu **Visual Studio Code -** je potrebný na editáciu zdrojových súborov v Zigbee2MQTT.

Obr. 5.1: Ukážka pripojenia Raspberry Pi do domáceho smerovača [8].

5.2 Nastavenie pre Windows

Pokiaľ boli všetky predošlé závislosti splnené, je možné centrálu na zariadení spustiť. Windows nevyžaduje ďalšie zásadné úpravy.

5.3 Nastavenie pre Raspberry Pi 4

Pri použitý na mikropočítači je najprv nutné nastaviť a stiahnuť OS. Celý operačný systém spolu so súborovým systémom je uložený na "mikro SD" karte zo spodnej strany mikropočítača. Použitý je Raspberry PI OS, nakoľko by mal byť prispoôsobený pre samotný mikropočítač a samotný výrobca ho odporúča [49].

SD karta by mala obsahovať aspoň 8 GB voľného miesta pre inštaláciu potrebných závislostí. Karta sa vloží do počítača a pomocou "Raspberry Pi Imager", dostupného na stránkach výrobcu, sa sformátuje. Počas formátovania je vybraný "image" operačného systému, ktorý sa na kartu uloží (je vhodné pri výbere "advanced options" nastaviť povolenie pre SSH a údaje pre budúci prístup). Po inštalácii je karta vložená do mikropočítača a ten je pripravený na spustenie [50]. Následne je nutné nakonfigurovať vyššie zmienené behové prostredie programu.

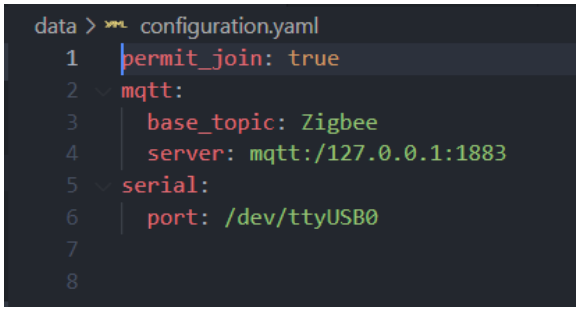
Obr. 5.2: Ukážka Raspberry Pi Imager [8].

5.4 Konfigurácia programu centrály pre Raspberry Pi 4

Po príprave behového prostredia je možné na mikropočítač nahrať zdrojové súbory. Použitý bol program WinSCP. Pomocou SSH sa k Raspberry pripojí a stiahne sa doň zložka so zdrojovými súbormi centrály. Z týchto súborov bola odstránená funkcionalita WPF použitá pri ladení. Následne je nutné stiahnuť súbory Zigbee2MQTT z repozitára projektu. Po stiahnutí je nutné pomocou npm manažéra doinštalovať Node.js závislosti. To je možné príkazmi "npm ci" (alebo "npm install").

5.5 Konfigurácia súborov Zigbee2MQTT

Nastavenie Zigbee2MQTT sa konfiguruje v súbore "/data/configuration.yaml". Ten je najlepšie otvoriť vo Visual Studio Code. Pre základnú funkcionalitu by mal súbor vyzerať ako na Obr. 5.3.



Obr. 5.3: Ukážka konfiguračného súboru pre Zigbee2MQTT s úplne minimálnou konfiguráciou [8].

V ňom "permit_join: true" udáva, aby sa do Zigbee siete mohlo pripojiť akékoľvek Zigbee zariadenie. "base_topic: Zigbee" špecifikuje MQTT topic, na ktorý sa bude Zigbee2MQTT pripájať a "publishovať" alebo "subscribovať správy". "Server: mqtt:/<adresa_ip>:1883" popisuje adresu MQTT broker a jeho port. Ten bude spustený v programe centrály a má mať adresu zariadenia (príkazom "ifconfig" je adresu možné zistiť). "port: /dev/ttyUSB0" alebo "port: //./COM3" (Raspberry

alebo Windows, pozor porty môžu byť odlišné) popisuje USB periférium s pripojeným Zigbee Dongle. Treba použiť aktuálne pripojený port. Po tejto konfigurácii je Zigbee2MQTT pripravený na činnosť.

5.6 Konfigurácia softvéru centrály

Nastavenie adresy softvéru centrály sa musí vykonať predom v zdrojovom súbore centrály, kde pre premennú "MQTTListenAdr" musí byť zvolená IP adresa (na danej adrese sa bude spúšťať MQTT broker). Predpokladá sa, že pri nasadení v domácnosti by bolo zabezpečené (rezerváciou), aby centrála dostávala vždy rovnakú adresu.

Taktiež je nutné pridať koncové zariadenia v súboroch "/AppData/DevsSensConfigs/...". Ich vzorová konfigurácia je znázornená na Obr. 5.4. Takto nakonfigurované prostredie je pripravené na spustenie.

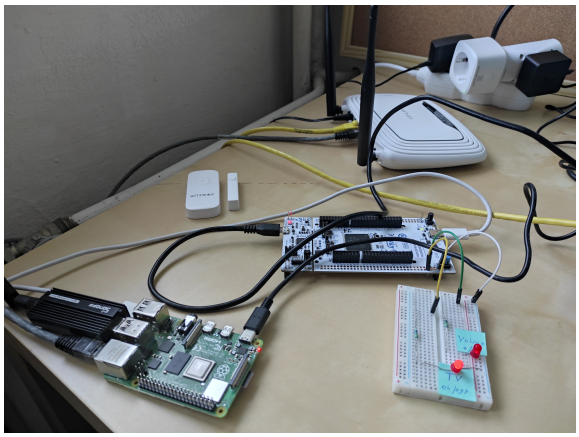
```
<?xml version="1.0" encoding="utf-8"?>
<Config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" Location="Room">
  <Devices>
    <DeviceConfigZigBee CmnType="ZigBee" Ident="0xec1bbdfffe6f1823" Name="Contact" Desc="Dvere">
      <Elements>
        <ElementConfigZigBee Ident="0xec1bbdfffe6f1823" Name="Contact" Desc="Dvere">
          <Datas>
            <DataConfigZigBee Ident="1" DataType="Bool" Dir="In" JsonValue="Contact" />
          </Datas>
        </ElementConfigZigBee>
      </Elements>
    </DeviceConfigZigBee>
  </Devices>
  <Devices>
    <DeviceConfigZigBee CmnType="ZigBee" Ident="0xccccccfffeeaaa78" Name="Power" Desc="Zasuvka">
      <Elements>
        <ElementConfigZigBee Ident="0xccccccfffeeaaa78" Name="Power" Desc="Zasuvka">
          <Datas>
            <DataConfigZigBee Ident="1" DataType="Bool" Dir="In" JsonValue="Power" />
            <DataConfigZigBee Ident="2" DataType="Bool" Dir="Out" JsonValue="Power" />
          </Datas>
        </ElementConfigZigBee>
      </Elements>
    </DeviceConfigZigBee>
  </Devices>
  <Devices>
    <DeviceConfigZigBee CmnType="ZigBee" Ident="0x00158d00080875d4" Name="Action" Desc="Help button">
      <Elements>
        <ElementConfigZigBee Ident="0x00158d00080875d4" Name="Action" Desc="Help button">
          <Datas>
            <DataConfigZigBee Ident="1" DataType="String" Dir="In" JsonValue="Action" />
          </Datas>
        </ElementConfigZigBee>
      </Elements>
    </DeviceConfigZigBee>
  </Devices>
</Config>
```

Obr. 5.4: Ukážka vzorovej konfigurácie Zigbee zariadení [8].

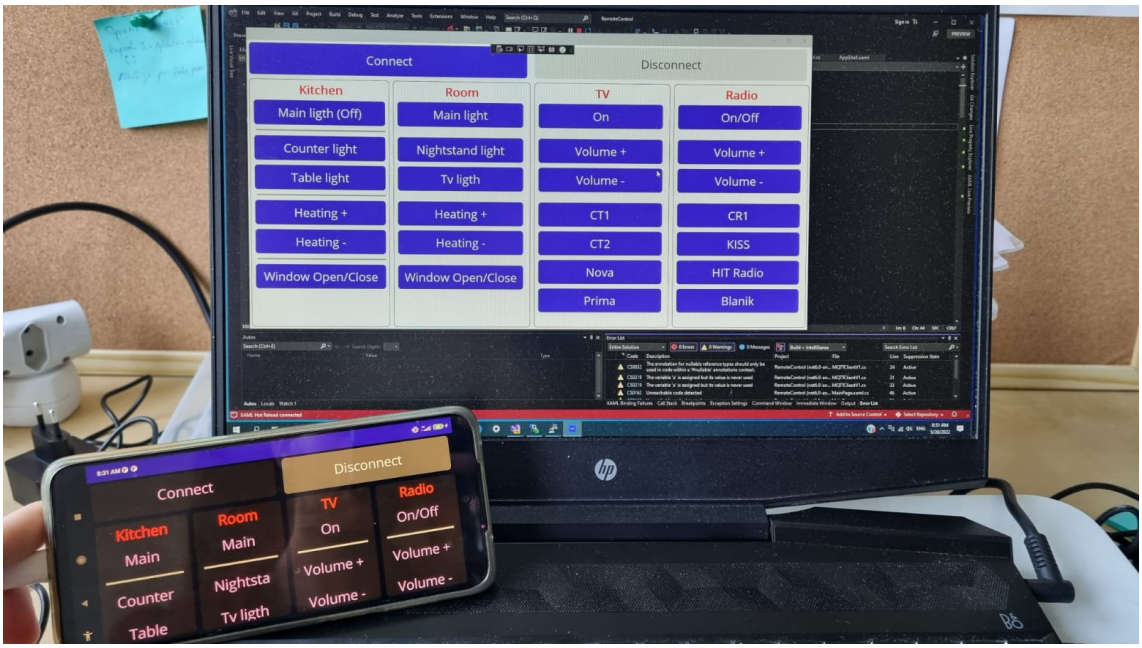
5.7 Spustenie programu centrály

Spustenie centrály prebieha v troch krokoch. V prvom sa pripojí USB dongle k mikropočítaču. V druhom sa v zložke so zdrojovými súbormi Zigbee2MQTT spustí terminálový príkaz "npm start", ten spustí Zigbee2MQTT. To bude hlásiť chyby o nedostupnom koordinátorovi. Ten bude spustený v kroku tri. V ňom sa v zložke so zdrojovými súbormi centrály zadá v príkazovom riadku "dotnet run", čo spustí

program centrály. V tomto okamihu je už centrála pripravená na ovládanie od aplikácií klientov, ktorí sa pripájajú pomocou MQTT na IP adresu Raspberry alebo Windows. Teraz je možné zariadenia ovládať pomocou ovládacích aplikácií, ktoré sa cez MQTT klienta pripoja na centrálu. Aplikácia môže byť napríklad vytvorená v Microsoft MAUI ako je zobrazené na Obr. 5.6 Experimentálne bolo overené na Raspberry Pi aj Windows.



Obr. 5.5: Ukážka zapojenia centrály, kedy je podporované ovládanie Zigbee zariadení [8].



Obr. 5.6: Ukážka použitia Microsoft MAUI aplikácie, ktorá má jeden zdrojový kód, ale je schopná sa kompilovať pre viacero platforiem [8].

Záver

Ako bolo v úvode spomenuté, tak v súčasnej dobe nie je žiaden z dostupných komplexnejších systémov vhodný pre ovládanie domáceho prostredia pre ľudí s pohybovým obmedzením a už vôbec nie pre ľudí s obmedzením jemnej motoriky rúk. Je to najmä z dôvodu, že sa výrobcovia primárne zameriavajú na zdravú populáciu predstavujúcu hlavnú skupinu užívateľov. Práca sa preto zaoberala iba návrhom zariadenia skôr pre menšiu skupinu užívateľov (s pohybovým obmedzením), namiesto návrhu celého komplexného systému

Dosiahnuté výsledky

V rámci záverečnej bakalárskej práce boli vybrané skutočne cenovo dostupné komponenty najmä pre ovládanie prostredia (aktuátory), ale tak isto aj komponenty pre snímanie akcie z tohto prostredia (senzory). Ako tie najvhodnejšie boli zvolené zariadenia využívajúce bezdrôtovú technológiu Zigbee. Jednak z dôvodu širokej dostupnosti na trhu, a taktiež pre veľmi prijateľné ceny. Aj keď v práci boli použité iba (niektoré) vzorové komponenty, je v nej urobený návrh na integráciu akýchkoľvek komponentov využívajúcich túto technológiu. Pri výbere vhodnej technológie bolo tiež myslené na dostupnosť aj celkovo špecifickosť komponentov, ako je napríklad tlačidlo na privolanie pomoci.

Ďalej bola navrhnutá centrála (takzvaný "HUB"), pracujúca ako skutočne centrálny bod pre zber informácií z okolitého prostredia pomocou senzorov IoT a súčasne vysielajúci povely pre IoT aktuátory. Centrála ukladá všetky dáta zo senzorov do jednotnej formy a tie poskytuje rôznym aplikáciam. Rovnako pokiaľ aplikácia nastaví stav nejakých dát, tak ich centrála odošle do príslušného výstupného zariadenia. Z obecného pohľadu teda centrála obsahuje najmä prevod komunikácie Zigbee na komunikáciu typu TCP/IP (MQTT cez LAN/WiFi). To jest medzi reálnymi zariadeniami v domácom prostredí a aplikáciami pre ich kontrolu (ovládaníe).

Veľmi dôležitou súčasťou centrály je takzvaný vstupný bod pre externé aplikácie. K tomu centrála využíva zabudovaného MQTT brokera. Na takto vytvorený vstupný bod sa je možné pripojiť s akoukoľvek aplikáciou špeciálne vyvinutou pre cieľovú skupinu užívateľov (napríklad aplikácia bežiaca na tablete pevne uchytenom na invalidnom vozíku). Tento vstupný bod nie je určený iba pre fyzicky postihnutých užívateľov, ale aj pre bežný dohľad (alebo dozor). Ten môže v prípade potreby (napríklad po slovnej dohode s užívateľom bytu) vzdialene zapnúť požadovanú činnosť ako svetlo, kúrenie a pod.

Významnou časťou práce je návrh pripojenia skutočne špecifického zariadenia

na vytvorenú centrálu. Pre demonštráciu unikátnosti tohoto využitia bolo vytvorené zariadenie simulujúce bežný diaľkový IR ovládač TV. Týmto sa vytvorený mini-systém stáva skutočne nielen unikátnym, ale rovnako tak veľmi vhodným pre cieľovú skupinu užívateľov.

V priebehu práce sa niekoľko, na prvý pohľad jednoduchých častí, ukázalo byť naopak veľmi zložitých. A teda bolo nutné vytvárané riešenie trochu upraviť. Jednou z najväčších zmien bolo odstúpenie od pôvodného zámeru na elimináciu prítomnosti MQTT pre prevod Zigbee správ na ich ďalšie použitie. Vytvoriť vlastný prevod by bolo však nad rámec tejto práce, a preto bola prítomnosť MQTT vo vytváranom riešení ponechaná. Z tohoto dôvodu bol tiež MQTT využitý ako jednotný prístupový bod pre externe sa pripojujúce aplikácie, čím došlo k veľmi vhodnému zjednoteniu externej komunikácie s vytvorenou centrálou.

Výstupom práce síce nie je celkovo hotové, a teda overené zariadenie spolu s jeho dokumentáciou, ale ucelený návrh obsahujúci nedostupné komponenty v existujúcich systémoch. Vytvorená centrála teda ponúka všetky zadané (požadované) činnosti a je vhodná pre ovládanie okolitého prostredia u ľudí s rôznou mierou obmedzenia pohybu.

Bibliografia

1. *Internet of Things: New Promises for Persons with Disabilities*. [online]. July 2015. [cit. 2022-03-02]. Dostupné tiež z: `https://g3ict.org/upload/publication/internet-of-things-new-promises-for-persons-with-disabilities/IoT_new-promises-for-PWD.pdf`.
2. AL-FUQAHA, Ala; GUIZANI, Mohsen; MOHAMMADI, Mehdi; ALEDHARI, Mohammed; AYYASH, Moussa. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*. 2015, roč. 17, č. 4, s. 2347–2376.
3. SAMIZADEH NIKOUI, Tina; RAHMANI, Amir Masoud; BALADOR, Ali; HAJ SEYYED JAVADI, Hamid. Internet of Things architecture challenges: A systematic review. *International Journal of Communication Systems*. 2021, roč. 34, č. 4, e4678.
4. ASHTON, Kevin et al. That 'internet of things' thing. *RFID journal*. 2009, roč. 22, č. 7, s. 97–114.
5. GILLIS, A. S. *What is the internet of things (IoT)?* [online]. March 2022. [cit. 2022-05-10]. Dostupné tiež z: `https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT`.
6. DODDA, R. The Evolution of Internet Of Things (IOT) And Its Impact on Existing Technology. *International Journal For Science Technology And Engineering*. 2016.
7. DORSEMAINE, Bruno; GAULIER, Jean-Philippe; WARY, Jean-Philippe; KHEIR, Nizar; URIEN, Pascal. Internet of things: a definition & taxonomy. In: *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*. 2015, s. 72–77.
8. *Ilustračné obrázky vytvorené autorom práce*. [B.r.].
9. PERWEJ, Yusuf; HAQ, Kashiful; PARWEJ, Firoj; MUMDOUH, M; HASSAN, Mohamed. The internet of things (IoT) and its application domains. *International Journal of Computer Applications*. 2019, roč. 182, č. 49, s. 36–49.
10. DA XU, Li; HE, Wu; LI, Shancang. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*. 2014, roč. 10, č. 4, s. 2233–2243.
11. SRIVASTAVA, Sudeep. *IoT in Manufacturing: Applications and Benefits Explained*. [online]. February 2022. [cit. 2022-03-12]. Dostupné tiež z: `https://appinventiv.com/blog/iot-in-manufacturing/`.

12. HERNÁNDEZ, M. *IoT: Consumer Commercial vs. Industrial - Main overview*. [online]. July 2019. [cit. 2022-03-15]. Dostupné tiež z: https://ubidots.com/blog/iot-consumer-vs-commercial-vs-industrial-main-overview/.

13. KARMAKAR, Avish; DEY, Naiwrita; BARAL, Tapadyuti; CHOWDHURY, Manojeet; REHAN, Md. Industrial internet of things: a review. In: *2019 international conference on opto-electronics and applied optics (optronix)*. 2019, s. 1–6.

14. KHAN, Wazir Zada; REHMAN, MH; ZANGOTI, Hussein Mohammed; AFZAL, Muhammad Khalil; ARMI, Nasrullah; SALAH, Khaled. Industrial internet of things: Recent advances, enabling technologies and open challenges. *Computers & Electrical Engineering*. 2020, roč. 81, s. 106522.

15. REDAKCIA, PR MIM. *Aj odpady môžu byť smart. Monitorovaný zber je efektívnejší.* [online]. February 2021. [cit. 2022-03-17]. Dostupné tiež z: https://www.odpady-portal.sk/Dokument/105905/aj-odpady-mozu-byt-smart-monitorovany-zber-je-efektivnejsi.aspx.

16. PUBLICATION, IAEME. A SUGGESTIVE STUDY FOR HOME AUTOMATION THROUGH IOT. 2019.

17. ALBULAYHI, Khalid; SMADI, Abdallah A; SHELDON, Frederick T; ABERCROMBIE, Robert K. IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors*. 2021, roč. 21, č. 19, s. 6432.

18. AL-GARADI, Mohammed Ali; MOHAMED, Amr; AL-ALI, Abdulla Khalid; DU, Xiaojiang; ALI, Ihsan; GUIZANI, Mohsen. A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE Communications Surveys & Tutorials*. 2020, roč. 22, č. 3, s. 1646–1685.

19. SATYABRATA, J. *Internet of Things (IoT) Gateways*. [online]. June 2020. [cit. 2022-03-19]. Dostupné tiež z: https://www.geeksforgeeks.org/internet-of-things-iot-gateways/.

20. *Papouch Store*. [online]. [cit. 2022-03-25]. Dostupné tiež z: https://en.papouch.com/about-us/.

21. *I/O moduly Quido*. [online]. [cit. 2022-03-19]. Dostupné tiež z: https://papouch.com/io-moduly/quido/.

22. *Digital Twins Definition Language (DTDL)*. [online]. [cit. 2022-03-19]. Dostupné tiež z: https://github.com/Azure/opendigitaltwins-dtdl/blob/master/DTDL/v2/dtdlv2.md.

23. *Introduction to Human Interface Devices (HID)*. [online]. [cit. 2022-05-07]. Dostupné tiež z: https://docs.microsoft.com/en-us/windows-hardware/drivers/hid/.

24. WEIS, O. *What is a COM port? - the main FAQs about serial port*. [online]. December 2019. [cit. 2022-03-20]. Dostupné tiež z: https://www.serial-over-ethernet.com/serial-to-ethernet-guide/what-is-com-port/.

25. *ZigBee Contact Sensor*. [online]. [cit. 2022-03-22]. Dostupné tiež z: https://www.blitzwolf.com/ZigBee-Contact-Sensor-p-439.html.

26. *ZigBee Contact Sensor image*. [online]. [cit. 2022-03-22]. Dostupné tiež z: url=https://www.blitzwolf.com/bg_os/other/upload_temp/products/original/201912/1577173478_26.jpg.

27. *Immax Neo Smart Plug (07048L)*. [online]. [cit. 2022-03-24]. Dostupné tiež z: url=https://www.zasuvka.eu/p/chytra-wifi-zasuvka-immax-neo-07048l.

28. *Návod k obsluze Immax Neo Smart Plug (07048L)*. [online]. [cit. 2022-03-24]. Dostupné tiež z: https://www.navod-k-obsluze.cz/chytra-zasuvka-immax-neo-smart-zigbee-3-0-61274-navod.

29. *Immax Neo bridge*. [online]. [cit. 2022-03-24]. Dostupné tiež z: https://www.immax.cz/immax-neo-bridge-pro-smart-zigbee-3-0-v2-p10213/.

30. *Immax Neo Smart Plug (07048L) image*. [online]. [cit. 2022-03-24]. Dostupné tiež z: https://zigbee.blakadder.com/assets/images/devices/Immax_07048L.jpg.

31. *Aqara Wireless Mini Switch*. [online]. [cit. 2022-03-24]. Dostupné tiež z: https://www.aqara.com/us/wireless_mini_switch.html.

32. *Aqara Wireless Mini Switch image*. [online]. [cit. 2022-03-24]. Dostupné tiež z: https://aqara.ru/wp-content/uploads/2019/10/product-photo-_0013s_0000_cnop.jpg.

33. KALAIVANI, T; ALLIRANI, A; PRIYA, P. A survey on Zigbee based wireless sensor networks in agriculture. In: *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)*. 2011, s. 85–89.

34. ASHRIT, L. *ZIGBEE Architecture (ZIGBEE Stack) – All Layers and its Functions*. [online]. [cit. 2022-03-25]. Dostupné tiež z: https://electricalfundablog.com/zigbee-architecture-zigbee-stack-layers/.

35. ASHRIT, L. *What is ZIGBEE Technology in IoT – Architecture, Network Topologies, Applications*. [online]. [cit. 2022-03-25]. Dostupné tiež z: https://electricalfundablog.com/zigbee-technology-architecture/.

36. *Raspberry Pi 4 Model B specifications*. [online]. [cit. 2022-03-31]. Dostupné tiež z: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/.

37. *Raspberry Pi 4 Model B image*. [online]. [cit. 2022-03-31]. Dostupné tiež z: https://i0.wp.com/www.miki-ie.com/wp-content/uploads/2019/07/RaspberryPi4.png.

38. *SONOFF ZigBee 3.0 USB Dongle Plus*. [online]. [cit. 2022-04-01]. Dostupné tiež z: https://sonoff.tech/product/diy-smart-switch/sonoff-dongle-plus/.

39. *SONOFF ZigBee 3.0 USB Dongle Plus image*. [online]. [cit. 2022-04-01]. Dostupné tiež z: https://images.restomods.com/00/s/ODAwWDgwMA/z/ULcAAOSwSNViYw32/57.jpg.

40. *SONOFF ZigBee Bridge*. [online]. [cit. 2022-04-01]. Dostupné tiež z: https://sonoff.tech/product/smart-home-security/zbbridge/.

41. *Texas Instruments Z-Stack 3.10 User's Guide*. [online]. [cit. 2022-04-16]. Dostupné tiež z: https://software-dl.ti.com/simplelink/esd/plugins/simplelink_zigbee_sdk_plugin/1.60.01.09/exports/docs/zigbee_user_guide/html/zigbee/index.html.

42. *Zigbee2MQTT*. [online]. [cit. 2022-04-15]. Dostupné tiež z: https://www.zigbee2mqtt.io/.

43. KANTERS, K. *Zigbee2MQTT GitHub repository*. [online]. [cit. 2022-04-16]. Dostupné tiež z: https://github.com/Koenkk/zigbee2mqtt.

44. *About Node.js*. [online]. [cit. 2022-04-18]. Dostupné tiež z: https://nodejs.org/en/about/.

45. MICHALEC, L. *MQTT: univerzální protokol nejen pro cloudové aplikace*. [online]. April 2019. [cit. 2022-03-30]. Dostupné tiež z: https://automatizace.hw.cz/mqtt-univerzalni-protokol-nejen-pro-cloudove-aplikace.html.

46. *MQTT*. [online]. [cit. 2022-03-30]. Dostupné tiež z: https://mqtt.org/.

47. *Zigbee2MQTT Windows installation*. [online]. [cit. 2022-04-22]. Dostupné tiež z: https://www.zigbee2mqtt.io/guide/installation/05_windows.html.

48. *STM32L552ZET6Q image*. [online]. [cit. 2022-04-22]. Dostupné tiež z: https://558936-1798772-raikfcquaxqncofqfm.stackpathdns.com/wp-content/uploads/2020/08/sklep_msalamon_NUCLEO-L552ZE-Q_2.jpg.

49. *Raspberry Pi OS*. [online]. [cit. 2022-05-01]. Dostupné tiež z: https://www.raspberrypi.com/software/.

50. *Setting up your Raspberry Pi*. [online]. [cit. 2022-05-01]. Dostupné tiež z: https://www.raspberrypi.com/documentation/computers/getting-started.html#setting-up-your-raspberry-pi.

Prílohy

A Prvá príloha

ReleaseHUB.zip - spustiteľný *.exe* súbor softvéru centrály pod Windows. Predkompilovaný na použitie adresy *127.0.0.1*.

B Druhá príloha

RemoteControlMAUI.zip - zdrojové súbory MAUI aplikácie pod windows.