



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

DEPARTMENT OF COMPUTER SYSTEMS

**KOMPAKTNÍ SYSTÉM PRO AUTONOMNÍ ODEČET  
MĚŘIDEL**

COMPACT SYSTEM FOR AUTONOMOUS METER READING

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**DAVID PODESZWA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**doc. Ing. ZDENĚK VAŠÍČEK, Ph.D.**

BRNO 2022

## Zadání bakalářské práce



Student: **Podeszwa David**  
Program: Informační technologie  
Název: **Kompaktní systém pro autonomní odečet měřidel**  
**Compact System for Autonomous Meter Reading**  
Kategorie: Vestavěné systémy

### Zadání:

1. Seznamte se s protokolem Wireless MBUS používaným pro bezdrátový odečet stavu měřidel jako jsou např. vodoměry, kalorimetry, apod. a dále bezdrátovým přenosovým protokolem LoRa a technologií LoRaWAN.
2. Na základě rešerše na trhu dostupných řešení navrhnete bateriově napájené zařízení, který bude periodicky provádět odečet měřidel a data odesílat prostřednictvím sítě LoRaWAN. Zaměřte se na minimalizaci počtu komponent a ceny řešení.
3. Zpracujte studii na téma uvedené v bodě 1 a 2 zadání.
4. Navržený zařízení implementujte na vhodně zvolené platformě a to např. s využitím vývojových kitů (STM32WL apod.). Při návrhu se pokuste minimalizovat spotřebu. Pro konfiguraci zařízení zvolte vhodnou techniku bezdrátové komunikace.
5. Činnost zařízení ověřte a diskutujte dosažené parametry, zejména spotřebu.

### Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů č. 1 a 2 zadání.

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

Vedoucí práce: **Vašíček Zdeněk, doc. Ing., Ph.D.**

Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 29. října 2021

## Abstrakt

Tato práce řeší problém s nedostatečným dosahem měřidel WMBUS, díky čemuž je jejich odečet stále složitý. Vytvořená realizace tento problém řeší použitím baterií napájeného zařízení, jenž tento dosah rozšíří použitím sítě LoRaWAN, kdy jsou touto sítí přijatá data z měřidel odesílána dále ke zpracování. Vytvořené zařízení je schopné příjmu WMBUS rámců a jejich následným odesláním po minimální dobu 9 let. Hlavním z přínosů je implementace takového zařízení na nové platformě STM32WL, jenž je díky integraci mikroprocesoru a Sub-GHz přijímače v jednom čipu unikátní.

## Abstract

This work solves the problem of the insufficient range of WMBUS meters, which makes their reading still difficult. The developed implementation solves this problem by using a battery-powered device that extends this range by using a LoRaWAN network, where the data received from the meters is sent over this network for further processing. The created device is capable of receiving WMBUS frames and sending them for a minimum period of 9 years. The main benefit of this work is the implementation of such a device on the new STM32WL platform, which is unique due to its integration of a microprocessor and a Sub-GHz receiver inside one chip.

## Klíčová slova

mikrokontrolér, mikroprocesor, vodoměry, vzdálený odečet, WMBUS, STM32WL, LoRa, LoRaWAN, NFC, NDEF

## Keywords

microcontroller, microprocessor, water meters, remote reading, WMBUS, STM32WL, LoRa, LoRaWAN, NFC, NDEF

## Citace

PODESZWA, David. *Kompaktní systém pro autonomní odečet měřidel*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Zdeněk Vašíček, Ph.D.

# Kompaktní systém pro autonomní odečet měřidel

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Zdeňka Vašíčka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....  
David Podeszwa  
9. května 2022

## Poděkování

Rád bych chtěl poděkovat mému vedoucímu doc. Ing. Zdeňkovi Vašíčkovi, Ph.D. za vedení práce a také za odborné konzultace poskytnuté hlavně v případě návrhu plošných spojů. Dále také své rodině za psychickou podporu v době psaní této práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Technologie bezdrátového odečtu WM-BUS</b>	<b>4</b>
2.1	Popis normy . . . . .	4
2.2	Režimy komunikace . . . . .	5
2.3	Fyzická vrstva . . . . .	6
2.4	Typy kódování . . . . .	8
2.5	Formáty rámců . . . . .	9
2.5.1	Rámec A . . . . .	9
2.5.2	Rámec B . . . . .	10
2.5.3	Popis polí rámců . . . . .	10
<b>3</b>	<b>Bezdrátová síť LoRaWAN</b>	<b>11</b>
3.1	Pokrytí v ČR . . . . .	12
3.2	Architektura sítě . . . . .	13
3.3	Třídy zařízení . . . . .	14
3.4	Zabezpečení . . . . .	16
3.5	Aktivace zařízení v síti . . . . .	16
3.5.1	APB - Aktivace personalizací . . . . .	16
3.5.2	OTAA . . . . .	17
3.6	Maximální velikost paketů . . . . .	17
<b>4</b>	<b>Komunikace na krátkou vzdálenost NFC</b>	<b>18</b>
4.1	Princip funkce . . . . .	18
4.2	Operační režimy . . . . .	19
4.2.1	Pasivní . . . . .	19
4.2.2	Aktivní . . . . .	19
4.3	Režimy komunikace . . . . .	19
4.3.1	Režim Čtení/Zápis . . . . .	19
4.3.2	Režim peer-to-peer . . . . .	19
4.3.3	Režim emulace karet . . . . .	19
4.4	Formát NDEF . . . . .	20
<b>5</b>	<b>Platformy dostupné pro implementaci</b>	<b>21</b>
5.1	STM32WLXX . . . . .	21
5.2	ESP32 . . . . .	23
<b>6</b>	<b>Návrh</b>	<b>24</b>

6.1	Současná řešení . . . . .	24
6.2	Obecný návrh . . . . .	25
6.3	Návrh zařízení . . . . .	26
6.4	Návrh konfigurační aplikace . . . . .	28
<b>7</b>	<b>Implementace</b>	<b>30</b>
7.1	Příjem WMBUS rámců . . . . .	31
7.2	Obsluha LoRaWAN sítě . . . . .	33
7.3	NFC rozhraní . . . . .	34
7.4	Ukládání do paměti FLASH . . . . .	35
7.5	Nastavení zařízení . . . . .	35
7.6	Hlavní smyčka programu . . . . .	36
7.7	Implementace webové aplikace . . . . .	37
	7.7.1 Uživatelské rozhraní . . . . .	37
	7.7.2 JavaScript komunikace NFC . . . . .	40
7.8	Hardware zařízení . . . . .	40
<b>8</b>	<b>Vyhodnocení</b>	<b>44</b>
<b>9</b>	<b>Závěr</b>	<b>49</b>
	<b>Literatura</b>	<b>50</b>
<b>A</b>	<b>Obsah přiloženého paměťového média</b>	<b>52</b>

# Kapitola 1

## Úvod

V dnešní době rozšiřujících se technologií a automatizace se snažíme udělat naše životy jednodušší. Ať už to jsou méně komplexní věci jako automatické dveře v supermarketech, nebo ty složitější jako jsou chytré domácnosti. Díky neustálé nutnosti pro zefektivnění práce tak vznikly také chytré měřidla. Ty umožňují poskytovatelům komunálních služeb odečet spotřeby z našich elektroměrů, vodoměrů či podobných odečtových zařízení na dálku bez nutnosti vstoupit do bytu zákazníka.

Tyto měřidla již velmi zjednodušují odečet už teď, avšak i přesto mají jeden velký nedostatek. Jejich dosah stále vyžaduje vyslání odečítající osoby do okruhu zhruba 50 metrů od měřidla.

Cíl této práce je tedy prodloužit jejich dosah pomocí zařízení, které bude přijímat informace z těchto měřidel a odesílat je dále pomocí sítě s daleko větším dosahem. Díky tomu nebude nutné aby byli zaměstnanci dodavatelů energií a služeb posíláni do terénu k odečtu. Protože je nutné aby bylo zařízení co nejvíce soběstačné, bude napájeno baterií zajišťující provoz po 7-10 let. Ke konfiguraci a instalaci nebude technik potřebovat žádný speciální nástroj a postačí si pouze s mobilním telefonem jenž má podporu bezdrátové komunikace NFC.

K výběru tématu mě vedl můj zájem o Internet věcí a chytrých zařízení. Víím, že tento obor roste velmi rychle a jakákoliv práce v něm má obrovský potenciál zlepšit nám všem život.

Struktura práce je tedy následující. V druhé kapitole se seznámíte s informacemi o technologii Wireless MBUS, jak tato technologie funguje, systém adresace měřidel a jak vypadají zasílané data. V kapitole č. 3 se nachází informace o síti LoRaWAN, konkrétně rozdělení do tříd a architekturu sítí. Také je popsána komunikace technologií NFC, a to v kapitole 4. Nadále jsou v kapitole 6 popsány požadavky a návrh finálního zařízení, následuje implementace zařízení popsané kapitolou 7. Poslední kapitolou s číslem 8 je vyhodnocení výsledků práce.

## Kapitola 2

# Technologie bezdrátového odečtu WM-BUS

Díky rychlé expanzi Internetu věcí a snižující se ceně vestavných systémů začalo být pro dodavatele služeb velice výhodné pořizování senzorů a zařízení s bezdrátovou komunikací ve větším množství.

Na trh se začaly dostávat chytré vodoměry, které umožňují odečet bez navštívení zákazníka. Evropskou unií byl tedy navržen standard WM-BUS, který je definován normou EN 13757-4. Jedná se o bezdrátovou evoluci již dřívějšího standardu MBUS, jenž ovšem používá k přenosu pouze fyzické médium [11].

### 2.1 Popis normy

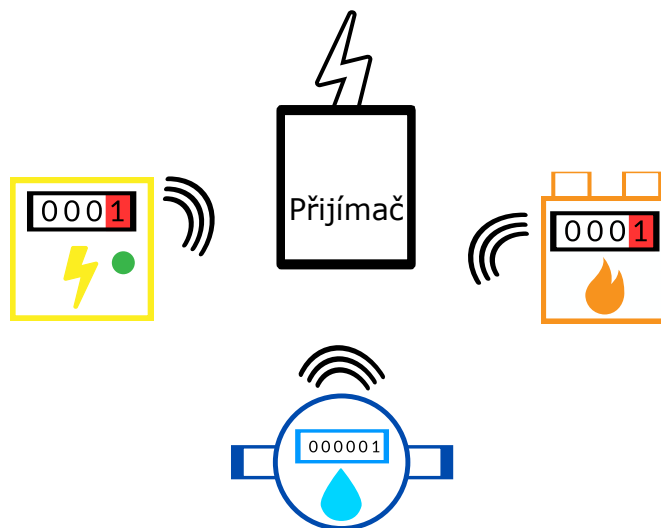
Wireless M-BUS (wM-BUS) je otevřený standard vytvořený za účelem bezdrátového sběru dat a jeho výhodou je velmi nízká spotřeba. Velmi rychle se šíří Evropou, a nejčastější využití má právě ve vodoměrech. Jedná se o standard využívající pásmo sub-1Ghz<sup>1</sup>, což je skvělý kompromis mezi dosahem přenosu a také velikostí potřebné antény [12]. Konkrétně využívá frekvence v nelicencovaných pásmech a tudíž není nutné získávat žádná povolení pro jejich operaci. Standard definuje způsob komunikace na fyzické vrstvě a také formát data-link vrstvy. Obsah aplikační vrstvy není standardem specifikován, je pouze popsáno jeho rozdělení do bloků dle typu rámce viz. 2.5

---

<sup>1</sup>Frekvence do 1Ghz



Obrázek 2.1 zobrazuje hvězdicovou topologii standardu WM-BUS.



Obrázek 2.1: Hvězdicová topologie WM-BUS

## 2.2 Režimy komunikace

Samotný standard definuje několik možných režimů komunikace. Každý z nich má konkrétní účel a využívá konkrétní frekvenci. Část parametrů, jako například frekvence je u některých režimů stejná, a umožňuje tak souběžným příjem dvou režimů zároveň za použití stejného hardware. Dále se pak liší použitým typem kódování. Posledním rozdílem je typ rámce (viz 2.5) přičemž jeden z režimů umožňuje použití obou typů rámců a konkrétní typ je určen použitým synchronizačním slovem viz 2.3.

Tabulka 2.1: Režimy Wireless M-BUS [13]

Režim	Frekvence	Kódování	Popis
S	868,3	Manchester	Stacionární zařízení. Odesílá párkrát za den
T	868,95 / 868,3	3-z-6	Časté odesílání malých rámců s krátkou periodou.
R	868,8	Manchester	Režim určený pouze k obousměrnému provozu.
N	169	NRZ	Úzkopásmový režim. Určen pro větší dosah
C	868,95	NRZ	Kompaktní. Obdoba režimu T, avšak umožňuje rychlejší přenos a větší velikost rámce.
F	433	NRZ	Častá komunikace dvěma směry. Určen pro větší dosah

Každý z režimů může podporovat jak jedno nebo obousměrnou komunikaci. Za identifikačním písmenem režimu se nachází číslice, která určuje, zda je komunikace jednosměrná

(1), nebo obousměrná (2).

Podrobnější popis jednotlivých režimů: [13]

- **S** - Stacionární režim - Je určen pro jednosměrnou i obousměrnou komunikaci mezi měřičem a odečítacím zařízením jež může být stacionární či v pohybu. Speciálním podrežimem je režim S1 jenž je optimalizován pro stacionární zařízení napájená baterií a vysílající delší hlavičku v rámci.
- **T** - Režim častého vysílání - V tomto režimu měřič vysílá velmi krátký rámec (typicky s délkou 3 až 8 ms) v intervalech několika sekund. Tím je umožněn odečet pouhým průjezdem v dosahu vysílače.  
V podrežimu T1 měřič periodicky odesílá pouze svou adresu a data aplikační vrstvy. Ve speciálním podrežimu T2 je komunikace zahájena odesláním krátkého rámce obsahující adresu měřiče. Druhé zařízení pak může reagovat odesláním
- **R** - Režim častého příjmu - V tomto režimu je relevantní pouze jeho podrežim R2 (podrežim R1 neexistuje z toho důvodu že tento režim slouží pouze na oboustrannou komunikaci). Měřič je periodicky probouzen a očekává přijetí tzv. wakeup rámce od odečítatele, po kterém následuje obousměrná komunikace se zařízením.
- **C** - Kompaktní režim - Tento režim je velmi podobný režimu T, avšak hlavním rozdílem je použití kódování NRZ (viz. 2.4) což umožňuje odeslat více dat a to za stejný přenosový čas. Díky podobným parametrům fyzické vrstvy je také možné přijímat rámce režimu C1 a T1 použitím stejného hardware a software.
- **N** - Úzkopásmový režim - Optimalizován pro použití v úzkém pásmu na frekvenci 169Mhz. V tomto režimu je možné použití repeaterů signálu pro zvětšení dosahu.
- **F** - Režim častého přijímání a odesílání - Měřič se periodicky probouzí a čeká na přijetí rámce o vzbuzení. Po přijmutí je otevřeno komunikační okno po dobu několika sekund

## 2.3 Fyzická vrstva

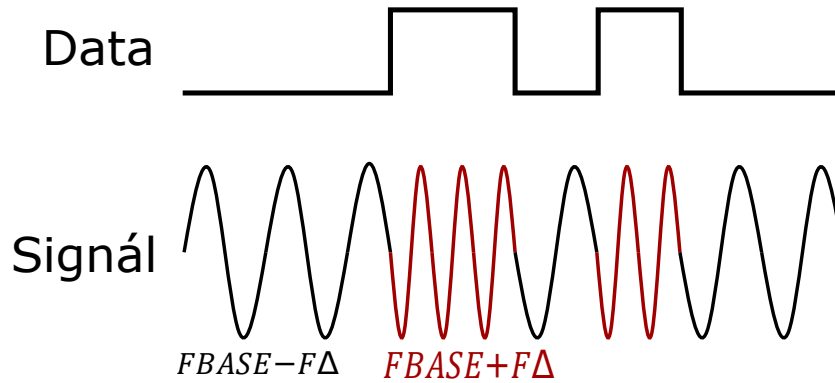
Standard pro každý režim definuje parametry bezdrátového vysílání. Nejdůležitějším parametrem této fyzické vrstvy je frekvenční pásmo. Mezi další parametry pak patří typ modulace, délka preamble a synchronizační slovo.

### Modulace

WM-BUS využívá následující modulace:

- **2-FSK** - Frequency shift-keying - jedná se o modulaci využívající posun frekvence k přenosu informací.
- **2-GFSK** - Gaussian frequency shift-keying - využívá Gaussovský filtr pro vyhlazení změn frekvencí
- **4-GFSK** - GFSK avšak se 4 stavy

Tyto modulace patří mezi digitální modulace. Ve všech výše zmíněných modulacích je k přenosu informací využit posun frekvence oproti frekvenci základní. Při přenosu je tedy logická 1 reprezentována nosnou vlnou s frekvencí  $F_1 = F_{BASE} + F\Delta$  a logická 0 nosnou vlnou s frekvencí  $F_0 = F_{BASE} - F\Delta$  kde  $F_{BASE}$  je základní frekvence a  $F\Delta$  je odchylka [23]. Fungování modulace FSK zobrazuje obrázek 2.2.



Obrázek 2.2: Nosné vlny FSK modulace

Různé WM-BUS režimy využívají rozdílné odchylky, avšak například režimy T1 a C1 mají jak frekvenci tak odchylku stejnou a je tudíž možný souběžný příjem za stejné konfigurace přijímače.

### Preamble a synchronizační slovo

Aby bylo možné zachytit přijímaný rámec správně, je nutné aby došlo k synchronizaci přijímače s vysílaným signálem. K tomuto účelu slouží preamble. Jedná se o sekvenci jedniček a nul, nejčastěji sekvence "01" opakovaných několikrát za sebou. Po detekci této sekvence dle nastavení je přijímač synchronizován se signálem a přijímač je připraven na přijetí synchronizačního slova.

Synchronizační slovo je sekvence bajtů, kterou je dále možné odlišit rámce, které chceme přijímat. Aby přijímač pokračoval v příjmu, je nutné aby bylo synchronizační slovo shodné s tím konfigurovaným.

Tabulka 2.2: Fyzická struktura Wireless M-BUS rámce

Preamble	Synchronizační slovo	Data vyšších vrstev	Postamble
(0x01) * N	0xXXXXXXXX		0x01

Synchronizační slovo WM-BUS rámců se liší dle režimu. Jeho délka může být taktéž rozdílná.

## 2.4 Typy kódování

Standard Wireless M-Bus používá tři různé typy kódování. Jejich použití se opět liší na základě použitého režimu i směru komunikace (zda se jedná o režim jednosměrný nebo obousměrný).

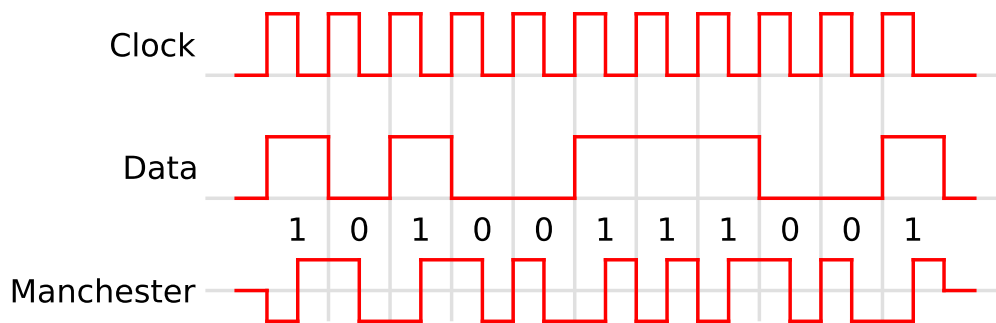
### Non Return to Zero - NRZ

V tomto kódování jsou jednotlivé bity dat reprezentovány sekvencí logických úrovní. Logická 1 zůstává logickou 1 a stejně je tak u logické 0. Nevýhodou tohoto kódování je situace kdy je přenášena sekvence neměnicích se bitů, např. "00000". Na straně přijímače totiž může dojít k desynchronizaci hodinového signálu s vysílaným signálem, protože se logická úroveň nemění.

Jeho výhodou je ovšem efektivnost. Pro každý bit dat je přenesen pouze jeden bit [3].

### Manchester kódování

Při tomto kódování dochází při přenosu k transformaci bitu na sekvenci přechodů. Logická 1 je reprezentována přechodem z 0 na 1, logická 0 pak přechodem z 1 na 0. Díky tomu dochází během přenosu k častým přechodům a přijímač má tak možnost zůstat synchronizován. Nevýhodou je však zdvojnásobení doby přenosu [6].



Obrázek 2.3: Manchester kódování. Převzato a upraveno z [https://en.wikipedia.org/wiki/Manchester\\_code#/media/File:Manchester\\_encoding\\_both\\_conventions.svg](https://en.wikipedia.org/wiki/Manchester_code#/media/File:Manchester_encoding_both_conventions.svg)

### Kódování 3-ze-6

Toto kódování je dobrým kompromisem mezi NRZ a Manchester kódováním. Pro každé 4 bity dat je dle lookup tabulky přiřazeno 6-bitové slovo. Standard pro toto kódování tuto tabulku definuje.

## 2.5 Formáty rámců

Standard pro linkovou vrstvu specifikuje dva formáty rámců, formát A a formát B. Formát přijatého rámce je detekován preambulí a synchronizačním slovem. Oba formáty rámců jsou rozděleny do bloků, kdy prvních 10 bajtů je společných, následuje CRC (Cyclic redundancy check)<sup>2</sup> a poté jednotlivé bloky, vždy o velikosti až 16 bajtů + CRC.

Všechna pole s délkou větší než 1 bajt jsou přenášena s nejméně významným bajtem první. Výjimkou jsou pole obsahující CRC a také pole vyšších vrstev následujících po poli CI jehož hodnota není ani jedna z 80<sub>h</sub>, 82<sub>h</sub>, 83<sub>h</sub>, 88<sub>h</sub> a 8F<sub>h</sub>. V tom případě je tedy u těchto vícebajtových polí prvně přenášen nejvíce významný bajt.

### 2.5.1 Rámec A

V rámci formátu A jsou dle standardu definovány 3 bloky, kdy první blok obsahuje informace linkové vrstvy, druhý informaci o obsahu dat, společně s uvedenými daty a třetí již jen data. Bloky č.1 a č.2 musí být obsaženy v rámci vždy, blok č.3 je volitelný, a v případě, že se vyskytuje vícekrát, všechny bloky č.3 krom toho posledního musí mít délku datového pole vždy 16 bajtů.

Název	L	C	M	A	CRC pole
Délka	1 bajt	1 bajt	2 bajty	6 bajtů	2 bajty

Tabulka 2.3: Rámec A, blok č.1

Název	CI	Data	CRC pole
Délka	1 bajt	1 - 15 bajtů	2 bajty

Tabulka 2.4: Rámec A, blok č.2

Název	Data	CRC pole
Délka	1-16 bajtů	2 bajty

Tabulka 2.5: Rámec A, blok č.3

---

<sup>2</sup>CRC — Cyclic redundancy check — způsob detekce chyb v datech. Za data je umístěna hodnota vypočtená z těchto dat a na straně příjemce se musí tato umístěná hodnota shodovat s vlastní vypočtenou

### 2.5.2 Rámec B

Pro tento rámec standard definuje 3 bloky. První blok opět obsahuje informace linkové vrstvy, avšak neobsahuje CRC. Definice druhého a třetího bloku se od rámce formátu A neliší, avšak CRC v druhém bloku je vypočteno z dat i prvního bloku, neboť ten vlastní CRC neobsahuje.

Název	L	C	M	A
Délka	1 bajt	1 bajt	2 bajty	6 bajtů

Tabulka 2.6: Rámec B, blok č.1

Název	CI	Data	CRC pole
Délka	1 bajt	1 - 115 bajtů	2 bajty

Tabulka 2.7: Rámec B, blok č.2

Název	Data	CRC pole
Délka	1-126 bajtů	2 bajty

Tabulka 2.8: Rámec B, blok č.3

### 2.5.3 Popis polí rámců

#### Pole L

Toto pole udává délku rámce. V závislosti na formátu rámce má jiný dopad.

V případě formátu A délka obsažená v tomto poli nezahrnuje délku CRC polí a sebe sama. U rámců formátu B obsahuje délku všech bajtů rámce krom sebe sama.

#### Pole C

Definuje typ rámce.

#### Pole M

Pokud je nejvíce významný bit tohoto pole roven 1, je adresa obsažená v poli A globálně unikátní, v opačném případě by měla být unikátní v maximálním vysílacím dosahu měřiče. Dalších 15 bitů pole určuje výrobce měřiče. Těchto 15 bitů je rozdělených do tří 5-bitových slov, kdy každé reprezentuje znak A-Z dle ISO/IEC 646.

#### Pole A

Obsahuje adresu měřiče. Ta se skládá z 4 bajtů unikátního ID zařízení, verze a typu měřiče.

#### CRC pole

Cyklický redundantní součet je počítán z dat bloku s použitím následujícího CRC polynomu a výchozí hodnotou 0.

$$x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$$

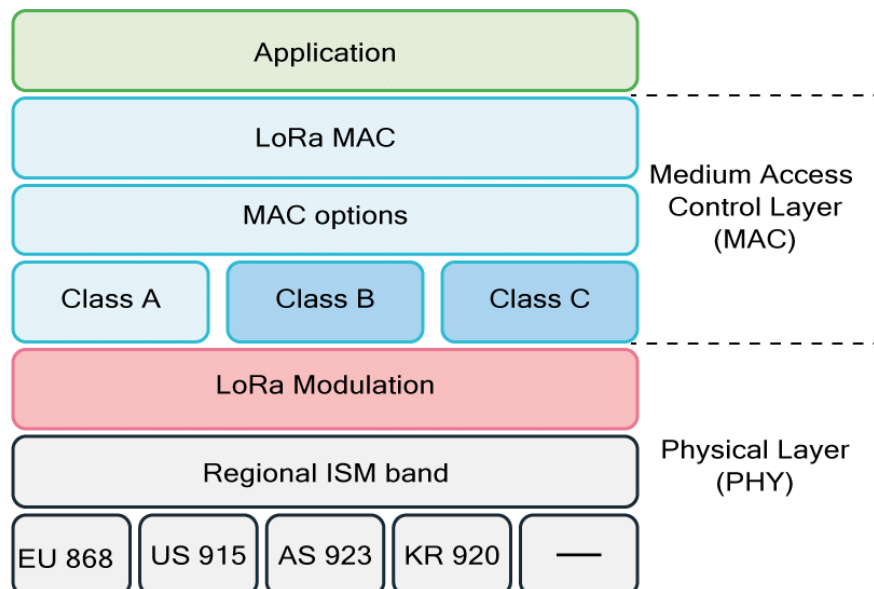
## Kapitola 3

# Bezdrátová síť LoRaWAN

LoRaWAN je protokol pro bezdrátovou síť v kategorii tzv. LPWAN (Low-power Wide-area Network). Jde o IoT (Internet věcí) síť vytvořenou s ohledem na nízkou energetickou náročnost a zároveň velkým dosahem včetně zastavěných oblastí. Díky speciální modulaci vytvořené přímo pro tuto síť je také schopna fungování v oblastech s vyšším rušením.

Přenášená data obvykle mívají délku v řádu desítek bajtů což je dostatek prostoru pro odesílání informací z běžných senzorů či zasílání příkazů koncovým zařízením. Díky takovýmto nízkým objemům dat a také intervalu odesílání jsou baterie napájená zařízení typicky schopna provozu několika let [9].

Protokol je spravován neziskovou organizací LoRa Alliance jejímž zakladatelem je společnost Semtech [24]. Samotný definuje pouze softwarovou vrstvu LoRaWAN MAC. Ta kontroluje připojení a přihlášení zařízení do sítě, odesílání a přijímání zpráv aplikačních vrstev či změny tříd zařízení. Veškerý provoz na této síti je také zabezpečen šifrováním a je tudíž vhodný pro komerční využití v průmyslu [10].

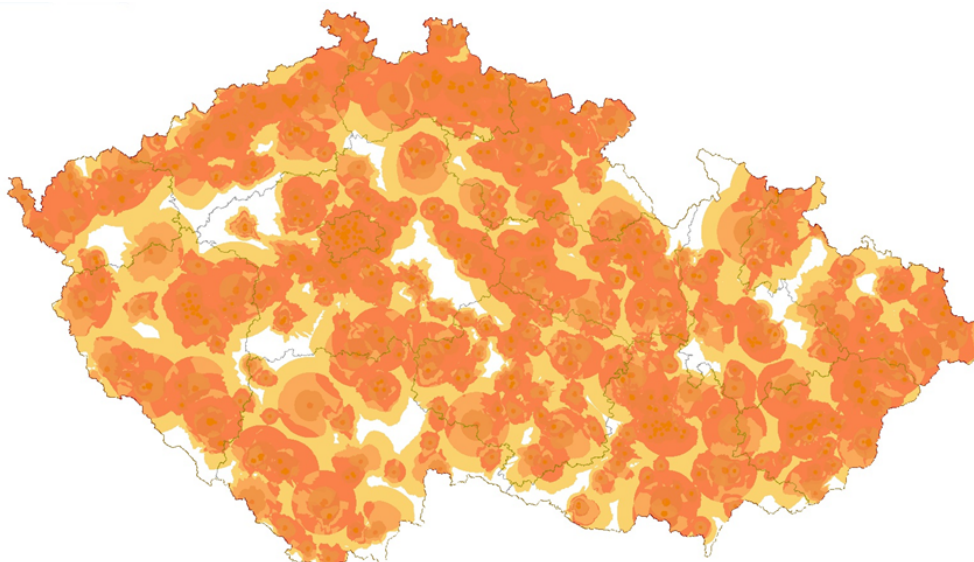


Obrázek 3.1: Vrstvy sítě LoRaWAN. Převzato z [21]

Pro přenos je využíváno frekvencí v SubGhz pásmu, konkrétně těch pásem, které jsou v jednotlivých zemích nelicencovány, tedy 433 a 868 Mhz v Evropě a 915 Mhz v Severní Americe. Také díky tomu, je narozdíl od jiných sítí, možné provozovat tuto síť lokálně pro vlastní užití, kupříkladu v průmyslovém areálu firmy. Další z výhod tohoto protokolu je jeho rozšířenost neboť existuje několik sítí a to jak komerčních tak také těch komunitních, jako je například TheThingsNetwork. Ta je v provozu hlavně díky dobrovolníkům, kteří provozují přístupové brány (viz 3.2) pro ostatní.

### 3.1 Pokrytí v ČR

Aktuálně lze v České Republice využít minimálně dvě sítě LoRaWAN. Tou nejrozšířenější je síť společnosti *České Radiokomunikace* jenž začala být budována v roce 2016. Pokrývá minimálně všechna krajská města a celkově pokrytí dosahuje přes 80% celé republiky [2]. Nejlevnější možný tarif poskytovaný touto společností začíná na ceně 200 Kč/měsíc se závazkem na 1 rok. Součástí tohoto tarifu je možnost připojení až 10 zařízení s celkovým maximem odeslaných zpráv 10 000 za den. Mapu pokrytí lze vidět zde na obrázku 3.2



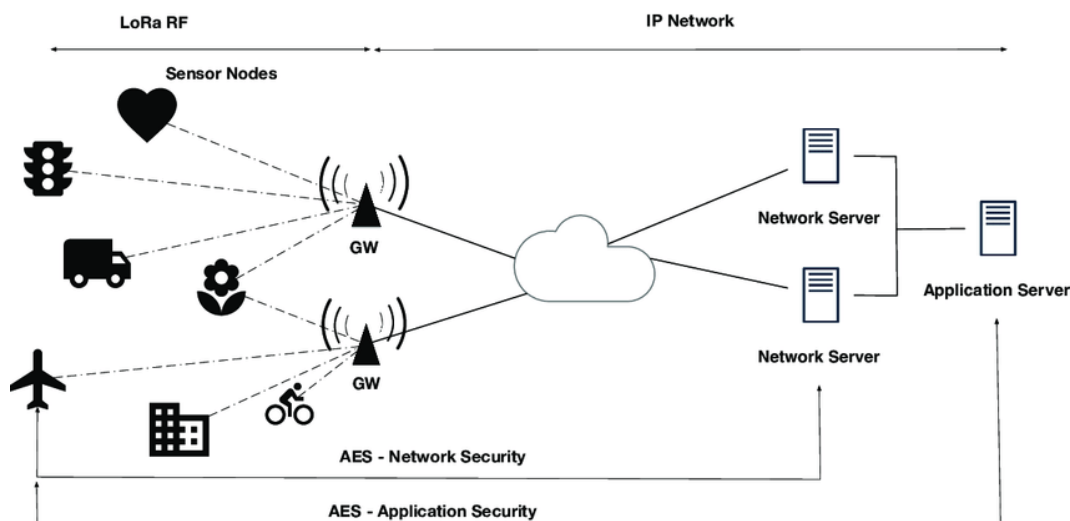
Obrázek 3.2: Pokrytí sítě LoRaWAN od Českých Radiokomunikací. Převzato z <https://www.iotport.cz/lorawan-sit-pro-iot> dne 26.3.2022

Další rozšířenou sítí je komunitní TheThingsNetwork. V případě, že uživatel dodržuje podmínky tzv. fair-use, je tato síť zdarma, a je tak vhodná pro hobbyistické projekty. Výchozí brány jsou provozovány většinou také hobbyisty a to plně dobrovolně na vlastní náklady. Na této síti není limitován počet registrovaných zařízení, ovšem je velmi limitován maximální čas po který může zařízení aktivně vysílat za jeden den. Aktuálně je tento limit nastaven na 30 sekund uplink (odesílání ze zařízení). Downlink (příjem do zařízení) je pak limitován 10ti zprávami za den. Pokrytí je na našem území menší než to od *Českých Radiokomunikací* avšak ve větších městech by neměl být problém se k síti připojit.



## 3.2 Architektura sítě

Každá síť LoRaWAN se skládá z několika zařízení, kde každé splňuje svůj účel aby bylo možné doručit informace napříč ní. Tyto zařízení se dělí na koncová zařízení, brány, síťové servery, join (připojovací) servery a aplikační servery. Strukturu této architektury znázorňuje obrázek 3.3.



Obrázek 3.3: Architektura sítě LoRaWAN. Převzato z [20]

### Koncové zařízení

Mezi tato zařízení patří všechna, která jsou vybavena LoRa transceiverem a jsou připojena do sítě za pomoci LoRaWAN bran. Jedná se ve většině případů o plně autonomní senzory či zařízení ovládající nějaký nástroj (spínání světel apod.). Častou vlastností je také napájení bateriemi pro delší dobu provozu bez nutnosti údržby. Každé ze zařízení by taktéž mělo mít unikátní identifikátor v rámci sítě, čehož je běžně docíleno přiřazením tohoto ID již při výrobě zařízení či čipu vysílače [5].

### Výchozí brána

Brány (gateway) umožňují koncovým zařízením připojit se do sítě. Jsou rozmístěny ve městech či krajině tak aby pokrývaly pokud možno co největší oblast.

S koncovými zařízeními komunikují použitím modulace LoRa a veškeré přijaté zprávy jsou dále odesílány na síťové servery ke zpracování. Každá takováto zpráva však může být přijmuta více branami najednou čímž se zvyšuje šance že zpráva opravdu dorazí na síťový server.

V případě, že síťový server odesílá zprávu koncovému zařízení, je dle historie přijatých zpráv vybrána ta brána u níž byla síla signálu nejvyšší. Tato brána pak zprávu odešle opět pomocí LoRa.

Ke komunikaci mezi bránou a síťovým serverem je typicky použit protokol IP.

## Síťový server

Tento server pracuje s LoRaWAN MAC vrstvou, zpracovává tedy LoRa rámce koncových zařízení a řídí provoz na síti. Jeho úkolem je zajistit správné fungování sítě za měnících se provozních podmínek. Ověřuje autentičnost zasláných zpráv, jejich integritu a také deduplikaci v případě, že je zpráva přijata více bránami najednou. V případě požadavku autentizace koncového zařízení přeposílá požadavek na *join server*. Také spravuje zabezpečený kanál mezi aplikačním serverem a koncovým zařízením. Neposlední funkcí je správa ADR (Adaptive data rate, viz 3.6).

## Join server

Join server obstarává proces OTAA (over the air activation). Při přijmutí OTAA join požadavku jsou z předsdílených klíčů vygenerovány klíče nutné k běžné komunikaci. Tyto klíče jsou pak zaslány síťovému serveru (kde jsou následně uloženy) společně s potvrzovací zprávou zaslánou koncovému zařízení.

## Aplikační server

Servery tohoto typu jsou již vlastněny zákazníkem a starají se o zpracování přijatých zpráv. Ty jsou mu zasílány ze síťového serveru.

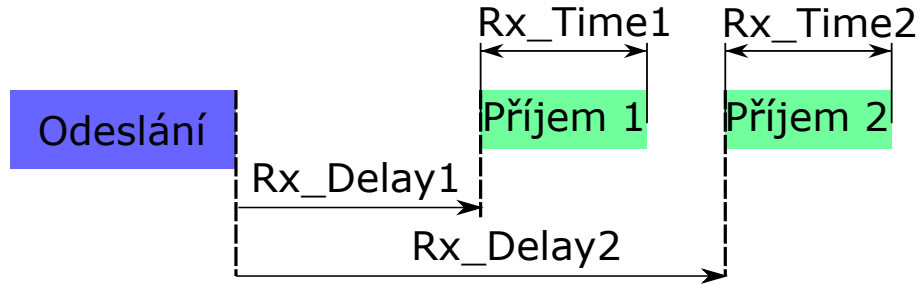
## 3.3 Třídy zařízení

Třídy zařízení určují jak se koncové zařízení chová v době své operace. Konkrétně specifikuje dobu ve kterou se zařízení probouzí a tudíž také čas během kterého může být zařízení zaslán downlink.

### Třída A

Zařízení operující v této třídě se probouzejí pouze v době, kterou si specifikuje samo zařízení na základě svého okolí. Může se jednat například o senzor pohybu který se při detekci probouzí a zasílá uplink zprávu. Po odeslání jsou pak vždy aktivovány dvě příjmové okénka.

První z příjmů začíná po čase `Rx_Delay1` a trvá dobu `Rx_Time1`. V případě, že koncové zařízení úspěšně přijme v tomto okénku downlink, je uvedeno do spánkového režimu. V opačném případě je zahájeno čekání na druhé okénko, na které se čeká `Rx_Delay2` od původního času dokončení vysílání a trvá `Rx_Time2`. V tom může opět dojít k přijmutí downlinku. Neohledě na to zda je v druhém okně nějaký downlink přijmut, zařízení je uspáno. Hodnoty jednotlivých časů jsou nastavitelné zařízením a typicky se řídí časy, které udává konkrétní síť. Časování okének znázorňuje obrázek 3.4.



Obrázek 3.4: Přijímací okénka třídy A

## Třída B

V této třídě je zařízením možné zasílat downlink zprávy bez toho aniž by byla komunikace zahájena jím samotným. Této funkcionality je dosaženo otevřením příjmového okénka v předdefinovaných časech. Zařízení se tedy v určitý čas probouzí, otevírá příjmové okénko a čeká na zprávu. Po ukončení tohoto okénka je zařízení opět uvedeno do spánkového režimu. Díky těmto vlastnostem se tedy třída hodí pro zařízení, které lze na dálku ovládat. Všechna zařízení této třídy musí taktéž podporovat třídu A.

## Beaconing

Aby mohla zařízení v této třídě správně fungovat, je zapotřebí aby měla vždy přesný aktuální čas uložený například v RTC (Real Time Clock)<sup>1</sup>. Aby byl tento čas vždy přesný, je síť LoRaWAN poskytován všem zařízením pomocí takzvaného beaconu. LoRaWAN brány vysílají neustále rámce obsahující aktuální čas. Tyto rámce mohou být zachyceny zařízením. K udržení synchronizace pak stačí aby byly tyto rámce přijaty párkrát denně díky čemuž je také šetřena baterie.

## Třída C

Zařízení pracující v této třídě jsou permanentně zapnutá. Během fungování jsou jejich přijímače stále aktivní (výjimkou je pouze doba kdy dochází k vlastnímu uplinku) a mohou tak přijmout zprávu kdykoliv. Díky tomu je tak možné zařízení zaslat zprávu s nejmenší možnou latencí. Nevýhodou je ovšem právě zmíněná nutnost být stále zapnut, kvůli čemuž je spotřeba vyšší než u nižších tříd. To znemožňuje napájení bateriemi a tato zařízení tak bývají napájena permanentně z elektrické sítě. Tato třída je tak ideální například pro chytrá pouliční světla, která mohou být ovládána dálkově.

<sup>1</sup>RTC — Hodiny reálného času. Obvod či periferie, která uchovává a poskytuje aktuální čas.

## 3.4 Zabezpečení

Síťe LoRaWAN jsou zabezpečeny několika způsoby, tak aby byl zajištěn jak obsah jednotlivých zpráv tak také k ověření zda jsou zařízení ta, se kterými opravdu chceme komunikovat. Pro zajištění těchto vlastností síť používá během komunikace dva relační klíče, `NwkSKey` a `AppSKey`, které jsou známy zařízením, síťovým serverem a ten druhý také aplikačním serverem koncového uživatele. S těmito klíči jsou pak prováděny operace použitím běžného algoritmu jako je právě AES<sup>2</sup>. Tyto klíče jsou do koncových zařízení a síťového serveru buď uloženy manuálně uživatelem a nebo jsou derivovány při aktivaci pomocí OTAA (podrobněji popsáno v 3.5).

### Klíč `NwkSKey`

`NwkSKey` (Network Session Key) je klíč, díky němuž je u všech zasláných rámců zajišťována integrita zpráv. Dále je s pomocí klíče chráněno také proti tzv. replay útokům, což jsou útoky kdy se útočník pokusí zaslat již dříve odeslanou zprávu znova bez toho aniž by znal obsah šifrované zprávy. Poslední ochranou tvořenou tímto klíčem je zajištění autenticity.

LoRaWAN rámec se skládá z MAC hlavičky, počítadla rámce (již zmíněná ochrana proti replay útokům), dat aplikační vrstvy a poté MIC (Message Integrity Code). Hodnota MIC je vždy vypočtena kombinací MAC hlavičky, počítadla a dat aplikační vrstvy právě s použitím `NwkSKey` klíče a to algoritmem AES-CMAC. Tato hodnota je pak umístěna na konec celého rámce. Po přijmutí zprávy druhou stranou (která také zná klíč `NwkSKey`) je z přijatého rámce opět MIC vypočteno a je porovnáno s tou hodnotou v přijatém rámci. V případě, že se shodují, si může být druhá strana jistá, že nedošlo ke změně dat v rámci a také že byl zaslán správným zařízením. Pokud by útočník či chyba během přenosu změnila jen jediný bit dat, vypočtené MIC by se již neshodovalo a rámec by byl zahozen.

### Klíč `AppSKey`

`AppSKey` (Application Session Key) je klíč, kterým jsou šifrována data aplikační vrstvy algoritmem AES. Je znám koncovým zařízením a aplikačním serverem. V některých případech může být uložen také na síťovém serveru či administračním portálu sítě, například během testování pokud chceme zprávy dešifrovat předtím než je uveden do provozu vlastní aplikační server [15].

## 3.5 Aktivace zařízení v síti

K identifikaci zařízení a správně fungujícímu šifrování, je jak již bylo zmíněno v předchozí sekci, zapotřebí dvou klíčů a také adresy zařízení `DevAddr`. To jak se koncové zařízení do sítě připojuje a jak získá tyto klíče je proveditelné dvěma způsoby, těmi sou připojení pomocí OTAA nebo aktivací personalizací.

### 3.5.1 APB - Aktivace personalizací

V režimu aktivace APB (Activation By Personalization) jsou klíče `NwkSKey`, `AppSKey` a adresa `DevAddr` vložena přímo do zařízení při výrobě či konfiguraci správcem. Výhodou

---

<sup>2</sup>AES (Advanced encryption standard) — Symetrické blokové šifrování využívající stejný klíč pro šifrování i dešifrování

této metody je jednoduchost (pouze tři hodnoty k uložení a zařízení funguje), nevýhodou je ovšem situace kdy dojde k naplnění počítadla rámců. V takovém případě je totiž nemožné dále komunikovat a je nutné manuálně změnit klíče [1].

### 3.5.2 OTAA

Při použití aktivace pomocí OTAA (Over the air activation) jsou do zařízení a Join serveru uloženy kořenové klíče. Pro aktivaci zasílá koncové zařízení na začátku komunikační relace `Join request` (požadavek na připojení k síti). Ten obsahuje `JoinEUI`, `DevEUI` (ID zařízení uložené v paměti zařízení od výroby) a `DevNonce`. V případě, že je ověřena identita koncového zařízení dle databáze registrovaných zařízení, je vygenerována `Join accept` (přijmutí požadavku připojení) zpráva a odeslána zpět zařízení. Koncové zařízení je následně schopno z kořenových klíčů, `DevNonce` a polí obsažených v této zprávě vygenerovat relační klíče. Stejný způsobem jsou tyto klíče vygenerovány také na Join serveru a zaslány síťovému serveru pro správné fungování. Tento způsob aktivace narozdíl do APB umožňuje znovupřipojení kdy jsou vygenerovány nové relační klíče a také resetováno počítadlo rámců což dovoluje zařízení komunikovat i po dosažení jeho maxima. Další z výhod je možnost stanovit komunikační rychlost v závislosti na síle signálu.

## 3.6 Maximální velikost paketů

Síť LoRaWAN při komunikaci může využít rozdílných komunikačních rychlostí. Ty jsou odvozeny jak od hodnoty `Spreading factor` tak také zvolené šířky pásma. Nižší `Spreading factor` zvyšuje přenosovou rychlost ovšem snižuje efektivní dosah, neboť má přijímač méně času rozlišit jednotlivé úrovně během přenosu a okolní šum tak může způsobit chyby. Kombinace `Spreading factoru` a šířky pásma následně udává `Data rate`. Na základě používaného `Data rate` je síť omezena maximální délkou jednoho paketu. Tyto maximální délky pro každý z `Data rate` lze vyčíst z tabulky 3.1.

Data rate	Spreading factor	Šířka pásma	Max délka
0	SF12	125 kHz	51
1	SF11	125 kHz	51
2	SF10	125 kHz	51
3	SF9	125 kHz	115
4	SF8	125 kHz	242
5	SF7	125 kHz	242
6	SF7	250 kHz	242

Tabulka 3.1: Maximální délky paketů v závislosti na zvoleném spreading factoru. Převzato z <https://www.thethingsnetwork.org/docs/lorawan/regional-parameters/> dne 7.5.2022

## Kapitola 4

# Komunikace na krátkou vzdálenost NFC

NFC, celým názvem Near field communication je technologie bezdrátové komunikace umožňující komunikaci mezi zařízeními na velmi krátkou vzdálenost, obvykle až 4 cm, avšak teoreticky až 20cm. Veškeré standardy spojené s touto technologií jsou spravovány neziskovou organizací NFC forum, jež byla založena společnostmi Nokia, Philips a Sony v roce 2004.

Jejím předchůdcem je technologie RFID (radio-frequency identification), která podporuje komunikaci až na vzdálenost 10 metrů. Neboť byla jedním ze záměrů tvorby NFC větší bezpečnost komunikace, je komunikace omezená na krátké vzdálenosti. Díky tomuto omezení je značně obtížné takovou komunikaci odposlouchávat, odpadá nutnost jakéhokoliv párování a komunikace je taktéž velmi spolehlivá. Tyto vlastnosti pomohly k rozšíření NFC do mobilních zařízení kde se začaly používat hlavně k bezkontaktním platbám.

Komunikace pomocí technologie NFC se dělí do dvou režimů operace, aktivní a pasivní. Více je popsáno v 4.1. Aktivní zařízení jsou například NFC čtečky, ty pasivní pak NFC tagy (značky), které umožňují uschování libovolných dat.

NFC pak podporuje tři režimy komunikace a to *peer-to-peer* u nějž probíhá komunikace mezi dvěma aktivními zařízeními, *čtení/zápis* pro komunikaci s pasivními NFC tagy a *card emulation* pro emulaci pasivních zařízení tím aktivním, např. emulace přístupových karet. [25].

### 4.1 Princip funkce

Ke komunikaci je používána smyčková anténa (cívka) na obou stranách komunikace. Aktivní zařízení vytváří svou anténou elektromagnetické pole, které následně indukuje proud v anténě druhého zařízení. Druhé zařízení pak může být tímto proudem napájeno a nepotřebuje tak svůj vlastní zdroj energie. Elektromagnetické pole používané ke komunikaci má frekvenci 13,56 MHz. Na tuto frekvenci musí být taktéž vyladěn obvod s anténou.

Pro přenos informací směrem k NFC tagu je aktivním zařízením vysílané elektromagnetické pole modulováno pomocí ASK (amplitude shift keying - klíčování amplitudovým posuvem). Pro přenos informací zpět je použita modulace zátěží. NFC tag obsahuje rezistor, kterým je měněn odebíraný proud, a také kondenzátor, kterým je měněna rezonanční frekvence obvodu. Tyto změny jsou detekovány demodulátorem na straně aktivního zařízení [18] [19].

## 4.2 Operační režimy

### 4.2.1 Pasivní

V tomto režimu je elektromagnetické pole vytvářeno pouze zařízením, které zahájilo komunikaci. Toto zařízení je tedy aktivní a jedná se o NFC iniciátor. Druhé zařízení je vždy pasivní a jedná se o zařízení cílové.

Cílové zařízení je napájeno elektromagnetickým polem ze zahajujícího, a odpovídá zátěžovou modulací tak jak bylo popsáno v kapitole 4.1.

Konkrétní rychlosti a dosah komunikace záleží na velikosti antény a také síle elektromagnetického pole. Obvyklá rychlost ve vzdálenosti 20cm je 6 až 53 kbit/s, ve vzdálenosti 10cm pak 106 až 848 kbit/s [14].

### 4.2.2 Aktivní

Během komunikace v aktivním režimu je elektromagnetické pole vytvářeno oběma zařízeními. Každé ze zařízení, v případě, že chce poslat informace druhé straně, vytváří své vlastní elektromagnetické pole, které pro přenos dat používá opět ASK modulaci. Režim také podporuje využití VHBR (very high bit rate) dosahující rychlostí až 6,78 Mbit/s kdy je pro přenos místo ASK použita modulace PSK (phase shift keying).

Zátěžová modulace není v tomto režimu použita pro běžný přenos dat, je pomocí ní ale odpovídáno na zahájení komunikačního kanálu [14].

Aby během komunikace nedocházelo ke kolizím, je využit *listen before talk* protokol. Zařízení před vlastním odesláním naslouchá zda-li zrovna druhé zařízení nevytváří elektromagnetické pole. Vlastní odeslání proběhne až poté co toto magnetické pole druhé strany zaniká. Komunikace v tomto režimu je tedy half-duplex [4].

## 4.3 Režimy komunikace

Tyto režimy mohou být využity v různých situacích.

### 4.3.1 Režim Čtení/Zápis

Komunikace probíhá v pasivním operačním režimu a typickým zařízením iniciující komunikaci je chytrý telefon. Zařízení může v tomto režimu zapsat do paměti NFC tagu například adresu URL či přístupové údaje k Wi-Fi síti. Jiné zařízení je pak schopno tyto data vyčíst a zobrazit uživateli či dle nich provést nějakou akci, například připojit se ke zmiňované Wi-Fi síti. Pro ukládání těchto dat je typicky využíván formát NDEF 4.4.

### 4.3.2 Režim peer-to-peer

Komunikace probíhá v aktivním operačním režimu a v takovém případě je možné využít VHBR (very high bit rate) viz. 4.2.2. Jedno ze zařízení zahajuje komunikaci, po potvrzení je otevřen komunikační kanál použitelný např. pro odesílání souborů mezi chytrými telefony.

### 4.3.3 Režim emulace karet

V tomto režimu se může NFC zařízení chovat jako pasivní NFC karta. Jednou ze situací jsou mobilní telefony chovající se jako kreditní karta při přiložení k platebnímu terminálu. Komunikace je vždy zahajována NFC čtečkou, v tomto případě platebním terminálem.

## 4.4 Formát NDEF

NDEF, neboli *NFC Data Exchange Format* je specifikace formátu zprávy používaného pro režim čtení/zápis. Jedná se o jednoduchý binární formát. Jedna NDEF zpráva (NDEF message) může obsahovat několik NDEF záznamů (NDEF record) a každý ze záznamů má svůj vlastní typ a délku. Délka obsahu jednotlivých záznamů je maximálně  $2^{32} - 1$  avšak reálně je délka limitována velikostí paměti zapisovaného NFC tagu.



## Kapitola 5

# Platformy dostupné pro implementaci

Tato kapitola pojednává o možných platformách a vysílačů/přijímačů použitelných pro implementaci zařízení, konkrétně těch dvou nevhodnějších. Jako platforma se v tomto kontextu považuje mikrokontrolér společně se softwarovým vybavením.

Mikrokontrolér je kombinací mikroprocesoru a všech periférií jako FLASH paměť a RAM v jednom balení. Jeho úkolem je vykonávat programátorem vytvořený kód reprezentovaný instrukcemi, jenž jsou uloženy v paměti. Různé mikrokontroléry pak mohou být ještě kombinovány společně s dalšími složitějšími perifériemi jako např. Wi-Fi, Bluetooth nebo Sub-GHz transceiver, čímž vzniká tzv. SoC (System-on-chip).

### 5.1 STM32WLXX

Tato platforma a SoC řady STM32WLXX od výrobce STM jsou mikrokontroléry s velmi nízkou spotřebou kombinované právě s Sub-GHz transceiverem v jednom integrovaném obvodu. Na trhu jsou velmi krátce, vydány byly teprve v roce 2020. Sub-GHz transceivery v těchto SoC jsou schopny pracovat na frekvencích 433MHz či 868MHz ovšem tato skutečnost záleží na konkrétním vybraném SoC, neboť hardware je vždy uspůsoben pouze na jednu z nich.

Dále se procesory dělí dle počtu jader. Existují jednojádrové či dvoujádrové verze jenž jsou postaveny na jádrech ARM Cortex-M4 a v případě dvoujádrové verze je k tomuto jádru dostupné taktéž jádro ARM Cortex-M0+. Oba tyto jádra jsou 32-bitové a dokáží běžet až na frekvencí 48 MHz.

Napájecí napětí je typicky 3,3V avšak procesor má vnitřní regulátor a tak je schopen pracovat v rozsahu 1,71V až 3,6V. Díky jeho efektivnímu návrhu je v nejvyšším režimu spánku STANDBY spotřeba pouhých  $2\mu A$ . Zabudovaná paměť FLASH je v případě nejvyšších variant velká až 256 KB přičemž paměť SRAM dosahuje velikostí 64KB. Tyto velikosti jsou dostačující pro většinu programů vzhledem k tomu, že například využití paměti SRAM dodávaného LoRaWAN stacku dosahuje pouze 8KB [8].

#### GPIO porty

Součástí je samozřejmě několik GPIO (generických vstupně/výstupních) portů, každý obsahují dále 16 pinů. Některé z těchto pinů lze také využít pro alternativní funkce jako jsou

různé periferie. Mezi tyto periferie patří 3 sběrnice I2C, 2 rozhraní SPI a v neposlední řadě 2 sériová komunikační rozhraní UART [8].

## Časovač

Pro události řízené časem, případně pro udržení času samotného disponuje mikrokontrolér hodinami reálného času RTC. Pro časované události lze pak využít přerušení, jenž je spuštěno v určitý čas a je vykonána obslužná rutina. Ty se dají využít taktéž pro plánované probuzení mikrokontroléru z hlubokého režimu spánku [8].

## Sub-GHz rádio

Jak již bylo zmíněno, řada STM32WL je první SoC, jenž kombinuje mikrokontrolér společně s LoRa/Sub-GHz transceiverem. Tento transceiver funguje na frekvencích 433MHz či 868MHz s vysílacím výkonem 15dBm v tzv. Low Power režimu či 22dBm v High Power režimu.

Mezi modulace podporované transceiverem patří LoRa, (G)FSK (frequency shift keying), (G)MSK (minimum-shift keying), a BPSK (binary-hhase shift keying) [8].

## Zabezpečení

Vzhledem k použitému šifrování v sítích LoRaWAN je mikrokontrolér vybaven také hardwarovou akcelerací výpočtu 128 a 256 bitového symetrického šifrování AES. Pro případ, že je nutné chránit firmware před extrakcí je možné aktivovat jednu ze dvou úrovní ochrany vyčtení paměti FLASH [8].

## Rozdělení paměti

Paměť FLASH je vždy rozdělena do stránek o velikostech 2KB. Těchto stránek může být až 128 v případě varianty mikrokontroléru s 256KB paměti. Tyto stránky jsou samostatně smazatelné v případě, že je potřeba do nich zapisovat zvlášť data za chodu [8] [17]. Tabulka 5.1 zobrazuje rozdělení paměti FLASH společně s počátečními adresami .

Číslo stránky	Adresy	Velikost
0	0x0800 0000 - 0x0800 07FF	2 KB
1	0x0800 8000 - 0x0800 0FFF	2 KB
2	0x0800 1000 - 0x0800 17FF	2 KB
...	...	...
126	0x0803 F800 - 0x0803 F7FF	2 KB
127	0x0803 F800 - 0x0803 FFFF	2 KB

Tabulka 5.1: Rozdělení paměti FLASH v mikrokontrolérech STM32WL [17]

## 5.2 ESP32

ESP32 je série velmi levných SoC mikrokontrolér, vyráběný firmou Espressif od roku 2016. Jde o SoC se zabudovaným Bluetooth a Wi-Fi rádiem. Jedná se o nástupce velmi populárního ESP8266, jenž se původně proslavil pouze jako levný Wi-Fi modul avšak později se firma rozhodla uvolnit pro něj vývojové nástroje a umožnila tak psát pro něj vlastní firmware [22].

Z tohoto důvodu se tyto mikrokontroléry velmi rozšířily jak v hobbystické tak profesionální sféře, kde jsou nejčastěji využívány pro IoT aplikace. Jejich rozšíření pomáhá hlavně oficiální podpora Arduino frameworku a také to, že se mikrokontroléry prodávají hlavně ve formě modulů, které se pak jednoduše integrují do zařízení.

Tyto mikrokontroléry jsou poháněny jádrem Tensilica Xtensa LX6 či Tensilica Xtensa LX7, jenž jsou založené na architektuře RISC-V, používají 32 bitovou instrukční sadu a jejich maximální frekvence dosahuje až 240 MHz. Společně s hlavním procesorem je k dispozici také tzv. ULP (ultra low power) koprocesor s velice úspornou instrukční sadou jenž může běžet během hlubokého spánku a hlavní procesor případně probudit. Ve všech verzích mikroprocesoru je součástí paměť SRAM s velikostí 320KB, která je doplněna o 448KB paměti ROM použité pro uschování naprogramovaného firmwaru.

Napájecí napětí, je taktéž 3,3V s možným rozsahem 2,2V až 3,6V. Po uvedení do nejnižšího režimu spánku je spotřeba tohoto mikrokontroléru  $5\mu A$ .

### GPIO a periferie

ESP32 je vybavena 34 GPIO porty pro digitální vstupy a výstupy. Některé z nich dokáží pracovat v režimu alternativních funkcí a pracovat tak jako výstup či vstup jedné z periférií. Mezi tyto periferie patří ty standardní jako SPI, I2C, UART, ADC (analog to digital converter) převodník ale také nestandardní jako jsou například řadiče SD karty, Ethernet MAC vrstvy či hallova sonda.

### Wi-Fi a Bluetooth

Jedním z největších využití ESP32 je právě díky jeho schopnosti připojit se k Wi-Fi sítím a komunikaci s ostatními zařízeními včetně hostování lokálního serveru. Wi-Fi rádio splňuje standardy Wi-Fi 802.11 b/g/n se zabezpečeními WPA, WPA2 a WPA3. Toto rádio je sdíleno zároveň s rozhraním Bluetooth verze 4.2 s podporou Bluetooth Low Energy pro aplikace vyžadující nižší spotřebu [22].

# Kapitola 6

## Návrh

Před samotným návrhem je dobré seznámit se již s existujícími řešeními podobného problému, podívat se na jejich aktuální vlastnosti a také je případně porovnat mezi sebou. Na základě zjištěných informací pak identifikovat jejich slabiny, výhody a nevýhody.

Dále je důležité se podívat na tyto nevýhody a zjistit jakým způsobem mohou být buď kompletně či alespoň částečně odstraněny. Také je nutné stanovit si své vlastní cíle práce. Poslední částí návrhu je pak činnost při níž jsou navržené změny zakomponovány do vlastních cílů a je navrženo zařízení, které by ve výsledku mělo být alespoň v několika ohledech lepší než současná řešení.

Tato kapitola tedy popisuje současná řešení, jejich výhody, a návrh vlastního řešení. To se skládá ze tří částí: obecný návrh, návrh hardware a návrh konfigurační aplikace.

### 6.1 Současná řešení

Na trhu je k dispozici pouze několik zařízení, která nějakým způsobem prodlužují dosah standardu WMBUS.

Jedním z nich je **LAN-WMBUS-uR** od společnosti LANSEN. Jedná se o opakovač WMBUS rámců, který po přijetí daný rámec odešle dvakrát znova. Nejedná se tak o žádné zásadní zvýšení dosahu, ten je stále limitován regulacemi o vysílacím výkonu a použité modulaci použité standardem WMBUS. Vzhledem k tomu, že zařízení ve své podstatě jen zesiluje již vyslaný signál, je stále zapotřebí manuálního odečtu měřičů v dosahu tohoto opakovače. Dále zařízení podporuje rámce o velikosti až 255 bajtů, a počet možných WMBUS adres, které mohou být opakovány je až 932. Zařízení je napájeno baterií nebo z elektrické sítě a konfigurováno musí být na místě pomocí počítače přes rozhraní USB [7].

Dalším zařízením je **Wireless M-Bus Bridge V2 (LoRaWAN)** od společnosti Lobaró. Toto zařízení je rozdílné od toho předchozího síťový můstek. Po přijetí všech WMBUS rámců měřičů jejichž adresa je uložena v paměti tohoto zařízení jsou postupně tyto rámce odesílány pomocí sítě LoRaWAN dále ke zpracování. Takto je dosaženo opravdového zvětšení dosahu měřičů až na 15km k nejbližší bráně sítě LoRaWAN. Napájení je řešeno 19Ah baterií a konfigurace je prováděna opět připojením počítače sběrnici USB.

## 6.2 Obecný návrh

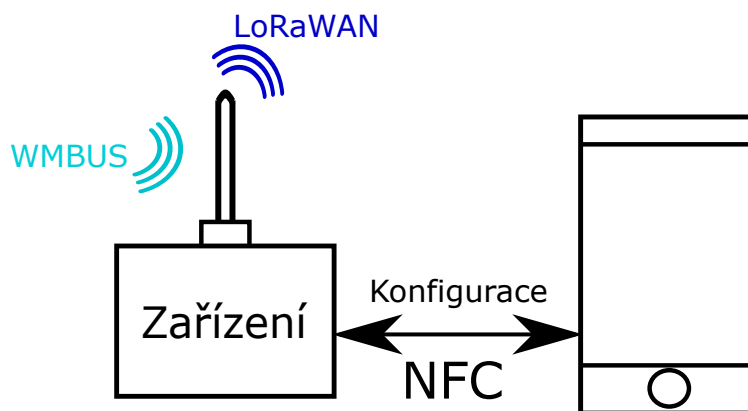
Součástí práce jsou dvě hlavní části. První částí je samotné zařízení, které bude přijímat rámce standardu WMBUS. Po přijetí budou tyto rámce filtrovány dle adres uložených v konfiguraci aby mohly být později odeslány dále. Druhou částí je pak aplikace se kterou lze načíst aktuální konfiguraci, pozměnit jí a upravené hodnoty uložit zpět do zařízení.

Dle požadavků ze zadání a také mých vlastních stanovených cílů byl vytvořen návrh na celé zařízení a konfigurační aplikaci. Návrh samotný musí tedy splňovat následující:

- **Samostatnost** - Na výsledném zařízení by mělo stačit provést konfiguraci pouze jednou a dále dokáže zařízení fungovat bez vnějšího zásahu
- **Robustnost** - Během provozu by se zařízení nemělo dostat do stavu, kdy bude zaseklé a potřebuje manuální reset správcem
- **Životnost** - Zařízení bude napájeno bateriemi a z principu jeho existence umístěno v místech kde se nebude správce často vyskytovat. Z toho důvodu by měla být výdrž na těchto bateriích zhruba 7-10 let
- **Cena** - Aby mělo zařízení šanci případně rozšířit se na trhu, je zapotřebí aby byla celková cena jednoho kusu co nejmenší. Pro dosažení tohoto cíle je nutné využít co nejmenší možný počet elektronických součástí a jejich ceny samotné.
- **Zvýšení dosahu** - Oproti normálnímu vysílacímu dosahu standardu WMBUS by mělo být zařízení schopné rámce odesílat data na vzdálenosti i několika kilometrů.
- **Jednoduchost konfigurace** - Konfigurační aplikace by měla vzhledem k účelu zařízení obsahovat menší počet nastavitelných hodnot a provedení konfigurace by mělo být rychlé.

Obzvláště na základě posledního cíle jsem se rozhodl pro účely komunikace použít technologii NFC. V dnešní době je její rozhraní integrované do téměř všech mobilních zařízení a narušil od současných řešení, kdy je ke konfiguraci zapotřebí laptopu se sběrníci USB, tak není nutné aby s sebou uživatel instalující zařízení nosil zbytečnou váhu navíc. Pro prodloužení dosahu se jeví jako nejlepší řešení bezdrátová síť LoRaWAN. Její velké pokrytí, energetická efektivita a také velikost rámců, které lze zasílat splňují stanovené požadavky nejlépe.

Obecné schéma celého systému po zvážení všech cílů lze spatřit na obrázku 6.1.



Obrázek 6.1: Obecné schéma řešení

## 6.3 Návrh zařízení

Jako centrální prvek, který bude součástí hardwarového návrhu jsem zvolil SoC (system-on-chip) z řady *STM32WL5X* od známého výrobce čipů STM. Ten byl vybrán na základě stanovených cílů z toho důvodu, že díky jeho integraci samotného mikroprocesoru a také sub-GHZ transceiveru v jednom čipu je snížena komplexnost na desce plošných spojů. Díky absenci dlouhé propojovací sběrnice je také eliminován vliv rušení okolních signálů na komunikaci mezi procesorem a transceiverem. Vybraná řada mikrokontrolérů je taktéž oproti ostatním řešením, kde by byl transceiver umístěn zvláště, výrazně cenově výhodnější.

Jak již bylo dříve zmíněno, pro komunikaci s konfigurační aplikací je použita technologie NFC. Neboť neexistuje mikrokontroler jenž by obsahoval sub-GHZ transceiver společně s NFC komunikací, je zapotřebí použít externí periférii.

Vzhledem k použitému mikrokontroléru jsem se rozhodl použít čip *ST25DV* od stejné společnosti STM. Jedná se o integrovaný obvod obsahující dynamický NFC tag s EEPROM pamětí o velikostech začínajíc na 4 kilobitech a končících 64 kilobitech. Neboť jsou 4 Kbity (512 bajtů) dostačující pro účely přenosů konfiguračních dat, byla vybrána tato varianta.

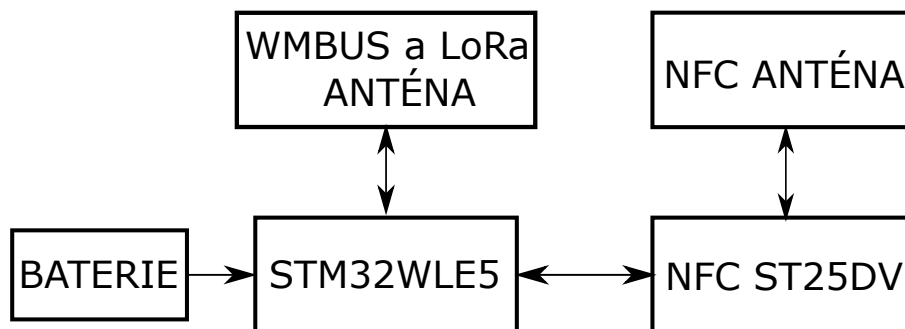
Dále obvod disponuje registry pomocí nichž je možné za chodu měnit chování NFC tagu. Lze jej například nastavit tak aby nereagoval na NFC příkazy v případě, že je potřeba pracovat pouze s v něm uloženými daty. Integrovaný obvod používá ke komunikaci s hlavním mikrokontrolérem sběrnici I2C což je jednoduchá sběrnice potřebující ke svému provozu pouze dva vodiče s pull-up rezistory a společnou zemi.

NFC tag potřebuje ke svému provozu taktéž anténu. Protože má být výsledné zařízení na jednotné desce plošných spojů, anténa bude vytvořena z jedné z vrstev desky měděnou trasou. Samotný návrh je popsán v 7.8

Jako zdroj energie pro napájení celého zařízení je nejvhodnější baterie. Rozsah napájecího napětí SoC *STM32WL5X* je 1,71V až 3,6V. Z tohoto důvodu nabíjecí Lithium-iontové či Lithium-Polymerové baterie nepřichází v úvahu neboť jejich napětí za stavu plného nabití dosahuje až 4,2V. V takovém případě by bylo zapotřebí napěťového regulátoru což zvyšuje cenu výrobku a většina takových regulátorů navíc spotřebovává více energie než je spotřeba samotného mikrokontroléru ve spánkovém režimu.

Po krátkém průzkumu jsem našel baterie které jsou pro tuto aplikaci nejvhodnější. Jedná se o Li-SOCl<sub>2</sub> (Lithium Thionyl Chloride) baterie jejichž nominální napětí je 3,6V a jejich operační teplota je v rozsahu -55°C až +85°C. Obrovskou výhodou je pak také malé samovybití.

Výsledné blokové schéma popisující návrh hardwaru zařízení lze vidět na obrázku 6.2.



Obrázek 6.2: Blokové schéma hardwaru

## Návrh softwaru zařízení

Chod zařízení bude řídit mnou vytvořený firmware pro mikrokontrolér *STM32WL5X*. Jeho hlavním úkolem je obstarávat všechny periferie tak aby plnily stanovené úkoly.

### Příjem dat z měřidel

První funkcionalitou, která je ze všech ta nejdůležitější je příjem WMBUS rámců. Pro příjem jsem se rozhodl cílit pouze na rámce režimů C1 a T1 standardu WMBUS (viz sekce 2.2) a to obzvláště kvůli jejich velké rozšířenosti v měřidlech po České republice a možnosti přijímat oba režimy najednou bez nadbytečné režie s přepínáním mezi frekvencemi a synchronizačními slovy. Protože je nutné zajistit příjem obou typů rámců (typ A a B, viz 2.5), firmware musí být schopen příjmu rámců s délkou, která je větší než 255 bajtů.

### Filtrování WMBUS měřidel

Po přijetí jsou rámce filtrovány dle **pole A**, které obsahuje adresu měřiče. Aby rámeček pokračoval dále ke zpracování, musí být tato adresa shodná alespoň s jednou z adres uložených v konfiguraci zařízení. Vyfiltrované rámce pak musí být uloženy do paměti společně s informací o jaký typ se jedná, neboť tato informace je důležitá pro pozdější zpracování. Uchovávání všech rámců stejné adresy od počátku přijímání je zbytečné a tak je vždy starší rámeček nahrazen tím novějším.

### Odesílání

Příjem je dokončen vždy po uplynutí časového limitu nebo v případě kdy došlo k přijetí rámců od všech adres uložených v konfiguraci.

Následuje odesílání přijatých dat pomocí sítě LoRaWAN. Postupně dochází k rozdělení rámců na části jejichž velikost nepřesahuje maximální možnou délku LoRaWAN paketu pro aktuální nastavený LoRa data rate. Tyto části jsou odesílány za sebou a to bez potvrzení přijetí druhou stranou. Po dokončení se zařízení převádí do režimu spánku na dobu určenou v konfiguraci a po něm je probuzeno a začíná opět příjem.

### Konfigurace a její uložení

Uživateli je umožněno uložit do zařízení několik hodnot ovlivňujících chování zařízení během provozu. Všechny tyto hodnoty je nutné zachovat i po uložení a je tedy nezbytné ukládat

je do nevolatilní<sup>1</sup> paměti. NFC rozhraní musí být funkční vždy a to včetně doby kdy je zařízení v režimu spánku. Uživatel tak může provést konfiguraci za každých okolností, nejen v relativně krátký interval kdy je zařízení probuzeno.

## 6.4 Návrh konfigurační aplikace

Mobilní platformy nabízejí několik možných technologií vývoje aplikací. Jako první přichází logicky v úvahu vytvoření konfiguratoru jako nativní aplikace pro jednotlivé mobilní operační systémy. V takovém případě běží nejrychleji a oba dva nejrozšířenější operační systémy Android a iOS poskytují dostatečně rozsáhlé API (aplikační rozhraní) pro nízkoúrovňový přístup k NFC periferiím.

V poslední době je však stále více oblíbený vývoj aplikací jako webových stránek. Nejenže jsou takovéto aplikace multiplatformní, ale jejich vývoj bývá často jednodušší díky rozsáhlé komunitě a letům historie. Počátkem roku 2019 začala vznikat experimentální verze webové API zprostředkovávající přístup k zápisu a čtení NFC tagů. Tato API je aktuálně podporována pouze v malé skupině webových prohlížečů operačního systému Android. Na platformě iOS od společnosti Apple tato API zatím není implementována údajně z důvodu bezpečnosti. Navzdory absenci této API na iOS jsem se rozhodl konfigurator vytvořit jako webovou aplikaci v jazycích HTML a JavaScript. Je možné, že bude podpora této experimentální API v budoucnu přidána také na iOS a nebude tak potřeba vyvíjet separátní verzi pro tento systém.

### Funkcionalita aplikace

Hlavním úkolem aplikace je poskytnout uživateli rozhraní, kterým nastaví zařízení. Aplikace po svém startu požádá uživatele o přiložení mobilního telefonu k hlavnímu zařízení. Po přiložení je aplikací detekován NFC tag a je načtena aktuální uložená konfigurace v zařízení.

Aplikace musí podporovat úpravu následujících vlastností:

- **Časování** - Upravuje časové události zařízení. První hodnota stanovuje dobu trvání příjmu WMBUS rámců v minutách. Druhá hodnota určuje dobu mezi odečty. Dobou mezi odečty je myšlen čas mezi začátkem příjmu a začátkem příjmu následujícího. Z obou hodnot lze tedy odvodit čas, který zařízení stráví v režimu spánku. Rozsah obou nastavitelných hodnot začíná na 1 minutě a maximálně je povoleno 65 535 minut.
- **Údaje sítě LoRaWAN** - Parametry připojení zařízení k síti pomocí aktivace OTAA (viz 3.5. Mezi ně patří kořenové klíče *AppEUI* a *AppKey*. Dále aplikace uživateli v této sekci zobrazí *DeviceID*, které má každé zařízení nepřepisovatelně zapsané v paměti od výroby.
- **Seznam WMBUS adres** - Uživateli jsou nejdříve zobrazeny adresy měřidel, jejichž rámce byly v době skenování alespoň jednou zachyceny. Každá ze zobrazených adres pak může být přesunuta do seznamu povolených adres. Povolené adresy lze také zpětně odebrat.

---

<sup>1</sup>Nevolatilní - paměť jejíž obsah není vymazán při ztrátě napájecího napětí či resetu procesoru



## **Formát komunikace mezi zařízením a aplikací**

Komunikace se zařízením používá pro přenos formát NDEF (viz 4.4). Veškerá data jsou přenášena binárně a délku jednoho NDEF záznamu jsem se rozhodl omezit na 242 bajtů. Toto specifické číslo je odvozeno od maximální délky (255 bajtů) jednoho sektoru NFC tagu z níž je odečteno 13 bajtů hlavičky formátu NDEF.

## Kapitola 7

# Implementace

Pro implementaci hlavního firmware, jenž bude obstarávat správný chod mikrokontroléru *STM32WL5X*, jsem se rozhodl použít jazyk C/C++ jenž je přímo podporován výrobcem. Existují také jiné možnosti vývoje využívající programovací jazyk *Rust* avšak ten je spravován komunitně a není tak jistá podpora. Výrobce každopádně pro jazyk C/C++ poskytuje všechny nástroje potřebné k vytvoření, kompilaci a nahrání projektů do zařízení. Všechny tyto zmiňované funkce lze nalézt v integrovaném vývojovém prostředí *STM32CubeIDE*.

Pro víceúrovňový přístup k perifériím používám HAL (Hardware Abstraction Layer) poskytnutý ve vývojovém balíčku taktéž výrobcem STM. Ten umožňuje pracovat s perifériemi voláním funkcí místo přímého přístupu k registrům. Dále je součástí vývojového balíčku také několik middleware (software poskytující funkce nad rámec HAL) pro snadnější ovládání Sub-GHz rádia a poté hlavně LoRaWAN stack kontrolující odesílání a příjem LoRa rámců pro správný provoz LoRaWAN zařízení.

Pro jednoduchý vývoj využívám nástroje pro ladění vestavěné přímo do vývojového prostředí *STM32CubeIDE*. Dále je používán čip *ST-LINK*, jenž tvoří můstek mezi počítačem a mikrokontrolérem. Díky němu lze z počítače nahrát program a také provádět ladění. Pro komunikaci mezi mikrokontrolérem a čipem *ST-LINK* je použito rozhraní *SWD*.

### Konfigurace periférií nástrojem CubeMX

Součástí prostředí *STM32CubeIDE* je konfigurační nástroj *CubeMX*. Pomocí GUI jsou vybrány a nastaveny periferie či povoleny jednotlivé middleware a na základě těchto nastavení jsou pak nástrojem vygenerovány zdrojové soubory s kódem, který periferie inicializuje.

Pro mou práci jsou oproti výchozí konfiguraci povoleny následující periferie:

- **RTC** - Použit pro probouzení zařízení ze spánku a plánování časovaných událostí. Hodinový signál přivedený na vstup (RTCCLK) má frekvenci 32 768 Hz. Konfigurace povoluje kalendář a také **WakeUp** signál jehož **WakeUpCLK** je nastaven na RTCCLK / 16. Jak pro kalendář tak **WakeUp** je v NVIC (Nested vector interrupt control) povoleno přerušování.
- **I2C3** - Využíván pro komunikaci s NFC čipem. Komunikační frekvence nastavena na 100 KHz.
- **CRC** - Slouží k výpočtu CRC WMBUS rámců po jejich příjmu. Polynom byl nastaven dle standardu WMBUS uvedeného v sekci [2.5.3](#)

Jako middleware je povolen jen ten pro LoRaWAN komunikaci. Díky jeho povolení je však do projektu jako závislost zahrnut také middleware obstarávající Sub-GHz transceiver.

## Struktura projektu

Projekt obsahující firmware se skládá z několika složek. Mezi ty nejdůležitější patří **Core**, **Drivers** a **RelayApp**.

### Core

Složka obsahuje převážně vygenerovaný kód vygenerovaný nástrojem *CubeMX*. Soubor `main.c` obsahuje vstupní bod programu a jsou v něm inicializovány dříve zmíněné periferie. Po inicializaci je spuštěn vstupní bod aplikace.

### Drivers

V této složce se nachází soubor definující GPIO piny pro specifickou desku.

### RelayApp

Zde se nachází veškeré zdrojové kódy definující chování aplikace. Složka je rozdělena na dvě další, `Inc` (hlavičkové soubory) a `Src` (zdrojové soubory). Vstupní bod aplikace je obsažen v souboru `Src/app.cpp` a zde je prováděna hlavní smyčka programu.

Provádění úkolů je rozděleno dle funkce do jednotlivých souborů, kdy každý z nich obsahuje kód pro obsluhu jedné periferie a zastává pouze jediný úkol pro fungování celého firmwaru.

## 7.1 Příjem WMBUS rámců

Kód pro příjem a zpracování WMBUS rámců je implementován v souboru `wmbus.cpp`. Příjem WMBUS rámců se skládá z několika fází, přijetí surových dat, extrakci délky rámce a výpočet CRC. V případě režimu T také provedeno dekodování 3-z-6.

První částí implementace je funkce `Wmbus_Init()`. Jak již její název napovídá, provádí inicializaci přijímání a to hlavně konfiguraci Sub-GHz transceiveru. Nejdříve jsou do proměnné `radioEvents` uloženy všechny ukazatele na funkce reprezentující události, které mohou nastat při příjmu. Tato proměnná je pak předána funkci `MRadio.Init` jenž inicializuje samotnou periferii přijímače. Poté je přijímač přepnut do režimu FSK a jsou nastaveny následující parametry:

- **Synchronizační slovo** - Nastaveno na pole bajtů `{0x54, 0x3D}`. Odvozeno od společného synchronizačního slova režimů C1 a T1
- **Délka preamble** - Standard WMBUS pro tyto režimy stanovuje délku preamble 4 bajty neboli 32 bitů. Detektor preamble je však nastaven tak aby očekával validní rámeček již po 8 bitech, neboť při nastavení vyšších hodnot dochází z neznámých důvodů k posunu dat v rámcích o 4 bity.

- **Základní frekvence a odchylka** - Frekvence dle režimů C1 a T1 nastavena na 868,95 MHz. Jedná se o střední frekvenci na jejímž základě jsou společně s odchylkou (nastaveno na 100 kHz) vypočteny frekvence pro modulaci FSK.
- **CRC a whitening** - Hardwarový výpočet a kontrola těchto vlastností je kvůli ne-standardnímu formátu WMBUS rámců zcela vypnut a je později prováděn softwarem.

Samotný příjem je vždy zahájen voláním funkce `Wmbus_StartReceiving()`. Po jejím dokončení čeká přijímač na přijetí jednoho rámcu. Po dokončení či v případě chyby je volána jedna z funkcí oznamující změnu stavu. Rámec je zpracován a pokud je validní, je předán dále k uložení do paměti uloženou funkcí.

Dalšími funkcemi dostupnými pro zbylou část firmwaru jsou `Wmbus_Standby()` a `Wmbus_RegisterCallback(callback)`. První funkce uvede transceiver do standby režimu kdy není nic přijímáno. Ta druhá nastavuje funkci jenž je předáván přijatý rámec po úspěšném zpracování.

## Upravený middleware

Režim T1 používá kódování 3-z-6 díky čemuž mohou být rámce delší než 255 bajtů. Zabudovaný transceiver ovšem příjem takto dlouhých rámců nepodporuje a je nutné jej přepnout do režimu nekonečného příjmu a takovéto rámce z transceiveru extrahovat po částech tzv. chunkích.

Součástí Sub-GHz middleware je přesně pro takovéto účely softwarové rádio. To je ovšem vytvořené pouze pro příjem rámců v nichž je délka uložena na začátku rámcu ve dvou bajtech. WMBUS rámce však mohou mít délku v režimu T1 kódovanou a toto softwarové rádio jí tak není schopné zpracovat.

Rozhodl jsem se tedy tento middleware upravit pro vlastní účely. Jeho upravená kopie se nachází ve složce `CustomRadio`. Konkrétně bylo nutné do funkce `RFW_GetPacketLength(uint8_t*)` vložit kód jenž po získání prvních 3 bajtů rámcu volá funkci `WMBUSGetPacketLen` ve které je délka vypočtena.

## Získání délky rámcu

Na základě druhého bajtu v rámci je zjištěno zda se jedná o typ A či typ B. Následuje bajt jenž reprezentuje pole `L`. V případě typu B je celková délka rámcu jednoduše hodnota obsažená v tomto poli. Typ A je ovšem komplikovanější a je nutné délku vypočíst. Pole obsahuje pouze délku bez CRC polí a tak je nutné z něj zjistit počet bloků a ten vynásobit 18 (16 bajtů na blok + 2 bajty pro CRC).

## Výpočet CRC

Výpočet je prováděn s hardwarovou akcelerací a to za použití CRC periferie dostupné přímo v mikrokontroléru. Její inicializace je součástí funkce `Wmbus_Init()`. Pro každý z typů rámců existuje funkce provádějící kontrolu. Jsou postupně procházeny všechny bloky v rámci, z dat je vypočteno 2 bajtové CRC a porovnáno s hodnotou nacházejícím se za daty. Pokud se jen jedno CRC neshoduje, celý rámec je zahozen.

## 7.2 Obsluha LoRaWAN sítě

Přijaté rámce musí být odesílány bezdrátovou sítí LoRaWAN dále na aplikační servery. K připojení k síti a následnému zasílání částí rámců slouží skupina funkcí implementována v souboru `lorawan.cpp`.

Pro implementaci obsluhy této sítě jsem využil LoRaWAN middleware od společnosti STM dodaný společně s vývojovým balíčkem. Ten je navržen tak aby dodržoval veškerá pravidla LoRaWAN sítě stanovená LoRa aliancí a podporuje tak všechny možné způsoby autentifikace na síti a také všechny LoRaWAN třídy zařízení.

Před odesíláním jakýchkoliv dat dále do sítě je nutné aby byl LoRaWAN stack správně inicializován a zařízení bylo do této sítě připojené.

Toho je docíleno voláním funkce `LoraWan_Init()`. Ten jako první provede inicializaci a základní konfiguraci třídy zařízení (třída A), použitého regionu (v tomto případě EU) a povolení ADR (adaptive data rate). Následně je nutné nastavit jednotlivé LoRaWAN klíče. Ty jsou získány z nastavení voláním

`AppSetting_GetLoraAppEui()` a `AppSetting_GetLoraAppKey()`.

Připojení je zahájeno pouhým voláním `LmHandlerJoin` jenž jako argument akceptuje typ aktivace, což je v tomto případě aktivace OTAA. Middleware v tu chvíli zasílá požadavek o připojení do sítě, pokud již v nějaké síti připojený není.

Zpracování událostí middlewaru je asynchronní a pouze oznamuje, že je nutné nějakou událost zpracovat zpětným voláním `OnMacProcessNotify`, které je vykonáno v přerušení. Firmware pak musí v nejbližší možný moment událost zpracovat v běžné programové smyčce mimo přerušení. K tomuto slouží utilita *Sequencer* jenž běží v hlavním vlákne firmwaru a v případě takovéto situace je jí oznámeno, že má při následující iteraci spustit právě zpracování událostí.

### Odesílání

Odesílání je prováděno vždy po částech dat. Data předaná funkci je nutné nejdříve skopírovat do připraveného bufferu. Společně s daty je odesíláno také číslo značící o jakou část dat se jedná (číslo reprezentuje, který WMBUS rámec je zasílán). K tomuto je využito tzv. *Portu* jenž je posílán s každým LoRaWAN paketem. Krom čísla 0 jenž je vyhrazeno pro testovací účely je možné použít jakoukoliv hodnotu z rozsahu 1-255. Proto je tento port vhodný právě pro identifikaci WMBUS rámců.

### Uložení kontextu

LoRaWAN middleware si musí v každou chvíli uchovávat aktuální počítadlo paketů a také stav připojení do sítě. Tento stav je však za provozu ukládán pouze v paměti SRAM a při přechodu do režimu spánku mezi odečty tak dochází k jeho ztrátě. Z toho důvodu bylo nutné implementovat ukládání tohoto stavu do nevolatilní paměti. Neboť je už využívána paměť FLASH k uložení nastavení zařízení (viz sekce 7.4) rozhodl jsem se tuto paměť použít k uložení stavu middlewaru.

Middleware k účelům zachování stavu poskytuje dvě zpětné volání jenž bylo zapotřebí implementovat. V nich je předán ukazatel na data reprezentující stav a také jejich délka. Data jsou uložena do paměti FLASH a v případě obnovení jsou všechna načtena zpět.

## 7.3 NFC rozhraní

Pro komunikaci s konfigurační aplikací byla zvolena technologie NFC za pomoci čipu *ST25DV*. Komunikace s tímto čipem však probíhá sběrnici I2C jenž je inicializována na začátku chodu zařízení.

Čip *ST25DV* ve výchozím stavu (po zapnutí jeho napájení) neodpovídá na žádné jiné příkazy krom autorizace heslem. Proto je nutné tuto autorizaci provést zapsáním 17 bajtů do konfiguračního registru čipu. Těchto 17 bajtů se skládá z 8 bajtů hesla následovaných bajtem s hodnotou 0x09 a opět 8 bajty stejného hesla.

### Čekání na dokončení operace

Po dokončení jakéhokoliv ze zápisů do registrů je nutné počkat než čip operaci dokončí celou. Po dobu vykonávání neodpovídá na I2C komunikaci. K tomuto účelu je vytvořena funkce `waitForChip()`, jenž se snaží zaslat čipu 1 bajt do té doby dokud čip operaci nedokončí a nezapne zpět svou I2C periferii.

Během testování této funkce však někdy docházelo k zaseknutí čipu z neznámých důvodů a tak došlo k nekonečnému zacyklení celé funkce. Kvůli tomuto jsem se rozhodl implementovat limit v počtu iterací čekání a při jeho překročení dojde k resetu čipu vypnutím a zapnutím jeho napájení.

### Čtení a zápis NDEF zpráv

Data zasílaná konfigurační aplikaci jsou zapisována a v případě čtení z aplikace, čtena z paměti EEPROM NFC čipu. K odeslání je využíváno formátu NDEF. NDEF záznam je sestaven jednoduše vytvořením pole o velikosti dat + délka hlavičky NDEF zprávy.

Pokud konfigurační aplikace zašle příkaz zařízení, je jeho obsah uložen do paměti EEPROM. Čip *ST25DV* neposkytuje možnost detekovat přijetí celé zprávy neboť data jsou do paměti zapisována po blocích. Čip jde však nastavit tak aby v případě jednoho tohoto zápisu vyvolal přerušení. Ve chvíli, kdy je přerušení vyvoláno, je spuštěn pomocí utility *TIMER* speciální časovač s dobou trvání 200 milisekund. Tento časovač je při každém přerušení resetován a začíná počítat od začátku. Po jeho uplynutí je jasné, že zpráva byla přijata celá.

Tabulka 7.1 popisuje obsah NDEF zprávy čtené/zapisované v paměti EEPROM.

Identifikuje zprávu jako NDEF	Délka dat + 3	NDEF flagy	NDEF délka typu	Data NDEF záznamu
0xE1 0x40 0x40 0x01 0x03	--	0xD5	0x00	--

Tabulka 7.1: Formát zapisované NDEF zprávy

### Průběh komunikace

Komunikace jako taková je defacto zahajována na straně zařízení, a to zasláním uvítací zprávy s obsahem `WMBUSRelay_HELLO` jenž je do NFC tagu zapsána na začátku provozu zařízení. Konfigurační aplikace pak načítá NFC tag a po jeho přečtení ověří, že načtená zpráva je uvítací. Následně odesílá svůj požadavek, jenž je zapsán do NFC tagu. Zařízení

požadavek přečte, vyhodnotí a odesílá odpověď zpět. Po odeslání má konfigurační aplikace 5 sekund na přečtení odpovědi neboť po uplynutí této doby je odpověď přepsána uvítací zprávou aby mohlo dojít k další komunikaci.

## Formát konfigurační zprávy

Komunikace mezi konfigurační aplikací a zařízením používá speciální formát zprávy. Zprávy zasílané směrem k zařízení se skládají z 2 bajtů identifikující příkaz, heslo o délce 8 bajtů a následují data příkazu. Zprávy zasílané zpět k aplikaci začínají ASCII znakem „!“ a následují libovolná binární data. Heslo je zasíláno z důvodu bezpečnosti, aby nemohla konfiguraci provést neoprávněná osoba. Aktuálně jsem však z časových důvodů nestihl tuto funkcionalitu plně implementovat a zařízení tak akceptuje hesla všechna.

## 7.4 Ukládání do paměti FLASH

Nastavení zařízení musí zůstat zachováno i v případě odpojení zařízení od napájení. K tomuto účelu jsem se rozhodl ukládat jej přímo do paměti FLASH v němž je uložen také samotný firmware. Není tak zapotřebí připojení externích pamětí a přístup k takto uloženým datům je bez zbytečného spoždění. Pro tyto data jsou v paměti vyhrazeny dvě celé stránky, každá o velikosti 2048 bajtů. Jedna ukládá nastavení a druhá je vyhrazena pro již dříve zmiňovaný (sekce 7.2) stav LoRaWAN middleware.

Protože je paměť chráněna proti zápisu, je nutné jí nejdříve před takovouto akcí odemknout voláním `HAL_FLASH_Unlock()`. Z principu paměti FLASH musí být stránka před zápisem vymazána. Mazání je prováděno na základě čísla stránky a stránka je takto vymazána celá.

Zápis do paměti musí být prováděn po slovech o velikosti 64 bitů. Je proto vhodné před uložením kombinovat menší než 8 bajtů do jednoho takového slova. Ukládání pole o velikosti 8 bajtů je z tohoto důvodu prováděno spojením jednotlivých bajtů do jednoho 64 bitového slova za použití bitových operátorů.

## 7.5 Nastavení zařízení

Všechna nastavení jsou uložena v paměti flash. Tyto hodnoty mohou být měněny za chodu zařízení vykonáním několika příkazů z konfigurační aplikace. Seznam jednotlivých implementovaných příkazů a jejich chování je uvedeno v této sekci.

### Příkaz načtení konfigurace

Identifikován kódem LC, tento příkaz zašle konfigurační aplikaci aktuální uložené hodnoty konfigurace. Společně s hodnotami konfigurace je zasláno také unikátní ID zařízení, které je extrahováno z paměti. Toto ID je v paměti zapsáno nepřepisovatelně výrobcem. Hodnoty časů jsou zaslány jako 2 bajty.

### Příkaz načtení adres

Identifikován kódem LA, tento příkaz zašle konfigurační aplikaci skenované a uložené adresy. Neboť seznam adres může dosáhnout takové délky, že jeho velikost přesáhne maximální

velikost zprávy, tento příkaz požaduje jako argument číslo stránky jenž je zrovna načítáno. Jedna stránka obsahuje 30 zařízení. Po sestavení bufferu obsahujícího seznam adres je ke zprávě také přiloženo, zda existuje další stránka.

## Příkaz uložení konfigurace

Identifikován kódem `SC`, tento příkaz přijme od konfigurační aplikace celou konfiguraci včetně povolených adres. Konfigurační aplikace sestaví buffer dat jenž je po segmentech odesílán. Každý příkaz obsahuje dva flagy, jedna oznamující reset příjmového buffer a druhá říkající zda má být po tomto segmentu celý buffer zpracován a konfigurace uložena.

## 7.6 Hlavní smyčka programu

V této sekci je popsána implementace hlavní smyčky programu, neboli způsob, kterým jsou kombinovány a koordinovány dříve zmíněné části firmwaru. Veškerý kód popisovaný v této sekci se nachází v souboru `app.cpp`

Chod firmware začíná nejdříve načtením konfigurace zařízení z paměti FLASH. Dále je inicializována obsluha NFC a utility *Sequencer* a *TIMER*. Časovače za použití utility *TIMER* jsou vytvořeny voláním `TimerInit(timer, func)` a jejich interval (doba mezi spuštěním jejich funkcí) je nastaven funkcí `TimerSetValue(timer, interval)`. Po dokončení následuje zahájení příjmu WMBUS zároveň se spuštěním jednoho z časovačů s intervalem doby sbírání.

Pokud je přijat validní WMBUS rámec, je následně zpracován na základě aktuálního režimu v němž se zařízení nachází. Pokud se jedná o režim skenování(konfigurace), je pouze zjištěna adresa měřiče z tohoto rámce a přidána do seznamu skenovaných adres. V režimu preposílání je dle adresy měřiče zjištěna pozice v povoleném seznamu adres, pokud je tato pozice jakékoliv číslo mezi 0-254, znamená to že je adresa povolená.

Tento index je využit pro uložení celého rámce do předalokovaného pole. Toto předalokované pole má velikost `MAX_ULOŽENÝCH_ZAŘÍZENÍ * (MAX_VELIKOST_RÁMCE+1)`. K maximální velikosti rámce je přičtena 1 neboť před samotným rámcem je na nultou pozici uložen jeho typ (A nebo B). Pole obsahuje vždy nejnovější přijatý rámec.

Po uplynutí času pro sbírání nebo v případě, že byly přijaty rámce od všech povolených adres, dojde k ukončení příjmu a je zahájen proces inicializace LoRaWAN. Od LoRaWAN middleware je funkcí `LoRaMacQueryTxPossible()` vyžádána maximální velikost zasláního paketu `MaxTxSize` (ten se totiž liší dle aktuálního spreading factoru).

Z pole WMBUS rámců je vyjmuta první část o velikosti `MaxTxSize` a odeslána sítí LoRaWAN. Po dokončení je *sequencerem* spuštěn proces odeslání další části. S každou částí je odeslán také zmiňovaný index celého rámce aby bylo možné zprávu rekonstruovat na straně aplikačního serveru. Části jsou takto odesílány dokud nedojde k odeslání rámce celého. Postupně jsou odeslány všechny rámce a ihned po dokončení poslední částí posledního rámce dojde k uložení stavu LoRaWAN middleware. Posledním příkazem je přechod do režimu spánku `STOP2`. Spánek trvá dobu stanovenou v konfiguraci z níž je odečten čas uplynulý od startu firmwaru.



## 7.7 Implementace webové aplikace

Konfigurační aplikace jenž slouží správci zařízení k jeho správnému nastavení. Uživatelé zobrazí požadavek o přiložení k zařízení, po načtení zobrazí aktuální konfiguraci a umožní jí měnit.

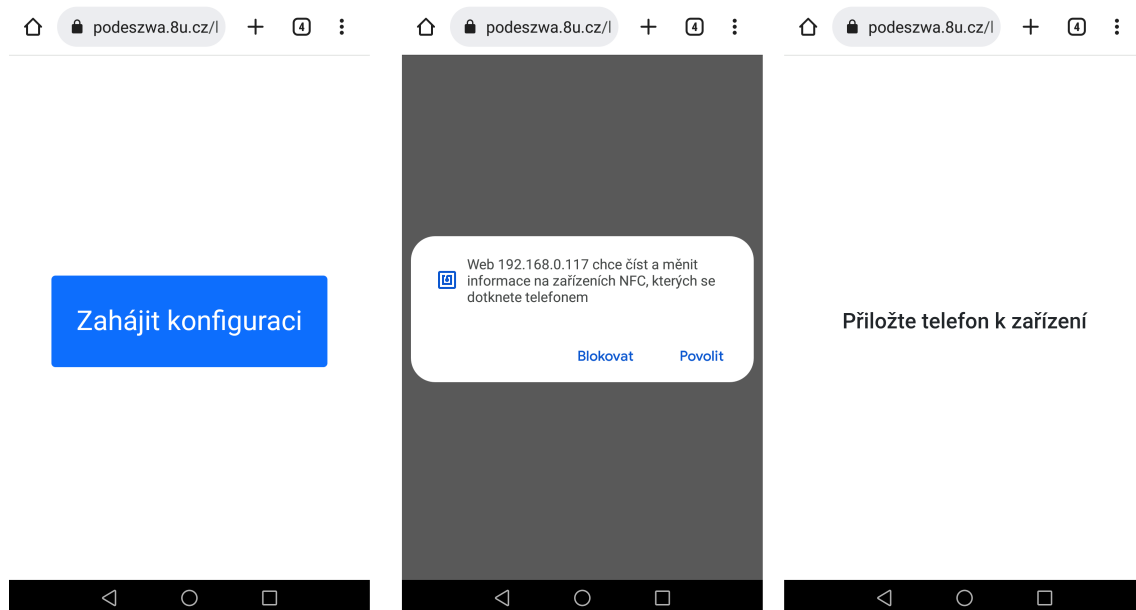
Rozhraní je vytvořeno v jazyce HTML za použití JavaScriptu pro NFC komunikaci a také dynamickou úpravu rozhraní. Pro zjednodušený návrh vzhledu rozhraní jsem se rozhodl využít CSS/JS knihovnu Bootstrap.

Rozhraní aplikace se skládá ze dvou částí, výzvou uživateli pro přiložení mobilního telefonu k zařízení a poté samotné konfigurační rozhraní. V druhé z částí je pak obrazovka rozdělena pomocí záložek na dva další pohledy konfigurace.

### 7.7.1 Uživatelské rozhraní

Vzhled rozhraní je implementován pouze v jednom souboru `index.html` společně s CSS styly v souboru `style.css`, jenž je však velmi krátký neboť většina stylování je prováděna přímo v HTML kódu za použití Bootstrap tříd.

Jak již bylo řečeno, první částí jenž je zobrazena uživateli po načtení stránky je tlačítko s nápisem „Zahájit konfiguraci“. Po jeho zmáčknutí může být uživatel prohlížečem požádán o povolení přístupu k NFC. Pokud je toto povolení uděleno je uživatel vyzván k přiložení mobilního telefonu k zařízení. Obrázek 7.1 zobrazuje vzhled jednotlivých fází této části rozhraní.



Obrázek 7.1: Fáze rozhraní s požadavkem přiložení

Druhá část rozhraní se skládá ze záložek *Konfigurace* a *Adresy*.

Záložka *Konfigurace* je rozdělena na sekce *Časování* a *LoRaWAN - OTAA*. Zde je zobrazeno aktuální nastavení zařízení a uživateli je umožněno hodnoty měnit ve vstupních polích. Pro zadávání hodnot časování je použito běžného HTML `<input type="text">`. Následně je použitím JavaScriptu implementována validace vstupu a rozsahu. Tato validace se nachází v souboru `input.js`.

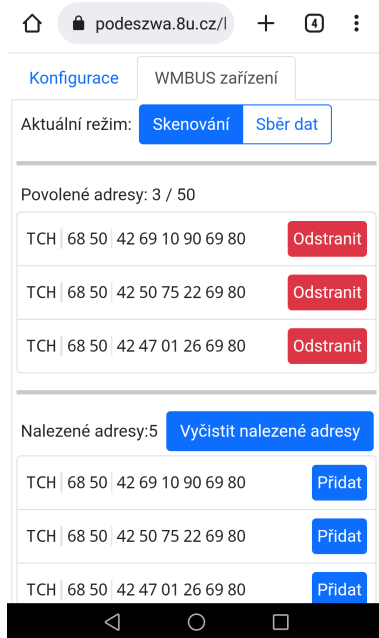
Pro vstup klíčů sítě LoRaWAN je taktéž použit běžný HTML input, ovšem validace hexadecimálního vstupu je komplexnější a použil jsem proto knihovnu `jquery.inputmask.js`. Touto knihovnou je pro vstupní pole nastaven regulární výraz `XX XX XX XX XX XX XX XX` kde `X` je `/[0-9a-fA-F]/` neboli řetězec obsahující 8 hexadecimálních čísel.

Obrázek 7.2 ukazuje vzhled záložky konfigurace.

The image shows two side-by-side screenshots of a web configuration interface for a WMBUS device. Both screenshots are from the URL `podeszwa.8u.cz/l`. The left screenshot shows the 'Konfigurace' tab with a sub-tab 'WMBUS zařízení'. Under the 'Časování' section, 'Délka spánku' is 1440 and 'Délka sbírání' is 15. The 'LoraWAN - OTAA' section has 'Unikátní ID zařízení', 'AppEUI', and 'AppKey' all set to '0x FF 01 02 03 04 05 06 07'. The right screenshot shows the 'Délka sbírání' set to 15. The 'LoraWAN - OTAA' section has 'Unikátní ID zařízení', 'AppEUI', and 'AppKey' all set to '0x A1 B2 C3 D4 E5 F6 A7 B8'. At the bottom of the right screenshot are two buttons: 'Uložit' (Save) and 'Zavřít zařízení' (Close device).

Obrázek 7.2: Vzhled záložky konfigurace

Záložka *WMBUS zařízení* obsahuje přepínač aktuálního režimu. Pod tímto přepínačem se nachází seznam povolených adres následovaný adresami nalezenými v režimu skenování. Tyto seznamy jsou generovány dynamicky použitím JavaScriptu a to konkrétně funkce `CreateGenericRow` jenž jako argumenty přebírá adresu, výrobce, text tlačítka a také index v seznamu. Tento index je používán pro identifikaci konkrétního řádku při přidávání či odebírání položky ze seznamu. Seznam s adresami v záložce lze vidět na obrázku 7.3.



Obrázek 7.3: Záložka zobrazující seznam uložených a skenovaných adres.

```

1 function CreateGenericRow( manBytes, addrBytes, buttonText, isDanger, dataIndex)
2 {
3     var li = gen("li", "list-group-item px-2");
4     var flexDiv = gen("div", "d-flex flex-row align-items-center");
5     var paragraph = gen("p", "m-0 bg-scondary");
6
7     var manNameSpan = gen("span", "border-end border-2 pe-1 me-1 font-monospace");
8     manNameSpan.textContent = Utils.GetWMBUSHumanManName(manBytes);
9
10    var manSpan = gen("span", "border-end pe-1 me-1 addressor");
11    manSpan.textContent = Array.from(manBytes, (e) => Utils.ToHex(e)).join(" ");
12
13    var addrSpan = gen("span", "addressor");
14    addrSpan.textContent = Array.from(addrBytes, (e) => Utils.ToHex(e)).join(" ");
15
16    var btn = gen("button", "btn p-1 ms-auto " + ((isDanger) ? "btn-danger": "btn-primary"));
17    btn.setAttribute("data-idx", dataIndex);

```

Výpis 7.1: Část funkce `CreateGenericRow`

Výpis 7.1 obsahuje část této funkce v níž je možné vidět jakým způsobem je vytvářen jeden záznam v seznamu. Pro přehlednější tvorbu HTML elementů je použita pomocná funkce `gen(nodeName, classes)`.

## 7.7.2 JavaScript komunikace NFC

Pro NFC komunikaci v podobě transakcí byla vytvořena třída `NFCService`. Jejím hlavním úkolem je provádět obecné čtení a zápis do zařízení v podobě NDEF záznamu obsahujícího příkazy a jejich odpovědi.

Implementace využívá experimentální JavaScript API `NDEFReader`. Tato API pracuje pouze s NDEF záznamy a operace s ní jsou prováděny použitím asynchronních metod.

Čtení dat z NFC tagu provádí metoda `ReadOnce()`, jenž vytváří na samém začátku instanci `NDEFReader` použitou pro jednu operaci. Kód metody je součástí JavaScript `Promise` pro asynchronní chování. Skenování NFC tagu je následně zahájeno voláním metody `ndef.scan(cancelSignal)`. Jako argument je předán `cancelSignal`, kterým je možné skenování kdykoliv nenásilně ukončit. Pokud je NFC tag úspěšně načten, je vyvolán delegát a dojde ke kontrole správnosti obsahu NDEF dat. Výpis 7.2 zobrazuje provedení kontroly NDEF záznamu (jeho samotné existence a také nulové délky). Také ukazuje jak dochází k vyřešení `Promise` jak v případě validního tak nevalidního záznamu.

```
1 if(message.records.length!=1)
2 {
3     resolve(null);
4     return;
5 }
6 if(message.records[0].data !=null && message.records[0].data.buffer!=null)
7 {
8     let buffer = message.records[0].data.buffer;
9
10    ndef.removeEventListener("reading",handler);
11    resolve(new Uint8Array(buffer));
12 }
```

Výpis 7.2: Kontrola validity načtené NDEF zprávy

Pro zapsání dat do NFC tagu je používána metoda `WriteOnce()` jenž jako argument přebírá objekty typu `Uint8Array`, neboli pole bajtů. Nejdříve je zkonstruován objekt NDEF záznamu s vlastnostmi jeho obsahu a typu. Finální zapsání proběhne předáním tohoto objektu metodě `ndef.write()`.

Neboť může při jakékoliv z těchto operací dojít k chybě, například z důvodů špatného umístění mobilního telefonu nad NFC periferií zařízení, je nutné aby jedna chyba neukončila celou operaci a zmátla tak uživatele. Kvůli těmto situacím jsou operace prováděny ve smyčce jenž se pokusí o vykonání až 20 krát.

## 7.8 Hardware zařízení

Pro výsledný plošný spoj zařízení byl zvolen modul Seed studio LoRa E5 založený na mikrokontroléru STM32WLE5. Jako NFC tag byl zvolen čip ST25DV v pouzdru typu SO8N. Pro návrh schématu byl využit program Altium Designer 22 a to s využitím studentské licence. Návrh byl rozdělen na obecné schéma s komponenty a poté samotné navádění tras mezi komponenty na desce.

## Zapojení modulu

Schéma a také rozložení podložek na desce modulu je oficiálně dostupné od výrobce Seed Studio jako knihovna pro odlišný návrhový program Eagle. Altium Designer však poskytuje možnost tuto knihovnu importovat a převést na vlastní formát.

V zapojení je na RFIO pin modulu připojen SMA konektor pro anténu napřímo a nejsou použity ladící kondenzátory ani cívka. Pro možnost využití manuálního resetu je pin RST sériově spojen společně s rezistorem o velikosti  $2\text{ k}\Omega$  přes 2.54mm patici k zemi. Stejným způsobem je pak zapojen pin PB13 jenž je připojen k patici označenou jako BOOT\_PIN. Velikost těchto rezistorů byla zvolena stejná jako již použité rezistory pro I2C sběrnici, neboť jejich hodnota nemá zásadní vliv na chování, protože na tyto piny stačí přivést zem.

Pro rychlý způsob ladění bez použití ladičky je z modulu také vyvedeno sériové rozhraní a přivedeno na patici UART. Samozřejmostí je pak společná zem taktéž přivedená k této patici.

## Integrace NFC tagu

NFC čipu ST25DV je nutné přivést napájení. Kvůli možnosti vypínat toto napájení za chodu procesorem, je připojeno přímo na GPIO pin modulu. Toto řešení se může zdát nestandardní, avšak je přímo doporučeno v dokumentaci výrobcem čipu.

Pro správné zapojení I2C sběrnice mezi čipem a modulem je nutné použít pull-up rezistory jejichž hodnota se musí nacházet v intervalu  $R_{min}$  a  $R_{max}$  vypočtenými následujícími vzorci.

$$R_{min} = \frac{V_{CC} - V_{OL}}{I_{OL}}, \quad R_{max} = \frac{t_r}{0,8379 \times C_b} \quad (7.1)$$

$$R_{min} = \frac{3,3\text{ V}}{40\text{ mA}}, \quad R_{max} = \frac{1000\text{ nS}}{0,8379 \times 400\text{ pF}} \quad (7.2)$$

$$R_{min} = 82,5\ \Omega, \quad R_{max} = 2983\ \Omega \quad (7.3)$$

$V_{CC}$  je napájecí napětí I2C sběrnice

$V_{OL}$  je napětí sběrnice při nízké úrovni

$I_{OL}$  je maximální možný proud při nízké úrovni

$t_r$  je doba trvání jednoho hodinového signálu

$C_b$  je maximální kapacita sběrnice

Na základě vypočteného intervalu byla pro pull-up rezistory zvolena hodnota  $2\text{ k}\Omega$ . Poslední částí zapojení NFC čipu je vývod GPO pinu což je pin přerušení použitý k detekci událostí zápisu do NFC tagu.

## Napájení

Napájení modulu je přivedeno z baterie za použití konektoru typu JST-EH jenž je běžným konektorem baterií používaných v praxi pro takovéto účely. Pro snížení šumu a vyrovnání proudových špiček je mezi VCC a GND umístěn kondenzátor o velikosti  $4,7\mu\text{F}$  jehož hodnota byla převzata z dokumentace modulu LoRa E5.

## Ladící port

Pro nahrání firmwaru a také jeho případné ladění je nutné použít rozhraní SWD pro ladění. Neboť jsem nenašel schéma ani rozložení již existujícího SWD/JTAG konektoru, bylo nutné jej navrhnout. Vytvořena byla tedy knihovna `Connector4048SWD.LibPkg` obsahující schéma a půdorys tohoto konektoru.

Ke konektoru jsou vyvedeny signály `SWDIO`, `SWO`, `SWCLK` samotného rozhraní SWD, poté signál `RESET` a napájení.

## Stavové diody

K zobrazení aktuálního stavu ve kterém se nachází zařízení byly přidány dvě stavové diody. Tyto stavové diody slouží k signalizaci aktuálního režimu a také signalizaci odesílání sítí LoRaWAN. Společně s diodou jsou sériově zapojeni rezistory pro limitaci proudu o velikosti  $470\Omega$ .

## Převod schématu na výslednou desku

Převod schématu na výsledný plošný spoj a trasování bylo taktéž provedeno v programu Altium Designer 22.

Téměř všechny součástky na plošném spoji jsou typu SMD (Surface Mount Device) krom použitého konektoru pro napájení a také patice pro `BOOT_PIN`, `RST` a `UART`. Tento konektor a patice jsou umístěny okolo veškeré elektroniky pro snadnější přístup

Na desce byl modul umístěn tak, aby jeho `RFIO` pin směřoval nahoru a to k místu kde je umístěn SMA konektor pro anténu. Šířka této trasy byla nastavena tak, aby se její impedance co nejvíce blížila  $50\Omega$ .

## Návrh NFC antény

Správné fungování NFC tagu zajišťuje také správně navržená anténa, což je v tomto případě cívka v podobě měděné trasy na plošném spoji. Důležitou vlastností takovéto cívky je její indukance, jež musí mít takovou hodnotu, aby celková frekvence obvodu dosahovala  $13,56\text{MHz}$ . K výpočtu indukance slouží tento vzorec [16].

$$L_{ant} = \frac{1}{(2\pi \times f)^2 \times C_c} \quad (7.4)$$

$$L_{ant} = \frac{1}{(2\pi \times 13,56\text{MHz})^2 \times 28,54\text{pF}} = 4,83\mu\text{H} \quad (7.5)$$

$f$  je cílová frekvence

$C_c$  je vnitřní kapacita NFC tagu

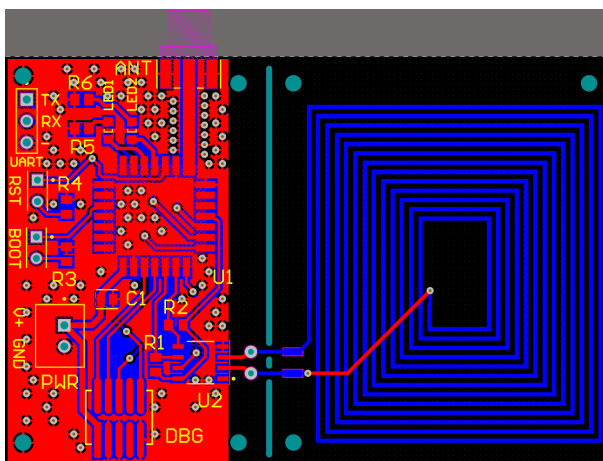
Induktance navržené cívky pro frekvenci  $13,56\text{MHz}$  s vnitřní kapacitou NFC tagu  $28,54\text{pF}$  (hodnota z dokumentace čipu `ST25DV`) musí být  $4,83\mu\text{H}$ .

K návrhu NFC antény, byl využit nástroj *NFC Inductance / eDesignSuite*<sup>1</sup> společnosti STM. Ten na základě vložených parametrů cívky vypočítá její indukanci. Mezi tyto parametry patří:

- Počet otáček
- Celková šířka a délka antény
- Šířka trasy
- Mezera mezi trasami
- Tloušťka materiálu (mědi)
- Materiál a tloušťka substrátu plošného spoje

Princípem pokus omyl bylo cílené indukance dosaženo a na základě těchto parametrů byla nakreslena cívka, jež byla následně programem Altium Designer 22 převedena na trasu na plošném spoji. Anténa je umístěna na dolní vrstvě aby bylo možné plošný spoj otočit anténou směrem k víku krabice se zařízením.

Obrázek 7.4 zobrazuje výsledný plošný spoj s anténou v pravé části.



Obrázek 7.4: Výsledný plošný spoj

<sup>1</sup><https://eds.st.com/antenna/#/?app=analogsimulator>

## Kapitola 8

# Vyhodnocení

Tato kapitola popisuje výsledek práce a také vyzkoušení zařízení v testovacím provozu k změření jeho funkčních vlastností jako je dosah zařízení v závislosti na typu antény, či jeho spotřeba.

### Výsledné zařízení

Po dokončení výroby plošného spoje, byly vytvořeny celkově tři verze zařízení. Každá z těchto verzí se mírně liší počtem osazených součástek či jejich typem.

V případě první z desek byly osazeny součástky všechny a k místu s vývodem antény byl osazen konektor typu SMA na nějž je možné přišroubovat anténu pro konkrétní frekvenci. Jedná se tedy o podobu zařízení tak, jak by

Druhá deska byla také osazena celá ovšem místo SMA konektoru byl připájen drát o délce 8,6cm což je  $1/4$  vlnové délky frekvence 868 MHz.

V případě třetího plošného spoje, byl osazen pouze modul Seed Studio LoRa E5, kondenzátor na napájecí větvi, a SWD konektor pro nahrání programu.

Důvodem realizace tří odlišných verzí byla potřeba změření spotřeby samotného modulu a dále také zjištění vlivu antény na kvalitu přijímaného signálu. V případě použití drátu místo SMA konektoru a externí antény se totiž může jednat o rozdíl v nákladech až sto korun.



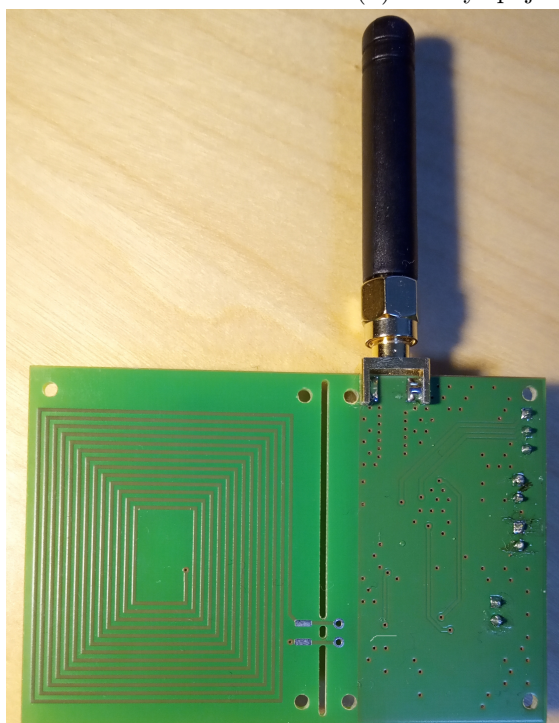
Do osazených modulů LoRa E5 na desce byl nahrán vytvořený firmware a postupně otestována funkčnost celého zařízení včetně změření parametrů. Obrázky 8.1 zobrazují dvě z osazených plošných desek včetně pohledu na desku z opačné strany.



(a) Plošný spoj s SMA konektorem a anténou



(b) Plošný spoj s vodičem místo antény



(c) Zadní strana osázeného plošného spoje

Obrázek 8.1: Výsledné osazené plošné spoje

## Spotřeba zařízení

Spotřeba výsledného zařízení je velmi důležitou vlastností, neboť z ní vyplývá doba, po kterou dokáže toto zařízení pracovat bez vnějšího zásahu na připojené baterii. Z tohoto důvodu bylo provedeno měření spotřeby v jednotlivých stavech zařízení.

K měření byl využit přípravek Power Profiler Kit II <sup>1</sup> společnosti Nordic Semiconductor, jenž umožňuje měřit aktuální odebíraný proud v rozsahu 100nA až 1A. Pro měření je přípravek schopen poskytovat také napájecí napětí nastavitelné v rozsahu 0,8V až 5V.

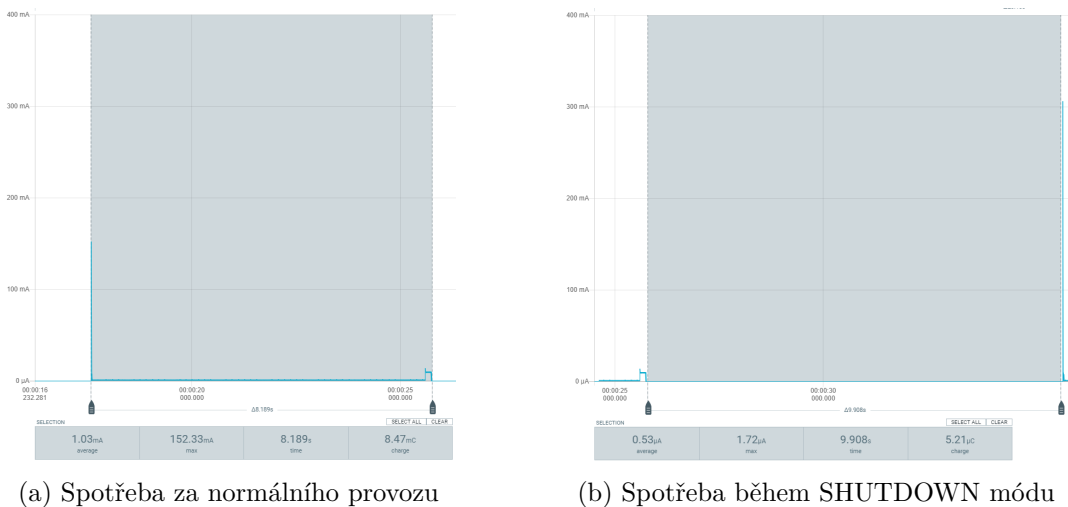
Pro měření odběru proudu bylo jako napájecí napětí zařízení zvolena hodnota 3,3V.

### Základní spotřeba

Prvně byla změřena orientačně spotřeba desky ve stavu kdy byl mikroprocesor v běžném provozu bez jakýchkoliv povolených periferií a hlavní frekvence procesoru byla nastavena na 48MHz. V tomto případě byl naměřen odebíraný proud 5,7mA.

Dále byla změřena spotřeba ve stejném režimu, ovšem tentokrát s frekvencí procesoru pouze 4MHz. V této situaci byla naměřena spotřeba pouze 1mA. To znamená, že pouhým snížením frekvence procesoru je možné snížit spotřebu ve chvílích kdy procesor není zrovna v žádném z režimů spánku.

Protože stráví zařízení většinu svého času v nejhlubším režimu spánku, byla změřena jeho spotřeba v nízkopříkonovém režimu *SHUTDOWN* společně s běžícími hodinami reálného času RTC, jenž jsou potřeba pro opětovné probuzení. Naměřená spotřeba dosahovala pouze  $0,53\mu A$  což je dostatečně nízká hodnota pro dlouhodobý provoz. Obrázky 8.2 zobrazují spotřeby během těchto režimů provozu.



Obrázek 8.2: Grafy spotřeby za provozu bez periferií

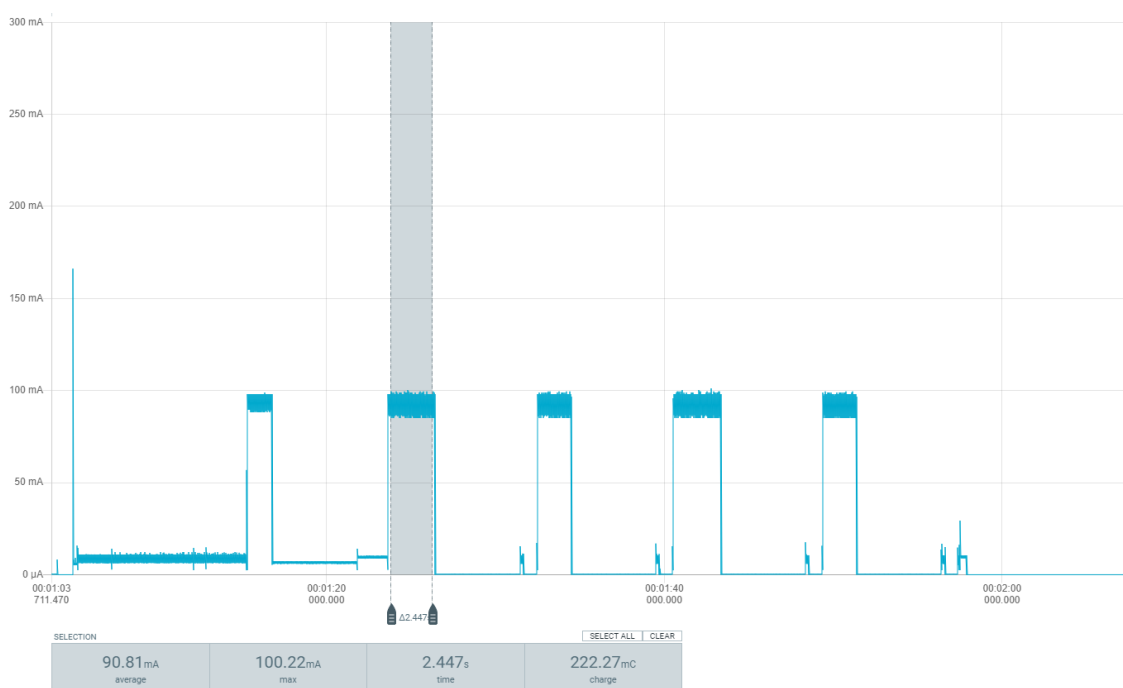
<sup>1</sup><https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2>

## Spotřeba během reálného provozu

Spotřeba za reálného provozu je velmi důležitá, neboť během stavů přijímání či odesílání rozhodně není zanedbatelná. Pro zjednodušení měření, byly sníženy časy po které je zařízení v určitých stavech. Hned po spuštění zařízení je zahájeno přijímání WMBUS rámců, v tomto stavu je průměrná spotřeba 7mA. Po dokončení příjmu je provedeno připojení k LoRaWAN síti, po jeho úspěšnosti pak dojde k odesílání dat.

V této části programu dosahovalo zařízení odebíraného proudu až 90,8mA, což je oproti příjmu velmi vysoká hodnota. Během odesílání byl transceiver nastaven na vysílací výkon +14dBm a naměřená spotřeba téměř odpovídá hodnotě 92mA jenž je pro tento výkon uvedena v katalogovém listu SoC<sup>2</sup>.

Graf spotřeby v celém provozu je možné spatřit na obrázku 8.3. Na počátku grafu lze vidět stav přijímání, následuje vysílání LoRa mezi nimiž je procesor v režimu *STOP2* pro snížení spotřeby. Zvýrazněný úsek reprezentuje spotřebu během vysílacího okna.



Obrázek 8.3: Graf spotřeby jedné hlavní smyčky programu

V případě, že počet uložených zařízení bude 60 a průměrná velikost jednoho WMBUS rámce bude 60 bajtů, bude při nejhorším možném využitím *Data rate* sítě celková doba transceiveru v režimu odesílání 156 sekund (*Data rate* 0 dokáže odeslat 18,2 bajt). Za předpokladu, že je doba sbírání nastavena na 15 minut, bude během přijímání spotřebováno 1,75 mAh ( $7mA \times 15min$ ). Během odesílání pak 3,94mAh ( $90,2mA \times 156s$ ) a po zbytek dne v režimu spánku pouhých 0,0125mAh ( $0,53\mu A \times (24h - 15min - 156s)$ ). Za jeden cyklus tohoto zařízení je tedy spotřebováno celkem 5,71mAh což v případě použité baterie s kapacitou 19Ah stačí pro zhruba 9 let provozu. Takováto spotřeba je však pro nejhorší situaci, kdy se zařízení nachází v oblasti s horším pokrytím sítě LoRaWAN a není schopné využít nejrychlejší *Data rate*.

<sup>2</sup><https://www.st.com/resource/en/datasheet/stm32wle5c8.pdf> strana 70

## Závislost síly signálu na anténě

Neboť přidává SMA konektor a anténa na ceně zařízení, byla druhá deska osazena pouze drátem. Následně byla změřena síla signálu při přijímání v závislosti na typu desky.

Měření probíhalo následovně. Do seznamu uložených zařízení byla přidána adresa měřidla. Následně byly ve vzdálenostech 5, 10 a 15 metrů, po dobu 15 minut přijímány WMBUS rámce a bylo vypočteno průměrné RSSI<sup>3</sup>.

Vzdálenost [m]	SMA + anténa [dBm]	Drát místo antény [dBm]
5	-55	-55
10	-65	-73
15	-75	-77

Tabulka 8.1: Naměřené průměrné RSSI pro rozdílné vzdálenosti

Tabulka 8.1 ukazuje naměřené hodnoty. Z těchto hodnot, lze usoudit, že použití drátu místo externí antény má mírný negativní vliv na sílu přijatého signálu. Ve vzdálenosti 15 metrů je rozdíl v řádech jednotek dBm. Takovýto rozdíl však může být akceptovatelný bereme-li v potaz to, že nebyla provedena kompenzace impedance anténního výstupu na desce. Cenu celého zařízení lze tak snížit bez většího dopadu na jeho dosah.

---

<sup>3</sup>Received Signal Strength Indication — Indikátor síly přijímaného signálu udávaný v dBm

## Kapitola 9

# Závěr

Cílem této práce bylo vytvořit baterií napájené a konfigurovatelné zařízení, jenž po určitou dobu přijímá data z WMBUS měřidel a přijaté rámce odesílá sítí LoRaWAN čímž efektivně prodlužuje dosah takovýchto měřidel. Všechny tyto cíle byly úspěšně splněny a v případě přidání krabice do níž by bylo zařízení uzavřeno, by bylo možné jej reálně využít pro vzdálený odečet.

Na začátku práce byl proveden průzkum standardu WMBUS a jeho využití v moderních měřicích vody. Dále byly zjištěny informace o bezdrátových technologiích LoRa a konkrétně sítím LoRaWAN a pro účely konfigurace také zvolené technologie NFC.

Po porovnání využitelných platforem byla na základě jejich výhod vybrána řada procesorů STM32WL pro níž byl implementován firmware. Zařízení je plně schopné příjmu WMBUS rámců režimů C1 a T1. Tyto rámce je schopno dle konfigurace filtrovat a uložit do paměti. Po dokončení příjmu jsou rámce po částech odeslány sítí LoRaWAN.

Pro účely konfigurace zvolenou technologií NFC bylo implementováno webové rozhraní v jazycích HTML a JavaScript umožňující rychlé nastavení parametrů zařízení pro jeho fungování.

Pro finální podobu zařízení byla navržena a vyrobena deska plošných spojů s modulem LoRa E5. Součástí je také cívková NFC anténa v podobě měděné trasy na plošném spoji.

Výsledná implementace dosahuje velmi nízké spotřeby v režimech spánku s odebíraným proudem  $0,53\mu A$ , jenž společně s nízkým odběrem během příjmu zaručuje dlouholetý provoz v nejhorších případech až 9 let.

Práce mi dala spoustu nových zkušeností, hlavně se jedná o zkušenosti s bezdrátovými technologiemi v oblasti mikrokontrolérů, ale také návrh plošného spoje s modulem mikrokontroléru. Dále jsem se naučil pracovat s nástroji a mikroprocesory společnosti STM.

V práci bych chtěl do budoucna implementovat komplexnější správu událostí jako je probuzení a odesílání za použití kalendáře do něhož by šlo vkládat přesně časované spouštění událostí. Další možností rozšíření je konfigurace zařízení na dálku použitím právě již využitě sítě LoRaWAN.

# Literatura

- [1] *ABP vs OTAA / The Things Stack for LoRaWAN* [online]. [cit. 2022-04-08]. Dostupné z: <https://www.thethingsindustries.com/docs/devices/abp-vs-otaa/>.
- [2] *CRA IoT Services - LoRa Alliance®* [online]. [cit. 2022-04-05]. Dostupné z: [https://lora-alliance.org/lora\\_products/cra-iot-services/](https://lora-alliance.org/lora_products/cra-iot-services/).
- [3] *Data Encoding Techniques* [online]. [cit. 2022-03-17]. Dostupné z: [https://www.tutorialspoint.com/digital\\_communication/digital\\_communication\\_data\\_encoding\\_techniques.htm](https://www.tutorialspoint.com/digital_communication/digital_communication_data_encoding_techniques.htm).
- [4] *Listen Before Talk in Wireless / LBT Advantages disadvantages*. [cit. 2022-04-09]. Dostupné z: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Listen-Before-Talk-in-Wireless-Communication.html>.
- [5] *LoRa and LoRaWAN: Technical overview / DEVELOPER PORTAL* [online]. [cit. 2022-04-05]. Dostupné z: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan>.
- [6] *Manchester Encoding* [online]. [cit. 2022-03-17]. Dostupné z: <https://erg.abdn.ac.uk/users/gorry/course/phy-pages/manc.html>.
- [7] *REPEATER uR-series OMS Wireless M-Bus*. [cit. 2022-05-01]. Dostupné z: <https://www.lansensystems.com/assortments/infrastructure/wireless-m-bus-repeaters>.
- [8] *STM32WL Series - STMicroelectronics*. [cit. 2022-05-07]. Dostupné z: <https://www.st.com/en/microcontrollers-microprocessors/stm32wl-series.html>.
- [9] *What is LoRaWAN and how it works, architecture and specifications* [online]. [cit. 2022-03-17]. Dostupné z: <https://telemetric.tech/en/lorawan-technology-and-specifications/>.
- [10] *What is LoRaWAN® Specification - LoRa Alliance®* [online]. [cit. 2022-04-05]. Dostupné z: <https://lora-alliance.org/about-lorawan/>.
- [11] *What is Wireless M-Bus: Meter Bus Technology > Electronics Notes* [online]. [cit. 2022-03-12]. Dostupné z: <https://www.electronics-notes.com/articles/connectivity/wireless-meter-bus/what-is-wireless-m-bus-basics.php>.
- [12] *Wireless M-Bus (wM-Bus) - STMicroelectronics* [online]. [cit. 2022-03-12]. Dostupné z: <https://www.st.com/en/applications/connectivity/wm-bus.html>.
- [13] *Komunikační systémy pro měřidla – Část 4: Bezdrátová komunikace M-BUS*. Norma ČSN EN-13757-4. Česká agentura pro standardizaci, únor 2014.

- [14] *ST25 NFC guide*. DocID027940. STMicroelectronics, říjen 2016. Rev. 2.
- [15] *LoRaWAN Security Whitepaper*. GEMALTO, ACTILITY AND SEMTECH, únor 2017. Dostupné z: [https://lora-alliance.org/wp-content/uploads/2020/11/lorawan\\_security\\_whitepaper.pdf](https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_security_whitepaper.pdf).
- [16] *How to design a 13.56 MHz customized antenna for ST25 NFC / RFID Tags*. AN2866. STMicroelectronics, srpen 2021. Rev. 5.
- [17] *STM32WLEx advanced Arm®-based 32-bit MCUs with sub-GHz radio solution*. RM0461. STMicroelectronics, duben 2022. Rev. 5.
- [18] ATTARAN, A., ALTUNYURT, N., LOCKE, J. a DELONG, A. NFC Reader Antenna Design and Considerations for Automotive Applications. In: *2021 Antenna Measurement Techniques Association Symposium (AMTA)*. 2021, s. 1–6. DOI: 10.23919/AMTA52830.2021.9620682.
- [19] CHO, J.-H., KIM, J., KIM, J.-W., LEE, K., AHN, K.-D. et al. An NFC transceiver with RF-powered RFID transponder mode. In: *2007 IEEE Asian Solid-State Circuits Conference*. 2007, s. 172–175. DOI: 10.1109/ASSCC.2007.4425758.
- [20] ERTÜRK, M. A., AYDIN, M., BÜYÜKAKKAŞLAR, T. a EVIRGEN, H. A Survey on LoRaWAN Architecture, Protocol and Technologies. *Future Internet*. Říjen 2019, sv. 11, s. 216. DOI: 10.3390/fi11100216.
- [21] KIM, D.-H., LEE, E.-K. a KIM, J. Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed. *Sustainability*. Březen 2019, sv. 11, s. 1917. DOI: 10.3390/su11071917.
- [22] WIKIPEDIA CONTRIBUTORS. *ESP32 — Wikipedia, The Free Encyclopedia* [online]. [cit. 2022-05-08]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=ESP32&oldid=1086524212>.
- [23] WIKIPEDIA CONTRIBUTORS. *Frequency-shift keying — Wikipedia, The Free Encyclopedia* [online]. [cit. 2022-05-07]. Dostupné z: [https://en.wikipedia.org/w/index.php?title=Frequency-shift\\_keying&oldid=1083766660](https://en.wikipedia.org/w/index.php?title=Frequency-shift_keying&oldid=1083766660).
- [24] WIKIPEDIA CONTRIBUTORS. *LoRa — Wikipedia, The Free Encyclopedia* [online]. [cit. 2022-05-07]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=LoRa&oldid=1086603573>.
- [25] WIKIPEDIA CONTRIBUTORS. *Near-field communication — Wikipedia, The Free Encyclopedia* [online]. [cit. 2022-05-07]. Dostupné z: [https://en.wikipedia.org/w/index.php?title=Near-field\\_communication&oldid=1082332561](https://en.wikipedia.org/w/index.php?title=Near-field_communication&oldid=1082332561).

## Příloha A

# Obsah přiloženého paměťového média

/	
—firmware/	- Zdrojové kódy firmwaru pro STM32WL
—pcb/	
—RelayPCB/	- Schéma a návrh hlavního PCB
—Connector4048SWD/	- Schéma a půdorys SWD konektoru
—LoraE5Lib/	- Importovaná knihovna schématu a půdorysu modulu
—web/	
—index.html	- Zdrojový kód stránky
—js/	- Skripty obsluhující stránku
—libraries/	- Použité knihovny
—latex/	- Zdrojové texty technické zprávy
—zprava.pdf	- Výsledná technická zpráva