České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra radioelektroniky

Bezdrátové hlasovací zařízení
Wireless voting device

Bakalářská práce

Vypracoval: Michal Hašl
Vedoucí práce: Doc. Ing. Stanislav Vítek, Ph.D.
Rok: 2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hašl** Jméno: **Michal** Osobní číslo: **498835**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Bezdrátové hlasovací zařízení

Název bakalářské práce anglicky:

Wireless Voting Device

Pokyny pro vypracování:

Navrhněte a implementujte bezdrátové hlasovací zařízení, které bude využívat vhodnou nízkopříkonovou bezdrátovou technologii, např. LPWAN. Vhodnou platformou pro implementaci je např. STM32. Při vypracování se řiďte následujícími pokyny:
1) Proveďte rešerši dostupných LPWAN technologií a vyhodnoťte jejich vhodnost pro daný účel
2) Na základě rešerše navrhněte a implementujte bezdrátového zařízení, které bude schopno odeslat do centrální řidící jednotky hlasování (nejméně 3 stavy - ano, ne, zdržel se)
3) Centrální jednotka umožnuje plánování, provedení, a archivaci jednotlivých hlasování
4) Diskutujte možnosti zabezpečení hlasování

Seznam doporučené literatury:

[1] ADEFEMI ALIMI, Kuburat Oyeranti, et al. A survey on the security of low power wide area networks: Threats, challenges, and potential solutions. Sensors, 2020, 20.20: 5800
[2] OGBODO, Emmanuel Utochukwu; ABU-MAHFOUZ, Adnan M.; KURIEN, Anish M. A Survey on 5G and LPWAN-IoT for improved smart cities and remote area applications: From the aspect of architecture and security. Sensors, 2022, 22.16: 6313.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Stanislav Vítek, Ph.D. katedra radioelektroniky FEL

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

Datum zadání bakalářské práce: **13.02.2023** Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **22.09.2024**

doc. Ing. Stanislav Vítek, Ph.D.	doc. Ing. Stanislav Vítek, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	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

„Prohlašuji, že jsem predloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.“

V Praze dne ....

Michal Hašl

Poděkování

Děkuji Doc. Ing. Stanislavu Vítkovi, Ph.D za jeho rady při vedení mé práce a děkuji také mé rodině a přátelům za jejich podporu.

Michal Hašl

Abstrakt

Tato bakalářská práce se zabývá realizací bezdrátového hlasovacího zařízení. Návrh je proveden jako systém IoT. Použitou komunikační technologií je nízkopříkonová LPWAN. Rešeršní část práce je zaměřena na výběr vhodné LPWAN a rozbor dílčích částí komunikačního řetězce. Praktická část práce je zaměřena na použití vybraných technologií při realizaci systému, zejména pak platformu TTN, komunikační protokol MQTT, LoRaWAN a framework Qt. Výsledkem práce je funkční hlasovací jednotka a řídicí aplikace.

Klíčová slova: IoT, LPWAN, LoRa, LoRaWAN, MQTT, TTN, Qt, bezdrátová komunikace.

Abstract

This bachelor´s thesis deals with an implementation of the wireless voting device. The device is designed as a IoT system. The used low power communication technology is known as LPWAN. The research part of thesis is focused on a selection of suitable LPWAN and an analysis of the communication chain. The practical part of thesis is focused on application of used technologies, especially the TTN platform, the MQTT communication protocol, LoRaWAN and the Qt framework. The result of this thesis is a functional voting unit and an application which controls the device.

Key words: IoT, LPWAN, LoRa, LoRaWAN, MQTT, TTN, Qt, wireless communication.

Obsah

Seznam obrázků

Seznam zkratek

Zkratka	Význam
BS	Base Station
CSS	Chirp Spread Spectrum
eNB	eNodeB
EPC	Evolved Packet Core
EUI	Extended Unique Identifier
GSM	Groupe Spécial Mobile
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
ISM	Industrial, Scientific and Medical
LED	Light-Emitting Diode
LPWAN	Low Power Wide Area Network
LR-WPAN	Low Rate Wide Personal Area Network
LTE	Long Term Evolution
MQTT	Message Queue Telemetry Transport
NB-IoT	Narrowband IoT
OFDMA	Orthogonal Frequency Division Multiple Access
P-GW	Packet data network Gateway
PRB	Physical Resource Block
PSD	Power Spectral Density
QoS	Quality of Service
S-GW	Serving Gateway
SC-FDMA	Single Carrier Frequency Division Multiple Access
SF	Spreading Factor
SNR	Signal to Noise ratio
TCP/IP	Transmission Control Protocol/Internet Protocol
TTN	The Things Network
UTF	Unicode Transformation Format
3GPP	3rd Generation Partnership Project

Úvod

Hlavními požadavky na hlasovací systémy jsou spolehlivost, jednoduchost při jejich užívání a dostupnost pro všechny uživatele bez ohledu na jejich polohu. Zařízení používaná aktuálně v běžné praxi, jsou často staticky zabudována v místě použití nebo potřebují k provozu robustní bezdrátovou komunikační technologii. Přitom z principu zaznamenat výběr hlasu nepředstavuje žádný složitý proces nebo přenos velkého objemu dat. V tom případě přichází v úvahu využít technologie používané v IoT. Na podobném principu je založeno již spoustu projektů, a to hlavně kvůli dobré konektivitě a pohyblivosti koncových zařízení. Tyto existující systémy a technologie lze použít k návrhu bezdrátového hlasovacího zařízení a aplikovat je při jeho implementaci. Příkladem takového IoT projektu může být bezdrátová síť měřičů teplot.

Teoretická část této práce je zaměřena na návrh bezdrátového hlasovacího zařízení a rozbor LPWAN a použitých komunikačních technologií. Praktická část uvádí realizaci celého systému rozděleného do tří bloků: komunikačního rozhraní, koncová hlasovací jednotka, centrální řídicí jednotka.

Kapitola 1

Teoretická část

1.1 IoT

IoT je technologický směr reprezentující propojení každodenně používaných objektů s Internetem. Internet je celosvětová síť propojující počítačové sítě. Vytváří všudypřítomnou možnost připojení chytrých zařízení jako jsou mobilní telefony nebo počítače. IoT rozšiřuje tento koncept o propojení nejen chytrých zařízení, ale i obyčejných věcí. Tyto věci jsou obecně každodenně používané, například dveře, světla, nábytek, kuchyňská elektronika, ruční nářadí, kancelářské potřeby, atd. V IoT sítích mohou tyto věci působit jako senzory, aktuátory nebo obojí najednou. Jejich propojení s Internetem umožní sdílení dat mezi sebou, efektivnější správu nad jejich funkcemi a tím i lepší odezvu dalších zařízení při různých změnách v jejich prostředí.

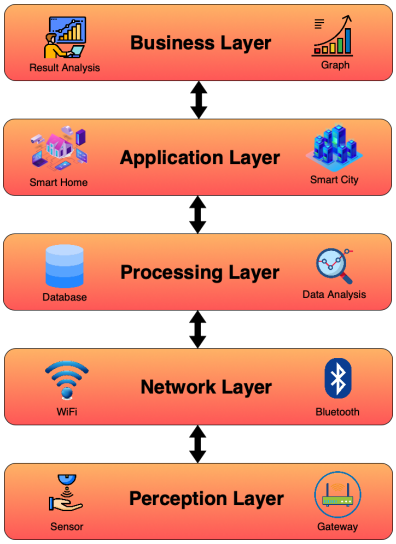
1.1.1 Princip IoT

Hlavním konceptem je vytvoření datové komunikace mezi dvěma objekty, které budou umístěny na libovolném místě a budou moci pracovat v jakémkoliv čase. Nejčastěji se jedná o propojení senzoru s aktuátorem. Aktuátor získává data ze senzoru a na základě těchto dat může proběhnout nějaká reakce. V komunikačním rozhraní bloku senzorů a aktuátorů se vyskytuje i prvek, který zpracovává data ze senzoru, zprostředkovává jejich uložení a přístup k Internetu. Stejným způsobem lze přistupovat ke konceptu návrhu hlasovacího zařízení. Jde o vytvoření více objektů, které budou vzájemně datově propojeny. Jeden z těchto objektů je řídicí člen a ostatní fungují jako koncová zařízení, sloužící k záznamu hlasu. Přenos dat se v IoT nejčastěji řeší pomocí bezdrátových technologií, jako například použití protokolu Wi-Fi, Bluetooth nebo nízkopříkonové technologie

LPWAN. Tyto technologie umožňují variabilnější aplikovatelnost a pohyblivost zařízení s využitím napájení ve formě baterií. Podrobnější struktura IoT je naznačena v sekci 1.1.2. Informace jsou čerpány ze zdroje [1].

1.1.2 IoT architektura

V IoT se využívá velmi široké spektrum technologií. Z tohoto důvodu to není ucelený standard, ale spíše vytváří kombinaci více různých standardů najednou. Proto ani není specifikována jednotná architektura, kterou je myšlen základní popis struktury systémů IoT. Je dělena do vrstev, kde každá vrstva reprezentuje různé funkce typického IoT systému. Níže je popsána jedna architektura IoT z mnoha. Skládá se z pěti vrstev a vyznačuje se tím, že je považována za jednu z nejvýstižněji navržených architektur pro IoT. Její strukturu je možné pozorovat na obrázku č. 1.1 a podle ní je navržen systém bezdrátového hlasovacího zařízení. Podrobný návrh a implementace jsou popsány v kapitole č. 2 této práce. Informace jsou čerpány ze zdrojů [2] a [3].



Obrázek 1.1: Pětivrstvá architektura IoT [4]

Senzorová vrstva

Je to první vrstva popisované architektury. Hlavní činností této vrstvy je získávání informací v reálném čase z okolí a předávání těchto informací do síťové vrstvy. Senzorová vrstva se skládá ze senzorů a aktuátorů. Senzory jsou používány například k měření

teploty, vlhkosti nebo detekci pohybu. Využívají nízkých datových toků k zajištění nižší spotřeby energie. Aktuátory jsou zařízení vykonávající práci, například elektrické motory a spínače nebo mohou jen reprezentovat upozornění například v chytrém telefonu.

Dle popisu této vrstvy by mělo hlasovací zařízení obsahovat senzor nebo prvek, který bude v reálném čase zaznamenávat rozhodnutí uživatele. Tímto prvkem může být například tlačítko nebo spínač. Nemají žádnou spotřebu energie, slouží jen ke změně stavu napětí, což návrhu vyhovuje. Aktuátor v tomto systému může představovat jen informační LED nebo jej není potřeba vůbec implementovat.

Síťová vrstva

Tato vrstva slouží jako komunikační rozhraní mezi senzorovou a zpracovávací vrstvou. Příkladem technologií, které v IoT představují síťovou vrstvu jsou třeba LPWAN sítě nebo technologie patřící do LR-WPAN jako Zigbee. IoT se skládá z obrovských sítí propojených zařízení vytvářejících data, která je potřeba zpracovat a uložit. Proto tato vrstva musí být robustní, aby zvládla velké objemy přenášených dat. Měla by podporovat požadavky QoS zpoždění komunikace a propojení IoT systémů. Zaručuje správu a zabezpečení přenosu dat.

Síťovou vrstvou je obslouženo předání dat zaznamenaných hlasů ke zpracování. Bezdrátová komunikace je nezbytná pro navrhovaný systém hlasování. Je potřeba zajistit přenos dat pro co nejvíce účastníků s co nejmenšími náklady na infrastrukturu. V takovémto případě přichází jako nejvhodnější řešení použití technologie LPWAN. Tato technologie přináší velmi dobré pokrytí pro malé toky dat, přičemž hlasování tyto podmínky skvěle splňuje. Jejich popis a výběr konkrétní technologie je uveden v sekci 1.2.

Zpracovávací vrstva

Zpracovávací vrstva obstarává analýzu, ukládání a zpracování hromadných dat přijatých ze síťové vrstvy. Hlavní funkcí je poskytnutí těchto služeb uživateli. Například analytiku služeb, statistickou analýzu dat, prediktivní analytiku nebo management konfigurace provozu a bezpečnosti.[1] Dovoluje interakci různorodých zařízení, která jsou založena na odlišných technologiích. Ukládá zpracovaná data s časovou známkou do vlastních databází nebo do cloudových úložišť.

Hlasovací zařízení posílá zprávy s hlasy technologií LPWAN, kde jsou přijímány. K zpracování nám můžou posloužit další služby IoT. Cílem je zajistit přeposílání zpráv

[1] Analytika je systematická výpočetní analýza dat. Odhaluje důležité spojitosti ve zpracovávaných datech na základě statistiky.

do centrální jednotky, kde bude provedeno vyhodnocení dat a výsledků. Nejjednodušším řešením se nabízí protokol MQTT, který poskytuje jednoduchou implementaci a rychlé přeposílání dat v reálném čase. Více informací o MQTT je popsáno v sekci 1.3.

Aplikační vrstva

Aplikační vrstva udává IoT systémům jejich podstatu. Využívá funkce nižších vrstev. Uceluje je do jednotného systému, který je aplikován dle zadaného problému. Různé aplikace v průmyslu a ve světě používají IoT při zlepšování svých služeb. Například inteligentní domy, inteligentní města, zdravotní péče, podpora životního prostředí, atd.

Jak již bylo naznačeno v předchozí vrstvě, navrhovaný hlasovací systém obsahuje i řídicí jednotku. Tato jednotka může být implementována jako spustitelná aplikace, která centralizuje koncová zařízení a vytváří aplikační vrstvu v systému bezdrátového hlasovacího zařízení. Její tvorba a popis je uveden v sekci 2.3.

Obchodní vrstva

Poslední vrstva je obchodní. Ze získaných dat vytváří obchodní modely, které jsou aplikovatelné v různých odvětvích naší společnosti. Na základě těchto modelů hodnotí výkonnost používaných systémů a prezentuje jejich možná zlepšení. Tato vrstva poskytuje údržbu nad celým systémem a zajišťuje představení výsledků v podobě grafů, vývojových diagramů nebo výsledkových tabulek.

1.1.3 IoT zpracování dat

V Internetu věcí je spousta různých požadavků na správu zpracování informací. Důležitá je spolehlivost příjmu dat. Musí být zajištěna dostatečná přesnost a kvalita senzorů, aby se předešlo vytváření chyb. Poté je potřeba informace třídit do struktur a čistit od duplikátů nebo chyb, které se mohou objevit. Při hlasování nesmí dojít z opětovnému záznamu již započtených hlasů, strukturalizací a kontrolou dat je potřeba tyto chyby eliminovat. Dále jsou uvedeny další procesy, které je v IoT nutno řešit. Informace jsou čerpány z [5] a [6].

Big data

Hlavní výzvou v IoT je rostoucí počet zařízení a s nimi i rostoucí toky dat, které je třeba zpracovat. Množství dat z IoT může být kolosální. Běžný IoT systém vygeneruje i jeden milion záznamů za sekundu. Takové množství dat se nazývá „big data“ v překladu

velká data. Proto systém IoT potřebuje silnou základní strukturu, aby tyto toky dat dokázal zvládat.

Strukturalizace

Data mohou být využívána pro procesy v aktuálním čase, jako třeba sledování pohybu objektů nebo monitorování zdraví v medicíně. Důležitou záležitostí je tedy zajištění nízké latence při zpracování velkých toků dat. K tomu napomáhá strukturalizace přenášených informací. Data přenášená obyčejně v síti Internet jsou nestrukturovaná, protože se přenáší spousta různých informací, které nejsou často typově stejné. V IoT senzory vysílají stále stejné zprávy. Například naměřená hodnota teploty je vždy numerická hodnota, která se dá strukturovat do buněk, jako datový typ float.

Časové řady

Zprávy jsou v IoT nepřetržitě generovány, a to nejčastěji periodicky. Při pravidelném odečtu dat vznikají datové toky, které nazýváme časovými řadami. Informace jsou v nich zaznamenávány i s časovou hodnotou, pojmenovanou „time stamp“ v překladu časová značka. Pro časové řady je vytvořeno mnoho speciálních metod pro modelování a analýzu. Tyto metody je možno využívat při zpracování informací v IoT.

Ukládání dat

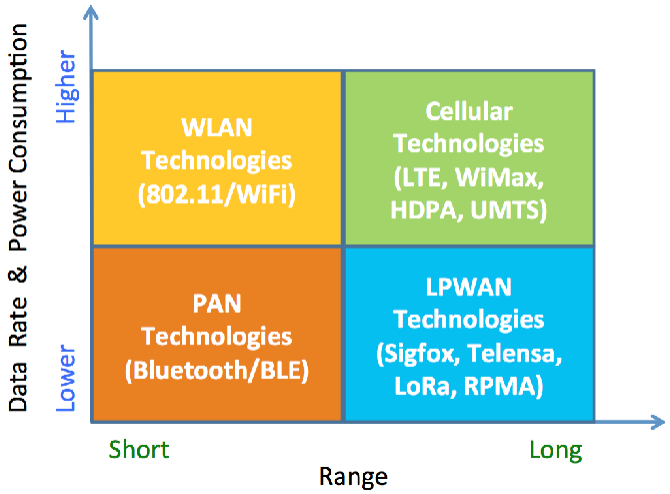
Výše popsané vlastnosti je potřeba dodržet při ukládání dat. Databáze pro IoT se bez správné strukturalizace neobejdou. Data mohou být využita až v budoucnu. Pokud by se z důvodu nedostatečné kapacity systému nezpracovala nebo by se chybnou časovou značkou znehodnocovala budou následně uživateli způsobovat potíže.

Je výhodné využívat databázových i cloudových služeb. Databáze jsou lokální úložiště a cloud představuje úložiště vzdálené. Při velkých datových tocích jsou cloudová úložiště nerentabilní pro většinu uživatelů. Pro menší objemy dat jsou ideální hlavně z důvodu jednoduchosti provozu.

1.2 LPWAN

LPWAN je v překladu pojem, který zastupuje spojení dvou slovních frází, a to nízkoenergetický a sítě velkého rozsahu. Pro IoT systémy jsou důležitými parametry bezdrátové připojení s možností příjmu informací na velké vzdálenosti, dlouhá výdrž baterie

a nízká cena koncových zařízení. Aktuálně existuje spousta technologií pro tvorbu bezdrátových sítí. Bohužel většina z nich přímo nevyhovuje systémům IoT, a to s ohledem na jejich příkon, cenu nebo dosah příjmu. Sítě LPWAN tyto vlastnosti splňují, a proto je o ně v odvětví IoT velký zájem. Vlastnosti technologií Wi-Fi, Bluetooth nebo LTE jsou porovnány s LPWAN na obrázku č. 1.2. Informace jsou čerpány z [7].



Obrázek 1.2: Porovnání technologií ve spotřebě energie a dosahu příjmu [7]

Existuje mnoho LPWAN technologií, které se dělí podle přístupu ve využití frekvenčních pásem. S frekvenčními pásmy je potřeba efektivně nakládat, protože jsou již obsazena dalšími technologiemi. LPWAN proto používají bezlicenční pásma a principy vysílání ve velice úzkém nebo naopak v rozprostřeném spektru. Nejpoužívanější technologie LPWAN v IoT jsou například NB-IoT, Sigfox nebo LoRa.

1.2.1 Sigfox

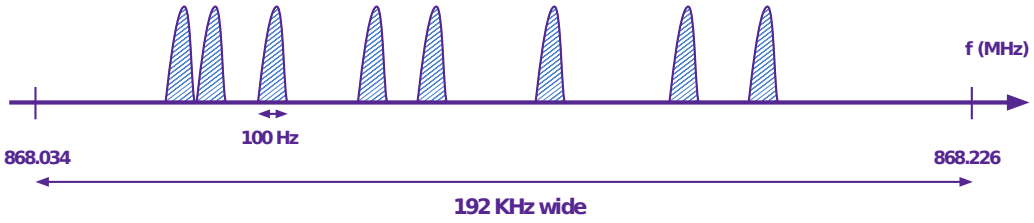
Sigfox je protokol tvořící LPWAN síť. Je založen na principu vysílání malých balíků dat na velmi úzkých frekvenčních pásmech. Díky tomu zajišťuje dobrou konektivitu, velký dosah a vysokou úsporu energie pro užívaná zařízení. Stojí za ním stejnojmenná francouzská společnost, která tuto unikátní komerční síť nabízí. Připojení k Sigfoxu je zpoplatněno, a to za každé zařízení a počet zpráv, které jsou denně vysílány. Koncová zařízení můžou vysílat až 140 zpráv denně a základnové stanice mohou vysílat až 4 zprávy denně. Maximální pokrytí tohoto protokolu se pohybuje od 3 km v městské zástavbě až po 50 km ve volné krajině. Informace jsou čerpány z [8] a [9].

UNB – Ultra narrow band

UNB je metoda přenosu signálů na velmi úzkém frekvenčním pásmu. Šířka pásma se pohybuje kolem 1kHz. Koncentrací energie v takto úzkém frekvenčním pásmu a díky nízkému šumu v pásmu, lze dosáhnout vysokých hodnot PSD i přes nižší výkony vysílače. Tímto je zajištěn velký dosah a nízká spotřeba energie. Zároveň s PSD roste i odolnost vůči rušení, což umožňuje soužití UNB na sdílených frekvenčních pásmech. Informace jsou čerpány z [7].

Fyzická vrstva

Sigfox využívá nelicencovaná frekvenční pásma ISM, v Evropě je to 863–870 MHz a v dalších částech světa 902–928 MHz. Technologie pracuje na 192 kHz širokém pásmu, kde každá zpráva je vysílána na náhodné frekvenci s šířkou modulovaného signálu 100 Hz, jak je naznačeno na obrázku č.1.3.



Obrázek 1.3: Příklad použitého pásma technologie Sigfox [9]

Modulace užívané na nosných jsou DBPSK a GFSK. Zpráva od koncového zařízení, neboli *uplink* je modulována pomocí DBPSK. Užitečná data pro *uplink* mohou být maximálně 12 Byte velká. Vysílání od základnové stanice se nazývá *downlink*, který užívá modulaci GFSK. Užitečná data pro *downlink* mohou být maximálně 8 Byte velká.

- DBPSK - Diferenční binární fázové klíčování je digitální modulace. Vysílaný symbol je prezentován změnou fáze.

- GFSK - Gaussovo frekvenční klíčování je digitální modulace. Vysílaný symbol je nejprve upraven do tvaru Gaussovy křivky a poté je prezentován postupnou změnou vysílané frekvence.

Vyšší vrstvy

Linková vrstva dává vysílaným zprávám strukturu, díky němuž jsou rozlišitelné pro ostatní zařízení, která v Sigfox síti pracují. Datový blok uplinků se nazývá *frame* a může být maximálně 25 Byte velký. Obsahuje počáteční preambuli, synchronizaci komunikace, ID a autentifikaci samotných zařízení. Tento *frame* je naznačen na obrázku č. 1.4. Dále je naznačena i stavba downlinku na obrázku č. 1.5.

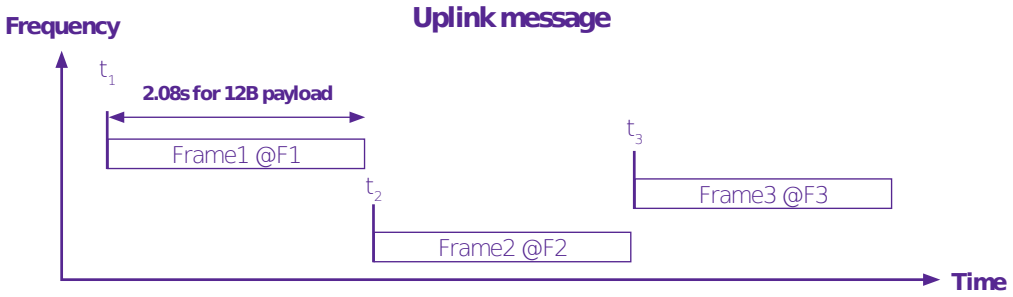
32	16	32	0–96	8	16
PREAMBLE	SYNC	ID	PAYLOAD	AUTH	FCS

Obrázek 1.4: Uplink Frame (hodnoty jsou v bitech)

32	13	2	8	8	8	0–64
PREAMBLE	SYNC	FLAGS	FCS	AUTH	EC	PAYLOAD

Obrázek 1.5: Downlink Frame (hodnoty jsou v bitech)

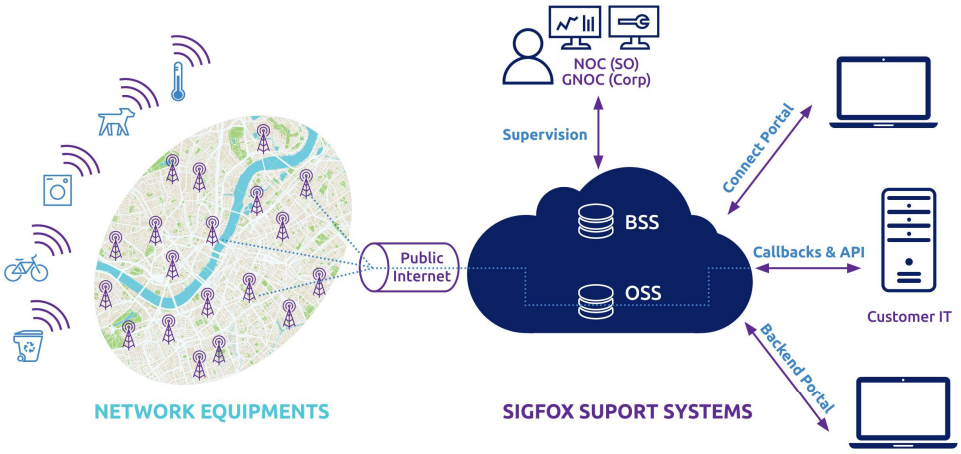
Rychlost přenosu se liší podle regionu, ve kterém se používá. V Evropě je to 100 bitů za sekundu a v ostatních částech světa jsou přenosy 600 bitů za sekundu. Komunikace je v Sigfoxu nesynchronizovaná. Je založena na síťovém principu *Aloha*, který je specifický v přístupu k synchronizaci vysílání signálů. Vysílání probíhá kdykoliv, tedy náhodně. Tato náhodnost vysílání v Sigfoxu slouží pro zajištění vyšší kvality příjmu, je naznačen na obrázku č. 1.6.



Obrázek 1.6: Náhodný přístup vysílání technologie Sigfox [9]

Uplink, je vysílán 3x na náhodné frekvenci, pokaždé v jiném čase. Následně 20 s po konci příjmu první zprávy je časové okno 25 s, které je použito základnovou stanicí pro vyslání *downlinku*.

Sigfox zajišťuje i síť základnových stanic. Má velmi dobré pokrytí. Průměrné množství základnových stanic přijímajících signál z koncového zařízení jsou tři. Touto vlastností je zajištěna ještě vyšší robustnost systému a příjmu v síti Sigfox. Topologie této sítě je naznačena na obrázku č. 1.7.



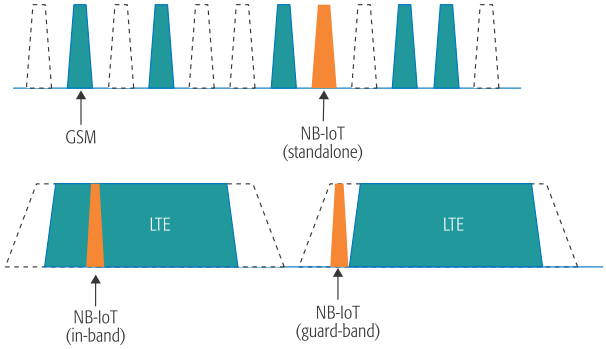
Obrázek 1.7: Topologie sítě Sigfox [9]

1.2.2 NB-IoT

NB-IoT je služba představena projektem 3GPP. Je postavena na architekturách sítí GSM a LTE s důrazem na velké pokrytí, nízkou spotřebu energie a dostupnost fyzických zařízení. NB-IoT poskytuje konektivitu pro velké množství zařízení na již pracující architektuře a síti, je však za tyto služby zpoplatněna. Zároveň je díky návrhu této služby zajištěna kompatibilita s možným nástupem modernějších verzí LTE a GSM. Informace jsou čerpány z [8] a [10].

Fyzická vrstva

Nízkou spotřebu a vysoký dosah umožňuje využití principu UNB, na kterém je založena fyzická vrstva NB-IoT. Princip UNB je vysvětlen výše v sekci o technologii *Sigfox*. Hlavním požadavkem je, aby tato technologie mohla být nasazena ve 200kHz frekvenčním pásmu standardu GSM. Díky tomu může být efektivně využita i v dalších pásmech licencovaného spektra. Ve výsledku má nosná NB-IoT šířku 180 kHz, což odpovídá šířce jednoho bloku pásma PRB z LTE, proto může nosná NB-IoT fungovat ve třech operačních módech naznačených na obrázku č. 1.8.



Obrázek 1.8: NB-IoT typy provozu [10]

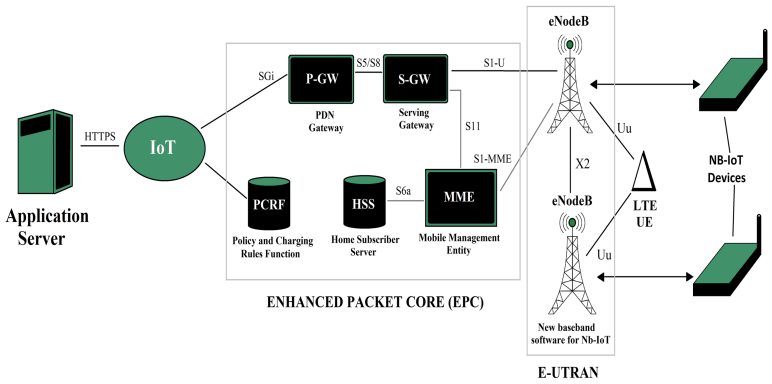
- *Standalone operation* - NB-IoT vysílá na nosném kanálu, který je typicky nasazen v pásmu GSM. V tomto provozu je využit maximální vysílací výkon.
- *In-band operation* - Při pásmovém provozu pracuje v pásmu LTE jako jeden blok PRB. V tomto módu musí NB-IoT sdílet vysílací výkon spolu s LTE provozními požadavky.
- *Guard band operation* - Ochranné pásmo LTE je nevyužité. Může posloužit pro vysílání NB-IoT při sdílení jedné základnové stanice BS a jejího vysílacího výkonu. Očekává se i nižší rušení oproti In-band módu, jelikož LTE může zarušit signál jen z jedné strany pásma NB-IoT.

NB-IoT používá modulační techniky frekvenčního dělení OFDMA a SC-FDMA, které jsou implementovány z LTE. Pro *downlink* se používá OFDMA a pro *uplink* SC-FDMA. *Downlink* a *uplink* je dělen na 12 subnosných kanálů po 15kHz. Konfigurace uplinku navíc podporuje dělení na 48 nosných kanálů po 3,75kHz.

- OFDMA - Vícenásobný přístup s ortogonálním frekvenčním dělením, je princip rozdělení nosného pásma ve frekvenci na subnosné kanály. Každý kanál vysílá různá data a vzájemně jsou postaveny vedle sebe tak, aby se co nejméně překrývaly. Je jednodušší než SC-FDMA a jsou s ním spojeny vyšší nároky na spotřebu energie.
- SC-FDMA - Frekvenční dělení s jednou nosnou frekvencí je sice složitější, ale má vyšší úsporu energie, a proto je aplikováno na uplink v NB-IoT. Principiálně se dělí signál ve frekvenci jako v OFDMA, ale kanály jsou dělené i v čase a nesou pro každý time slot stejný symbol ve všech subnosných kanálech.

Vyšší vrstvy

Topologie NB-IoT je založena na infrastruktuře LTE. Je uvedena na obrázku č. 1.9.



Obrázek 1.9: Topologie služby NB-IoT [11]

Infrastruktura LTE zahrnuje jádro sítě nazývané EPC a uživatelská zařízení UE. EPC obsahuje MME, což je entita řešící mobilitu připojených zařízení jako sledování, management přenosu a výběr služební gatewaye S-GW. S-GW trasuje přenášená data pakety sítí a slouží k zachování přenosu při handoveru. Handover je přechod mezi dvěma různými základnovými stanicemi eNB. Gateway P-GW obstarává rozhraní mezi sítí NB-IoT a externími sítěmi jako například Internet. HSS je domácí jednotka uživatelů umožňující přístup k nasbíraným datům, jejich management a celkovou správu komunikace. UE

představují všechna připojená uživatelská zařízení v síti NB-IoT. Zařízení NB-IoT pracují hlavně v úsporném módu nazývaném PSM. Většinu času jsou ve spánku a mají minimální spotřebu energie. V takovém stavu nejsou pro základní stanice dostupné a vysílají data jen v daných intervalech.

1.2.3 LoRa

Technologie *LoRa* je bezdrátová modulační technika. Její název vychází z dvou vlastností, kterými disponuje. Těmito vlastnostmi jsou *Long* v překladu dlouhý a *Range* v překladu dosah, tedy dlouhý dosah. Je vytvořena firmou Cycleo, následně odkoupenou firmou Semtech. LoRa dále obsahuje zavedení linkové vrstvy s názvem LoRaWAN. Dohromady tvoří spolehlivou síť LPWAN pro IoT aplikace. Je nejvhodnější možností pro návrh bezdrátového hlasovacího zařízení. Existují služby s implementací LoRaWAN serverů, které nabízejí volný přístup k jejich infrastruktuře. Pod podmínkou implementace nižšího počtu zařízení mohou být využity i bez placení, a to je z hlediska návrhu systému hlasování dostačující.

Fyzická vrstva

LoRa je odvozena z metody CSS. Namísto kódování se v modulaci používá takzvaný *Chirp*. Využívá nelicencovaná frekvenční pásma ISM jako technologie Sigfox. V Evropě je například pásmo okolo 868 MHz v USA je to pásmo okolo 915 MHz. Vysílání v pásmu ISM je omezeno takzvaným „Duty Cycle“, který specifikuje maximální denní dobu pro vysílání všech zařízení v tomto pásmu. Pro evropské pásmo EU863-870 je to 1 % tedy z celkových 86400 s je 1 % denně 864 s. Informace jsou čerpány z [14].

SS - Spread spectrum

Metoda rozprostřeného spektra anglicky *Spread spectrum* umožňuje záměnu přenosové rychlosti za citlivost příjmu signálu. Tento jev je možné popsat na Shannon-Hartleyově teorému. Ten vychází z rovnice

$$C = B \cdot \log_2(1 + \frac{S}{N}), \tag{1.1}$$

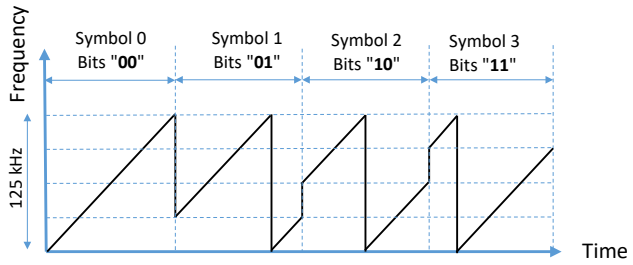
která stanovuje maximální rychlost přenesené informace komunikačního kanálu C na specifickém frekvenčním pásmu B při odstupu signálu od šumu $\frac{S}{N}$ nebo také SNR *signal to noise ratio*. Ta lze aproximovat pro malé hodnoty SNR na

$$\frac{C}{B} \approx \frac{S}{N}. \tag{1.2}$$

Z rovnice (1.2) lze vypozorovat, že při zvětšení šířky pásma B, lze zachovat určitý přenos i pro malé hodnoty SNR. Díky tomu je možné přenášet menší objem dat na velké vzdálenosti s minimální spotřebou energie pro vysílání, protože není potřeba vysílat signál s vysokým výkonem. Informace jsou čerpány z [12] a [13].

CSS – Chirp spread spectrum

Chirp je pulz s lineárně se měnící frekvencí v čase. V modulaci LoRa se používá k zakódování vysílaných symbolů do širokého spektra. Chirpy mají stejnou šířku jako pásmo technologie LoRa, která používá pásma od 125 kHz až po 500 kHz.



Obrázek 1.10: Příklad chirpů technologie LoRa [15]

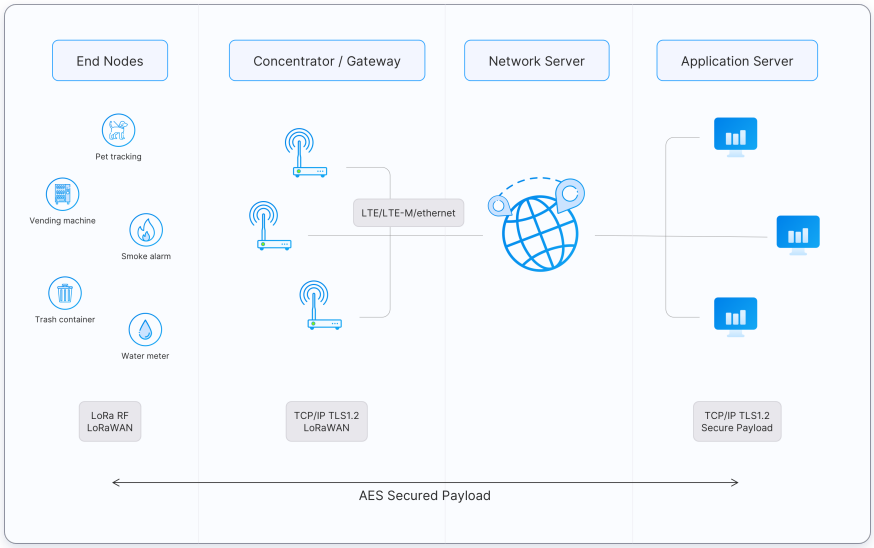
SF – Spreading factor

SF je parametr udávající kolik bitů bude přeneseno v jednom symbolu. Délka vysílání jednoho symbolu závisí na SF a šířce pásma. Vztah č. (1.3) popisuje tuto závislost. Je z něj názorně vidět, že s širším pásmem BW bude růst i symbolová rychlost F_{symbolu}.

$$F_{\text{symbolu}} = \frac{1}{T_{\text{symbolu}}} = \frac{BW}{2^{SF}} \tag{1.3}$$

LoRaWAN

Standard LoRaWAN je nadstavbou k modulační technologii LoRa. V ISO/OSI modelu reprezentuje linkovou vrstvu a vytváří celý komunikační řetězec od koncových zařízení až k LoRaWAN serveru. Obrázek č. 1.11 ilustruje topologii LoRaWAN. Je na ní naznačena sít základnových stanic a přijímačů takzvaných *gateways*, propojených s LoRaWAN serverem. Informace jsou čerpány z [15] a [16].



Obrázek 1.11: Topologie sítě LoRaWAN [17]

Síťový a aplikační server

LoRaWAN server je tvořen ze dvou částí, síťového a aplikačního serveru. Tyto servery dohromady zprostředkovávají zpracování přijatých zpráv, tedy autentizaci koncových zařízení a dešifrování zpráv. Síťový server obstarává příjem přijatých zpráv a mazání duplikátů. Následně zprávu autentizuje pomocí 128bit klíče nazývaného *NwkSKey* a ověří od jakého zařízení zpráva přišla. Aplikační server zajišťuje rozšifrování zpráv, a to díky podobnému klíči, který se nazývá *AppSKey*. Rozšifrované a ověřené zprávy jsou následně posílány uživateli. Více k zabezpečení je uvedeno v sekci 2.4.

Třídy koncových zařízení

Koncová zařízení se v LoRaWAN dělí do tří kategorií, a to podle pravidelnosti vysílání *uplinků* a příjmu *downlinků*.

- Třída A – Koncová zařízení vysílají *uplink* náhodně bez verifikace dostupnosti gatewaye. *Downlink* je vysílán jen v krátkém okně po příjmu *uplinku*. Koncové zařízení je jinak po zbytek času ve spánku.
- Třída B – Zařízení se chovají stejně jako v třídě A, ale vysílání *uplinků* je synchronizováno do časových oken. Gatewaye vysílají v těchto oknech synchronizační beacon zprávy.
- Třída C – Mezi náhodně vysílanými *uplinky* je koncové zařízení stále aktivní a čeká na příchozí *downlink*. Je tedy za cenu mnohanásobně vyšší spotřeby energie přístupné uživateli k řízení pomocí *downlinků*.

Aktivace koncových zařízení

K zahájení komunikace s LoRaWAN serverem je zapotřebí aktivace koncových zařízení. Aktivace proběhne pokud obě dvě části komunikačního spojení znají DevAddr a klíče popsané v sekci výše *síťový a aplikační server*, *NwkSKey* a *AppSKey*. DevAddr je 32 bitový identifikátor koncového zařízení v LoRaWAN síti. K dispozici jsou dvě metody ABP a OTAA.

ABP - Activation By Personalization Zavedení této metody je jednodušší. Provádí se statickým uložením klíčů do koncové jednotky i LoRaWAN serveru. Používá se však spíše při testování komunikace, neboť statickým uložením těchto citlivých klíčů je zvýšena šance jejich úniku a tím i narušení bezpečnosti komunikace.

OTAA - Over The Air Activation Při této metodě se klíče potřebné pro komunikaci generují během párování zařízení s LoRaWAN serverem. K úspěšnému spárování musí server znát Device EUI, AppEUI/JoinEUI a AppKey což jsou identifikátory koncového zařízení a klíče pro zakódování vzdálené aktivace.

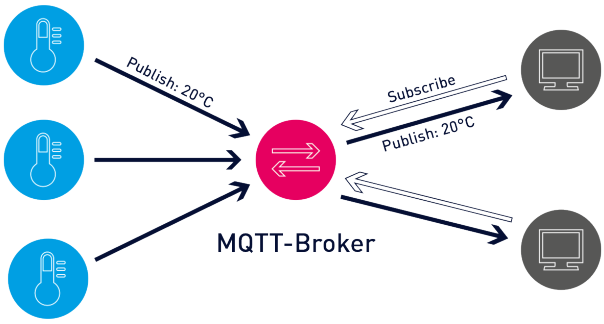
1.3 Komunikační protokol MQTT

MQTT je velmi populární a užitečný nástroj v implementaci IoT systémů. Bude využit v návrhu bezdrátového hlasovacího zařízení a proto je v této části uveden a rozebrán. Informace jsou čerpány z [18]. MQTT je transportní protokol založený na principu publish-subscribe. Publish vyznačuje publikaci zpráv a subscribe odběr zpráv. Je to velmi odlehčený protokol zakládající se na triviální implementaci s jednoduchou režií transferu dat, oproti například protokolu HTTP. Byl vytvořen v roce 1999 panem Andy Stanford Clarkem a Arlenem Nipperem, s požadavky na:

- jednoduchou implementaci
- kvalitu doručení dat
- efektivitu fungování
- nezávislost na přenosu různých typů dat
- nepřetržitý provoz

Publish a Subscribe model

Tento princip je specifický tím, že odstraňuje přímé spojení klientů a dělí je na propojení mezi klienty, kteří posílají zprávy (publisher) a klienty odebírající zprávy (subscriber). Klienti mezi sebou nemají přímý kontakt ani informaci jestli vůbec existují. Takto je velmi zjednodušena režie přeposílání dat. Propojení klientů je zajištěno komponentou zvanou broker. Funkcí brokeru je filtrovat všechny příchozí zprávy a přeposílat je klientům, kteří mají přihlášený odběr. Filtrace zpráv se organizuje pomocí speciální hierarchie témat nazývaných *topic*. Tato témata jsou tvořena UTF-8 řetězci a skládají se z více vrstev, přičemž každá vrstva je oddělována dopředným lomítkem. Příkladem témata může být: `domov/teplota/kuchyne`. Architektura MQTT je znázorněna na obrázku č. 1.12.



Obrázek 1.12: Topologie architektury MQTT [19]

Kapitola 2

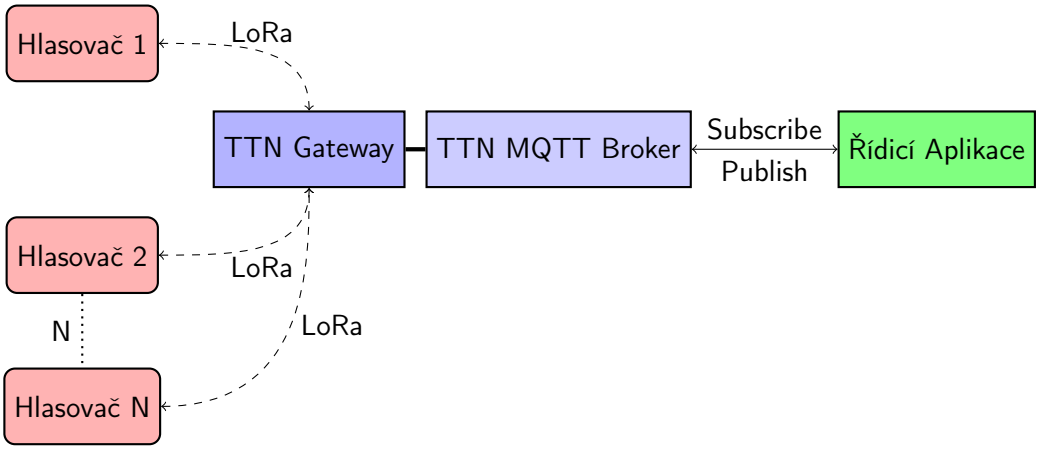
Praktická část

Praktická část popisuje tvorbu celého zařízení a implementaci technologií uvedených v teoretické části. Bezdrátové hlasovací zařízení se skládá ze dvou bloků. Prvním je hlasovací jednotka, která zajišťuje zpracování a odeslání volby uživatele do sítě LoRaWAN. Druhým je centrální jednotka zprostředkovávající řízení hlasování, zobrazení jeho výsledků a jejich archivaci. Na začátku této kapitoly je rozebráno komunikační spojení těchto dvou částí a následně bude popsána i tvorba obou bloků.

2.1 Komunikační rozhraní

V této sekci je popsána přenosová cesta záznamu hlasování od fyzického hlasovacího zařízení až po řídicí aplikaci. Datové propojení tvoří bezdrátová komunikační technologie LoRa, platforma TTN a protokol MQTT.

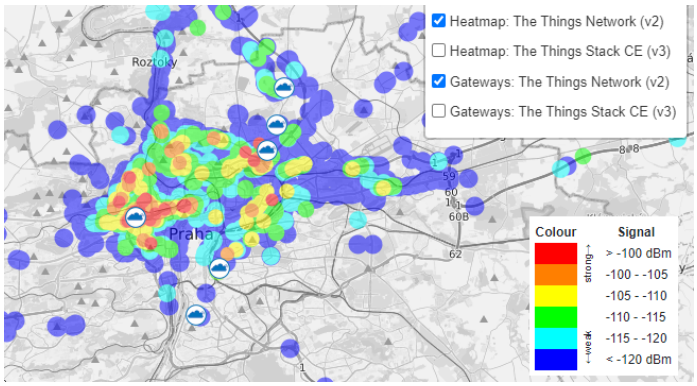
Hlasování je zprostředkováno stiskem tlačítka na hlasovací jednotce. Tímto stiskem je zároveň vyslána zpráva do sítě LoRaWAN. Platforma TTN poskytuje LoRaWAN server a síť LoRa bran, které naši zprávu přijímají a zpracovávají. Dále TTN obsahuje možnost tvorby vlastního MQTT serveru a jeho pomocí budou zprávy přeposílány centrální jednotce. Pokud je aktivní hlasování s přihlášeným odběrem MQTT zpráv, centrální jednotka obdrží informaci o hlasování a o zařízení, které hlasování zprostředkovalo. Schéma tohoto komunikačního rozhraní je ilustrováno na obrázku č. 2.1 a bude podrobněji popsáno v dalších sekcích.



Obrázek 2.1: Schéma přenosové cesty

2.1.1 TTN

TTN je globální uživatelská platforma pro tvorbu IoT systémů. Nabízí uživatelům infrastrukturu LoRaWAN a interaktivní prostředí ke správě projektů. V systému hlasovacího zařízení je využita proto, že obsahuje volně přístupný LoRaWAN a MQTT server. Zajišťuje příjem zpráv díky rozsáhlé síti bran. Tyto brány jsou v platformě tvořeny uživateli a jsou volně dostupné i pro ostatní uživatele. TTN umožňuje rozklíčování přijímaných dat, která jsou dále posílána do nastaveného MQTT serveru. Samotná služba je placená ale pro účely testování s nižším počtem zařízení je volně přístupná pro užití zdarma. Je třeba zajistit dostatečnou vzdálenost v dosahu příjmu bran služby TTN. V aglomeraci Prahy má služba TTN velmi vysoké pokrytí, které je možné sledovat na obrázku č. 2.2. Neměl by při připojení zařízení nastat žádný problém.



Obrázek 2.2: Teplotní mapa síly signálu v Praze [20]

2.1.2 Použití zařízení s TTN

Hlavní předností tohoto konceptu je možnost záznamu zpráv bez přístupu k Internetu. Aby uživatel mohl začít používat službu TTN musí se zaregistrovat a vytvořit si na webové platformě projekt. Pro implementaci hlasovacího zařízení byl vytvořen projekt s názvem „LoRa – Wireless vote device“ a do něj jsou přidána koncová zařízení mikrokontroléry Nucleo-WL55JC, které jsou popsány v sekci 2.2.1. K registraci koncových zařízení je potřeba specifikovat důležité parametry, a to používaný regionální frekvenční plán, verzi LoRaWAN, DevEUI, JoinEUI a AppKey.

- DevEUI je unikátní 64-bit identifikátor koncového zařízení.
- JoinEUI je 64-bit identifikátor sloužící k propojení koncového zařízení k serveru LoRaWAN metodou OTAA, která je více popsána v sekci 1.2.3. Pro hlasovací zařízení je použito JoinEUI: `01 01 01 01 01 01 01 01`.
- AppKey je šifrovací klíč sloužící k zakódování připojovací sekvence, která přeposílá klíče NwkSKey a AppSKey. Více o těchto klíčích je uvedeno v sekci 1.2.3.
- Verze LoRaWAN definuje jaká verze media access control bude podporována.
- Frekvenční plán je měněn dle regionu. Pro projekt hlasovacího zařízení je to Europe 863-870 MHz.

Pokud jsou tyto parametry správně zvoleny, tak se koncové zařízení při aktivaci připojí k serveru LoRaWAN a ten bude zobrazovat přijímané zprávy. Data z těchto zpráv jsou ukazovány ve výpisu v reálném čase a mohou být následně formátovány. Pro účel přeposílání hlasů je využit jednoduchý javascript formátovač, který jednobytový datový balík převede na číslo v desítkové soustavě. TTN dále nabízí spoustu integrací dalších služeb, například LoRa Cloud nebo protokol MQTT, který je vhodný pro přeposílání dat do centrální jednotky.

2.1.3 Integrace MQTT v TTN

Použití protokolu MQTT je velmi jednoduché. Je to díky jeho vlastnostem, které již jsou popsány v sekci 1.3. Integrace zahrnuje vytvoření vlastního serveru, prostřednictvím platformy TTN, který je automaticky propojen s již využívaným LoRaWAN serverem. MQTT následně přeposílá přijímaná a formátovaná data z hlasovačů. K serveru je možné se připojit pod veřejnou adresou `eu1.cloud.thethings.network:1883` a přihlašovacími údaji skládajícími se z uživatelského jména a vygenerovaného hesla.

2.2 Hlasovací jednotka

Hlasovací jednotka představuje v systému bezdrátového hlasovacího zařízení koncovou jednotku. Slouží uživateli k zaznamenání hlasu a jeho odeslání ke zpracování centrální jednotkou. Skládá se z mikrokontroléru umožňujícího zprostředkovat komunikaci v síti LoRaWAN a přidružené plošné desky, která obsahuje potřebná tlačítka k zaznamenání volby hlasu. Hlasování má tři možnosti: *pro*, *proti* nebo *zdržel se*. Pro každý hlas je implementováno záznamové tlačítko. Řízení jednotky je zajištěno mikrokontrolérem a aktivace probíhá připojením k napájení. Poté se na dálku připojí do sítě TTN a může proběhnout hlasování uživatelem. Napájení jednotky je zajištěno nabíječkou přes USB Micro-B kabel, nebo pomocí power banky, čímž je zajištěna plná mobilita hlasovací jednotky.

2.2.1 Mikrokontrolér

Množství zařízení, umožňujících vysílání technologií LoRa, je velké. V tomto projektu je použita vývojová deska Nucleo-WL55JC, která je založena na dvoujádrovém ARM Cortex -M4/M0+ od firmy STM32. Tento mikrokontrolér je vhodný pro vývoj a návrh koncových jednotek do IoT systémů stavějících na LoRaWAN technologii.

Základní specifikace

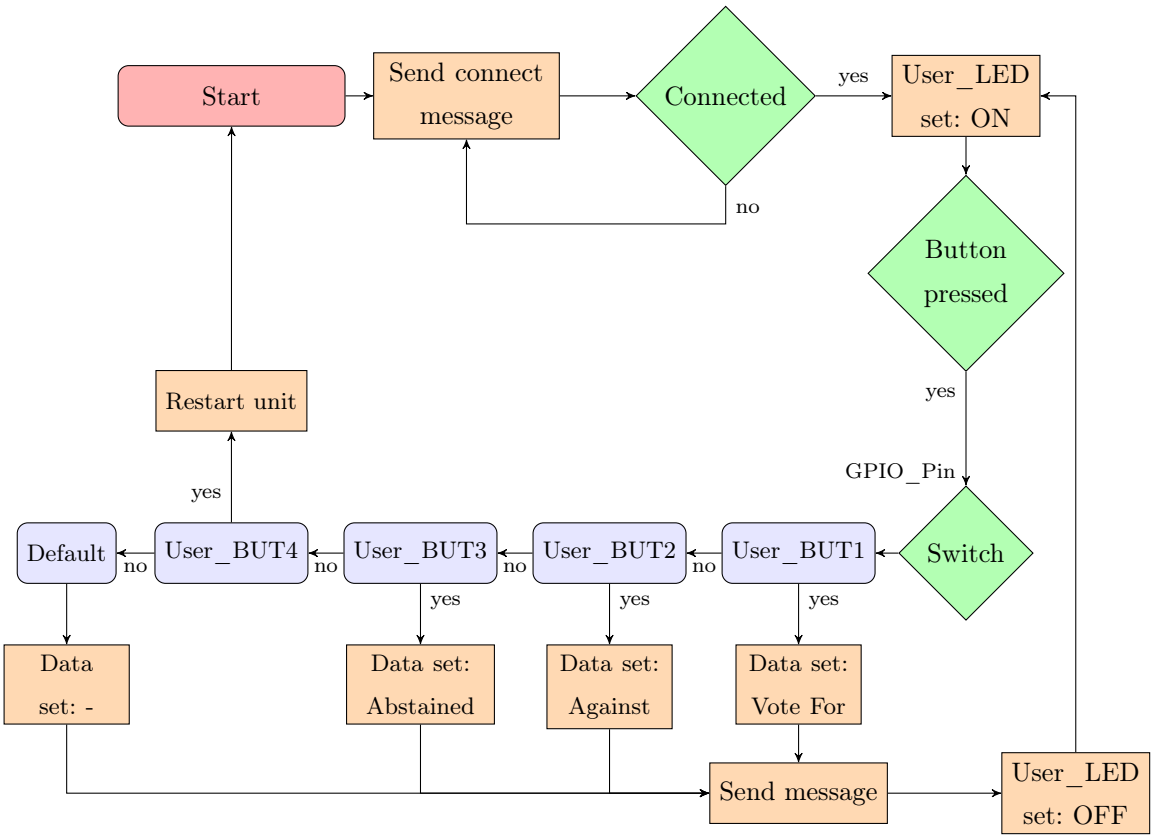
Je vybaven energeticky nenáročnými komponentami a vysílačem s frekvenčním rozsahem od 150 MHz do 960 MHz podporující modulace LoRa, (G)FSK, (G)MSK, BPSK. Obsahuje implementované konektory kompatibilní k připojení periferie s rozměry vývojové desky Arduino Uno nebo externí desky se stejnou strukturou konektorů. Tato struktura je použita k zapojení externího plošného spoje s hlasovacími tlačítky. Technická data jsou uvedena z [21]. Specifikace použitých pinů, jejich funkce a nastavení je uvedena v tabulce č. 2.1.

Tabulka 2.1: Tabulka použitých pinů a jejich nastavení

Jméno periferie	**GPIO**	**GPIO mód**	**Pull up/-down**	**Funkce**	**UNO**
User_BUT1	PB7	Ext. interrupt	Pull up	Záznam hlasu pro	D0
User_BUT2	PB5	Ext. interrupt	Pull up	Záznam hlasu proti	D4
User_BUT3	PB8	Ext. interrupt	Pull up	Záznam neúčasti	D5
User_BUT4	PA9	Ext. interrupt	Pull up	Servisní tlačítko	D9
User_LED	PA11	Output	-	Informační LED	D14

2.2.2 Zdrojový kód

K naprogramování koncové hlasovací jednotky je využit vzorový příklad od firmy STM pro Nucleo-WL55JC1 s názvem *LoRaWAN_End_Node*. Tento příklad zahrnuje implementaci koncového zařízení posílajícího data na LoRaWAN server. Je rozdělen do mnoha částí se zdrojovým kódem, přičemž dokument `lora\_app.c` obsahuje funkce zprostředkovávající hlavní činnost koncové jednotky, která je rozdělena do dvou částí. První probíhá inicializace zařízení a připojení k LoRaWAN serveru. Následně po potvrzeném spojení se serverem je umožněno posílat zprávy s uživatelskými daty. Konfiguraci aplikace lze nastavit na posílání zpráv se stiskem uživatelského tlačítka nebo cyklicky po časových úsecích. Pro účely hlasovacího zařízení je vzorový příklad upraven. Vývojový diagram výsledného kódu je uveden na obrázku č. 2.3.



Obrázek 2.3: Vývojový diagram činnosti hlasovací jednotky

Hlavní úprava kódu spočívá ve změně konfigurace koncové jednotky. Zařízení se nejdříve cyklickým vysíláním přihlašovacích zpráv připojí k serveru LoRaWAN. Poté se změní konfigurace na manuální vysílání a zprávy jsou vysílány s daty rozlišující stisknutá tlačítka. Uživatelská tlačítka při stisku spouští externí přerušení, na které mikrokontrolér reaguje vysláním zprávy. Servisní tlačítko v případě stisku restartuje mikrokontrolér, a tím se znovu zahájí sekvence připojení k serveru LoRaWAN. Informační LED je připojena na pin ve výstupním módu. Změna stavu tohoto pinu způsobí změnu stavu LED. K rozlišení zaznamenaných hlasů je do datového bufferu posílané zprávy ukládána osmibitová hodnota, která je specifická pro každé tlačítko. Tyto hodnoty jsou uvedeny v tabulce č. 2.2.

Tabulka 2.2: Tabulka hodnot ukládaných pro specifická tlačítka

Tlačítko	Hodnota
User_BUT1	11000001
User_BUT2	01010010
User_BUT3	11110000

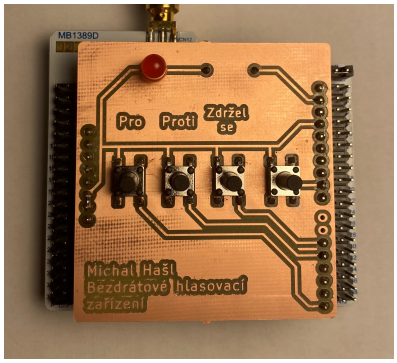
Níže je uvedena ukázka kódu č. 2.1 hlasovací jednotky. Příklad obsahuje část funkce `SendTxData()`, ve které jsou ukládána posílaná data v závislosti na stisknutém tlačítku. Celý zdrojový kód je k nahlédnutí v přílohách této práce.

```
1  static void SendTxData(void)
2  {
3    // AppButState uchovava informaci o stisku tlacitka
4    switch (AppButState)
5    {
6      case 1:
7        AppData.Buffer[i++] = (uint8_t)(0b11000001 & 0xFF); // 193
8        break;
9      case 2:
10       AppData.Buffer[i++] = (uint8_t)(0b01010010 & 0xFF); // 82
11       break;
12     ...
13   }
14   // funkce LmHandlerSend() spousti dalsi procesy pro vysilani
15   LmHandlerSend(&AppData, LmHandlerParams.IsTxConfirmed, false);
16 }
```

Kód 2.1: Část funkce SendTxData()

2.2.3 Plošný spoj

Součástí koncového zařízení pro bezdrátové hlasovací zařízení je plošný spoj se čtyřmi uživatelskými tlačítky a informační LED. K jeho osazení na mikrokontrolér Nucleo-WL55JC je použita struktura pinů připravena pro externí desku na Arduino UNO. Tato struktura je k dispozici v návrhovém systému pro návrh podobných desek. Je osazena čtyřmi tlačítky, LED a jejím předřadným rezistorem. Dále jsou upraveny okraje desky, přidány popisky tlačítek a jméno celého zařízení. Návrh byl proveden v otevřeném systému pro návrh elektroniky KiCad a jeho výroba proběhla na fréze plošných spojů v učebně IoT-Lab. Na obrázku č. 2.4 je fotografie hotového plošného spoje a v přílohách této práce je k nahlédnutí i jeho schéma a layout.



Obrázek 2.4: Externí plošný spoj

2.3 Centrální jednotka

Centrální jednotka představuje ve vytvořeném systému hlavní řídicí blok. Je třeba zajistit aby splňovala požadavky uživatelů hlasovacích zařízení. Hlavní inspirací při návrhu centrální jednotky je systém Ministr [23], což je komerční hlasovací systém používaný při konferencích a dalších hlasováních. Dle tohoto systému jsou definovány hlavní parametry řídící jednotky, které by měla pro správný chod systému hlasování splňovat, například:

- nastavení názvu hlasování a jeho podbodů
- třídění hlasování do podbodů
- správu podbodů po dobu schůze uživatelů
- zobrazení výsledků hlasování pro každou podotázku

- zobrazení výpisu spočtených hlasů
- možnost restartu hlasování
- uložení výsledků každého podbodu

Dále je potřeba zajistit možnost registrace hlasovacích jednotek a jejich přiřazení pro jednotlivá hlasování. Centrální jednotka je navržena jako spustitelná aplikace a díky protokolu MQTT může běžet na každém počítači připojeném k síti Internet.

2.3.1 Vývojová platforma Qt

Řídící aplikace je navržena ve frameworku Qt. Je to multiplatformní knihovna určená pro tvorbu aplikací. Podporuje více programovacích jazyků jako například C++, Python nebo Java. Jeho součástí jsou i vývojové nástroje jako Qt Creator a Qt Designer. Qt dodává velké množství knihoven a nabízí volně přístupnou licenci, proto je velmi vhodný pro tvorbu navrhované centrální jednotky. Dále obsahuje i realizaci MQTT komunikace pomocí knihovny QMQTT, která poskytuje implementaci klienta MQTT. Tento klient zajistí aplikaci odběr zpráv nebo i publikování zpráv na server MQTT. Knihovna QMQTT není přímo poskytnuta ve volné verzi Qt a proto musí být přidána externě. Aplikace byla napsána v programovacím jazyku C++, a to z důvodu již předešlé zkušenosti s jeho použitím při práci v Qt.

Framework Qt je založen na objektově orientované hierarchii, která vychází a dědí z třídy `QObject`. Tato třída implementuje model zajišťující komunikaci mezi objekty v aplikacích. Tento model se nazývá *signal – slot* a je popsán v sekci 2.3.1 uvedené níže. Uživatelské rozhraní je generováno za použití dalších tříd v Qt. Hlavní okno aplikace je vytvořeno třídou `QWidget`. Text, zobrazovaný v okně aplikace, je vykreslován pomocí třídy `QLabel` a tlačítka jsou vytvářena za použití třídy `QPushButton`. Informace jsou čerpány z dokumentace Qt [22], ve které jsou popsány i další použité třídy. Dále jsou uvedeny důležité třídy doplněné knihovnou QMQTT.

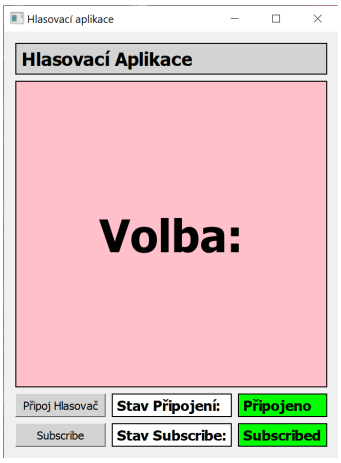
`QMqttClient`	Je třída vytvářející klienta a jeho připojení k požadovanému serveru MQTT.
`QMqttSubscription`	Je třída poskytující správu nad propojením klienta s konkrétním tématem. Při provedení odběru zpráv klientem je vytvořena instance této třídy, která obdrží odebírané zprávy prostřednictvím signálu `messageReceived()`.

Systém Signal-Slot

Tento model umožňuje komunikaci mezi jednotlivými objekty ve frameworku Qt. Je to sada funkcí, které se mohou vzájemně propojit. Funkce typu *signál* jsou volány při určitých událostech jako například stisknutí tlačítka nebo vybrání položky z listu. Funkce typu *slot* fungují jako odezva na konkrétní signály. Signály a sloty si po propojení mohou předávat proměnné a jejich propojení je provedeno funkcí `connect()`. Většina tříd v Qt obsahuje mnoho signálů i slotů, které jsou použity ke návrhu a fungování aplikací.

2.3.2 První verze aplikace

Hlavním motivem v počátku návrhu řídicí jednotky bylo vytvořit fungující komunikační spojení s hlasovací jednotkou. Grafické rozhraní sestává z dvou tlačítek a velké zobrazovací plochy, na které jsou prezentovány přijaté zprávy. Tlačítka slouží pro připojení aplikace k MQTT serveru a přihlášení odběru zpráv od tohoto serveru. MQTT server je již propojen s hlasovací jednotkou a na konkrétní téma může publikovat zprávy odběratelům. První verze aplikace je k nahlédnutí na obrázku č. 2.5. Pro otestování fungování komunikačního rozhraní a jeho rozchození byla první verze aplikace vyhovující. Implementace MQTT komunikace však nebyla v této verzi dostatečná k rozlišení zpráv od více koncových jednotek. Tento problém je řešen v následující sekci, kde je popsán i celý proces propojení aplikace se serverem MQTT a jeho tématy.



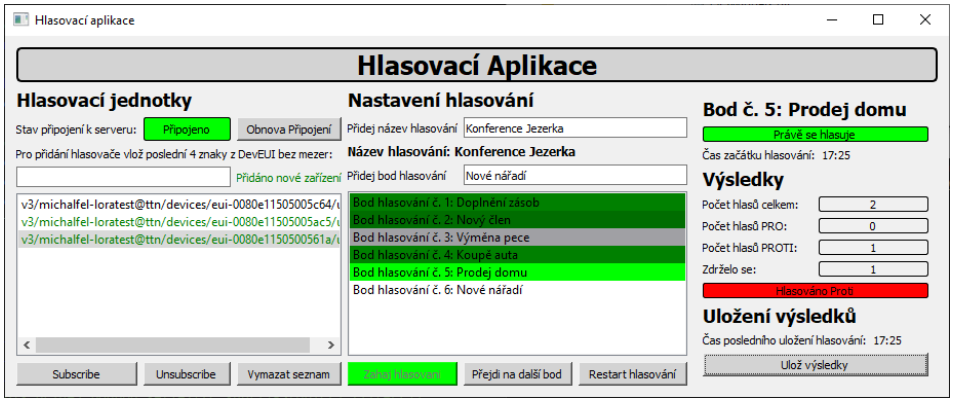
Obrázek 2.5: Centrální jednotka verze č.1

2.3.3 Druhá verze aplikace

Řídící aplikace verze č. 2 řeší problematiku rozlišení zpráv od více hlasovacích jednotek a obsahuje propracovanější grafické rozhraní s kompletní sadou možností pro uskutečnění hlasování. Kód aplikace je lépe strukturalizován do více tříd oproti první verzi. Tyto třídy obsahují specifické části implementace jako například správu klienta MQTT, volbu hlasovacích jednotek nebo okno s podotázkami k hlasování. V této části je nejprve naznačena finální podoba řídicí aplikace na obrázku č. 2.6. Dále je uveden popis všech segmentů řídicí aplikace spolu s dílčími částmi kódu. Celý zdrojový kód je k nahlédnutí v přílohách této práce.

2.3.4 Uživatelské okno

Uživatelské okno je rozděleno do více částí. K tomuto rozvržení je využita třída `QGridLayout`, která vytváří mřížkový layout oken. Aplikace je tímto layoutem strukturována do 3 sloupců s více řádky. První část, na kterou uživatel narazí vlevo, zajišťuje propojení aplikace s MQTT serverem a připojení hlasovacích jednotek. Druhá část uprostřed nabízí volbu názvu hlasování a možnost správy různých podbodů, které se budou po spuštění hlasování projednávat. Třetí část aplikace napravo zobrazuje informace o konkrétních podbodech hlasování.



Obrázek 2.6: Centrální jednotka verze č.2

2.3.5 Komunikace aplikace s MQTT

Hlavní část kódu propojující aplikaci a MQTT server je vedena ve třídě *Client*. Tato třída obsahuje implementaci klienta MQTT komunikace a vykreslení levé části uživatelského okna. Klient MQTT je vytvořen třídou `QMqttClient`. Po spuštění aplikace dojde

k zavolání vytvořeného slotu `Client::connect_to_server`, kterým se provede připojení klienta k serveru. Slot `Client::connect_to_server` je ukázán v kódu č. 2.2. Do klienta jsou uloženy: přihlašovací údaje, veřejná adresa MQTT serveru a jeho port. Tento slot může být znovu zavolán tlačítkem v horní části okna. Vedle tohoto tlačítka je umístěn informační text, který ukazuje aktuální status připojení se serverem. Aktualizace stavu připojení je zajištěna signálem třídy `QMqttClient::stateChanged`. Tento signál je propojen se slotem `Client::check_connection_status`, který provádí vypsání aktuálního stavu do informačního textového pole. Zprávy s hlasy jsou z MQTT serveru přijímány signálem `QMqttSubscription::messageReceived`. Data zpráv jsou uložena ve třídě `QMqttMessage`. Z této třídy je payload zprávy převeden na řetězec znaků, se kterým jsou porovnány řetězce reprezentující volené hlasy. Tímto způsobem jsou vyhodnoceny zaslané hlasy přenesené zprávou z MQTT. Dále je emitován signál `Client::push_message`, kterým se hlasy zapíšou do výsledků. Tento proces je popsán v sekci 2.3.8.

```
1  void connect_to_server(){
2    client->setHostname(Hostname);
3    client->setPort(port);
4    client->setUsername(Username);
5    client->setPassword(Password);
6    client->connectToHost();
7    QObject::connect(client, &QMqttClient::stateChanged, this, &Client::
     check_connection_status); //  indication of client connection state
8  }
```

Kód 2.2: Slot connect_to_server() třídy Client

2.3.6 Připojení hlasovacích jednotek

Další část okna zabírá kolonka pro vložení klíče DevEUI, rozlišujícího rozdílné hlasovací jednotky. Tento jednořádkový editor textu je vytvořen třídou `QLineEdit` a umožňuje vložení řetězce znaků do aplikace. Protože se klíče DevEUI liší jen v posledních čtyřech znacích, stačí vkládat do aplikace jen tyto čtyři znaky. Program následně doplní zbývající část klíče a vytvoří z něj řetězec znaků reprezentující *topic* konkrétní hlasovací jednotky. Přidaný *topic* je zobrazen v listu položek, který je vytvořen za pomoci třídy `QListWidget`. Před zahájením hlasování je nutné provést přihlášení k odběru zpráv hlasovacích jednotek, a to výběrem jejich *topicu* a stiskem tlačítka *Subscribe*. Tlačítko *Unsubscribe* analogicky zruší odběr zpráv od hlasovací jednotky, pro vybraná hlasování. Tlačítko *Vymazat seznam* smaže všechny uložené hlasovací jednotky v položkovém listu.

2.3.7 Nastavení parametrů hlasování

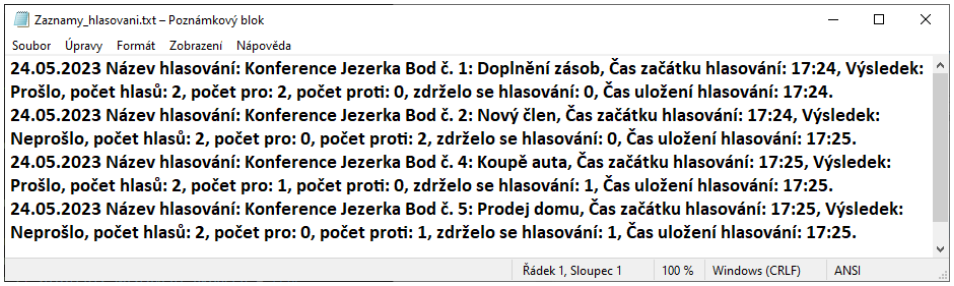
Druhá část okna aplikace poskytuje nastavení pro budoucí hlasování. První kolonka slouží k jeho pojmenování. Druhá kolonka dovoluje vytváření nových podbodů, které jsou následně zobrazeny v listu položek níže v aplikaci. Pokud je vytvořen alespoň jeden bod jednání, tak může být tlačítkem níže spuštěno hlasování. Na další body jednání se přechází tlačítkem uprostřed *Přejít na další bod*. Při stisku tlačítka *Restart hlasování* se vymažou všechny podbody a může se tak založit nový seznam. Provedením dvojkliku na položku v seznamu mohou být podbody během hlasování přeskočeny.

2.3.8 Výsledky hlasování

Výsledky hlasování jsou zobrazeny v pravé části řídicí aplikace. Podbody jsou vytvářeny spolu s třídou `Result_Widget`, která dědí z třídy `QWidget`. Díky tomu každá instance této třídy obsahuje všechny výsledky aktuálního podbodu a jejich vykreslení do okna. Zobrazení jen jednoho okna v danou chvíli je umožněno díky třídě `QStackedWidget`. Tato třída zobrazí konkrétní výsledky v závislosti na vybrané položce ze seznamu podbodů. Použití této třídy je inspirováno ze zdroje [24] s podobným tématem realizace hlasovací aplikace. Třída `Result_Widget` obsahuje funkci `Save_data_to_point()`, která v reálném čase ukládá přijatá data ze signálu `Client::push_message` a vypisuje je do výsledků hlasovaného podbodu.

2.3.9 Uložení výsledků

Ukládání výsledků je implementováno do třídy `Save_results`. Tato třída je obsažena v každé instanci třídy `Result_Widget`, a tím je pro každý podbod hlasování zajištěno uložení výsledků. Uložená data jsou prezentována v textovém dokumentu *Zaznamy_-hlasovani.txt*. Ve spodní části okna s výsledky je umístěno tlačítko *Ulož výsledky*. Se stiskem tohoto tlačítka je vygenerován signál `QPushButton::clicked`, kterým je otevřen dokument s výsledky a jsou do něj uloženy důležité údaje o aktuálním hlasování. Příklad uložených hlasování je ukázán na obrázku č. 2.7.



Obrázek 2.7: Výpis uložených výsledků

2.4 Zabezpečení hlasování

V současné době je nutné, aby veškerá elektronická zařízení byla zabezpečena proti zneužití osob třetích stran. V IoT se proto musí také počítat s zajištěním bezpečnosti přenosu zpráv. Pro hlasovací systémy to platí dvojnásobně, protože je to velmi citlivou záležitostí. Při použití elektronických hlasovacích zařízení v anonymních volbách je velmi důležité zaručení anonymity voličů a jejich zaznamenaných hlasů. Dále nesmí nastat průnik neplatných hlasů, které by se započítaly do výsledku.

Zabezpečení zpráv hlasovacího systému je již implementováno v technologii LoRaWAN. Komunikační propojení je zabezpečeno dvojicí klíčů. První klíč *NwkSKey* slouží k ověření zprávy a jejího vysílatele. Druhý klíč *AppSKey* slouží k zašifrování datové části zprávy. Je proto obtížné s takto zabezpečenou zprávou manipulovat.

Dalším nebezpečím hlasovacího systému je zachycení zpráv narušiteli, kteří mohou zprávu opětovně vysílat. LoRaWAN již obsahuje implementaci řešení tohoto problému. S každým *uplinkem* a *downlinkem* se provede přičtení hodnoty odeslaných a přijatých zpráv v čítači. Pokud dojde k přijetí zprávy, která má oproti té minulé nižší hodnotu počtu vysílaných zpráv, je tato přijatá zpráva ignorována. Informace jsou čerpány z [25].

Další možností úniku informace je při MQTT komunikaci. K zabezpečení MQTT komunikace je vhodné použít zašifrování datové zprávy formátovačem služby TTN, který je více popsán v sekci 2.1.2. Přeposlaná data mohou být zašifrována šifrovacím kódem, který následně aplikace dešifruje a vyhodnotí. Používaný MQTT server je sám navíc zabezpečen proti vniknutí vygenerovanými přihlašovacími údaji. Přístup k tomuto serveru má jen přihlášená centrální řídicí jednotka hlasování a služba TTN.

Závěr

Cílem této bakalářské práce byla realizace bezdrátového hlasovacího zařízení. Tento cíl jsem splnil ve všech bodech zadání. Provedl jsem rešerši LPWAN, a to tří dostupných technologií Sigfox, NB-IoT a LoRa. Na základě této rešerše jsem jako bezdrátovou komunikační technologii hlasovacích jednotek stanovil modulační techniku LoRa, protože nabízí nejjednodušší a nejdostupnější implementaci koncových zařízení. Vytvořil jsem dvě koncové hlasovací jednotky, realizované mikrokontrolérem Nucleo-WL55JC, které umožňují záznam tří stavů: pro, proti a zdržel se. Jednotky jsou doplněny externím plošným spojem s tlačítky záznamu hlasování a informační LED. Naprogramoval jsem řídicí centrální jednotku hlasování jako spustitelnou aplikaci, která přijímá zvolené hlasy z koncových jednotek. Aplikace umožňuje plánování různých hlasování, vyhodnocení provedených hlasování a uložení jejich výsledků. Nakonec jsem popsal a zhodnotil problematiku zabezpečení hlasování v dílčích blocích vytvořeného komunikačního rozhraní.

Při práci na tomto projektu jsem se potýkal s různými problémy, které se podařilo vyřešit. Nejobtížnější částí realizace bylo rozklíčování kódu koncových jednotek Nucleo-WL55JC a následná úprava jejich funkcí zajišťujících správný chod při hlasování.

Další rozšíření vytvořeného konceptu hlasovacího zařízení by mohla být implementace zpětné vazby z centrální jednotky do koncových hlasovacích jednotek. Dále také vytvoření vhodného krytu koncových jednotek a zajištění integrovaného bateriového napájení.

Literatura

[1] MADAKAM, Somayya, R. RAMASWAMY a Siddharth TRIPATHI, 2015. Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*. 03(05), 164-173. ISSN 2327-5219. Dostupné z: doi:10.4236/jcc.2015.35021

[2] AL-QASEEMI, Sarah A., Hajer A. ALMULHIM, Maria F. ALMULHIM a Saqib Rasool CHAUDHRY, 2016. *IoT architecture challenges and issues: Lack of standardization.* 2016 Future Technologies Conference (FTC). IEEE, 2016, 731-738. ISBN 978-1-5090-4171-8. Dostupné z: doi:10.1109/FTC.2016.7821686

[3] NAVANI, Deepika, Sanjeev JAIN a Maninder Singh NEHRA, 2017. *The Internet of Things (IoT): A Study of Architectural Elements.* IEEE, 2017, 473-478. ISBN 978-1-5386-4283-2. Dostupné z: doi:10.1109/SITIS.2017.83

[4] THE TECH PLATFORM, *IoT : Protocol and Architecture.* The Tech Platform [online]. Listopad 2021 [cit. 2023-04-27]. Dostupné z: https://www.thetechplatform.com/post/iot-protocol-and-architecture

[5] ZHENHUI, Li, 2020. *Data Processing Strategy in Internet of Things Application System.* IEEE, 2020, 86-89. ISBN 978-1-7281-9619-0. Dostupné z: doi:10.1109/ICBASE51474.2020.00026

[6] NIE, Zehua, Can SU, Yichen MAO a Kaigui BIAN, 2022. *IoTPass: IoT Data Management System for Processing Time-series Data.* IEEE, 5. 11. 2022, 288-293. ISBN 979-8-3503-0971-3. Dostupné z: doi:10.1109/CBD58033.2022.00058

[7] NAIK, Nitin, 2018. *LPWAN Technologies for IoT Systems: Choice Between Ultra Narrow Band and Spread Spectrum.*
2018 IEEE International Systems Engineering Symposium (ISSE).
IEEE, 29. 11. 2018, 1-8. ISBN 978-1-5386-4446-1.
Dostupné z: doi:10.1109/SysEng.2018.8544414

[8] HERRERO, Rolando. *Fundamentals of IoT Communication Technologies.*
Cham, Switzerland: Springer, 2021. Textbooks in telecommunication engineering.
ISBN 978-3-030-70079-9.

[9] SIGFOX, 2017. *Sigfox Technical Overview.* Francie, Květen 2017.
Dostupné z: https://storage.googleapis.com
/public-assets-xd-sigfox-production-338901379285/build_technicalOverview.pdf

[10] BEYENE, Yihenew Dagne, Riku JANTTI, Olav TIRKKONEN, Kalle RUTTIK, Sassan IRAJI, Anna LARMO, Tuomas TIRRONEN a Johan TORSNER.
NB-IoT Technology Overview and Experience from Cloud-RAN Implementation.
IEEE Wireless Communications. 24(3), 26-32.
ISSN 1536-1284. Dostupné z: doi:10.1109/MWC.2017.1600418

[11] IQBAL, Mehzabien, Abu Yousha Md ABDULLAH a Farzana SHABNAM, 2020.
An Application Based Comparative Study of LPWAN Technologies for IoT Environment. IEEE, 2020, 1857-1860.
ISBN 978-1-7281-7366-5. Dostupné z: doi:10.1109/TENSYMP50017.2020.9230597

[12] SEMTECH. *LoRa Modulation Basics.* 2015. Dostupné z:
https://web.archive.org/web/20190718200516/https:
//www.semtech.com/uploads/documents/an1200.22.pdf

[13] ZLEVOR, Jan. *Využití SDR v LoRaWAN sítích.* Praha, 2021. Bakalářská práce.
ČVUT v Praze, Fakulta elektrotechnická, Katedra radioelektroniky.
Dostupné z: http://hdl.handle.net/10467/96704

[14] THE THINGS NETWORK. Regional Parameters [online] [cit. 2023-05-16].
Dostupné z: https://www.thethingsnetwork.org/docs/lorawan/regional-parameters/

[15] MONTAGNY, Sylvain, 2022.*LoRa - LoRaWAN and IoT* [online].
Savoie Mont Blanc University [cit. 2023-05-11].
Dostupné z: https://www.univ-smb.fr/lorawan/en/free-book/

[16] LORA ALLIANCE, 2017. *LoRaWAN Specification v1.1* [online] [cit. 2023-05-11]. Dostupné z: https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/

[17] THE THINGS NETWORK. *LoRaWAN Architecture* [online] [cit. 2023-05-12]. Dostupné z: https://www.thethingsnetwork.org/docs/lorawan/architecture/

[18] HIVEMQ. *MQTT & MQTT 5 Essentials*. 2020. Ergoldinger Str. 2A 84030 Landshut Germany. ISBN 978-3-00-067913-1. Dostupné z: https://www.hivemq.com/downloads/hivemq-ebook-mqtt-essentials.pdf

[19] PAESSLER, The monitoring experts. *IT Explained: MQTT* [online] [cit. 2023-05-08]. Dostupné z: https://www.paessler.com/it-explained/mqtt

[20] THE THINGS NETWORK. *TTN Mapper* [online] [cit. 2023-05-19]. Dostupné z: https://ttnmapper.org/heatmap/

[21] STMICROELECTRONICS. *STM32WL Nucleo-64 board (MB1389) (UM2592).* Dostupné z: https://www.st.com/en/evaluation-tools/nucleo-wl55jc.html

[22] THE QT COMPANY. *Qt Documentation* [online] [cit. 2023-04-22]. Dostupné z: https://doc.qt.io/

[23] MINISTR SYSTEMS. *Hlasovací systémy Ministr* [online] [cit. 2023-05-25]. Dostupné z: https://www.ministr.cz/hlasys.html

[24] VYBÍRAL, Tomáš. *Hlasovací zařízení na bázi Bluetooth Low Energy* [online] [cit. 2023-05-24]. Dostupné z: http://hdl.handle.net/11012/69799

[25] THE THINGS NETWORK. *LoRaWAN Security* [online] [cit. 2023-05-23]. Dostupné z: https://www.thethingsnetwork.org/docs/lorawan/security/

Přílohy

Práce je doplněna o soubor obsahující zdrojový kód obou verzí aplikace, hlavní část zdrojového kódu hlasovací jednotky a schéma navržené externí desky. Struktura souboru je uvedena níže:

```
haslmich_BP_priloha.zip
├── 01_Central_unit ..... Zdrojový kód a projekt první verze aplikace
│   ├── 01_Centralni_jednotka.pro
│   └── 01_Centralni_jednotka.cpp
├── 02_Central_unit ..... Zdrojový kód a projekt druhé verze aplikace
│   ├── 02_Centralni_jednotka.pro
│   ├── Headers ..... Zdrojové soubory
│   └── Sources ..... Hlavičkové soubory
├── PCB_shield ..... Soubory návrhu PCB
│   ├── Shield for Wireless Vote Device.pdf
│   ├── Shield Uno.kicad_pcb
│   ├── Shield Uno.kicad_sch
│   └── Shield Uno.png
└── Vote_unit
    └── lora_app.c ..... Hlavní část zdrojového kódu hlasovací jednotky
```