



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

### ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

## BLUETOOTH LOKALIZACE S NÍZKOU SPOTŘEBOU ENERGIE

LOW ENERGY BLUETOOTH LOCALIZATION

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Jan Šlígł

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Tomáš Götthans, Ph.D.

BRNO 2022

# Diplomová práce

magisterský navazující studijní program **Elektronika a komunikační technologie**

Ústav radioelektroniky

**Student:** Bc. Jan Šlígľ

**ID:** 203352

**Ročník:** 2

**Akademický rok:** 2021/22

**NÁZEV TÉMATU:**

## **Bluetooth lokalizace s nízkou spotřebou energie**

### **POKYNY PRO VYPRACOVÁNÍ:**

V teoretické části práce prostudujte technologii otevřeného bezdrátového standardu Bluetooth. Seznamte se s verzemi standardu s důrazem na režim „Low Energy“ a funkce monitoringu signálu. Obzvláště se zaměřte na specifikaci Bluetooth 5.2. Vyberte si obvodové řešení, které umožní testování radiofrekvenční části a připojení dalších obvodů – například filtru. Realizujte hardware navrženého řešení (návrh, osazení). Dále navrhnete anténu (několik provedení) s všesměrovou charakteristikou.

Realizované zařízení zprovozněte, vytvořte firmware a ovládací software pro PC. Pomocí PC nebo MCU nastavujte a vyčítejte vybrané parametry (například údaje o síle signálu) obvodu a kvantifikujte jejich dopad na přenos signálu. Provedte měření charakteristik navržených antén, analyzujte vliv charakteristik na přenos signálu. Dále vytvořte algoritmus, který umožní lokalizovat vysílač v prostoru na základě naměřené síly signálu.

### **DOPORUČENÁ LITERATURA:**

[1] ALSMADI, L.; KONG, X.; SANDRASEGARAN, K.; FANG, G.; An Improved Indoor Positioning Accuracy using Filtered RSSI and Beacon Weight, in IEEE Sensors Journal, doi: 10.1109/JSEN.2021.3085323.

[2] Bluetooth Technology Website. Bluetooth Technology Website [online]. [cit. 2021-6-21]. Dostupné z: <https://www.bluetooth.com/>

**Termín zadání:** 11.2.2022

**Termín odevzdání:** 25.5.2022

**Vedoucí práce:** doc. Ing. Tomáš Götthans, Ph.D.

**prof. Dr. Ing. Zbyněk Raida**  
předseda rady studijního programu

### **UPOZORNĚNÍ:**

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

## **ABSTRAKT**

Tato diplomová práce se zabývá návrhem lokalizačního zařízení v Bluetooth 5.2 s nízkou spotřebou. Lokalizace je provedena pomocí vytvořených zařízení tvořících beacony a navržené aplikace pro mobilní telefon. Dále také návrhem více druhů antén s různými směrovými charakteristikami, ziskem a velikostmi. Návrh bere v potaz především nízkou spotřebu energie, kompaktnost a nezávislost zařízení.

## **KLÍČOVÁ SLOVA**

Lokalizace, Bluetooth 5.2, mobilní aplikace, nízká spotřeba energie, baterie

## **ABSTRACT**

This diploma thesis is focused on design of localization device using Bluetooth 5.2 with low power consumption. Localization is performed using created devices forming a beacons and designed application for mobile phone. Also design of multiple types of antennas with different directional radiation pattern, gain and size. Design is mostly focusing on low power consumption and compact independent device.

## **KEYWORDS**

Localization, Bluetooth 5.2, mobile application, low power consumption, battery

ŠLÍGL, Jan. *Bluetooth lokalizace s nízkou spotřebou energie*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2022, 87 s. Diplomová práce. Vedoucí práce: doc. Ing. Tomáš Götthans, Ph.D.

## Prohlášení autora o původnosti díla

**Jméno a příjmení autora:** Bc. Jan Šlígł  
**VUT ID autora:** 203352  
**Typ práce:** Diplomová práce  
**Akademický rok:** 2021/22  
**Téma závěrečné práce:** Bluetooth lokalizace s nízkou spotřebou energie

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno .....

.....

podpis autora\*

---

\* Autor podepisuje pouze v tištěné verzi.

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Tomáši Götthansovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

# Obsah

Úvod	12
<b>1 Bluetooth</b>	<b>13</b>
1.1 Standardy	13
1.1.1 Bluetooth Classic	13
1.1.2 Bluetooth Low Energy	14
1.1.3 Bluetooth 5.2	16
<b>2 Obvodové řešení</b>	<b>18</b>
2.1 Parametry	18
<b>3 Výběr antén pro zařízení</b>	<b>19</b>
3.1 Požadavky	19
3.2 Provedení	19
3.2.1 Mikropásková flíčková anténa	19
3.2.2 Mikropásková flíčková anténní řada	22
3.2.3 Planární invertovaná F anténa	26
3.2.4 Čipová anténa	29
3.3 Srovnání	30
<b>4 Návrh zařízení</b>	<b>33</b>
4.1 Napájení	33
4.2 Připojení vř části	34
4.3 Deska plošných spojů	36
<b>5 Firmware zařízení</b>	<b>38</b>
5.1 Bluetooth stack - Silicon Labs	38
5.2 Funkce programu	39
5.3 Optimalizace spotřeby baterie	41
5.3.1 Spotřeba energie	41
5.3.2 Spotřeba energie po optimalizaci	42
<b>6 Vyrobené antény</b>	<b>43</b>
6.1 Srovnání simulace a měření	43
<b>7 Aplikace pro telefon</b>	<b>45</b>
7.1 Funkce programu	45
7.2 Provedení lokalizace	48

<b>8</b>	<b>Lokalizace - měření</b>	<b>50</b>
8.1	Měření vzdálenosti . . . . .	50
8.2	Měření lokalizace . . . . .	53
	<b>Závěr</b>	<b>57</b>
	<b>Literatura</b>	<b>60</b>
	<b>Seznam příloh</b>	<b>63</b>
<b>A</b>	<b>Deska plošných spojů navrženého zařízení</b>	<b>64</b>
A.1	Motiv desky . . . . .	64
A.2	Osazovací plán desky . . . . .	66
A.3	Seznam součástek . . . . .	68
A.4	Fotografie . . . . .	69
A.5	Deska pro pomocné měření . . . . .	72
A.5.1	Motiv desky . . . . .	72
A.5.2	Fotografie . . . . .	74
<b>B</b>	<b>Zdrojový kód zařízení</b>	<b>76</b>
<b>C</b>	<b>Zdrojový kód aplikace</b>	<b>81</b>



# Seznam obrázků

1.1	Topologie Bluetooth Low-Energy . . . . .	15
1.2	Topologie Bluetooth Low-Energy detailněji . . . . .	15
1.3	Bluetooth Low Energy kanály a frekvence. . . . .	16
1.4	Synchronní kanály (ISOC) . . . . .	17
3.1	Flíček s vloženým vedením . . . . .	20
3.2	Návrh flíčku s vloženým vedením . . . . .	21
3.3	Vstupní činitel odrazu flíčku s vloženým vedením . . . . .	21
3.4	Vyzařovací charakteristika (v polárních souřadnicích) flíčku s vloženým vedením . . . . .	22
3.5	Paralelní flíčková řada . . . . .	23
3.6	Návrh paralelní flíčkové řady . . . . .	24
3.7	Vstupní činitel odrazu paralelní flíčkové řady . . . . .	24
3.8	Vyzařovací charakteristika (v polárních souřadnicích) paralelní flíčkové řady . . . . .	25
3.9	Invertovaná F anténa napájená koplanárním vlnovodem . . . . .	26
3.10	Návrh planární inverotvané F antény . . . . .	27
3.11	Vstupní činitel odrazu PIFA . . . . .	27
3.12	Vyzařovací charakteristika (v polárních souřadnicích) PIFA . . . . .	28
3.13	Vstupní činitel odrazu čipové antény . . . . .	30
3.14	Vyzařovací charakteristika čipové antény . . . . .	30
3.15	Vstupní činitel odrazu všech designů . . . . .	31
4.1	Pinout EFR32BG22 . . . . .	33
4.2	Schéma zařízení . . . . .	35
4.3	Zobrazení parametrů koplanárního vedení . . . . .	36
5.1	Silicon Labs Bluetooth stack . . . . .	38
5.2	Vývojový diagram programu pro EFR32BG22 . . . . .	39
5.3	Výchozí spotřeba energie zařízení . . . . .	41
5.4	Spotřeba energie zařízení po úpravě . . . . .	42
6.1	Změřený vstupní činitel odrazu všech vyrobených antén . . . . .	43
7.1	Ukázka aplikace ve fázi hledání beaconů . . . . .	46
7.2	Vývojový diagram aplikace pro telefon . . . . .	47
7.3	Screenshot aplikace ve fázi lokalizace . . . . .	49
8.1	Měření velikosti RSSI pro jednotlivé antény. . . . .	51
8.2	Vypočtená vzdálenost . . . . .	52
8.3	Průběh měření lokalizace . . . . .	54
A.1	Vrchní strana desky plošných spojů . . . . .	64
A.2	Spodní strana desky plošných spojů . . . . .	65

A.3	Vrchní strana desky - osazení . . . . .	66
A.4	Spodní strana desky - osazení . . . . .	67
A.5	Fotografie osazené desky - vrchní strana . . . . .	69
A.6	Fotografie osazené desky - spodní strana . . . . .	70
A.7	Vrchní strana . . . . .	72
A.8	Spodní strana . . . . .	73
A.9	Fotografie osazené pomocné desky - vrchní strana . . . . .	74
A.10	Fotografie osazené pomocné desky - spodní strana . . . . .	75

# Seznam tabulek

3.1	Parametry mikropáskové flíčkové antény . . . . .	22
3.2	Parametry mikropáskové flíčkové anténní řady . . . . .	25
3.3	Parametry planární invertované F antény . . . . .	28
3.4	Parametry čipové antény . . . . .	29
3.5	Srovnání parametrů antén . . . . .	31
4.1	Parametry koplanárního vedení . . . . .	36
6.1	Srovnání parametrů simulovaných a měřených antén . . . . .	43
8.1	Změřené hodnoty RSSI pro zvětšující se vzdálenost . . . . .	50
8.2	Vypočtené hodnoty vzdálenosti . . . . .	52
8.3	Velikost odchylky vypočtené hodnoty od reálné vzdálenosti . . . . .	53
8.4	Vzdálenosti Beaconů od bodů měření . . . . .	54
8.5	Velikost odchylky v lokalizaci . . . . .	55
8.6	Velikost odchylky v lokalizaci - odebrání krajních bodů . . . . .	56
A.1	Seznam součástí . . . . .	68
A.2	Fotografie osazené desky - srovnání velikosti . . . . .	71

# Úvod

Se stále rostoucím počtem mobilních zařízení v rukou uživatelů rostou také možnosti jak pomáhat lidem s určitým fyzickým postižením, konkrétně ztrátou zraku. Jelikož se jedná o nejdůležitější vjem pro lidské tělo, je žádoucí vymýšlet zařízení, která jsou schopná alespoň částečně tyto funkce nahrazovat.

Hlavní motivací pro tuto práci je utvořit základ navigačního systému, který bude schopný lokalizace zařízení (společně s uživatelem) v prostoru a tím usnadnění pohybu v tomto prostoru.

Předmětem této práce je navrhnout zařízení, které pomocí systému Bluetooth, konkrétně specifikaci 5.2, bylo tvořit článek nebo jeden z článků lokalizačního systému zaměřujícího se na nízkou spotřebu energie. Je nutné vybrat vhodné obvodové řešení, které zároveň umožňuje monitorování signálu, s požadovanou radiofrekvenční částí. Bylo vytvořeno zařízení s důrazem na nízkou spotřebu energie s bateriovým napájením. U zařízení je možnost použít buďto integrovanou anténu nebo různé typy antén, které mohou být připojeny externě.

Další součástí práce je návrh několika druhů antén, které byly změřeny a střídavě použity při finálním měření lokalizace a vzdálenosti, aby bylo možné kvantifikovat, které řešení je vhodnější v dané situaci. Pro následné provedení lokalizace a měření síly signálu byla vytvořena aplikace pro mobilní telefon se systémem Android.

# 1 Bluetooth

Bezdrátová technologie Bluetooth přenáší data na 2,4 GHz bezlicenčním pásmu ISM (od 2,4 do 2,4835 GHz). Tato technologie byla vynalezena v roce 1994, následně se stala součástí *IEEE 802.15.1* standardu, který mimo jiné zajišťuje, že nikdo tuto technologii nevlastní. Původním záměrem bylo především nahrazení kabelových spojení periférií počítače, tento záměr ale postupem času přerostl v mnohem více a nyní technologie Bluetooth umožňuje propojit velké množství zařízení.

Proces připojení dvou zařízení pomocí Bluetooth se děje v třech krocích: *dotazování/naslouchání*, *paging a spojení*. Před samotným dotazováním nemá ani jedno ze zařízení informace o zařízení druhém. Poté jedno ze zařízení vyšle dotaz do svého okolí, pokud je dotaz zachycen vyšle zpět druhé zařízení svůj název. V procesu *pagingu* dochází mezi dvěma mezi sebou detekovanými zařízeními k výměně adres a připravuje se na poslední proces, tedy spojení. Pokud je navázáno spojení, mohou zařízení fungovat ve více módech: *active*, *sniff*, *hold a park*. V *active* režimu si zařízení aktivně sdílejí data mezi sebou. Ve *sniff* módu dochází k šetření energie, kdy jsou zařízení ve spánkovém módu a naslouchají opakovaně po časových úsecích, zda nebude docházet k přenosu. Mód *hold* je podobný jako předchozí mód, čas zde však může být nastaven ve větším rozsahu a také může jedno zařízení druhému přikázat do tohoto módu vstoupit a následně *park mód* je také obdobný, ale zde může řídicí zařízení podřízenému nastavit spánkový mód do doby, než ho opět vzbudí.

Párování je proces, při kterém se vytvářejí vazby mezi zařízeními. A dvě vzájemná zařízení mohou navázat spojení okamžitě, pokud jsou ve svém dosahu. Pokud jsou zařízení spárována, sdílejí svoje adresy a názvy a ukládají si je do své paměti. Zároveň také sdílí společný tajný klíč, který jim umožňuje navázat spojení, pokud jsou ve svém dosahu. Tento proces autentizace provádí uživatel (a to i v případě prvního navázání spojení dvou zařízení), může se jednat pouze o stisknutí tlačítka, ale také o zadání vícemístného kódu.

## 1.1 Standardy

### 1.1.1 Bluetooth Classic

Jedná se o verze Bluetooth 1.0 a 2.0 (případně i 3.0). Z hlediska verzí rozlišujeme především to, jaký má daná verze Bluetooth: dosah, datovou rychlost a spotřebu energie. Datová rychlost v případě Bluetooth 1.0 dosahovala maximálně 1 Mbps (megabitů za sekundu) a dosah se v ideálních podmínkách přibližoval 10 m. Zároveň v této verzi byla používána modulace GFSK (Gaussian Frequency Shift Keying).

Verze 2.0 následně přešla na modulaci  $\pi/4$ -DQPSK, díky čemuž se bylo dosaženo datových rychlostí až 2 a 3 Mbps. V rámci verze 3.0 vzrostla datová rychlost až na 24 Mbps, avšak tento údaj může být relativně zavádějící vzhledem k tomu, že ve verzi 3.0 je užíváno Bluetooth pouze k nastolení a udržování spojení, ale samotný přenos se děje přes Wi-Fi.

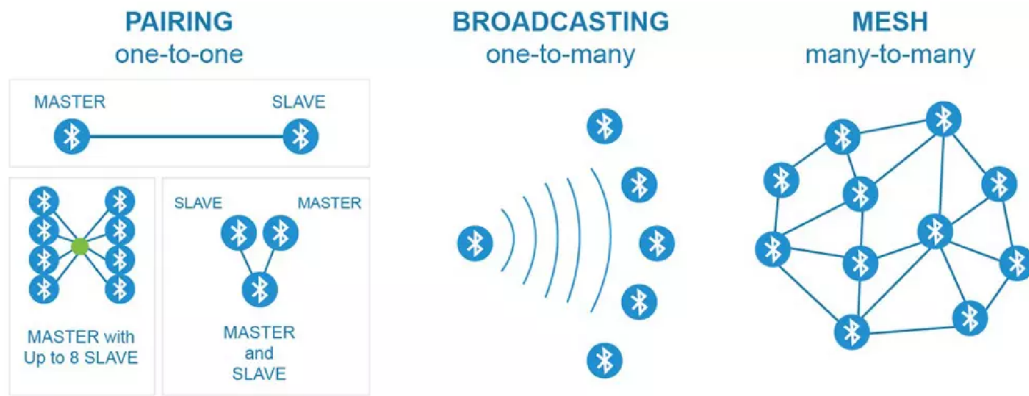
## 1.1.2 Bluetooth Low Energy

Bluetooth verze 4.0 rozdělilo Bluetooth do více kategorií: *Classic*, *High-Speed* a *Low-Energy*. Přičemž zásadní změnou a novinkou je režim Bluetooth Low-Energy (dále jen BLE). Zde došlo ke snížení datové rychlosti opět na 1 Mbps. Spojení typu BLE využívají především *sniff* módu, kdy nedochází k trvalému přenosu dat, ale pouze naslouchá, zda nebude docházet k přenosu. Toto umožňuje výhodné aplikace pro zařízení, která operují na malých bateriích jako jsou IoT aplikace.

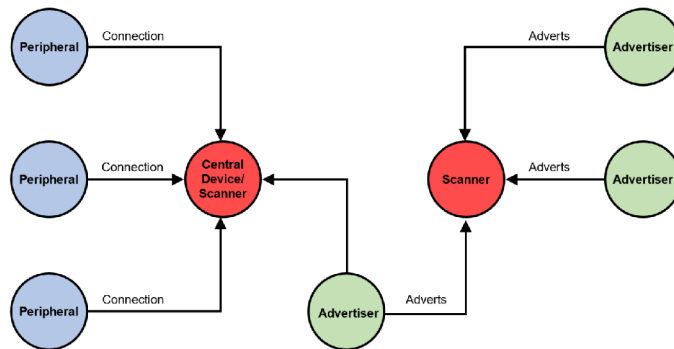
V další verzi 5.0 jsou vlastnosti BLE vylepšeny především k dalšímu snižování spotřeby energie. Avšak nabízí také zlepšení v dosahu a datové rychlosti. Verze 5.0 nabízí čtyři druhy rychlostí: 2 Mbps, 1 Mbps, 500 kbps (kilobitů za sekundu) a 125 kbps. Zároveň je také implementován větší rozestup mezi jednotlivými kanály, který nyní činí 2 MHz oproti standardnímu 1 MHz z předchozích verzí, to redukuje nároky na filtraci a činí přenos robustnější. Toto má však svou cenu v podobě snížení počtu kanálů na polovinu.

Jelikož je dosah závislý na datové rychlosti, je možné vždy volit takové nastavení pro konkrétní zařízení, aby byly pokryty nároky na vzdálenosti resp. datovou rychlost. Například v případě senzoru, který sbírá data jednou za minutu není nutná vysoká datová rychlost, avšak je pravděpodobné, že bude potřeba velký dosah a právě toto je umožněno. Již zde tedy není kladen důraz na přenos především hlasových služeb (případně kontinuálního proudu dat), pro který byl Bluetooth používán především.

Kromě základních rozdělení celkové sítě 1.1.2, můžeme na obrázku níže vidět jaké jsou například možné role v rámci Bluetooth Low-Energy. *Advertiser* - zařízení, které provádí vysílání paketů, ale není schopné je přijmout, zároveň může povolovat, nebo rušit spojení. *Scanner* - zařízení, které může pouze naslouchat vysílání a může být připojeno k advertiseru. *Peripheral* - zařízení, které může být připojeno k jednomu nebo více centrálním bodům. *Central* - zařízení, které může být připojeno k jedné nebo více periférií, zde tedy mohou být navzájem předávána data, příkazy a povely.



Obr. 1.1: Možné topologie sítě BLE: (zleva) párování, vysílání a smíšená síť (Mesh).  
Převzato z [3].



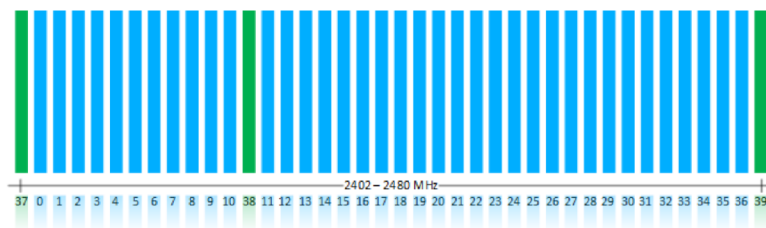
Obr. 1.2: Detailnější pohled na Bluetooth Low-Energy topologii s jednotlivými členy.  
Převzato z [4].

## Fyzická vrstva Bluetooth Low Energy

Celé 2,4 GHz ISM pásmo je rozděleno a 40 radiofrekvenčních kanálů s 2 MHz odstupem. Tři ze 40 kanálů se nazývají advertising kanály (1.1.2), používané pro nalezení zařízení, nastavení spojení a vysílání. Advertising kanály jsou vybrány takto z důvodu interference s kanály z IEEE 802.11 v některých zemích.

Zbylých 37 kanálů se používá buďto jako rozšířené advertising kanály nebo jako datové kanály. Ty se pak používají při obousměrné komunikaci mezi připojenými zařízeními. Zde se pro robustnost, spolehlivost a především odolnost vůči rušení, užívá AFH - Adaptive Frequency Hopping (Adaptive FHSS).

Všechny fyzické kanály užívají GFSK (Gaussian Frequency Shift Keying) modulaci s indexem modulace 0,5 (z důvodu redukce spotřeby ve špičkách).



Obr. 1.3: Bluetooth Low Energy - rozdělení kanálů a frekvencí. Převzato z [4].

### 1.1.3 Bluetooth 5.2

Bluetooth 5.2 nabízí tři vylepšení v podobě následujících funkcí:

- EAAT - Enhanced Attribute Protocol
- LEPC - LE Power Control
- ISOC - LE Isochronous Channels

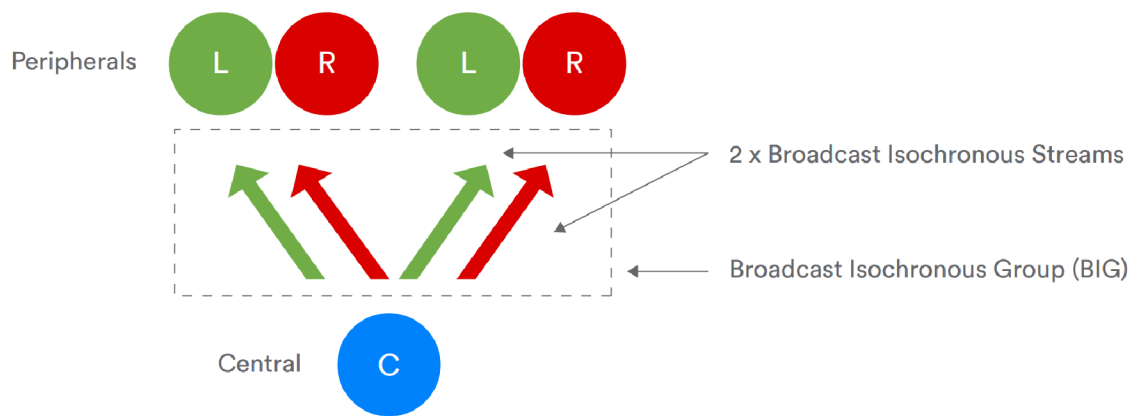
Originální protokol v BLE funguje sekvenčně, nový EAAT umožňuje, aby probíhaly paralelní operace v rámci klienta a serveru (master a slave, dvě zařízení apod.), čímž snižuje latenci operací. Toto je především výhodné pokud je na chytrý telefon připojeno více zařízení pomocí aplikací, které využívají BLE. Pokud tedy například z jednoho zařízení chytrý telefon přijímá data, není zároveň blokováno vysílání dat na jiné zařízení využívající jinou aplikaci.

LEPC umožňuje dynamicky optimalizovat vysílací výkon, který je používán při komunikaci mezi zařízeními. Příjímače v tomto systému mohou monitorovat sílu signálu a zažádat o změnu vysílacího výkonu. To umožňuje nastavovat výkon tak, aby byla zachována optimální síla signálu a zároveň nebylo spotřebováno nadměrně energie.

Výhody jsou tedy zřejmé, celkově nižší spotřebovaný výkon, zlepšení spolehlivosti spojení a také lepší koexistence ostatních bezdrátových zařízení (nejen Bluetooth), které pracují na rozsahu frekvencí okolo 2,4 GHz.

A konečně ISOC, které se zaměřuje na podporu Low-Energy audio zařízení. Tato verze umožňuje synchronizované přijímání dat na jedno nebo více zařízení. Prakticky to znamená, že je možné připojit více sluchátek na jeden vysílač audi-osignálu bez nutnosti odpojování jednoho připojeného zařízení. Nabízí taky nový standard, který podporuje připojování naslouchátek.





Obr. 1.4: Vysílání synchronního toku pro více periférií (sluchátek) [5].

## 2 Obvodové řešení

Pro optimální obvodové řešení by bylo vhodné, aby samotný mikroprocesor mohl zastávat co nejvíce funkcí a udržela se složitost zapojení na nižší úrovni. Minimálními požadavky jsou monitorování velikosti příchozího signálu a částečná lokalizace při nízké spotřebě celého zařízení.

Toto dobře umožní SoC (angl. System on Chip) typu *EFR32BG22 Wireless Gecko* od firmy *Silicon Labs*. Tento jednočipový systém umožňuje fungování v Bluetooth 5.2 pro IoT (Internet of Things) zařízení.

Tento systém byl vybrán z důvodů, které budou uvedeny dále, ale také díky možnosti testování čipu na vývojových deskách (v tomto případě typ *EFR32xG22 WG Starter* od *Silicon Labs*. Ty zároveň disponují vlastním vývojovým prostředím, které usnadňuje práci s jednotlivými deskami i čipy.

### 2.1 Parametry

Klíčové vlastnosti, kterými dle výrobce [10] disponují vybrané čipy jsou následující:

- 32-bit ARM Cortex-M33 s maximální operační frekvencí 76,8 MHz
- 512 kB Flash a 32 kB RAM
- Energeticky efektivní jádro s nízkými pracovními proudy a "sleep" proudy
- Bluetooth 5.2 umožňující Direction Finding neboli hledání směru
- Integrovaný výkonový zesilovač umožňující výkon na vysílači až 6 dBm

Vybráno bylo provedení typu QFN32, které svou velikostí 4 mm délky a 4 mm šířky umožňuje případně velmi malé provedení finálního zařízení.

Aby byla zachována samostatnost, nízká velikost a výdrž zařízení na místě po dlouhou dobu, byla pro napájení zvolena kruhová baterie typu CR2450 3V s kapacitou 580 mAh při průměru 24,5 mm. Velikost kruhové baterie v tomto případě přímo určí velikost vytvořené desky plošných spojů kvůli svému držáku.

Pro naprogramování obvodu slouží 5 vyvedených pinů na jumpery, zde může být připojen sériový programátor na vyvedené piny SWDIO, SWCLK, nRST, 3V3 a GND.

## 3 Výběr antén pro zařízení

### 3.1 Požadavky

Pro návrh antén jsou kladeny požadavky především z hlediska tvaru vyzařovacích charakteristik. Jedná se o grafické vyjádření směrových vlastností antén. Jejich tvar je důležitý především pro roviny E a H.

V tomto případě nás zajímá výběr takových antén, respektive jejich charakteristik, aby bylo možné si anténu vybrat pro různé aplikace.

### 3.2 Provedení

Při realizaci a výběru je třeba vybírat dle následujících faktorů antén

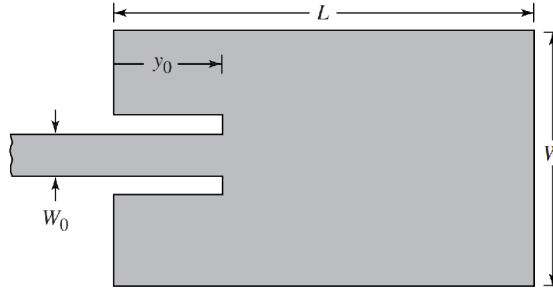
- Umístění: nutno zvážit, zda se anténa společně se zařízením bude nacházet např. v rohu místnosti, u stěny, v chodbě, ve volném prostoru.
- Cena: zda se vyplatí produkovat antény, kolik stojí nákup, jak drahý je materiál, z kterého bude anténa vytvořena, případně komponenty, které budou k anténě potřeba.
- Náročnost provedení: zda je anténa pouze vyfrézována na povrchu substrátu nebo se jedná o složitější technologie, které by mohli zvedat náročnost (a tím i cenu). Jak robustní je materiál, z kterého bude vyrobena.
- Velikost: podle umístění nás také zajímá, zda se anténa do daných prostor vejde fyzicky. Také aby nedošlo ke ztrátě kompaktnosti zařízení.
- Propojení antény s deskou: nutno také zvážit, zda je anténa přímo součástí desky plošných spojů zařízení, nebo je nutné spojit antény pomocí konektoru, který zvedá útlum. Zda není nutné, aby byla anténa nezávisle pohyblivá, pokud je zařízení pevně na jednom místě.

Většina těchto faktorů je zákonitě protichůdných a nikdy nebude dosaženo optimálních výsledků, jedná se tedy vždy o kompromisní řešení z hlediska stanovených nutných požadavků.

#### 3.2.1 Mikropásková flíčková anténa

V případě jednoduchosti provedení, výroby i relativně nízké ceny je vhodné zvolit mikropáskovou anténu. Vzhledem k tomu, že ani její návrh není složitý, je tato výchozí anténa dobrým referenčním bodem. Anténa byla navržena na substrát *CuClad 233* s relativní permitivitou  $\epsilon_r = 2,33$ , ztrátovým činitelem  $tg\delta = 0,0013$  při výšce substrátu  $h = 1,524$  mm. Anténa byla přizpůsobena vloženým vedením na  $50 \Omega$ . Požadovanou frekvenci stanovíme na  $f_r = 2,44$  GHz, aby bylo dosaženo středu pásma

Bluetooth. Zjednodušený návrh antény, který počítá se shodným prostředím substrátu a okolí, je možné provést pomocí vzorců uvedených v publikaci [6].



Obr. 3.1: Flíček s vloženým vedením, kde  $W$  je šířka flíčku,  $L$  délka flíčku,  $w_0$  šířka mikropáskového napájení a  $y_0$  hloubka zapuštěného vedení [7].

Nejprve proběhne výpočet šířky flíčku  $W$  odpovídající polovině délky vlny

$$W = \frac{c}{2f_r} \sqrt{\frac{2}{\varepsilon_r + 1}}, \quad (3.1)$$

zde  $\varepsilon_r$  je permitivitou substrátu,  $f_r$  požadovaná rezonanční frekvence a  $c$  rychlost světla. Ze vzorce vychází šířka flíčku  $W = 47,64$  mm. Následuje výpočet relativní efektivní permitivity substrátu pro danou šířku flíčku a výšku substrátu

$$\varepsilon_{reff} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2} \left[ 1 + \frac{12h}{W} \right]^{-1/2}, \quad (3.2)$$

zde  $h$  je výška substrátu. Dostáváme relativní efektivní permitivitu substrátu rovnu  $\varepsilon_{reff} = 2,23$ . Následný výpočet je proveden z důvodu rozšíření délky flíčku  $\Delta L$  na jeho hranách způsobený rozptylovými poli. Nyní

$$\frac{\Delta L}{h} = 0,412 \frac{(\varepsilon_r + 0,3) \left( \frac{W}{h} + 0,264 \right)}{(\varepsilon_r + 0,258) \left( \frac{W}{h} + 0,8 \right)}, \quad (3.3)$$

vyjádřením  $\Delta L$  obdržíme  $\Delta L = 0,79$  mm. Nyní je možné korigovat a zjistit tak skutečnou délku flíčku

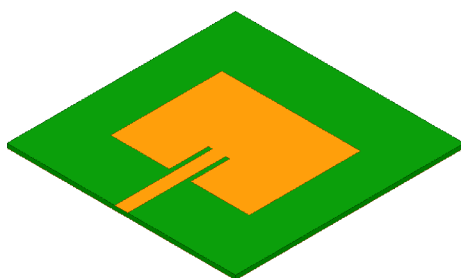
$$L = \frac{c}{2f_r \sqrt{\varepsilon_r}} - 2\Delta L, \quad (3.4)$$

po korekci je skutečná délka  $L = 39,58$  mm. Dále je třeba vyřešit nesymetrické mikropáskové vedení, které bude napájet flíček. Pro mikropáskové napájení bylo rozhodnuto z důvodu napájení SMA konektoru, aby bylo možné anténu případně zaměňovat za jinou. Pokud je třeba provést syntézu mikropáskového vedení, je třeba nejprve určit, která z následujících podmínek bude platit:  $\frac{w}{h} > 1$  nebo  $\frac{w}{h} \leq 1$ .

V tomto případě se jedná o první variantu. A pro syntézu bude použit následující vzorec:

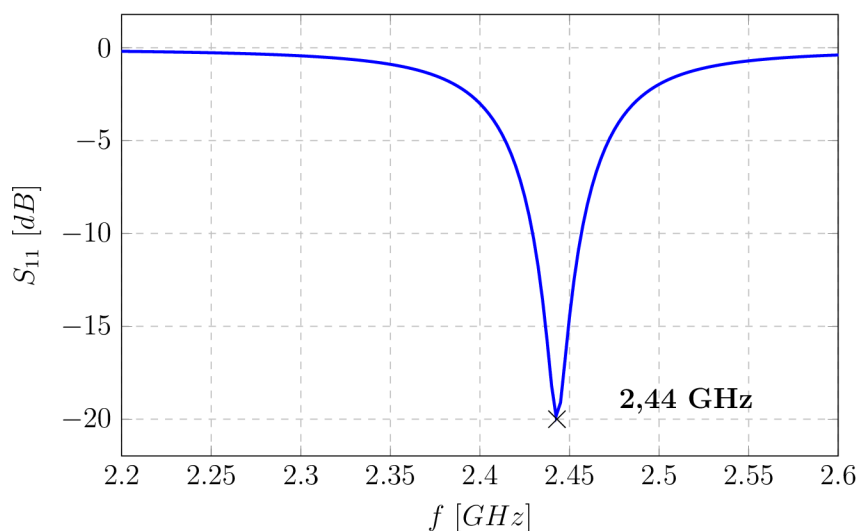
$$\frac{w}{h} = \frac{120\pi}{Z_0\sqrt{\epsilon_r}} - \frac{2}{\pi} - \left( \frac{2}{\pi} - \frac{\epsilon_r - 1}{3,7\epsilon_r} \right) \ln \left( \frac{120\pi^2}{Z_0\sqrt{\epsilon_r}} - 1 + 1,84 \frac{\epsilon_r - 1}{\epsilon_r} \right), \quad (3.5)$$

kde  $Z_0$  je charakteristická impedance, kde chceme dosáhnout  $50 \Omega$ . Při převodu výšky substrátu  $h$  na pravou stranu rovnice vychází výsledná šířka pásku  $w = 4,54$  mm. Parametr  $y_0$  (je vidět na 3.2) je hloubka zapuštění vedení, pomocí které je možné přizpůsobit vstupní impedanci flíčkové antény. Tento parametr bude nalezen empiricky z vycházející hodnoty  $y_0 = 10$  mm.

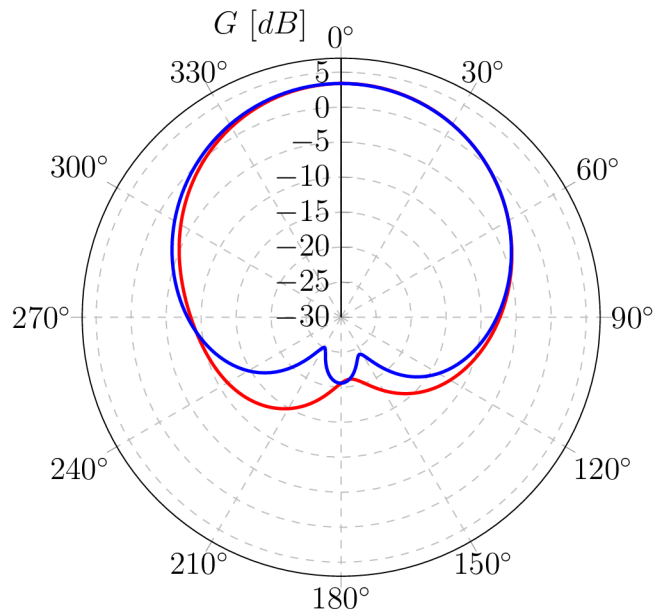


Obr. 3.2: Návrh flíčku s vloženým vedením, dle výše uvedených parametrů, v programu ANSYS HFSS.

Uvedené parametry byly následně rozmítány v jejich blízké oblasti, aby bylo dosaženo optimálních výsledků. A výsledky ze simulací antény jsou následující:



Obr. 3.3: Vstupní činitel odrazu ( $S_{11}$ ) flíčku s vloženým vedením.



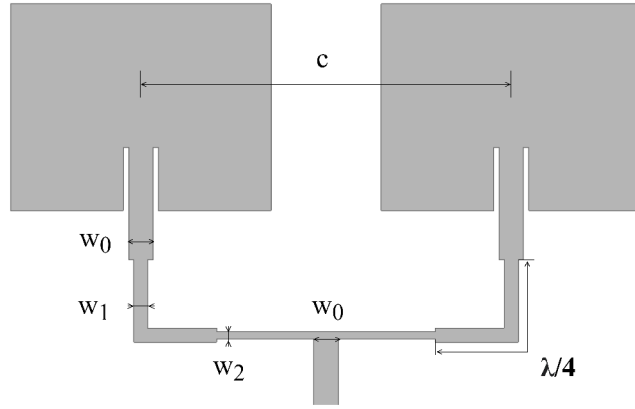
Obr. 3.4: Vyzařovací charakteristika realizovaného zisku flíčku v polárních souřadnicích. Odpovídá dvěma hlavním rovinám E (modře) a H (červeně).

Tab. 3.1: Srovnání parametrů vypočtených a dosažených simulací v HFSS.

Parametr		Výpočet	Simulace
Název	Zkratka		
Šířka napájecího mikropásku [mm]	$w_0$	4,54	4,53
Zapuštění mikropásku [mm]	$y_0$	10,00	13,70
Délka flíčku [mm]	$L$	39,58	38,90
Šířka flíčku [mm]	$W$	47,64	48,44
Délka substrátu [mm]	SubX	60	80
Šířka substrátu [mm]	SubY	60	80
Vstupní impedance [ $\Omega$ ]	$Z_0$	50,00	50,90

### 3.2.2 Mikropásková flíčková anténní řada

Vytvoření anténní řady zajistí větší směrovost, větší zisk a tvar vyzařovací charakteristiky. Z předchozího flíčku (zachování substrátu) byla vytvořena paralelně napájená anténní řada. Vzhledem k tomu, že parametry antén jsou vypočítány z rovnice 4.1 až 4.5, je pouze potřeba navrhnout vedení pro napájení antén.



Obr. 3.5: Paralelní flíčková řada s vloženým vedením, kde  $c$  vzdálenost středů flíčků,  $w_0$  šířka mikropáskového vedení o velikosti  $50 \Omega$ ,  $w_1$  šířka mikropáskového impedančního transformátoru o délce  $\frac{\lambda}{4}$  a  $w_2$  šířka mikropáskového vedení, které kvůli rozdělení na dvě větve dosahuje impedance  $100 \Omega$ .

Šířka mikropáskového vedení  $w_0$  (pro  $50 \Omega$ ) je již vypočítána. Nyní je třeba zjistit šířku  $w_2$ , ta vychází z toho, že pokud máme dva rozcházející se pásy, má každý z nich dvojnásobnou impedanci oproti pásku z kterého vycházejí. Díky této znalosti je možné vypočítat jejich šířku, kterou si vyjádříme z následující rovnice, neboť nyní se jedná o variantu  $\frac{w}{h} \leq 1$

$$\frac{h}{w} = \frac{1}{4} \left( \frac{e^H}{2} - e^{-H} \right), \quad (3.6)$$

kde v předcházející rovnici je parametr  $H$

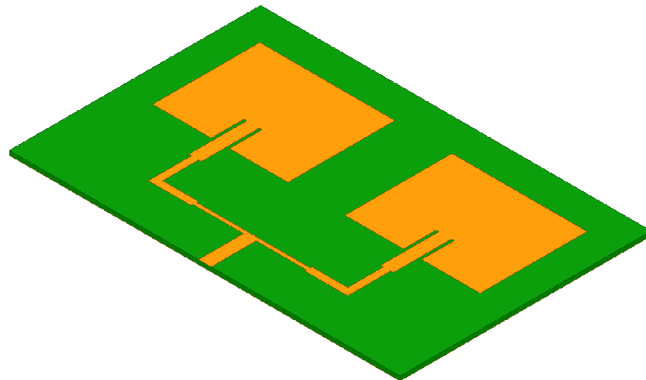
$$H = \sqrt{\frac{\varepsilon_r + 1}{2} \frac{Z_0}{60}} + \frac{0,9 \varepsilon_r - 1}{\pi \varepsilon_r + 1}, \quad (3.7)$$

a tedy  $w_2 = 1,294 \text{ mm}$ . Aby bylo možné vypočítat šířku mikropáskového vedení  $w_1$  je nutné znát impedanci mikropásku. V první řadě, aby bylo možné napojit vedení o šířce  $w_2$  na vedení o šířce  $w_0$  (při zachování impedančního přizpůsobení), provést vedení o šířce  $w_2$  jako impedanční transformátor. To je zaručeno tak, že délka tohoto vedení bude dána jako  $l = \frac{\lambda}{4}$ . Pokud toto platí je možné vypočítat impedanci mikropásku z následujícího vzorce

$$Z_{0T} = \sqrt{Z_k Z_{0V}}, \quad (3.8)$$

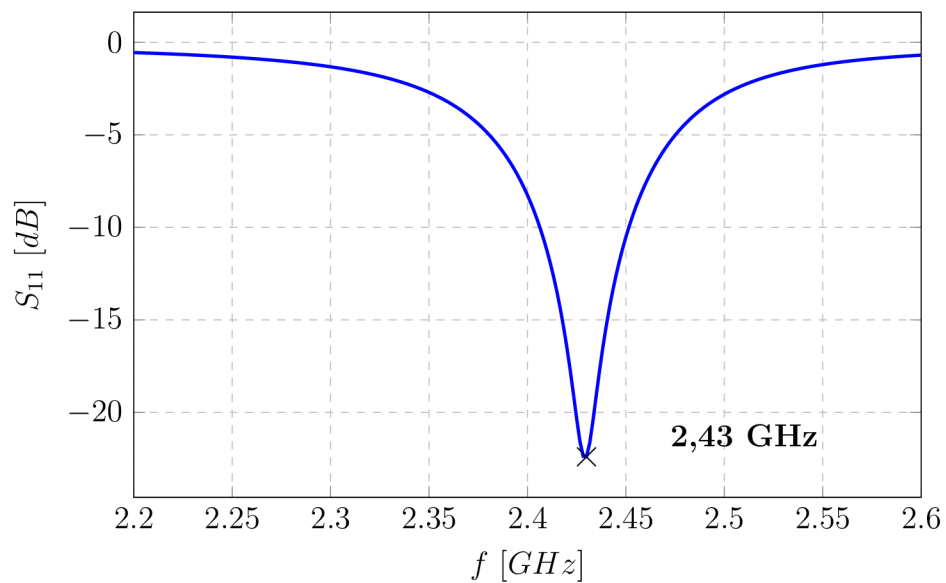
kde  $Z_{0T}$  je charakteristická impedance úseku vedení,  $Z_k$  je impedance zátěže a  $Z_{0V}$  je vstupní impedance. Obdržíme impedanci  $Z_{0T} = 70,7 \Omega$ . Jelikož lze očekávat, že mikropáskové vedení o šířce  $w_1$ , bude širší než vedení o šířce  $w_2$ , bude šířka

transformačního úseku vedení vypočítána pomocí vzorce 3.5 a šířka bude  $w_1 = 2,34$  mm. Parametr  $c$ , který udává vzdálenost středů flíčků od sebe, je doporučována jako  $c = \frac{\lambda}{2}$ .



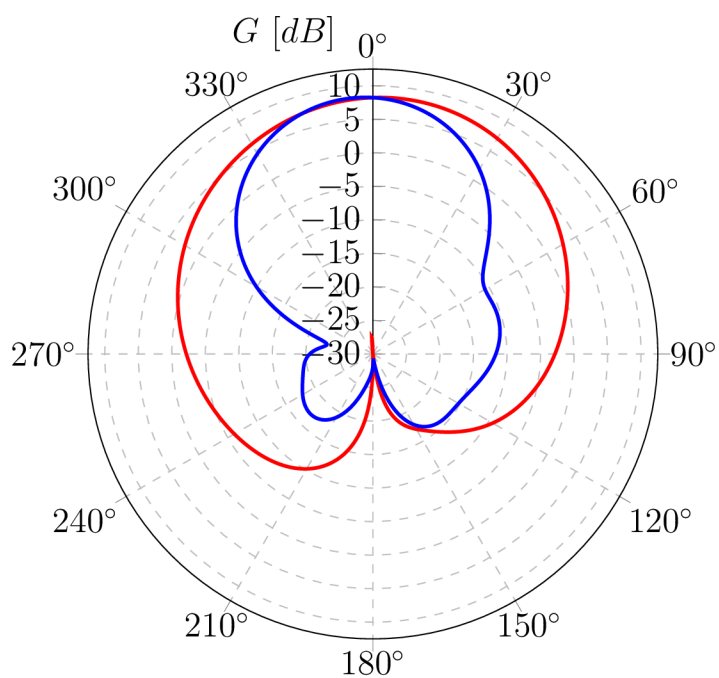
Obr. 3.6: Návrh mikropáskové flíčkové řady s vloženým vedením, dle výše uvedených parametrů, v programu ANSYS HFSS.

Uvedené parametry byly následně opět rozmítány (v případě nutnosti) v jejich blízké oblasti. Výsledky jsou následující:



Obr. 3.7: Vstupní činitel odrazu ( $S_{11}$ ) paralelní flíčkové řady.





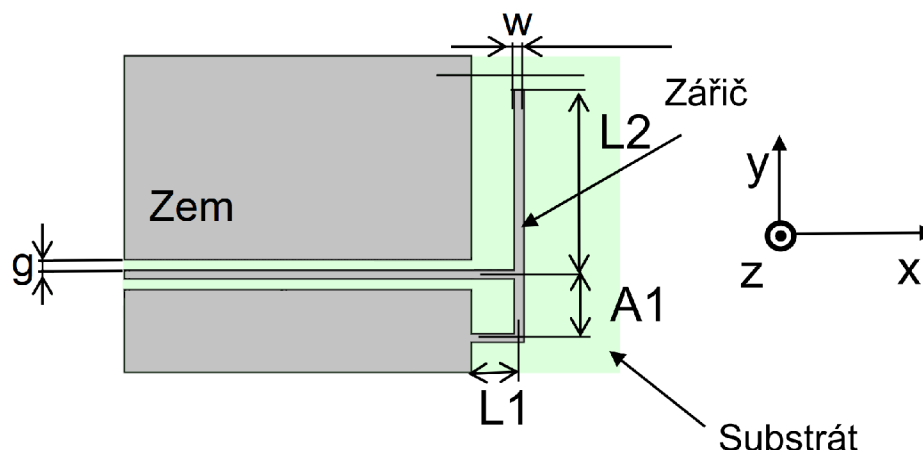
Obr. 3.8: Vyzařovací charakteristika realizovaného zisku paralelní flíčkové řady v polárních souřadnicích. Odpovídá dvěma hlavním rovinám E (modře) a H (červeně).

Tab. 3.2: Srovnání parametrů vypočtených a dosažených simulací v HFSS.

Parametr		Výpočet	Simulace
Název	Zkratka		
Šířka napájecího mikropásku [mm]	$w_0$	4,54	4,53
Šířka impedančního transformátoru [mm]	$w_1$	2,34	2,58
Šířka zdvojeného mikropásku [mm]	$w_2$	1,294	1,30
Zapuštění mikropásku [mm]	$y_0$	10,00	11,70
Vzdálenost středů flíčků [mm]	$c$	61,48	68,85
Délka flíčku [mm]	$L$	39,58	38,35
Šířka flíčku [mm]	$W$	47,64	48,44
Délka substrátu [mm]	SubX	100	90
Šířka substrátu [mm]	SubY	100	140
Vstupní impedance [ $\Omega$ ]	$Z_0$	50,00	51,20

### 3.2.3 Planární invertovaná F anténa

Předchozí flíčkové antény mají vyzařovací charakteristiky buďto v jedné polorovině (flíček) a nebo více směrové (flíčková řada). Planární invertovaná F anténa bude mít vyzařovací charakteristiku, která je více všesměrová (zároveň září do dvou směrů, bez zemní ploch), při zmenšení rozměrů antény, avšak při snížení zisku. Napájena bude opět mikropáskovým vedením v tomto případě se jedná o mikropáskový koplanární vlnovod.



Obr. 3.9: Planární invertovaná F anténa. Převzato z [8].

Parametry jsou stejné jako v předchozích návrzích, nyní však volíme substrát *FR4* o  $\epsilon_r = 4,4$ . Návrh bude proveden pomocí vzorců uvedených v [7], zde pro relativní efektivní permitivitu platí

$$\epsilon_{reff} < \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \frac{1}{\sqrt{1 + \frac{12h}{w}}}, \quad (3.9)$$

kde  $\epsilon_r$  je relativní permitivita substrátu,  $h$  je výška substrátu a  $w$  je šířka mikropáskového vedení, které tvoří planární invertovanou F anténu (dále jen PIFA), a  $\epsilon_{reff} = 3,34$ . Mikropáskové vedení bude zvoleno jako  $w = 2,91$  mm. Tuto hodnotu je dle publikace [8] možné zvolit ale vzhledem k tomu, že pokud je aplikován vzorec 3.5, dostáváme 2,91 mm, je tato hodnota adekvátní. Při vycházení z [7] jsou doporučené rozměry stanoveny pro  $L_1$

$$L_1 \approx 0,25 \frac{\lambda_0}{4\sqrt{\epsilon_{reff}}}, \quad (3.10)$$

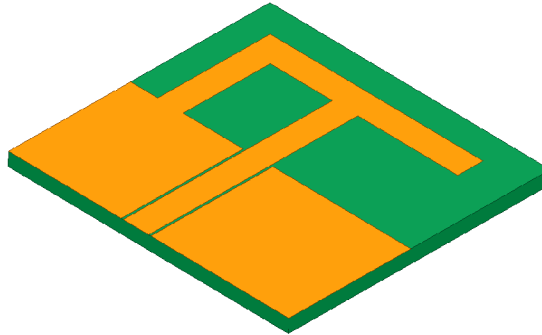
kde  $\lambda_0$  je vlnová délka pro danou frekvenci. Při aktuálních parametrech vychází  $L_1 = 4,21$  mm. Nyní je možné dopočítat také rozměr  $L_2$  vyjádřením z

$$(L_1 + L_2) \ll \approx \frac{\lambda_0}{4\sqrt{\epsilon_{reff}}}, \quad (3.11)$$

kde  $L_2 = 12,6$  mm .A rozměr  $A_1$  platí

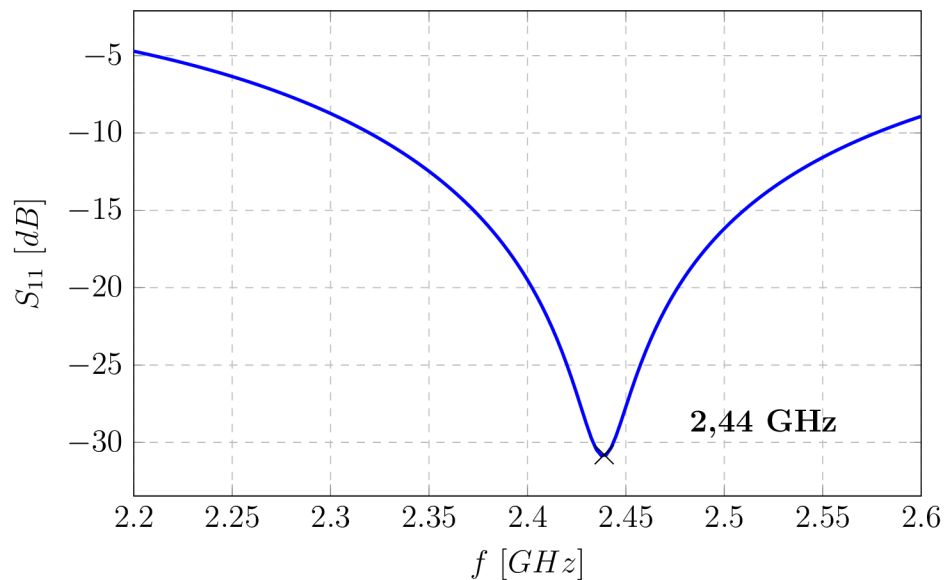
$$A_1 \ll \frac{\lambda_0}{\sqrt{\epsilon_{reff}}}, \quad (3.12)$$

zde pravá strana rovnice vychází 70 mm z tohoto tedy bude stanoveno  $A_1$  na 5 mm. Mezera v koplanárním vedení  $g$ , která je na obou stranách koplanárního vedení stejná, bude empiricky zvolena nejprve jako  $g = 0,5$  mm. Následně dochází k rozmítání parametrů k přizpůsobení antény.

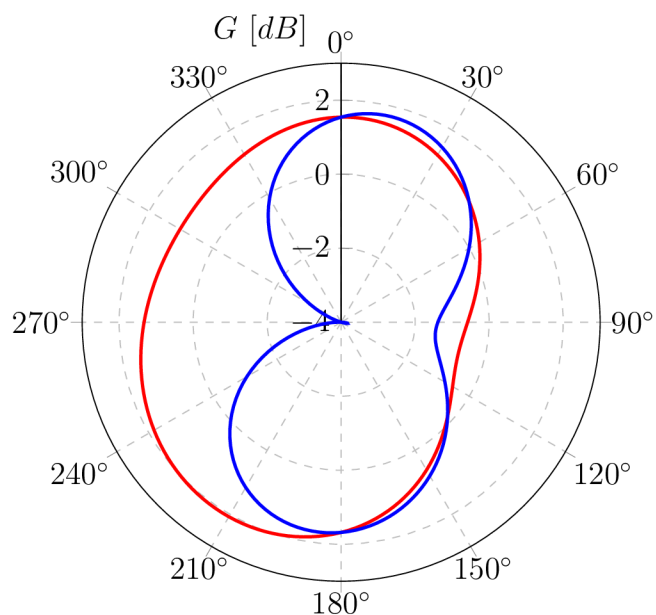


Obr. 3.10: Návrh planární inverotvané F antény napájené koplanárním mikropáskovým vlnovodem, dle výše uvedených parametrů, v programu ANSYS HFSS.

Z designu výše dostáváme následující výsledky



Obr. 3.11: Vstupní činitel odrazu ( $S_{11}$ ) PIFA.



Obr. 3.12: Vyzařovací charakteristika PIFA v polárních souřadnicích. Odpovídá dvěma hlavním rovinám E (modře) a H (červeně).

Tab. 3.3: Srovnání parametrů vypočtených a dosažených simulací v HFSS.

Parametr		Výpočet	Simulace
Název	Zkratka		
Délka hlavního zářiče [mm]	$L_2$	12,6	14,3
Vzdálenost zkrátového mikropásku [mm]	$A_1$	5,00	10,0
Vzdálenost zářiče od zemnicí plochy [mm]	$L_1$	4,21	10,0
Šířka mikropáskového vedení [mm]	$w$	2,91	2,90
Velikost mezery v koplanárním vedení [mm]	$g$	0,50	0,50
Délka zemnicí plochy [mm]	$GndX$	30	14
Délka substrátu [mm]	$SubX$	40	30
Šířka substrátu [mm]	$SubY$	40	32
Vstupní impedance [ $\Omega$ ]	$Z_0$	50,00	45,10

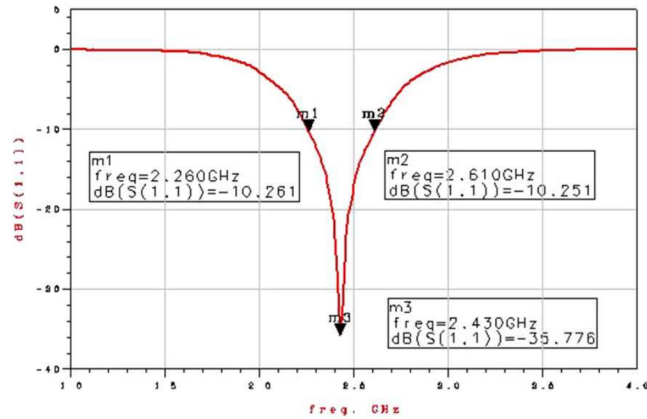
### 3.2.4 Čipová anténa

Další možností antény pro toto zařízení je čipová anténa. Hlavní výhodou je kompaktnost a povrchová montáž, která ušetří prostor desky. Základem těchto antén je dielektrikum o velké permitivitě ( $\epsilon_{reff} > 8$ ), ty se užívají kvůli koncentrování elektromagnetického pole do tohoto nízko ztrátového materiálu, a ten pak může být usazen na vysoce ztrátových deskách. Na základě doporučení z [14], bylo uvažováno několik druhů antén od firmy *Johanson Technology* ze série *2450AT*, kde byla nakonec vybrána *2450AT45A100E*, která již na povrchovou montáž dosahuje relativně větších rozměrů (9,5 x 2 mm), avšak vzhledem k tomu, že bude na desce zařízení uvažován také konektor UH.L, není tento rozměr problém. Anténa má dle výrobce následující parametry:

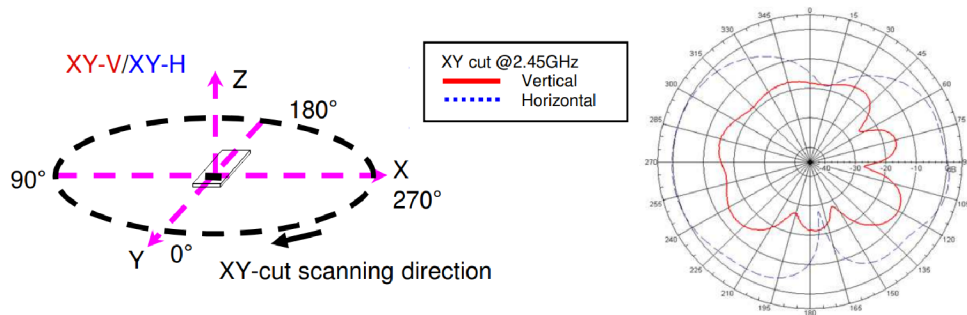
Tab. 3.4: Parametry čipové antény [9].

<b>Obecné parametry</b>	
Číslo součástky	2450AT45A100
Frekvenční rozsah [MHz]	2400 - 2500
Vstupní výkon [W]	3 max.
Vstupní impedance [ $\Omega$ ]	50
Operační teplota [ $^{\circ}\text{C}$ ]	-40 až +125
Délka, šířka a výška [mm]	9,5 x 2 x 1,2
<b>Maximální zisk na základě orientace antény</b>	
Horizontální montáž "A"- strana 5 [9]	1,5 dBi
Horizontální montáž "B"- strana 8 [9]	1,3 dBi

V tomto případě bude počítáno s horizontální montáží typu A, kde je nižší počet proměnných (velikostí napájecího vedení jak z hlediska šířky, délky a zahnutí) a tím pádem menší prostor pro chybný návrh i montáž. Pro tuto implementaci platí následující měření antény.



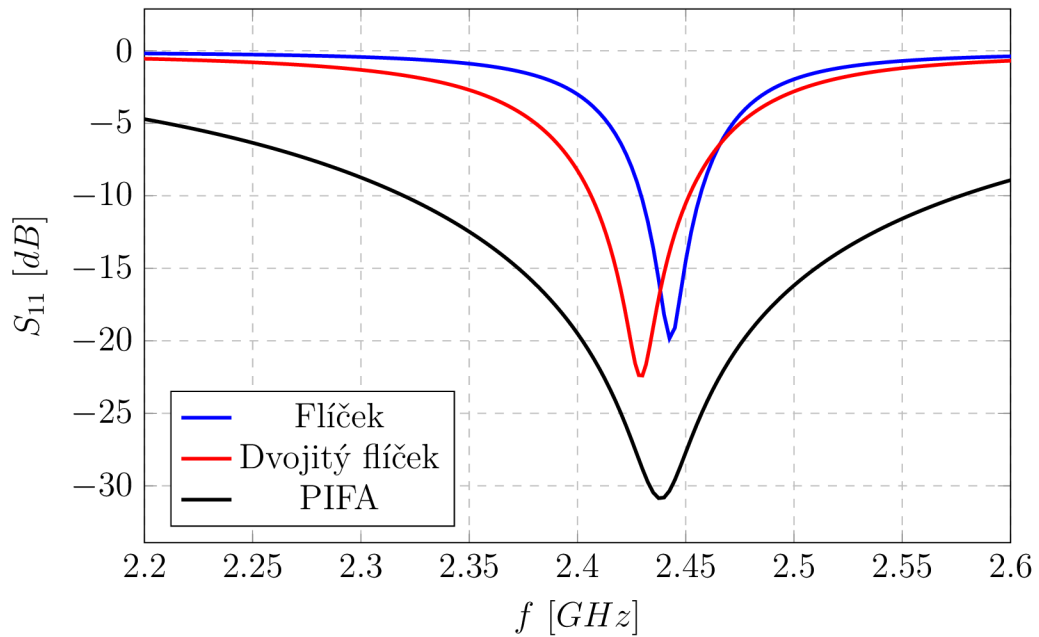
Obr. 3.13: Vstupní činitel odrazu ( $S_{11}$ ) čipové antény 2450AT45A100. Převzato z [9].



Obr. 3.14: Rovina záření (vlevo), vyzařovací charakteristika čipové antény 2450AT45A100. Převzato z [9].

### 3.3 Srovnání

Nyní je možné srovnat různé druhy navržených řešení antén pro toto zařízení. V následujícím grafu budou zobrazeny grafické průběhy činitele přizpůsobení pro jednodušší orientaci. Poté budou jednotlivé parametry vypsány v tabulce pro detailnější porovnání přizpůsobení, šířky pásma, zisku atp.



Obr. 3.15: Vstupní činitel odrazu ( $S_{11}$ ) pro navržené designy jednotlivých antén.

Tab. 3.5: Srovnání parametrů jednotlivých antén.

Parametr	Anténa			
	Flíček	Dvojitý Flíček	PIFA	Čipová
Rezonanční frekvence [GHz]	2,443	2,430	2,438	2,430
Vstupní činitel odrazu [dB]	-19,8	-22,4	-30,8	-36,8
Šířka pásma (-3 dB) [MHz]	8,90	10,8	22,1	350
Max. realizovaný zisk [dBi]	6,5	8,5	2,3	1,5
Rozměry substrátu (délka x šířka) [mm]	80 x 80	90 x 140	30 x 32	-
Směrovost [dB]	6,7 (max)	8,6 (max)	1,4 (mean)	-

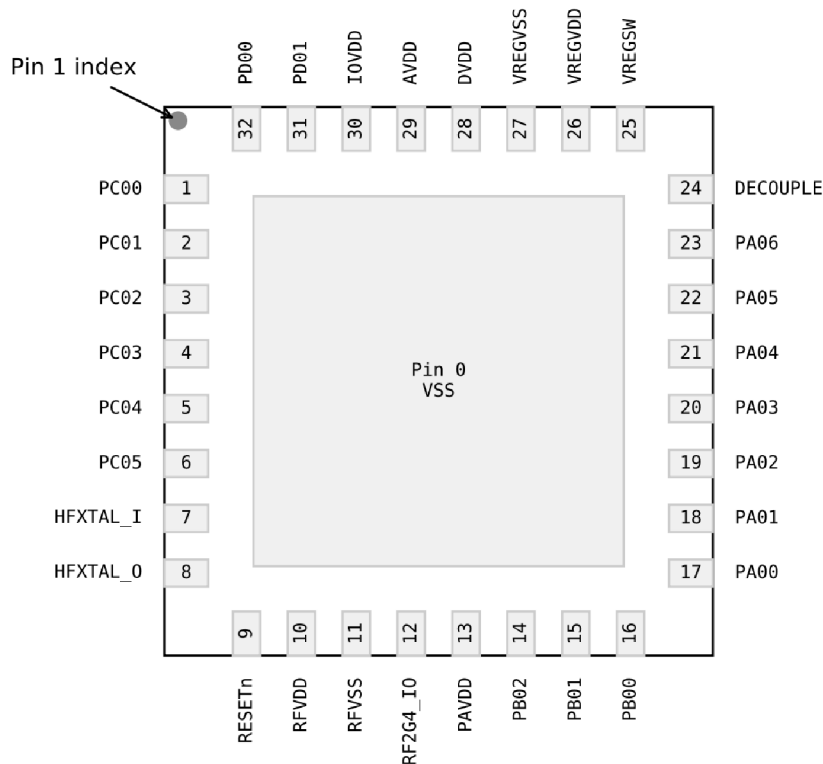
Na grafickém znázornění vstupního činitele odrazu vidíme 3 vytvořené antény, pro grafické porovnání můžeme srovnat 3.13 s 3.15. U všech antén je dosaženo přizpůsobení přibližně na frekvencích 2,43 až 2,44 GHz. Nejlepšího přizpůsobení dosahuje čipová anténa, která dosahuje při měření i hodnot až  $S_{11} = -37$  dB. Je viditelné, že v případě nižší hodnoty přizpůsobení zároveň roste šířka pásma na  $S_{11} = -10$  dB a tyto parametry jsou na sobě závislé. Zde naopak čipová anténa má šířku pásma až  $B = 350$  MHz. Realizovaného zisku je na samotném flíčku dosaženo  $G = 6,5$  dBi, při vytvoření dvojitého flíčku je realizovaný zisk  $G = 8,5$  dBi. Z toho vyplývá, že i

při dvojnásobném počtu stejné antény zisk nevzrostl razantně, avšak rozměry se ve směru šířky téměř zdvojnásobily. Je tedy otázkou, zda je výhodné použít dvojitý flíček. PIFA má při nejmenších rozměrech substrátu zisk  $G = 2,3$  dBi s relativně velkou šířkou pásma. Pro rozhodování volby antény je třeba vzít v potaz také vyzařovací charakteristiku (směrovost), tzn. jaký tvar bude vhodný pro jakou aplikaci.



## 4 Návrh zařízení

Ačkoliv má obvod vybraný v kapitole 2 velmi nízkou průměrnou spotřebu proudu, je nezbytné správné oddělení. Ve všech digitálních obvodech je proud odebírán v krátkých pulzech, které odpovídají hranám clocku. Přechodné proudové pulzy z napájení mohou mít řádově i několik stovek mA po několik nanosekund, ikdyž je definovaný průměrný odběr proudu nízký. Aby nebyl vnášen šum do napájecího napětí, je nutné použít oddělovací kondenzátory, které budou doplňovat proud během těchto jevů. Pro dosažení optimálních parametrů stanovených výrobcem, tedy budou brány v potaz doporučení při návrhu hardwaru [11].



Obr. 4.1: Pinout EFR32BG22 v provedení QFN32. Převzato z [10].

### 4.1 Napájení

Pin č. 28 (**DVDD**) je vstupem na vnitřní digitální LDO (Low-dropout regulator) a pin č. 24 (**DECOUPLE**) pak výstupem vnitřního LDO a zároveň digitální logický napájecí vstup. Pro potlačení přechodných jevů je v případě EFR32xG22 na DVDD doporučován kondenzátor  $2,7 \mu\text{F}$ . A zároveň pro lepší potlačení vysokofrekvenčního šumu bude přidán paralelně kondenzátor o hodnotě  $0,1 \mu\text{F}$ . Ačkoliv

je pin DECOUPLE připojen na výstup vnitřního LDO, je vyveden pouze za účelem oddělení pomocí kondenzátoru a není uvažováno žádné napájení tímto pinem. Doporučován je kondenzátor o hodnotě  $1 \mu\text{F}$ .

Pin č. 30 (**IOVDD**) je napájením GPIO. Zde je nutné, pokud je napájen z hlavního zdroje, pro správné oddělení připojit kombinaci paralelního  $1 \mu\text{F}$  a  $0,1 \mu\text{F}$ .

Dále pin č. 29 (**AVDD**) je napájením všech analogových periférií. V případě vylepšeného oddělení, které bude využito, je nutné připojit paralelně na vstup kondenzátory  $1 \mu\text{F}$  a  $1 \text{nF}$ . Jako další bude do série zapojen, pro dodatečné filtrování, rezistor hodnoty  $1 \Omega$  v sérii s feritovou perlou o impedanci  $1 \text{k}\Omega$  ( $100\text{MHz}$ ) s maximálním proudem  $200 \text{mA}$ .

Jako další jsou piny č. 10 (**RFVDD**) napájení analogového rádia a č. 14 (**PAVDD**) napájení rádiového výkonového zesilovače. V tomto případě se jedná o aplikaci na  $2,4 \text{GHz}$ , která nám na obou těchto pinech udává stanovenou kombinaci. Paralelní kondenzátory o hodnotách  $120 \text{pF}$  a  $100 \text{nF}$  a sériově zapojený feritovou perlu o stejné velikosti, jako bylo zmíněno v odstavci výše.

Jelikož nebude použit DC-DC převodník je potřeba ošetřit piny, které jsou mu přiřazeny. Pin č. 26 (**VREGSW**) je připojen k hlavního zdroji napájení přes kombinaci paralelních kondenzátorů  $10 \mu\text{F}$  a  $0,1 \mu\text{F}$ . Pin č. 25 (**VREGSW**) je nechán nezapojený a pin č. 27 (**VREGVSS**) je spojen se zemí.

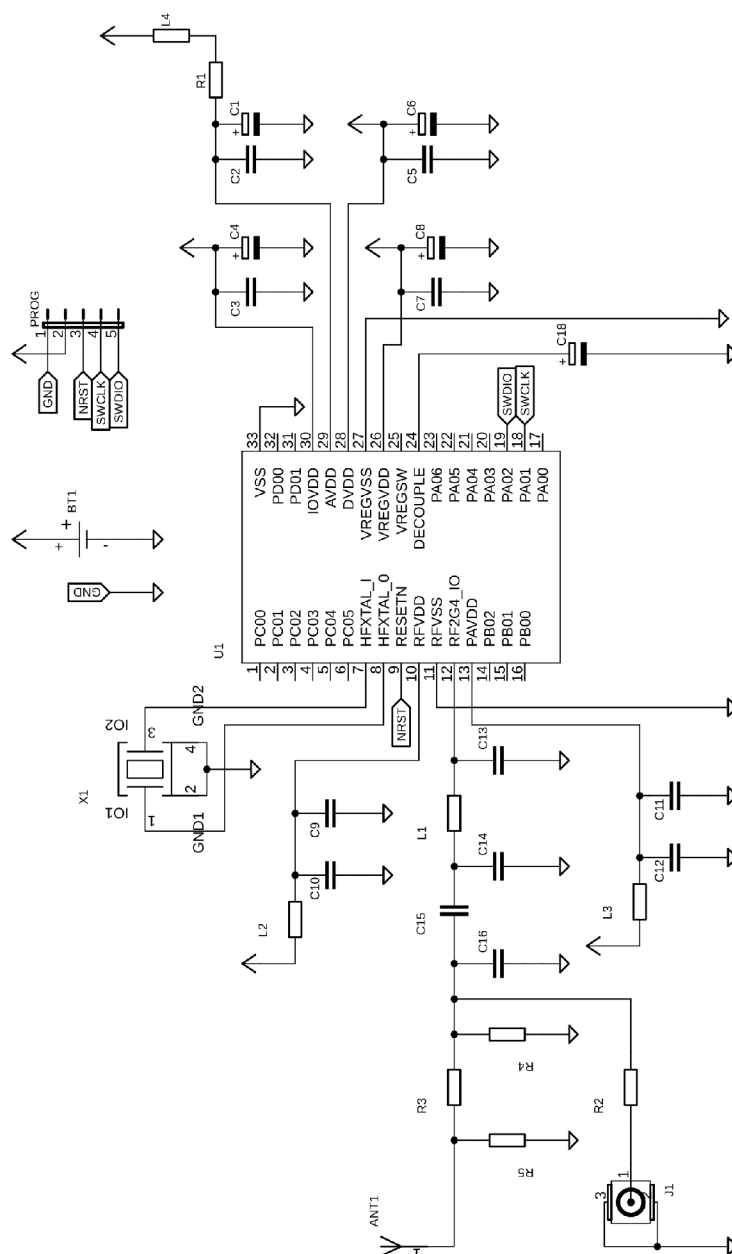
Pro možnost přesného externího clocku (v případě nutnosti) je na vstupy č. 7 (**HFXTAL\_I**) a 8 (**HFXTAL\_O**) připraveno připojení krystalu s frekvencí  $38,4 \text{MHz}$  při toleranci  $10 \text{ppm}$ , přičemž interní vysokofrekvenční oscilátor dosahuje  $500 \text{ppm}$ .

## 4.2 Připojení vf části

Aby bylo zařízení celkově více variabilní bude volen přístup, kdy bude možné připojit anténu přes U.FL konektor a následné napojení přes kabel na SMA konektor (díky čemuž se dosáhne i větší variability výběru místa případné antény). Avšak zároveň bude pro menší dosahy možnost, napájet čipovou anténu přímo na desku plošných spojů. Toto řešení umožní výběr pro danou specifikaci. Je tedy nutné vytvořit přizpůsobené vedení, které bude mít variabilitu napojení těchto dvou způsobů.

Pro přizpůsobení vedení z čipu (pin č. 12 **RF2G4\_IO**) bude využito doporučení z [13]. Doporučené přizpůsobení na  $50 \Omega$ , na které bude připojena anténa nebo konektor, je vytvořeno pomocí přizpůsobovacího vedení ( $\pi$ -článku). To je tvořeno dvěma paralelními kapacitami s hodnotami  $1,7 \text{pF}$  a  $1,3 \text{pF}$ , mezi kterými je v sériovém zapojení induktor o hodnotě  $2,7 \text{nH}$ . Na tento článek je napojen sériově kondenzátor o hodnotě  $18 \text{pF}$ , který je zde nezbytný pro blokování stejnosměrné složky.

Vedení je na několika místech doplněno sériovými i paralelními pájecími ploškami pro možnost případného doladění obvodu. Před rozvětvením pro připojení různých druhů antén jsou vloženy rezistory o hodnotách  $0 \Omega$  pro možnost rozhodnutí jaká anténa bude použita.



Obr. 4.2: Vytvořené schéma zařízení.

## 4.3 Deska plošných spojů

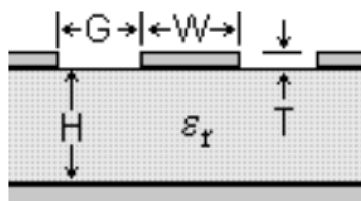
Deska plošných spojů byla navrhována na základě doporučení dokumentu [14]. Bylo dodržováno několik návrhových pravidel. Filtrační kapacitory u napájení byly usazovány co nejbližší pinům EFR32 a zároveň byl blízko ke kapacitorům (jejich uzemněného spoje) umístován prokov do spodní strany desky.

Napájecí vedení bylo umístováno tak, aby v ideálním případě bylo obklopené zemnicí plochou a aby nebylo vedeno po okrajích desky. V případě velkých rozměrů zemnicí plochy je vždy doplněn prostor větším počtem prokovů, v případě radiofrekvenční části desky je použití prokovů velmi vysoké. Okraje desky plošných spojů jsou pokryty tzv. 'stitching vias', kdy jsou umístěny prokovy s rozestupy nižšími než  $\frac{\lambda}{10}$  (kvůli nejvyšší harmonické - z [14]).

Přizpůsobovací koplanární vedení (pro 50  $\Omega$  vedení) má doporučené parametry Tab. 4.1 pro rozložení na Obr. 4.3 Z tabulky je patrné, že velikost impedance není

Tab. 4.1: Parametry koplanárního vedení. Převzato z [14].

Parametr	Hodnota
f [GHz]	2,4
T [mm]	0,018 - 0,035
$\varepsilon_R$ [-]	4,2 - 4,6
H [mm]	0,3
G [mm]	0,25 - 0,40
W [mm]	0,45



Obr. 4.3: Zobrazení parametrů koplanárního vedení. Převzato z [14].

příliš závislá na velikosti mezery koplanární vedení  $G$ , impedance se v daném rozmezí pohybuje mezi 47 až 53  $\Omega$ . V tomto případě byla zvolena mezera 0,40 mm, vzhledem k tomu, že navržené antény se pohybují přibližně okolo 51  $\Omega$  na jejich vstupu, a zároveň pro zbytek desky je odstup nastaven na 0,30 mm. Ostatní parametry jsou dány výběrem druhu desky plošných spojů, přičemž tloušťka měděné vrstvy bude 0,030 mm.

Pro přizpůsobovací článek složený (A.3) z  $C_{13}$ ,  $L_1$ ,  $C_{14}$  a  $C_{18}$  je nutné, aby byly jednotlivé komponenty umístěny co nejbližší u sebe a zemněné součástky měly u svého vývodu co nejbližší prokov na spodní zemní plochu. Jako další bylo zajištěno, aby okolo přizpůsobovacího vedení a konektoru U.FL, byly zemní plochy s co nejvíce prokvy, to je důležité především okolo samotného vedení.

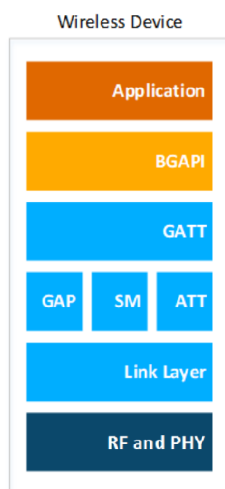
Motiv desky plošných spojů, osazovací plán a seznam součástek je obsažen v příloze.

## 5 Firmware zařízení

Firmware pro navržené zařízení byl vytvořen za pomoci *Simplicity Studio v5* a stojí základě jednoduchého počátečního projektu pro práci s Bluetooth.

### 5.1 Bluetooth stack - Silicon Labs

Bluetooth stack v3.x od firmy *Silicon Labs* je pokročilý Bluetooth protokol, který implementuje low energy standard páté verze bluetooth, která již byla popsána výše. Tato konkrétní verze v3.x je tvořena pro zařízení firmy *Silicon Labs* EFR32 a jejich moduly [15].



Obr. 5.1: Silicon Labs Bluetooth stack v3.x. Převzato z [15].

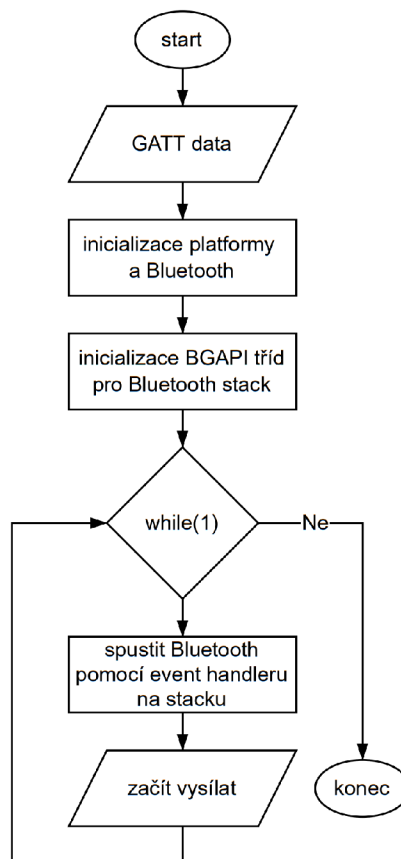
Níže budou stručně popsány vybrané části tohoto protokolu (převzato z [4]).

- **PHY (Physical Layer)** - Fyzická vrstva byla popsána v části 1.1.2. Přímou definujeme, jaký bude transport dat rádiovým prostředím, řídí rádiový příjem nebo vysílání.
- **Link Layer** - Linková vrstva přímo navazuje na fyzickou a určuje první stupeň kontroly nad daty a jejich strukturou. Předně definuje data a formáty paketů advertisement, operace na linkové vrstvě, připojování a časování paketů. Z hlediska operací na linkové vrstvě jsou základní důležité operace: advertising, skenování a zajištění připojení.
- **GAP (Generic Access Profile)** - zjednodušeně se umožňuje přístup k jednotlivým operacím linkové vrstvy. Definuje role jednotlivých zařízení, tyto přiřazené role, pak umožňují zařízením buďto vysílat, přijímat a nebo obojí.

- **ATT (Attribute Protocol)** - definuje komunikaci mezi dvěma zařízeními a určuje zařízením dvě možné role: *server* - zařízení ukládá data jako svoje atributy a *klient* - sbírá informace ze serverů. V případě navázaného připojení/spojení zařízení, poskytuje procedury ke čtení, zapsání, indikaci a oznamování atribut přes toto spojení.
- **GATT (Generic Attribute Profile)** - postaven na ATT, z kterého umožňuje shlukovat jednotlivé atributy do skupin a vytváří rámec pro transport dat. Dále také přináší informace o atributách, jak k nim přistupovat a jaká úroveň zabezpečení je nezbytná.
- **BGAPI (BG Application Programming Interface)** - tento protokol umožňuje pomocí vlastním příkazů komunikaci od uživatele na celý Bluetooth stack a přenáší odpovědi a události z Bluetooth stacku zpátky k uživateli.

## 5.2 Funkce programu

Následující vývojový diagram kódu ukazuje přibližnou funkčnost programu.



Obr. 5.2: Vývojový diagram programu pro zařízení na základě čipu EFR32BG22.

Samotný program se spustí ihned po nabootování zařízení. Nejprve dojde k mnohačetným inicializacím a následnému načtení GATT dat, která jsou díky Bluetooth GATT Konfiguratoru v rámci *Simplicity Studio 5*, jednoduše upravitelná. Nastaveno je systémové ID zařízení (číslo samotného čipu), název, výrobce, obecně tedy informace o zařízení.

Dále funkce *sl\_system\_init()* zapříčiní nejprve inicializaci platformy - čipu, přerušení, handleru událostí apod. A také inicializaci Bluetooth - zde se nejprve inicializuje, dle nastavené konfigurace, Bluetooth stack podle požadovaných parametrů.

Následně také inicializujeme BGAPI třídy na základě GATT dat, to znamená, data (rámce) jsou vytvořena do podoby advertising, aby mohla být vysílána a následně zachycena a zkoumána.

Po dokončení jednotlivých inicializací vstupuje program do opakující-se smyčky *while(1)*, kde je neustále volána funkce *sl\_system\_process\_action*, která vytvoří událost, která se bude ve smyčce opakovat.

Kód je uveden v příloze B. Můžeme vidět, že hlavní strukturou je *main.c* a *app.c*. Dále vidíme, že jsou nejprve převzata data z události, je extrahováno ID zařízení, vytvořen advertising set, který je následně vyslán v definovaných periodických intervalech, protože je neustále volán event handler.

Zařízení je nastavené jako obecně zjištěitelné a skenovatelné (je tedy možné ho detekovat).



## 5.3 Optimalizace spotřeby baterie

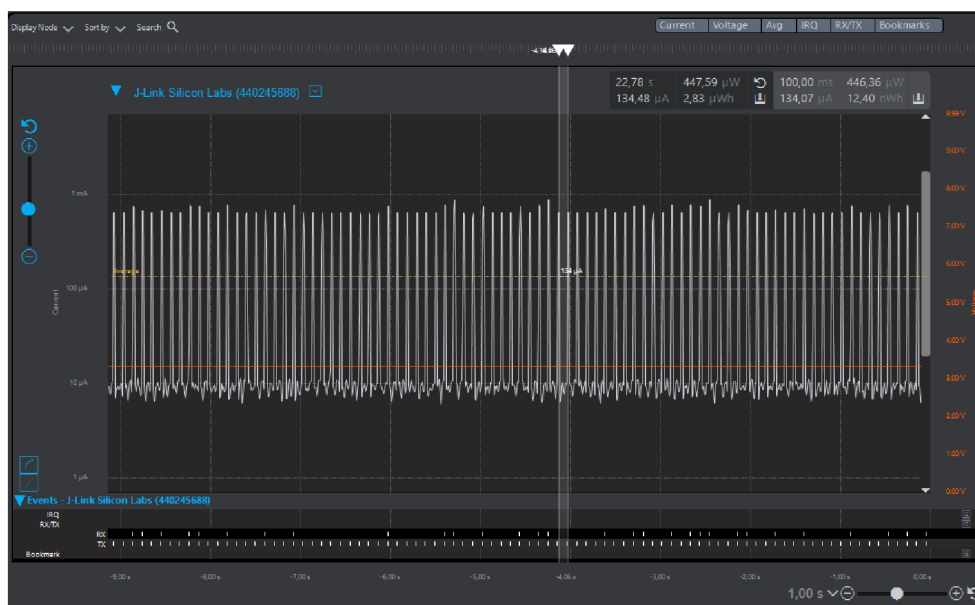
Vzhledem k tomu, že v případě tohoto zařízení záleží na spotřebě energie (i vzhledem k bateriovému napájení), je potřeba zajistit, co možná nejnižší spotřebu.

Jak již bylo zmíněno v 1.1.2, ze 40 kanálů v Bluetooth Low-Energy, jsou 3 kanály (hlavní advertising kanály 37, 38 a 39) vyhrazeny pro vysílání advertising paketů, které obsahují informace o zařízení, které je vysílá. Ve výchozím módu, dochází k vysílání na všech těchto kanálech. Avšak pokud bude nastaveno vysílání pouze na jeden z těchto kanálů, je možné dosáhnout snížení spotřeby. V tomto případě byl vybrán kanál 38 z důvodu umístění blíže středu rozsahu frekvence kanálů 1.1.2.

Ve výchozím nastavení dochází k vysílání advertising paketů každých 100 ms, což je velmi frekventovaný interval, proto je vhodné ho zvětšit s přihlédnutím na spotřebu energie (viz kapitoly níže). V B.3 na řádce 60 až 67 jsou prováděny výše popsané částečné optimalizace.

### 5.3.1 Spotřeba energie

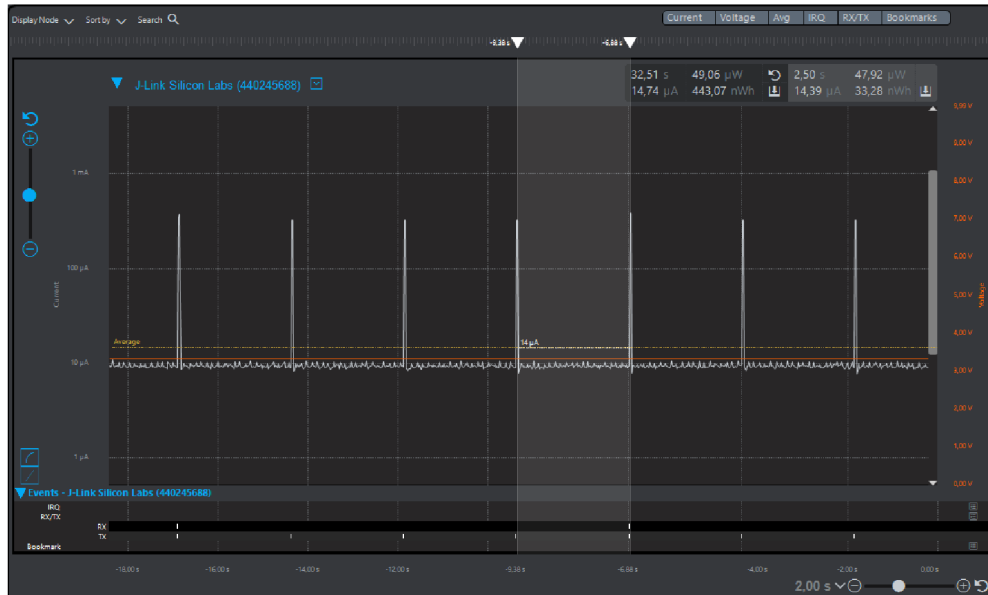
Následující měření jsou provedena díky hlavní vývojové desce *Silicon Labs*, která umožňuje *power consumption measurement* (měření odběru proudu). V případě výchozího nastavení, tedy periodickém vysílání 100 ms při všech kanálech, je viditelné, že odebíraný proud se pohybuje přibližně okolo 140  $\mu\text{A}$



Obr. 5.3: Výchozí spotřeba energie zařízení.

### 5.3.2 Spotřeba energie po optimalizaci

Při změně na jeden vysílací kanál a zároveň nastavení periodického opakování vysílání na 2,5 s je odebíraný proud výrazně snížen na průměrnou hodnotu 14  $\mu\text{A}$ .

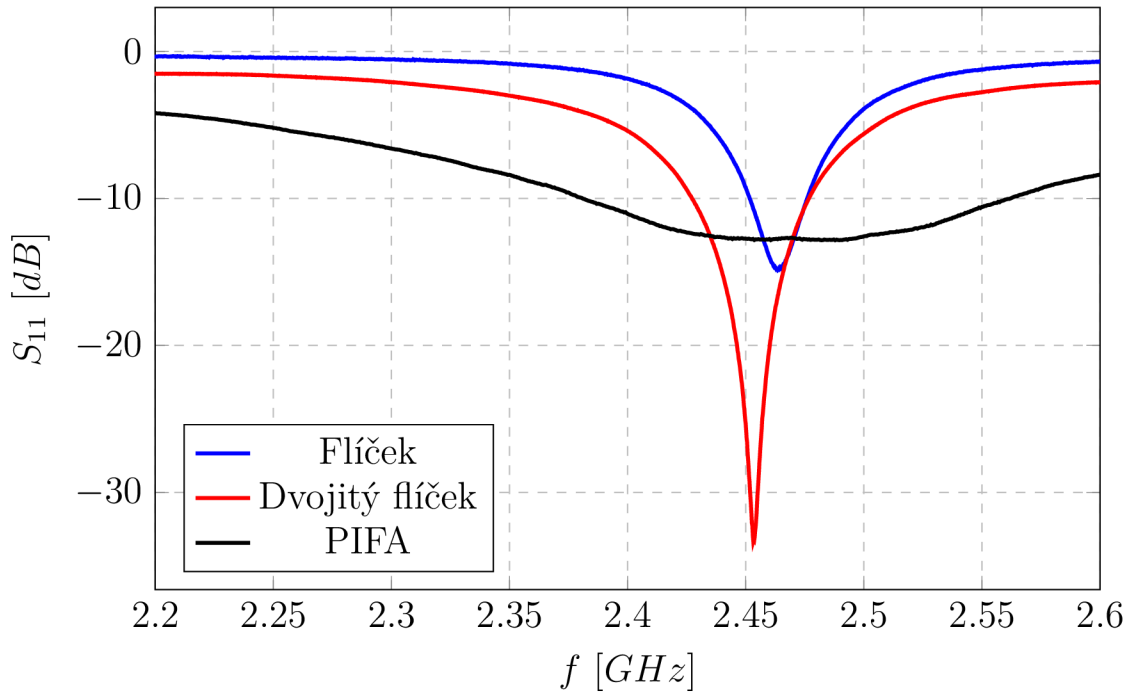


Obr. 5.4: Spotřeba energie zařízení po úpravě.

## 6 Vyrobené antény

### 6.1 Srovnání simulace a měření

V grafu a tabulce níže je viditelné, jaké je přizpůsobení vyrobených antén v požadovaném intervalu. Pro jednotlivé antény bylo vybráno vždy měření, které odpovídá přibližně střední hodnotě ze sady provedených měření, aby nebyly brány v potaz hodnoty, které můžou reprezentovat pouze výkyvy v měření.



Obr. 6.1: Změřený vstupní činitel odrazu ( $S_{11}$ ) navržených designů jednotlivých antén.

Tab. 6.1: Srovnání parametrů simulovaných a měřených antén.

Parametr	Anténa					
	Flíček		Dvojitý flíček		PIFA	
	Simulace	Měření	Simulace	Měření	Simulace	Měření
Rezonanční frekvence [GHz]	2,443	2,464	2,430	2,453	2,438	2,465
Vstupní činitel odrazu [dB]	-19,8	-14,9	-22,4	-33,3	-30,8	-12,8
Šířka pásma (-3 dB) [MHz]	8,90	16,4	10,8	2,91	22,1	186

V případě jednoduchého flíčku je rezonanční frekvence posunuta oproti simulované verzi přibližně o 20 MHz směrem k vyšší frekvenci. Přičemž velikost vstupního činitele odrazu se zvětšila zhruba o 5 dB, při dvojnásobné šířce pásma, než bylo dosaženo v simulaci. Z hlediska výsledků návrhu se jedná o relativně dobré výsledky, které zůstávají přibližně na středu rozsahu frekvencí Bluetooth.

Pro dvojitý flíček se podařilo dosáhnout rezonanční frekvence na téměř ideální hodnotě 2,45 GHz, při vstupním činiteli odrazu nižším než v případě simulace. Šířka pásma je velmi úzká, avšak toto částečně zavádějící vzhledem k velmi velké ostrosti charakteristiky, která i přes vícenásobná měření, nebude pravděpodobně stálá. S ohledem na to, že v reálných podmínkách lze očekávat větší velikost činitele odrazu, při větší šířce pásma, jsou výsledky velmi uspokojivé.

Naopak je tomu z hlediska planární invertované F antény. V tomto případě se jedná již o třetí vyrobenou verzi antény, kde až v posledním případě bylo alespoň na rozsahu 2,4 až 2,5 GHz dosaženo činitele odrazu menšího než -10 dB. V tomto případě se jedná o přibližně -12,8 dB na rezonanční frekvenci 2,44 až 2,47 GHz (dáno rozsahem, kvůli ploché povaze charakteristiky v této části). Šířka pásma této antény je velká, ale i vzhledem k simulovanému výsledku, bylo očekávané, že anténa bude mít velmi velkou šířku pásma, vzhledem k velmi pozvolnému růstu, resp. poklesu charakteristiky. Vzhledem k velmi malým rozměrům antény, odchylce při výrobě a také velké citlivosti na malé změny v rozměrech bude tento výsledek považován za dostatečný a s anténou bude nadále počítáno.

## 7 Aplikace pro telefon

Aplikace byla vytvořena za pomoci vývojového prostředí *Android Studio Bumblebee*, které umožňuje vývoj aplikací pro mobilní zařízení s operačním systémem Android ve velké šíři jejich verzí. Aplikace stojí na základě volně dostupného softwaru - aplikace pro simulování Bluetooth zařízení ([16]). Tato aplikace byla použita jako počáteční struktura pro vytvořenou aplikaci, především z důvodu práce s balíčky na platformě Android: *android.bluetooth* a *android.bluetooth.le*. Tyto balíčky ([22]) obsahují funkce pro práci s Bluetooth a jejich implementace výrazně zvýší šance na vyvarování se chybám při jejich aplikaci.

Při používání balíčku *android.bluetooth.le*, je možné při zapnutém Bluetooth na mobilním zařízení přijímat signály okolních dostupných zařízení. Z těch je možné získat, funkcí jako je *ScanResult*, jméno zařízení, vysílané ID zařízení a přijatou sílu signálu, spolu s dalšími parametry.

### 7.1 Funkce programu

Vývojový diagram na další straně přibližně popisuje funkci aplikace vytvořené v prostředí Android Studio.

Po zapnutí aplikace nejprve dojde k dotazu na zapnutí Bluetooth v telefonu. Pokud tak uživatel neučiní, není možné pokračovat dále a aplikace se ukončí.

Jako další nastává inicializace prostředí, tedy vytvoření celého kanvasu pro možnost vložení periférií, hlavní lišty, tlačítek, designů tlačítek, vytvoření grafického zobrazení beaconů apod. Zároveň se také vytváří proměnné pro uložení přijatých dat (názvů, ID zařízení, RSSI (Received Signal Strength Indication)), ale i vnitřní proměnné, které jsou užity v rámci programu.

Následuje vytvoření fragmentu, který se jmenuje *FragmentScanner*, ten obsahuje veškeré funkce pro práci s přijatými daty, jejich zpracování, vytvoření polí, do kterých budou uloženy, extrakce ID zařízení a jiných proměnných. Zároveň také obsluhuje zobrazení vyhledaných beaconů.

Po celkové inicializaci se program dostává do "hlavní smyčky". Nejedná se o smyčku jako takovou, aplikace čeká na stisknutí tlačítek od uživatele. V případě stisku tlačítka lokalizace (7.1 dole uprostřed) před vlastním skenováním nemůže dojít k postupu a aplikace zobrazí hlášku o neexistujících beaconech (přesněji "Not enough active beacons. None are active"). Pokud však dojde ke stisknutí tlačítka vyhledávání, spustí se vyhledávání Bluetooth zařízení v okolí.

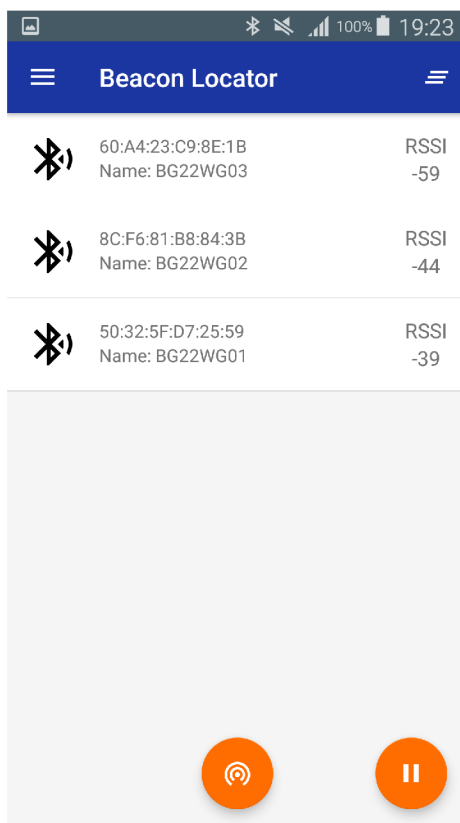
Ihned s tímto povelům se zobrazí zařízení v okolí, která jsou již selektována tak, aby byly zobrazovány (i ukládány do polí), jen požadované beacony s názvy *BG22*.

Následně se pak zobrazí ve vlastím řádku s ID zařízení, názvem a velikostí RSSI, tak jak je vidět na 7.1.

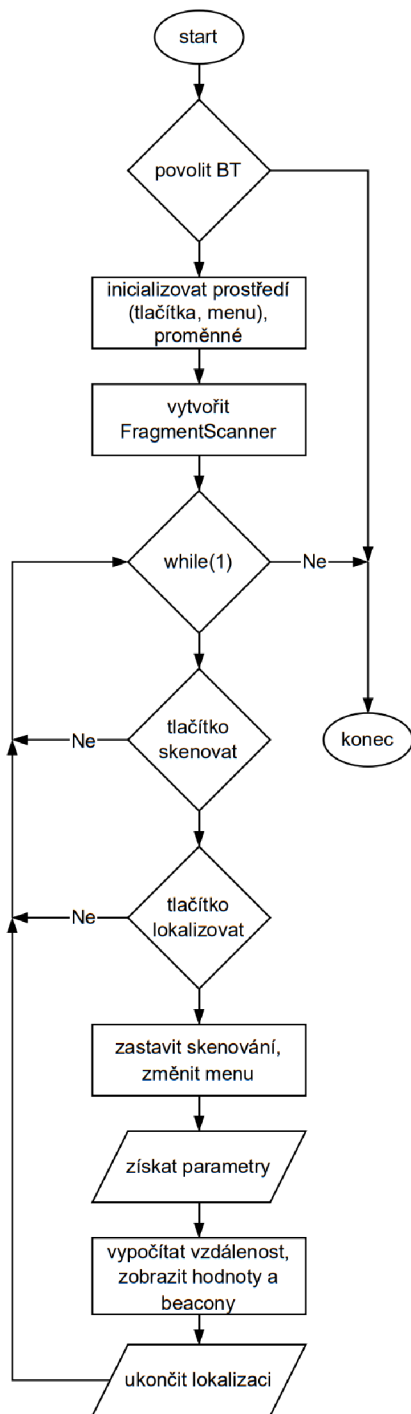
Nyní už je možné stisknout tlačítko lokalizace. Avšak to také funguje pouze v případě dostatečného počtu lokalizačních beaconů, tedy tří. V případě nižšího je zobrazena hláška o nedostatečném počtu (přesněji "Not enough active beacons. Only  $x/3$ ", kde  $x$  reprezentuje aktuální počet). V opačném případě však nejprve nastává zastavení skenování, změna rozhraní menu (zobrazení jednotlivých beaconů a boxů pro zápis vypočtených a přečtených dat).

Dojde k získání hodnoty RSSI z FragmentScanneru a názvů beaconů, kterým patří. Následuje výpočet vzdálenosti (bude podrobněji popsán níže), která je následně převedena na požadovanou pixelovou hustotu zařízení. Tato hodnota je pak aplikována na jednotlivé beacony a zobrazí se finální design lokalizace.

Pak už aplikace čeká na ukončení lokalizace, který znamená návrat do výchozího stavu menu i získaných dat a čeká se opět na spuštění vyhledávání beaconů.



Obr. 7.1: Ukázka aplikace ve fázi hledání aktivních beaconů. V tomto případě zobrazeny tři nalezené beacony s jejich ID, názvem a velikostí RSSI (Received Signal Strength Indication).



Obr. 7.2: Vývojový diagram aplikace pro telefon.

## 7.2 Provedení lokalizace

Již bylo zmíněno, že pro lokalizaci bude použit RSSI neboli Received Signal Strength Indicator. Jedná se o odhadovanou míru úrovně výkonu, kterou přijímá dané zařízení od okolních zařízení (v tomto případě Bluetooth zařízení).

Hodně technik (Wi-Fi triaterace, Bluetooth LE beaconing a jiné - vycházeno z [18]) pro lokalizaci používá RSSI, jako hlavní parametr pro získání vzdálenosti s akceptovatelnou přesností. Jiné druhy metod jsou přesnější, ale znamenají vyšší náklady a komplexnost.

Avšak hlavní nevýhodou při RSSI lokalizaci jsou především menší vnitřní prostory, kde může být signál lehce ovlivněn. Především pak náchylností k mnohacestnému šíření a stínění zapříčiněných objekty (nábytek, dveře apod.) v prostoru. Ve vnitřních prostorách se tedy jedná o velmi hrubé odhady vzdálenosti, kde zvětšování vzdálenosti mezi přijímačem a vysílačem nemusí nutně znamenat snižování velikosti signálu ([18] a [19]).

V tomto případě se tedy bude jednat především o venkovní aplikace z důvodu zvýšení přesnosti měření vzdálenosti.

Byl převzat vzorec č. 1 z [18] pro výpočet odhadu vzdálenosti. Ten je následující:

$$RSSI = -(10 \cdot n \cdot \log(d) - A). \quad (7.1)$$

Zde  $A$  je měřený výkon, který je očekávaný ve vzdálenosti do 1 m od Bluetooth LE beaconu, jenž se dle výrobce rovná 6 dBm pro EFR32BG22. Dále  $d$  je vzdálenost zařízení v metrech a  $n$  bezrozměrná konstanta prostření (ve zdrojích [18], [20] a [21] je uváděna v rozsahu 2-4, přičemž pro volný prostor je zvolena ve velikosti 2). Velikost konstanty bude entropicky nalezena a upravována po měření, aby bylo dosaženo co nejpřesnějších výsledků.

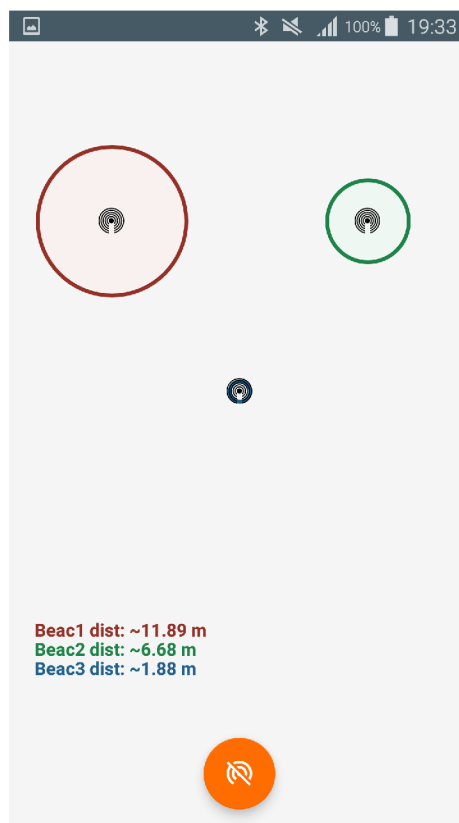
Aby bylo dosaženo výpočtu vzdálenosti je nutné vzorec upravit na následující tvar:

$$d = 10^{\frac{A - RSSI}{10 \cdot n}}. \quad (7.2)$$



Poté, co je tímto způsobem vypočtena vzdálenost, je převedena na pixelovou hustotu vybraného android zařízení, kterou jsou následně nastaveny velikosti kruhů symbolizujících jednotlivé beacons. Přičemž platí, že čím je velikost kruhu větší, tím se nachází zařízení, vysílající svůj Bluetooth signál, blíže. Na obrazovce zařízení je poté možné se částečně zorientovat, kde se zařízení v rámci prostoru nachází.

V tomto případě by se tedy mohlo jednat o uživatele, který drží mobilní zařízení v ruce a po otevření aplikace může nalézt svou přibližnou v rámci prostoru beaconů. Takto je možné se přibližně realizovat měření na ploše zhruba 25 x 25 m (takto je zároveň velký canvas zobrazený na displeji telefonu - viz obrázek níže). Tato velikost byla nastavena zpětně, až pomocí měření s anténami.



Obr. 7.3: Screenshot aplikace ve fázi lokalizování mobilního zařízení v rámci prostoru. Jsou zobrazeny jednotlivé beacons, přibližná vzdálenost od nich a velikost RSSI.

## 8 Lokalizace - měření

Všechna následující měření byla provedena ve volném prostoru na otevřeném prostranství. Nejprve bude zobrazeno, jak přesně je možné změřit vzdálenost při použití jednoho beaconu - při použití mobilní telefonu a jednotlivých typů antén. Přičemž měření s každým typem připojené antény bude provedeno tak, aby měla anténa svou hlavní osou směrem na přijímač. Zároveň bude provedeno vícenásobné měření, aby bylo možné aproximovat změřené hodnoty.

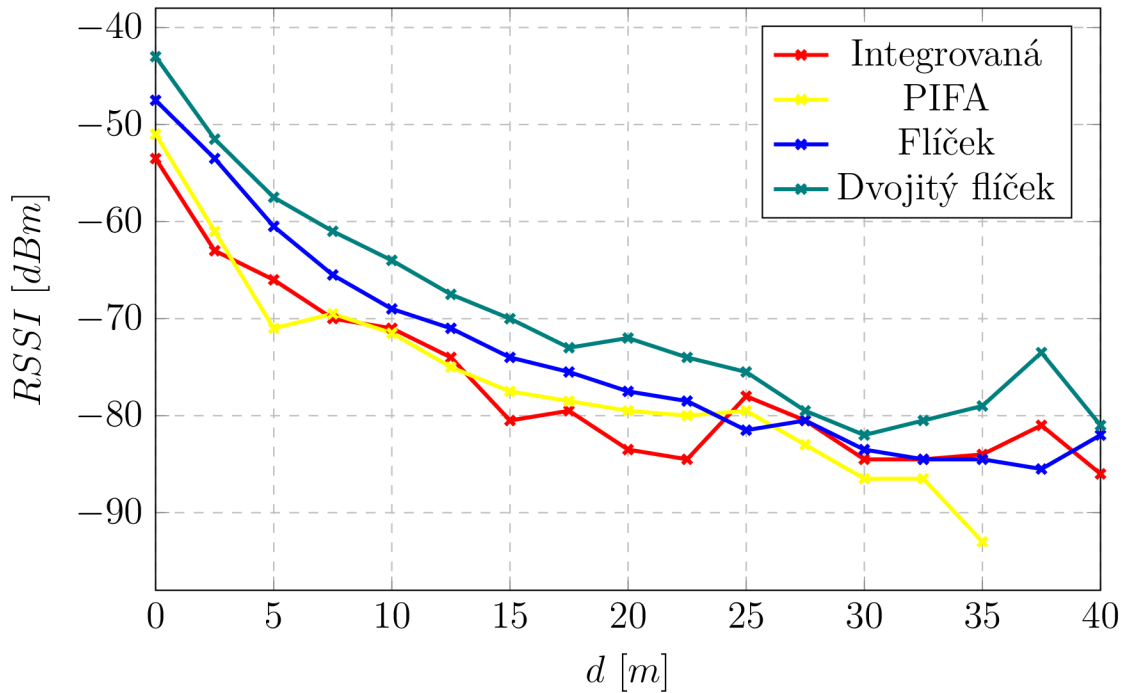
### 8.1 Měření vzdálenosti

V případě měření vzdálenosti byl beacon nejprve umístěn (nyní už napájený z knoflíkové baterie) do výšky 1,9 metrů od země tak, aby plocha desky byla kolmá se zemí. Beacon byl nastaven tak, aby mířil na vytyčenou vzdálenost 40 metrů. Původně byla vzdálenost zvolena empiricky s předpokladem, že bude v průběhu měření zvětšena. Tento předpoklad byl v průběhu měření vyvrácen a vzdálenost byla vyhodnocena jako dostačující. Zmíněná vzdálenost byla nakótována po 2,5 metrových úsecích.

Změřené hodnoty RSSI pro jednotlivé antény jsou zaznamenány v následující tabulce. RSSI bylo odečítáno za pomoci vytvořené aplikace pro telefon.

Tab. 8.1: Změřené hodnoty RSSI pro jednotlivé antény na zvětšující se vzdálenosti.

d [m]	Antény - RSSI [dBm]			
	Integrovaná	PIFA	Flíček	Dvojitý Flíček
0	-53,5	-51	-47,5	-43
2,5	-63	-61	-53,5	-51,5
5,0	-66	-71	-60,5	-57,5
7,5	-70	-69,5	-65,5	-61
10,0	-74	-71,5	-69	-64
12,5	-80,5	-75	-71	-67,5
15,0	-79,5	-77,5	-74	-70
17,5	-79,5	-78,5	-75,5	-73
20,0	-83,5	-79,5	-77,5	-72
22,5	-84,5	-80	-78,5	-74
25,0	-78	-79,5	-81,5	-75,5
27,5	-80,5	-83	-80,5	-79,5
30,0	-84,5	-86,5	-83,5	-82
32,5	-84,5	-86	-84,5	-80,5
35,0	-84	-93	-84,5	-79
37,5	-81	-	-85,5	-73,5
40,0	-86	-	-82	-81



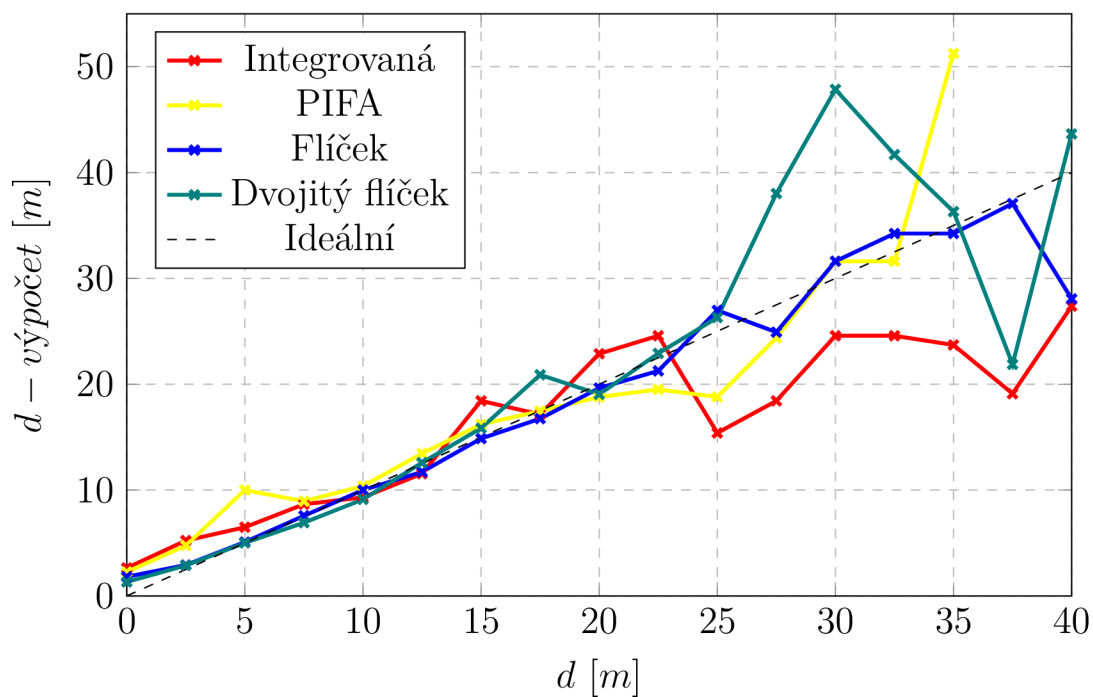
Obr. 8.1: Měření velikosti RSSI pro jednotlivé antény na vzdálenosti 40 metrů.

Z grafu je viditelné, že z hlediska síly signálu na vzdálenost, vychází jako nejlepší dvojitý flíček, jako další pak jednoduchý flíček a integrovaná anténa s planární integrovanou anténou mají přibližně stejný průběh. Dále je viditelné, že se průběhy signálu postupně vyrovnávají, respektive opět částečně roste RSSI s dále rostoucí vzdáleností. Tento efekt, který je nejvíce viditelný přibližně od značky 22,5 až 25 metru, je pravděpodobně způsoben rostoucími odrazy od země a konstruktivní interferencí vyslaných elektromagnetických vln.

Aby bylo možné ověřit s jakou přesností dochází k měření vzdálenosti, byly hodnoty z předchozího grafu, za pomoci 7.2, přepočteny na vzdálenost v metrech. Již bylo zmíněno, že ve vzorci figuruje bezrozměrná konstanta  $n$ , pro kterou je definován přibližný rozsah 2-4. Velikost této konstanty byla rozmítána s krokem 0,1 v tomto rozsahu a byl vybrán nejlepší výsledek pro případ každé měřené antény ( $n_{integ} = 3,2$ ;  $n_{pifa} = 3,1$ ;  $n_{flicek} = 2,9$ ;  $n_{dvojitýflicek} = 2,5$ ). Vypočtené hodnoty vzdálenosti jsou v následující tabulce.

Tab. 8.2: Vypočtené hodnoty vzdálenosti z RSSI.

d [m]	Antény			
	Vypočtená vzdálenost [m]			
	n = 3,2	n = 3,1	n = 2,9	n = 2,5
	Integrovaná	PIFA	Flíček	Dvojitý Flíček
0	2,6	2,3	1,8	1,3
2,5	5,2	4,8	2,9	2,9
5,0	6,5	10,0	5,1	5,0
7,5	8,7	8,90	7,6	6,9
10,0	9,3	10,4	10,0	9,1
12,5	11,5	13,5	11,7	12,6
15,0	18,4	16,2	14,9	15,8
17,5	17,2	17,5	16,8	20,9
20,0	22,9	18,8	19,6	19,1
22,5	24,6	19,5	21,3	22,9
25,0	15,4	18,8	27,0	26,3
27,5	18,4	24,4	24,9	38,0
30,0	24,6	31,6	31,6	47,9
32,5	24,6	30,5	34,2	41,7
35,0	23,7	51,2	34,2	36,3
37,5	19,1	-	37,1	21,9
40,0	27,4	-	28,1	43,7



Obr. 8.2: Vypočtená vzdálenost ze změřených RSSI hodnot pro jednotlivé antény.

V grafu bylo také zobrazeno, jak by měl vypadat ideální průběh měření vzdálenosti. Tento průběh ukazuje, že předpokládaná vzdálenost 22,5 až 25 metrů je hraniční pro alespoň uspokojivé měření vzdálenosti. V případě nulové vzdálenosti je vždy naměřena relativně vysoká odchylka a to pravděpodobně z důvodu, že při měření byl přijímač přímo pod měřeným beaconem. Pro lepší přehled je na následující tabulce zaznamenána velikost odchylky od skutečné hodnoty vzdálenosti.

Tab. 8.3: Velikost odchylky vypočtené hodnoty od reálné vzdálenosti.

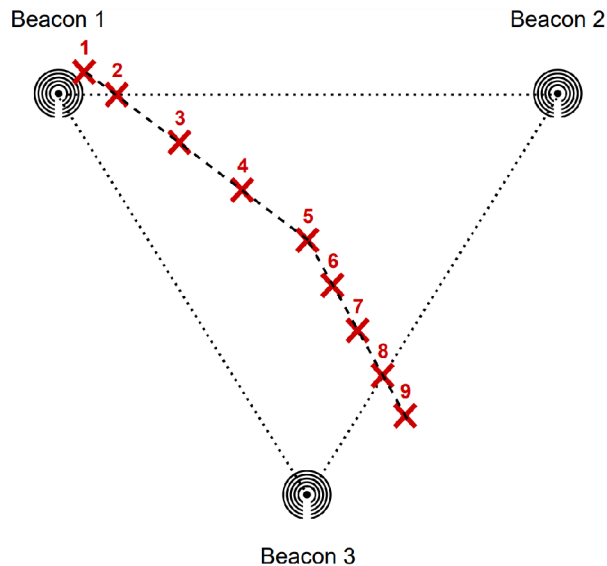
d [m]	Antény - Odchylka [m]			
	Integrovaná	PIFA	Flíček	Dvojitý Flíček
0	2,6	2,3	1,8	1,3
2,5	2,7	2,3	0,4	0,4
5,0	1,5	5,0	0,1	0,0
7,5	1,2	1,4	0,1	0,6
10,0	0,7	0,4	0,0	0,9
12,5	1,0	1,0	0,8	0,1
15,0	3,4	1,2	0,1	0,8
17,5	0,3	0,0	0,7	3,4
20,0	2,9	1,2	0,4	0,9
22,5	2,1	3,0	1,2	0,4
25,0	9,6	6,2	2,0	1,3
27,5	9,1	3,1	2,6	10,5
30,0	5,4	1,6	1,6	17,9
32,5	7,9	0,9	1,7	9,2
35,0	11,3	16,2	0,8	1,3
37,5	18,4	-	0,4	15,6
40,0	12,6	-	11,9	3,7
<b>Průměrná odchylka (do 22,5 m)</b>	<b>1,8</b>	<b>1,8</b>	<b>0,6</b>	<b>0,9</b>

## 8.2 Měření lokalizace

Na základě znalostí získaných v měření vzdálenosti, bude provedeno měření lokalizace uvnitř útvaru (pomyslného trojúhelníku), jehož vrcholy budou tvořeny jednotlivými beacony - tak jak je definováno v aplikaci. Trojúhelník, v kterém bude provedena lokalizace, bude rovnostranný o velikosti strany 20 metrů. Tato hodnota byla vybrána na základě předchozího měření, kde se jako limit vzdálenosti jevila vzdálenost 22,5 - 25 metrů.

Na následujícím obrázku je zobrazen náčrt měření lokalizace s body v místech, kde bude provedeno měření.

Následující tabulka obsahuje reálné změřené vzdálenosti do jednotlivých bodů měření od Beaconů 1, 2 a 3.



Obr. 8.3: Průběh měření lokalizace. Jednotlivé zaznamenané body zobrazují místa měření signálu v rámci vymezeného prostoru, přičemž bod č. 5 se nachází uprostřed pomyslného trojúhelníku.

Tab. 8.4: Vzdálenosti bodů měření od Beaconů.

	Vzdálenost [m]		
	Beacon		
bod	1	2	3
1.	2,0	19,1	21,0
2.	<b>3,0</b>	<b>17,0</b>	<b>18,7</b>
3.	5,7	14,8	16,1
4.	8,6	12,9	13,7
5.	<b>11,5</b>	<b>11,5</b>	<b>11,5</b>
6.	13,5	11,7	9,8
7.	15,6	12,2	8,2
8.	<b>17,6</b>	<b>13,0</b>	<b>7,0</b>
9.	19,5	14,0	6,3

Tento formát měření a umístění bodů byl vybrán tak, aby bylo malé množství bodů, které jsou stejně vzdálené od všech beaconů (zde se jedná pouze o střed pomyslného trojúhelníku, který byl ponechán pro srovnání). Zároveň jsou zde i body (1. a 9.), které se nachází mimo vytyčenou oblast, aby bylo možné pozorovat vliv toho, pokud přijímač částečně zmizí z vytyčené oblasti.

V tabulce 8.2 jsou vyznačeny tučně body 2., 5. a 8., protože se jedná o body, které dostali přesně definované vzdálenosti již v rámci náčrtu a zároveň pak sloužily jako body pro vytvoření úseček skrze vytyčenou oblast. Zároveň bod 5. je středem.

Pro lokalizační měření byly použity 3 různé antény a to z důvodu zjištění podobnosti odchylky v 8.1, které bylo nalezeno u prvních dvou antén. Bylo provedeno vícenásobné měření v jednotlivých bodech. Lokalizační data byla zpracovávána pomocí aplikace a zaznamenávána jejich velikost.

Pro zjednodušení vyhodnocení a především lepší orientaci ve výsledcích (z důvodu velkého množství dat), následující tabulka obsahuje již mobilní aplikací vypočtené hodnoty, které byly přepočteny jako odchylky od reálných vzdáleností. Přepočet proběhl stejným způsobem, jako při měření samotné vzdálenosti, kde byly zároveň použity poznatky o bezrozměrném koeficientu prostředí. Jeho velikost byla v aplikaci vždy nastavena pro požadované antény (velikost odpovídá 8.1).

Tab. 8.5: Velikost odchylky od reálné vzdálenosti při lokalizaci.

	Odchylka [m]								
	PIFA			Flíček			Dvojitý flíček		
	Beacon								
<b>bod</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>1.</b>	0,7	1,9	2,9	4,8	3,0	7,3	7,1	14,0	6,5
<b>2.</b>	1,8	2,5	8,7	0,9	4,8	7,9	3,9	3,9	0,4
<b>3.</b>	4,3	0,8	8,1	0,3	1,6	4,4	1,7	1,6	5,1
<b>4.</b>	5,9	3,9	2,9	1,3	1,7	3,3	3,8	0,9	4,5
<b>5.</b>	0,8	0,8	1,5	1,9	0,7	1,5	4,0	4,6	4,0
<b>6.</b>	2,1	1,7	1,8	1,8	0,8	1,6	2,5	4,1	2,9
<b>7.</b>	5,5	2,2	1,8	1,3	2,6	1,5	0,3	3,9	2,4
<b>8.</b>	10,7	3,8	2,3	0,5	3,1	1,2	1,4	5,4	2,2
<b>9.</b>	13,3	2,4	0,6	2,6	7,2	6,4	13,6	1,8	1,3
<b>Průměrná odchylka</b>	<b>5,01</b>	<b>2,23</b>	<b>3,40</b>	<b>1,71</b>	<b>2,84</b>	<b>3,89</b>	<b>4,27</b>	<b>4,47</b>	<b>3,25</b>
<b>Celková průměrná odchylka</b>	<b>3,54</b>			<b>2,81</b>			<b>4,03</b>		

Veškeré hodnoty zaznamenány v tabulce jsou v metrech. Všechny beacons byly při jednotlivých měřeních nastaveny na jeden typ antény a do pozic, jak je naznačeno v rámci nákresu 8.2. Ve spodní části tabulky jsou vypočítány průměrné odchylky v rámci jednoho beaconu při měření určitým typem antény. Na posledním řádku tabulky je vypočtená celková průměrná odchylka celého lokalizačního měření pro danou anténu, aby bylo možné nějakým způsobem kvantifikovat, jak přesné či nepřesné toto měření bylo.

V případě planární invertované F antény je největší odchylka v rámci beaconu č. 1. Zde je odchylka až 5 metrů i vzhledem k posledním bodům, kde hodnoty odchylky dosahují i více než 10 metrů. V případě beaconu č. 2 dostáváme přibližně o 0,5

metrů horší průměrnou odchylku oproti měření vzdálenosti v 8.1. Rozsah hodnot se pohybuje mezi 11 až 19 metry, kde byla i planární invertovaná anténa nejpřesnější ze svého rozsahu měření vzdálenosti. Pro beacon č. 3 je průměrná odchylka vyšší, avšak pravděpodobnou příčinou jsou body měření č. 1 až č. 3, kde je odchylka nejvyšší. Jinak by se její hodnota pohybovala okolo 1,8 metru, což koresponduje s měření vzdálenosti. Celková průměrná odchylka zde tedy vychází okolo 3,5 metru.

Při použití jednoduchého flíčku na beaconu č. 1 jsou hodnoty relativně přesné k reálným. Největším odchylkám dochází v případě bodu č. 1, kde je zřejmě signál slabý kvůli umístění mimo hlavní část vyzařovací charakteristiky a pak také v posledním bodě č. 9, kde je již vyšší vzdálenost. Pro beacon 2 a 3 dochází k horším výsledkům, které jsou opět nejvíce ovlivněny krajními body měření. Celková průměrná odchylka je zde nejlepší z celého měření a to přibližně 2,8 metru.

V posledním případě dvojitého flíčku dosahujeme horších výsledků, než v případě předchozích antén. Zde se zřejmě nejvíce projevuje umístění mimo hlavní vyzařovací charakteristiku antén, která způsobí, že celková průměrná odchylka se pohybuje okolo 4 metrů.

Obecný poznatkem je, že nejpřesnější bývají antény pro měření v bodě č. 5, který se nachází pro všechny antény v rovině hlavní osy vyzařování. Naopak nejméně přesné jsou v průměru antény na okrajových bodech měření (č. 1 a č. 9). Pokud dojde k opomenutí krajních bodů měření, jakožto bodů, které se nenacházejí v rámci pomyslného lokalizačního trojúhelníku dostáváme následující výsledky.

Tab. 8.6: Velikost odchylky od reálné vzdálenosti při lokalizaci v případě odebrání krajních bodů měření.

	Odchylka [m]		
	PIFA	Flíček	Dvojitý flíček
<b>Celková průměrná odchylka</b>	<b>3,52</b>	<b>2,12</b>	<b>3,02</b>
<b>Rozdíl [%]</b>	<b>0,8</b>	<b>24,5</b>	<b>24,3</b>

Zde je vidět, že v případě odebrání krajních bodů měření v případě planární invertované F antény je rozdíl menší než 1 %, tedy dá se říct, že krajní body neovlivňují měření v případě této antény, což může být způsobeno tvarem vyzařovací charakteristiky. Naopak v případě jednoduchého flíčku a dvojitého flíčku krajní body zapříčiňují zvětšení odchylky až o téměř 25 %.



# Závěr

Tato diplomová práce byla zaměřena na vytvoření základu pro lokalizační zařízení fungující v Bluetooth 5.2, vytvoření více druhů antén, umožňujících přenosu radiofrekvenční části, a také vytvoření aplikace pro mobilní telefon, která provádí částečnou lokalizaci měření signálu.

Pro napájení zařízení byla zvolena knoflíková anténa 3V s kapacitou 580 mAh, je tedy očekávána relativně velká výdrž zařízení z hlediska samostatnosti. Čipem, který pracuje s Bluetooth 5.2, byl zvolen nový programovatelný EFR32BG22, který podporuje i lokalizační protokoly. Vzhledem k funkčnosti na přibližných 2,4 GHz byly dodržovány pokyny pro návrh z jednotlivých datasheetů pro EFR32BG22. S šířkou 32 mm a délkou 52 mm byla dodržena nízká velikost zařízení.

Výsledkem předchozího návrhu je deska plošných spojů, přizpůsobená pro frekvenční pásmo 2,4 GHz. Součástí desky plošných spojů je footprint konektoru, přes který je možné připojit zařízení k hlavní vývojové desce od Silicon Labs a tou pak vyrobenou desku resp. čip EFR32BG22 naprogramovat. Při návrhu desky byly dodržovány doporučené postupy, aby byla maximalizována pravděpodobnost dobrého přizpůsobení a nízkého rušení.

Následně byly vytvořeny podklady pro tvorbu tří druhů antén, které byly vyrobeny a poté měřeny. Planární invertovaná F anténa byla před dosažením uspokojivých parametrů na požadovaném frekvenčním pásmu, vyrobena ve 3 různých verzích. V první verzi byla nejprve zkrácena délka hlavního zářiče, což stále nevedlo k uspokojivým výsledkům. Ve druhé verzi byla opět zkrácena délka hlavního zářiče a zároveň byla zemní plocha antény posunuta blíže k hlavnímu zářiči. Tyto změny již vedly na uspokojivou velikost vstupního činitele odrazu. Vzhledem k malým rozměrům antény a odchylkám v technologii, byly tyto chyby očekávanější. V případě fličkové antény nebylo třeba vytvářet změny nebo úpravy. V případě dvojitého fličku byla rezonanční frekvence antény posunuta až na hranici 2,5 GHz. Z tohoto důvodu byla vytvořena další verze antény, kde došlo k přiblížení středů antén pro částečné snížení frekvence na finální rezonanční frekvenci 2,43 GHz.

Firmware zařízení byl vytvořen za pomoci *Simplicity Studio 5*. Zde byly nejdříve vytvořeny jednoduché programy pro spouštění periférií, aby bylo možné ověřit, zda lze desku oživit naprogramováním. Tohoto bylo dosaženo za pomoci spouštění vnitřního oscilátoru na výstupním pinu čipu. Následně byl vytvořen program pro komunikaci v Bluetooth. V tomto případě umožňuje *Simplicity Studio 5* usnadnění v podobě vytvoření bootloADERU pro Bluetooth na programovací paměť o velikosti (v tomto případě) 352 kB. *Simplicity Studio 5* zároveň umožňuje kontrolu, zda proběhlo k nahrání programu do programovací paměti. Za pomoci *Simplicity commanderu* je možné zobrazit obsah flash paměti (zde nazváno *Flash Map*), který následně zob-

razí obsazené pozice programem a bootloaderem v rámci paměti. Samotný program provádí jednoduché vysílání názvu svého čipu společně s ID zařízení. Díky tomu je pak možné zařízení rozpoznat a následně lokalizovat.

Zde se nepodařilo dosáhnout uspokojivých výsledků. Z hlediska testování a časové úspory bylo nejprve prováděno testování vytvořeného programu na *EFR32xG22 Wireless Gecko Radio Board*, což je vývojová deska pro rádiové aplikace. Ačkoliv bylo nahrávání bootloaderu a samotného programu prováděno bezproblémově na zmíněné desce pro rádiové aplikace, nepodařilo se stejný program spustit na vytvořeném zařízení, i přes viditelné změny v rámci *Flash Map*. Pravděpodobně byl tento problém způsoben tím, že samotný čip nebyl schopný bootloader použít pro nastavení a zapnutí vysílání pomocí Bluetooth. Aby bylo možné dosáhnout výsledků měření a ověření funkčnosti kódů zařízení i aplikace, byl vytvořen shield pro napájení *Wireless Gecko Radio Board* z knoflíkové baterie (stejně jako pro původní návrh), která vychází ze schématu v příloze , kde je zobrazena i fotografie pomocné desky.

Aby bylo možné zpracovat vysílaná data, byla vytvořena aplikace pro mobilní telefon systému Android, která stojí na základě volně dostupného softwaru pro simulování Bluetooth zařízení. Tato aplikace ukazuje dostupné beacons, jejich ID, názvy a především sílu signálu v podobě RSSI. Jsou zobrazovány pouze beacons, které mají nastavené unikátní názvy pro tuto lokalizaci. Aplikace zobrazuje měnící se RSSI, kde dochází k vysílání v pravidelných intervalech 2,5 sekundy, stanovených programem samotného beaconu pro optimální spotřebu energie. Pokud jsou přítomny všechny beacons, je možné provést lokalizaci. Po stisku tlačítka se zobrazují beacons s velikostí odpovídající silám signálů. Pomocí graficky zobrazených dosahů beaconů je možné se podle grafického znázornění přibližně orientovat v prostoru, podle velikosti signálů beaconů.

V kategorii měření bylo nejprve provedeno měření vzdálenosti, aby bylo možné stanovit, jaké bude třeba nastavit koeficienty prostředí pro antény a aby bylo možné porovnat, jak přesné jsou antény v různých případech. Ukázalo se, že pokud je měřená vzdálenost všemi druhy antén, je při Integrované (čipové) anténě a planární invertované anténě průměrná odchylka (do 22,5 metru, kde nastane již velká odchylka ve všech případech, proto je uvažováno jen do této vzdálenosti) od měřené vzdálenosti 1,8 metru. Pro flíčkovou anténu je průměrná odchylka 0,6 metru a v případě dvojitého flíčku se jedná o 0,9 metru. Veškeré hodnoty, které se při průměrné chybě pohybují pod 1 metrem, lze považovat za relativně přesné z hlediska určování vzdálenosti a uspokojující pro tuto metodu.

Lokalizační měření již dosahovalo větších výchylek, kde se jako nejpřesnější anténa pro určování vzdálenosti jeví jednoduchý flíček, následně planární invertovaná F anténa a poté dvojitý flíček. Pokud nezohledníme body na krajních částech měření, je průměrná odchylka přibližně 2 metry pro jednoduchý flíček, 3 metry pro

dvojitý flíček a při planární invertované F anténě se jedná o 3,5 metru.

Bylo vytvořeno zařízení pro komunikaci v Bluetooth, které vysílá svoje parametry, následně pak byla vytvořena aplikace, která vyhodnocuje data a převádí je do lokalizační podoby pro orientaci v prostoru. Závěrem by se dalo říci, že pro určování vzdálenosti by se v případě antén tvořených flíčky mohlo jednat o relativně přesnou metodu s odchylkou pod 1 metr, avšak z hlediska lokalizace jsou odchylky relativně vysoké. Při uložení beaconů výše nad zemí by pravděpodobně bylo možné dosáhnout této přesnosti i na větším vytyčeném území a tím pádem by bylo možné tolerovat i odchylky okolo 2 metrů pro lokalizaci z RSSI.

Z důvodu velikosti jsou veškeré přílohy dostupné na:  
<<https://github.com/jsligl/diplomova-prace-prilohy>>.

# Literatura

- [1] Bluetooth® Wireless Technology [online]. [cit. 2021-10-21]. Dostupné z URL: <<https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>>.
- [2] Bluetooth® SIG Proprietary *Specification of the Bluetooth System, v5.0* [online]. [cit. 2021-10-21]. Dostupné z URL: <<https://www.mouser.it/pdfdocs/bluetooth-Core-v50.pdf>>.
- [3] WU, J. *How Bluetooth Module Works? M-B14001 BLE Module Introduction* [online]. [cit. 2021-10-21]. Dostupné z URL: <<https://www.iot-rf.com/how-bluetooth-module-works-m-b14001-ble-module/-introduction.html>>.
- [4] Bluetooth LE Fundamentals [online]. [cit. 2021-10-21]. Dostupné z URL: <<https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf>>.
- [5] Bluetooth® SIG *Bluetooth Core Specification Version 5.2 Feature Overview* [online]. [cit. 2021-10-21]. Dostupné z URL: <<https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-version-5-2-feature-overview/>>.
- [6] JAMES, J. R. a P. S. HALL, ed. *Handbook of Microstrip Antennas, Volume 1* [online]. The Institution of Engineering and Technology, Michael Faraday House, Six Hills Way, Stevenage SG1 2AY, UK: IET, 1989 [cit. 2021-11-10]. ISBN 9780863417597. Dostupné z: doi:10.1049/PBEW028F
- [7] BALANIS, Constantine A. *Antenna theory: analysis and design*. 3rd ed. Hoboken: Wiley-Interscience, 2005. [cit. 2021-11-10]. ISBN 978-0-471-66782-7.
- [8] LÁČIK J. *Antény a rádiové spoje - P7: Antény pro RFID, antény pro mobilní aplikace, antény pro sensory*. [cit. 2021-11-10].
- [9] 2.45 GHz High Gain SMD Chip Antenna P/N 2450AT45A100 [online]. [cit. 2021-11-5]. Dostupné z URL: <<https://cz.mouser.com/datasheet/2/611/2450AT45A100-1519708.pdf>>.
- [10] EFR32BG22 Wireless Gecko SoC Family Data Sheet [online]. [cit. 2021-10-16]. Dostupné z URL: <<https://www.silabs.com/documents/public/data-sheets/efr32bg22-datasheet.pdf>>.

- [11] EFM32 and EFR32 WirelessGecko Series 2 Hardware Design Considerations [online]. [cit. 2021-10-16]. Dostupné z URL: <https://www.silabs.com/documents/public/application-notes/an0002.2-efr32-efm32-series-2-hardware-design-considerations.pdf>.
- [12] EFR32 Series 2 Minimal BOM [online]. [cit. 2021-10-16]. Dostupné z URL: <https://www.silabs.com/documents/public/application-notes/an933.2-efr32-series-2-minimal-bom.pdf>.
- [13] EFR32 Series 2 2.4 GHz Matching Guide [online]. [cit. 2021-10-16]. Dostupné z URL: <https://www.silabs.com/documents/public/application-notes/an930.2-efr32-series-2.pdf>.
- [14] EFR32 Series 2 Layout DesignGuide [online]. [cit. 2021-10-16]. Dostupné z URL: <https://www.silabs.com/documents/public/application-notes/an928.2-efr32-series2-layout-design-guide.pdf>.
- [15] Bluetooth SDK v3.x Quick-Start Guide [online]. [cit. 2022-04-21]. Dostupné z URL: <https://www.silabs.com/documents/public/quick-start-guides/qsg169-bluetooth-sdk-v3x-quick-start-guide.pdf>.
- [16] Beacon Simulator Android [online]. [cit. 2022-04-21]. Dostupné z URL: <https://github.com/vhiribarren/beacon-simulator-android>.
- [17] *EFR32 RSSI measurement process*. Knowledge Article. Silicon Labs. [online]. [cit. 2022-04-21]. Dostupné z URL: <https://www.silabs.com/documents/public/quick-start-guides/qsg169-bluetooth-sdk-v3x-quick-start-guide.pdf>.
- [18] GOLESTANIAN, Mehdi, Josuha SIVA a Christian POELLABAUER. *Radio Frequency-Based Indoor Localization in Ad-Hoc Networks*. ORTIZ, Jesus Hamilton a Alvaro Pachon de la CRUZ, ed. Ad Hoc Networks [online]. In-Tech, 2017, 2017-05-11 [cit. 2022-04-21]. ISBN 978-953-51-3109-0. Dostupné z: doi:10.5772/66523
- [19] Hossain AM, Soh WS. *A comprehensive study of bluetooth signal parameters for localization*. In 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications. 2007 Sep 3 (pp. 1–5). IEEE. [online]. [cit. 2022-04-21].

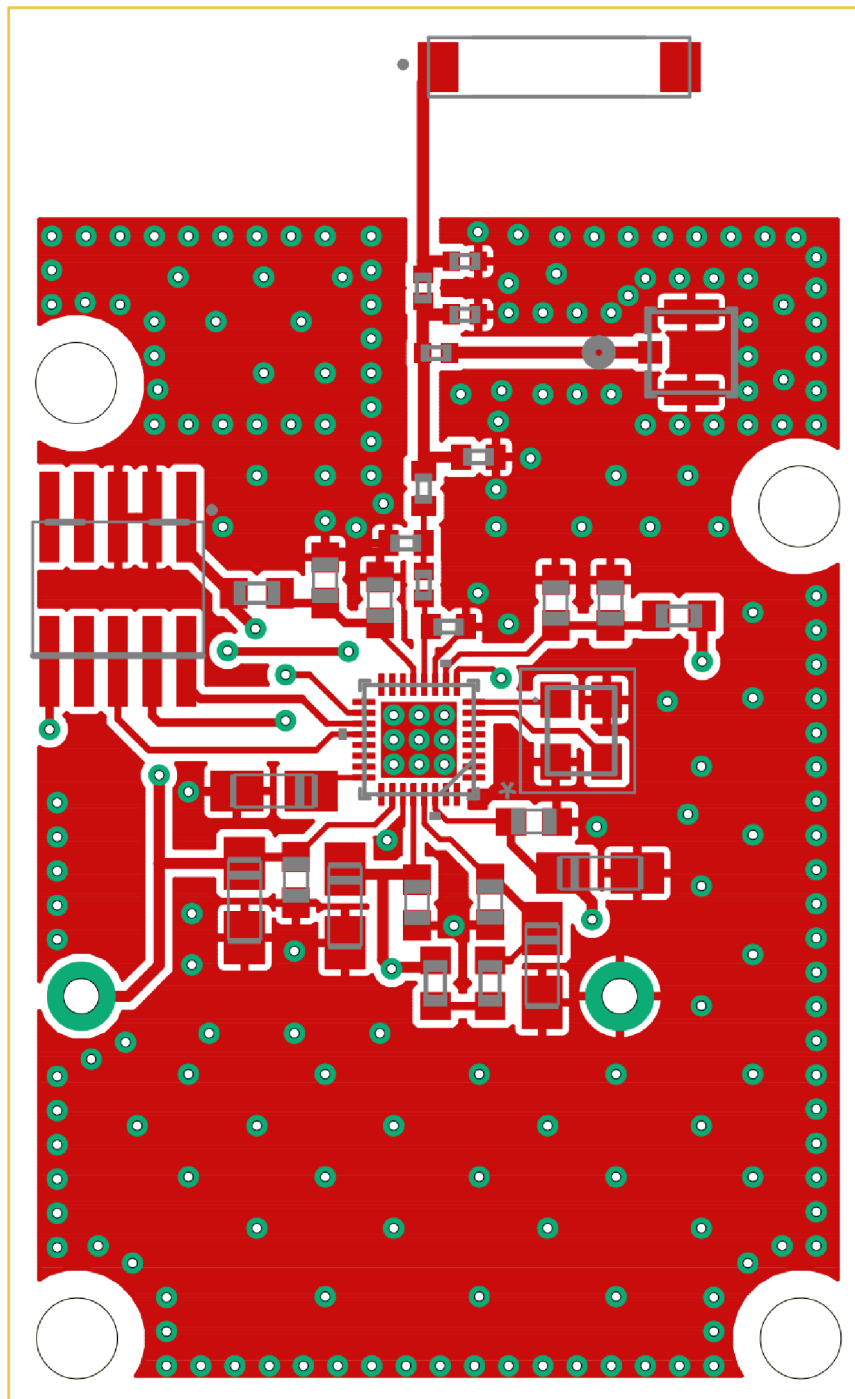
- [20] Subhan F, Hasbullah H, Rozyyev A, Bakhsh ST. *Handover in bluetooth networks using signal parameters*. Information Technology Journal. 2011;10(5):965-73. [online]. [cit. 2022-04-21].
- [21] Maytham Fahmi. *IELS: Indoor Equipment Localization System: System Design (Part 2/6) (IoT)*. itbackyard.com [online]. [cit. 2022-04-21]. Dostupné z URL: <<https://cutt.ly/QHgrBvb>>.
- [22] Android developers documentation - Android Platform Packages [online]. [cit. 2022-04-21]. Dostupné z URL: <<https://developer.android.com/reference/android/bluetooth/le/package-summary>>.

# Seznam příloh

<b>A</b>	<b>Deska plošných spojů navrženého zařízení</b>	<b>64</b>
A.1	Motiv desky . . . . .	64
A.2	Osazovací plán desky . . . . .	66
A.3	Seznam součástek . . . . .	68
A.4	Fotografie . . . . .	69
A.5	Deska pro pomocné měření . . . . .	72
A.5.1	Motiv desky . . . . .	72
A.5.2	Fotografie . . . . .	74
<b>B</b>	<b>Zdrojový kód zařízení</b>	<b>76</b>
<b>C</b>	<b>Zdrojový kód aplikace</b>	<b>81</b>

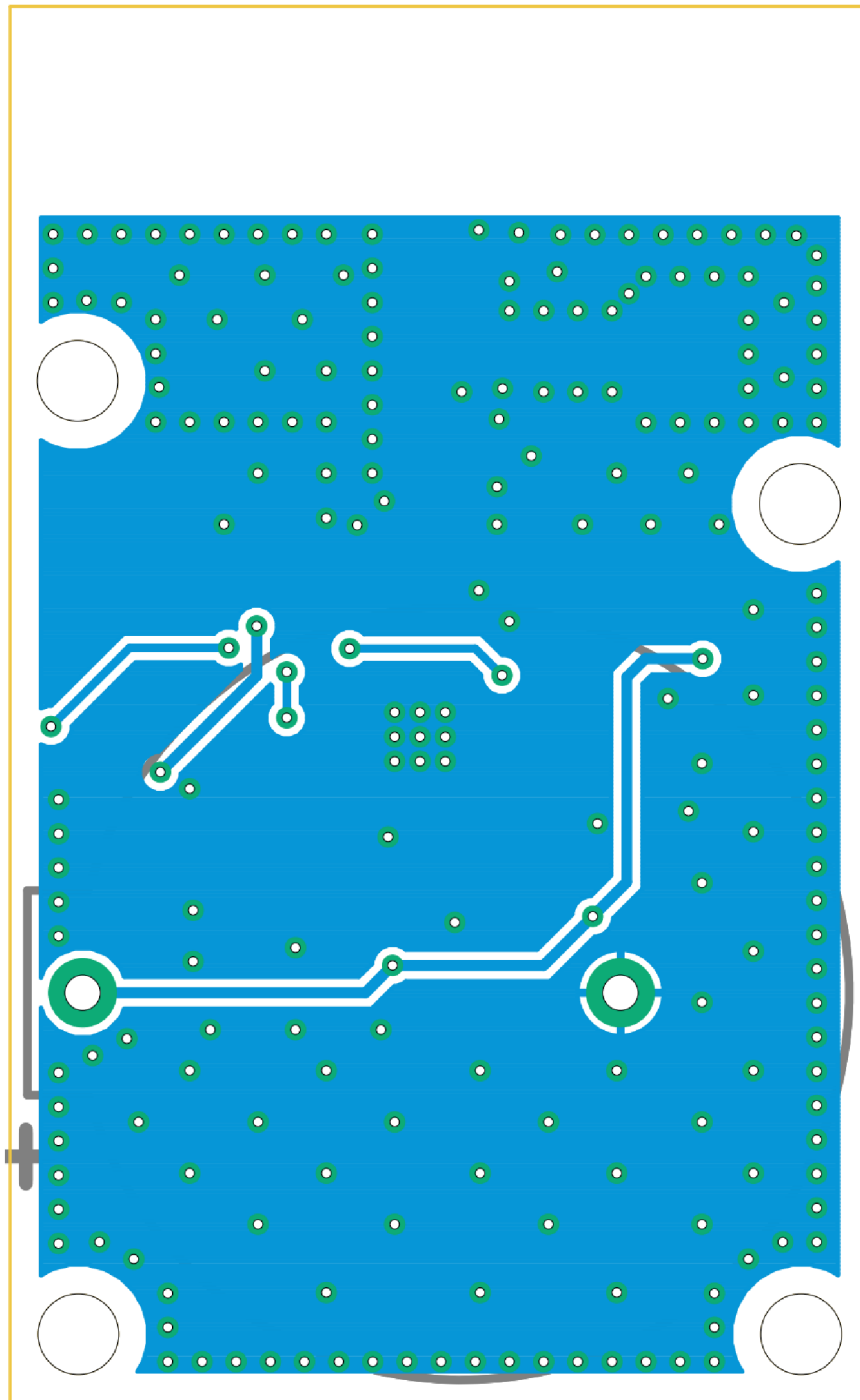
# A Deska plošných spojů navrženého zařízení

## A.1 Motiv desky



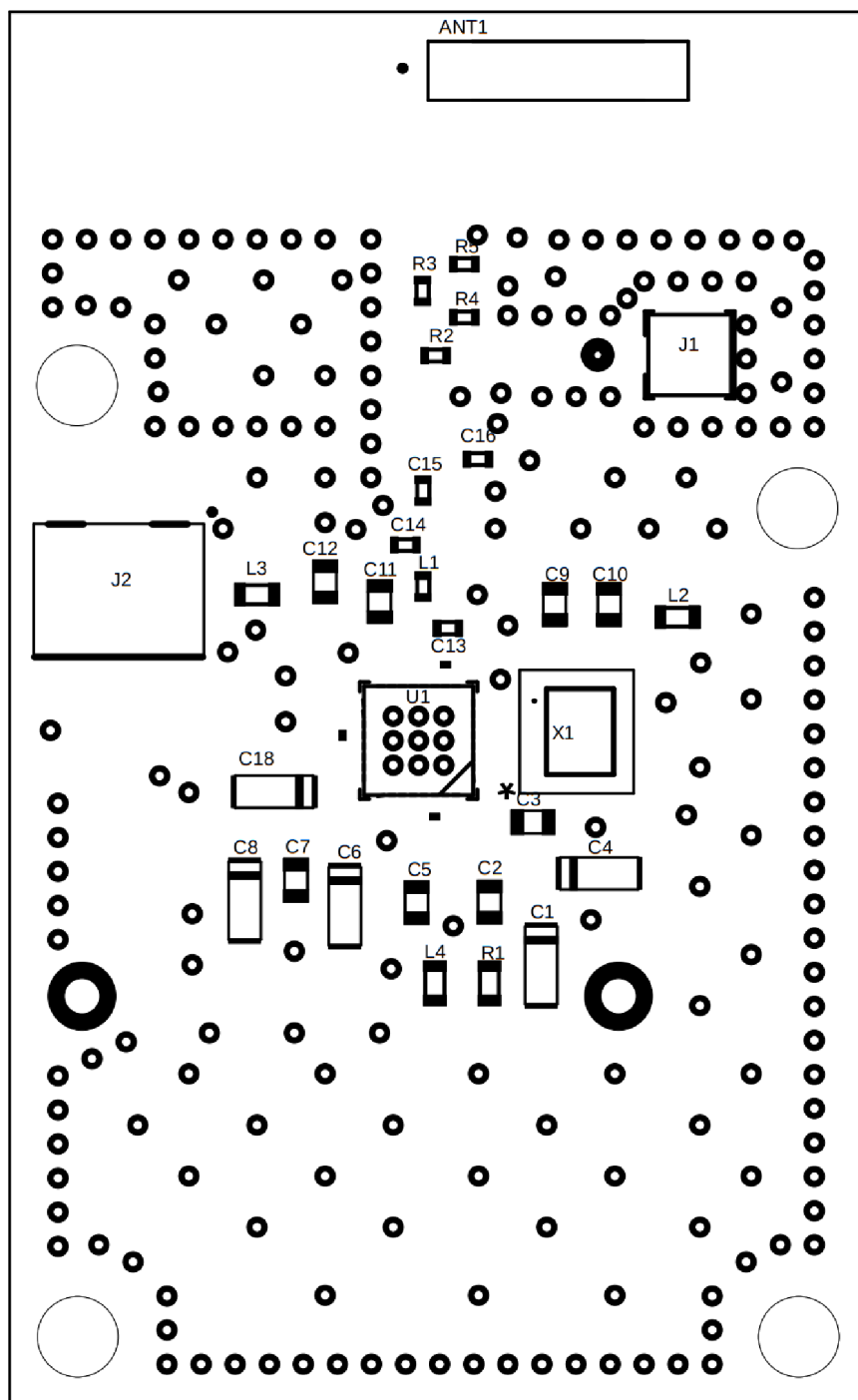
Obr. A.1: Vrchní strana desky plošných spojů vytvořená dle doporučení a na základě schématu.



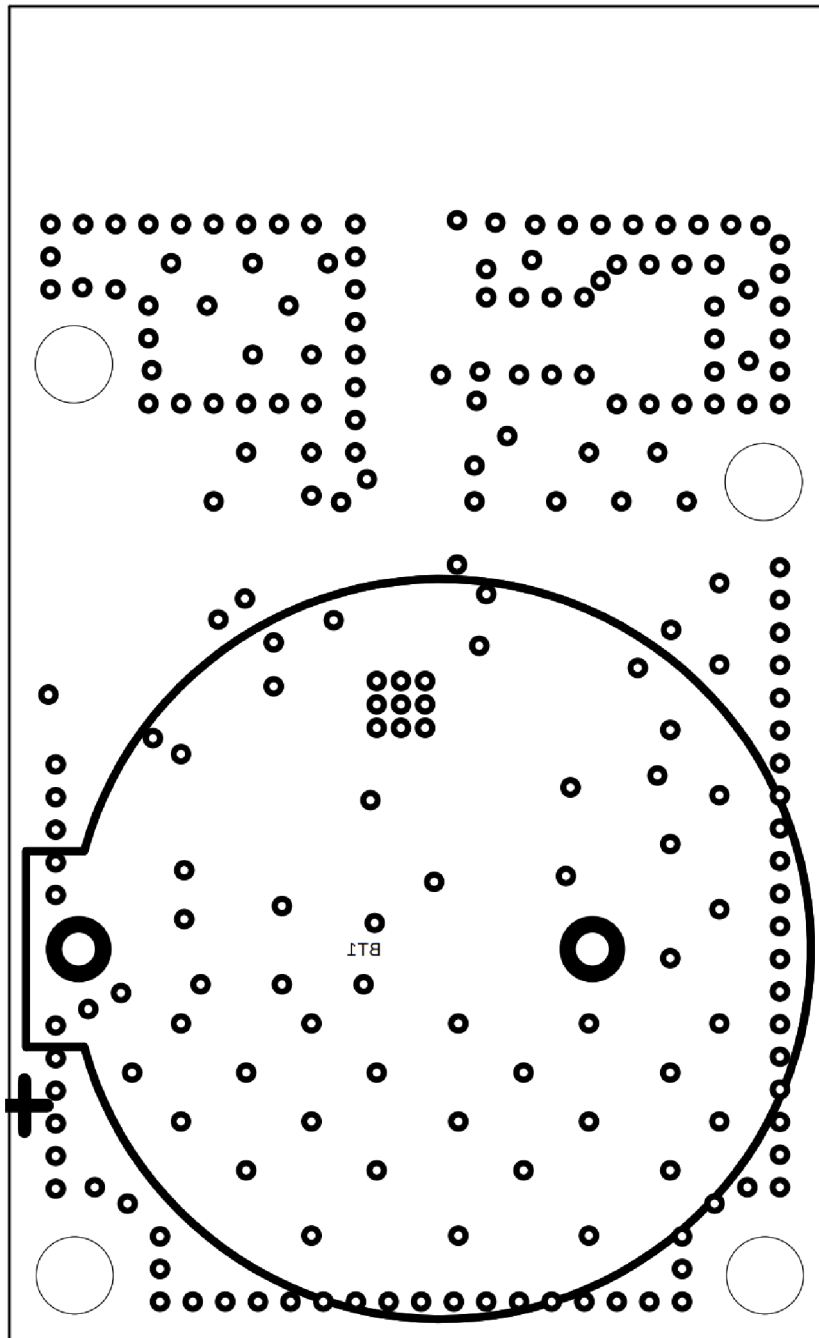


Obr. A.2: Spodní strana desky plošných spojů vytvořená dle doporučení a na základě schématu.

## A.2 Osazovací plán desky



Obr. A.3: Vrchní strana desky - osazovací plán.



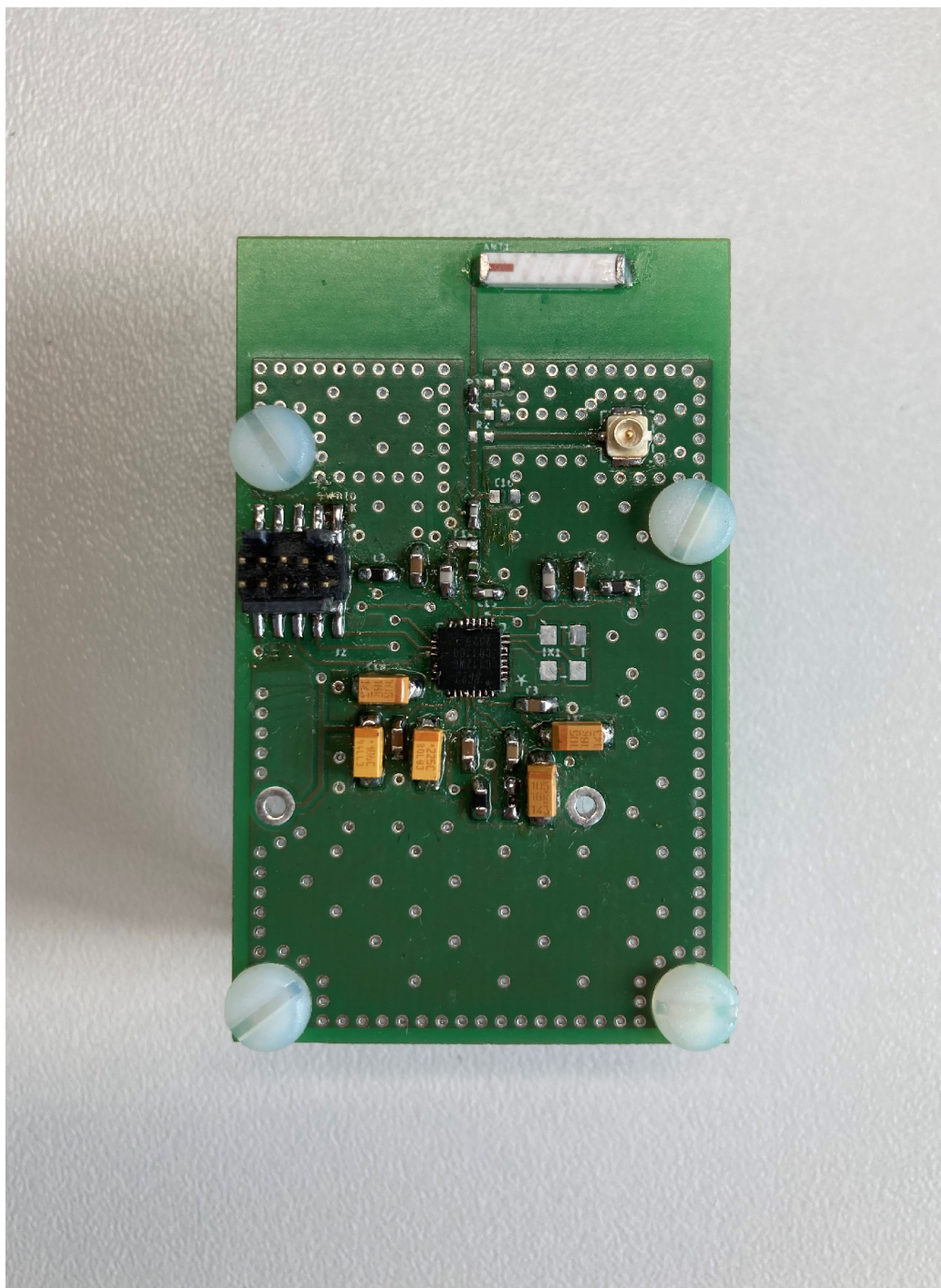
Obr. A.4: Spodní strana desky - osazovací plán.

## A.3 Seznam součástek

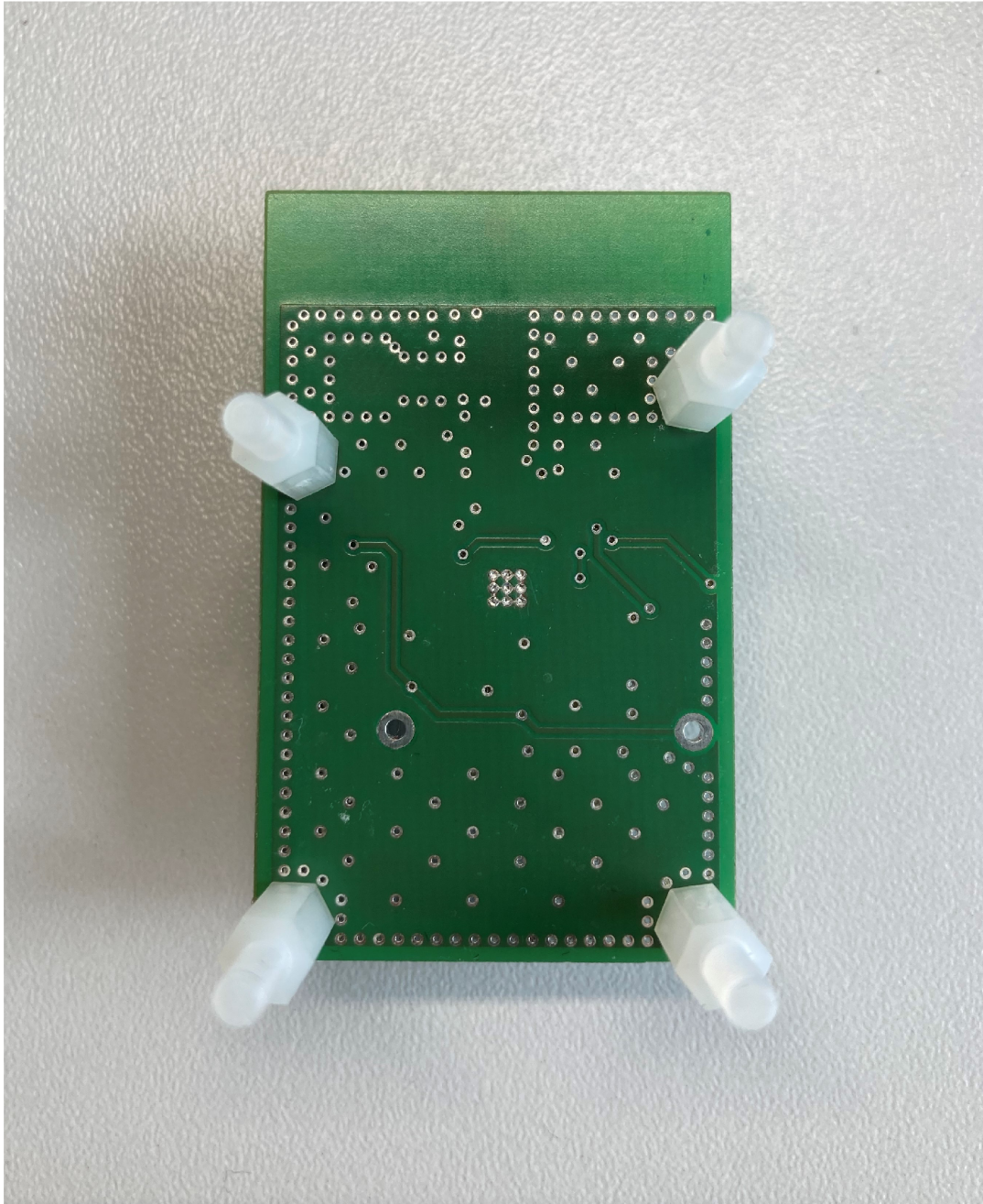
Tab. A.1: Seznam součástek.

Počet	Hodnota	Součástka	Pouzdro	Součástky
2	0R	R-EU_R0402	R0402	R2, R3
5	100n/50V	C-EUC0603	C0603	C3,C5,C7,C10,C12
1	BH2430	BH2430T-C	BAT_BH2430T-C	BT1
1	10n/50V	C-EUC0603	C0603	C2
1	10u/16V	CPOL-EUA/3216-18R	A/3216-18R	C8
2	120p/50V	C-EUC0603	C0603	C9,C11
1	18p	C-EUC0402	C0402	C15
1	1R	R-EU_R0603	R0603	R1
1	1p3	C-EUC0402	C0402	C14
1	1p7	C-EUC0402	C0402	C13
3	1u/16V	CPOL-EUA/3216-18R	A/3216-18R	C1,C4,C18
1	2450AT45A100E	2450AT45A100E	2450AT45A100E	ANT1
1	2n7	R-EU_R0402	R0402	L1
1	2u7/16V	CPOL-EUA/3216-18R	A/3216-18R	C6
1	38M4,40R,10p	ECS-160-9-33B-CKM-TR	ECS2501033BCKMTR	X1
1	EFR32BG22	EFR32BG22QFN32-C_SIL-M	QFN32-C_SIL-M	U1
3	FB-1k	R-EU_R0603	R0603	L2,L3,L4
1	4048	4048	FTR-105-01-S-D	PROG
1	OPT	C-EUC0402	C0402	C16
2	OPT	R-EU_R0402	R0402	R4,R5
1	U.FL-R-SMT-1(10)	U.FL-R-SMT-1(10)	U.FL-R-SMT-1(10)	J1

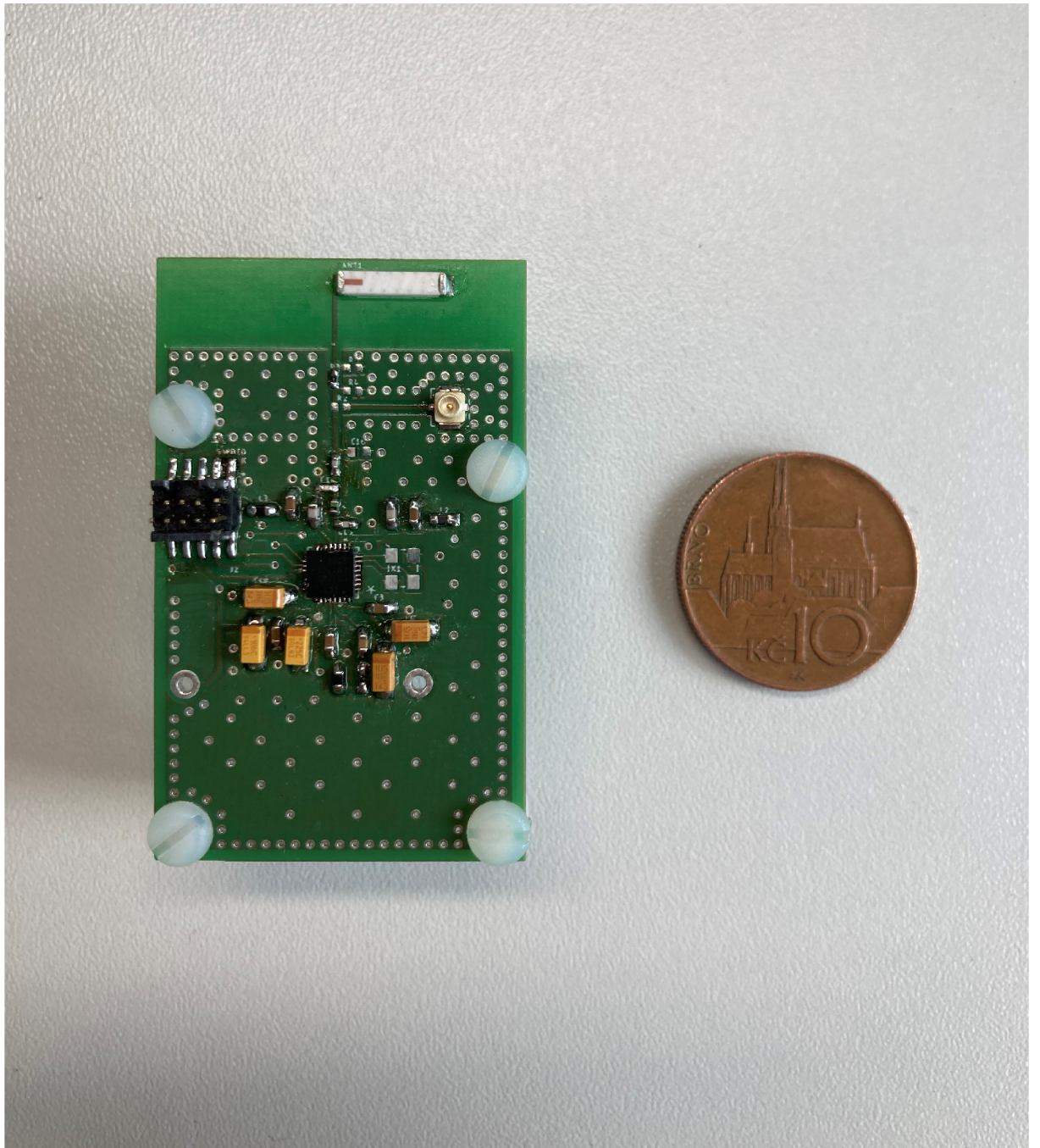
## A.4 Fotografie



Obr. A.5: Fotografie osazené desky - vrchní strana.



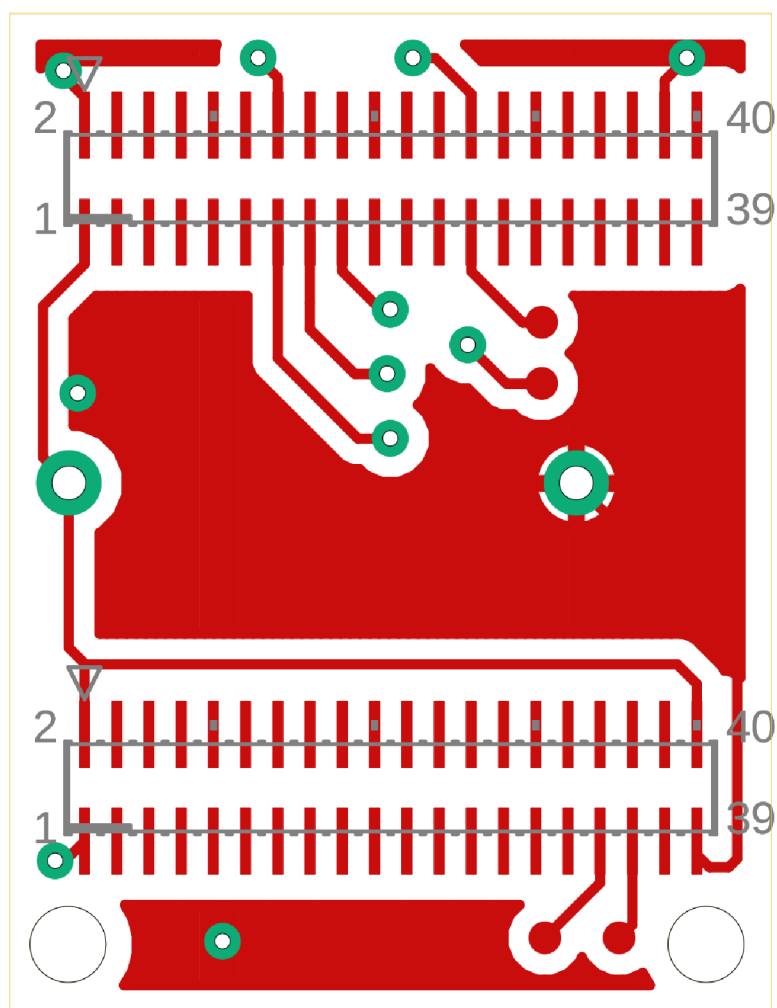
Obr. A.6: Fotografie osazené desky - spodní strana.



Tab. A.2: Fotografie osazené desky - srovnání velikosti.

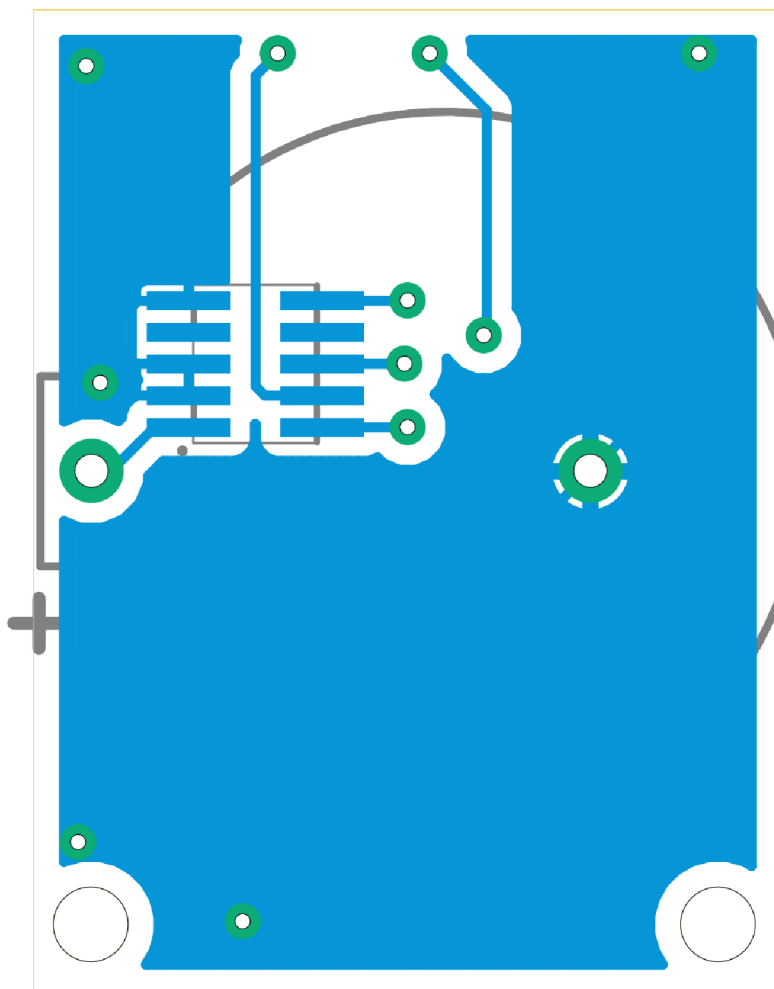
## A.5 Deska pro pomocné měření

### A.5.1 Motiv desky



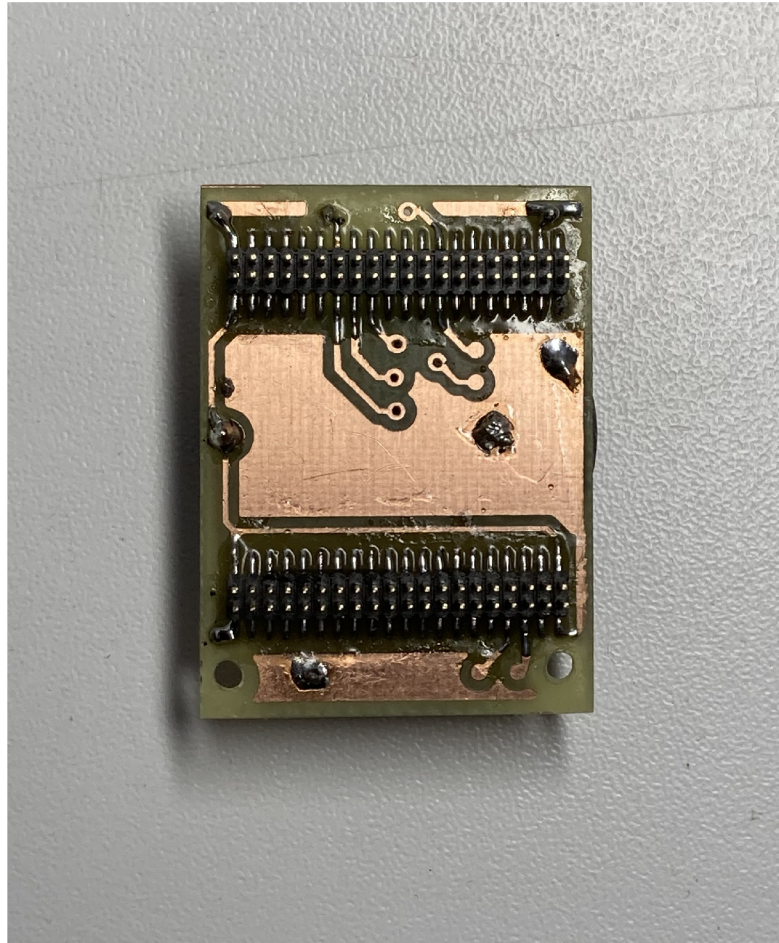
Obr. A.7: Vrchní strana desky plošných spojů pomocné desky pro měření.



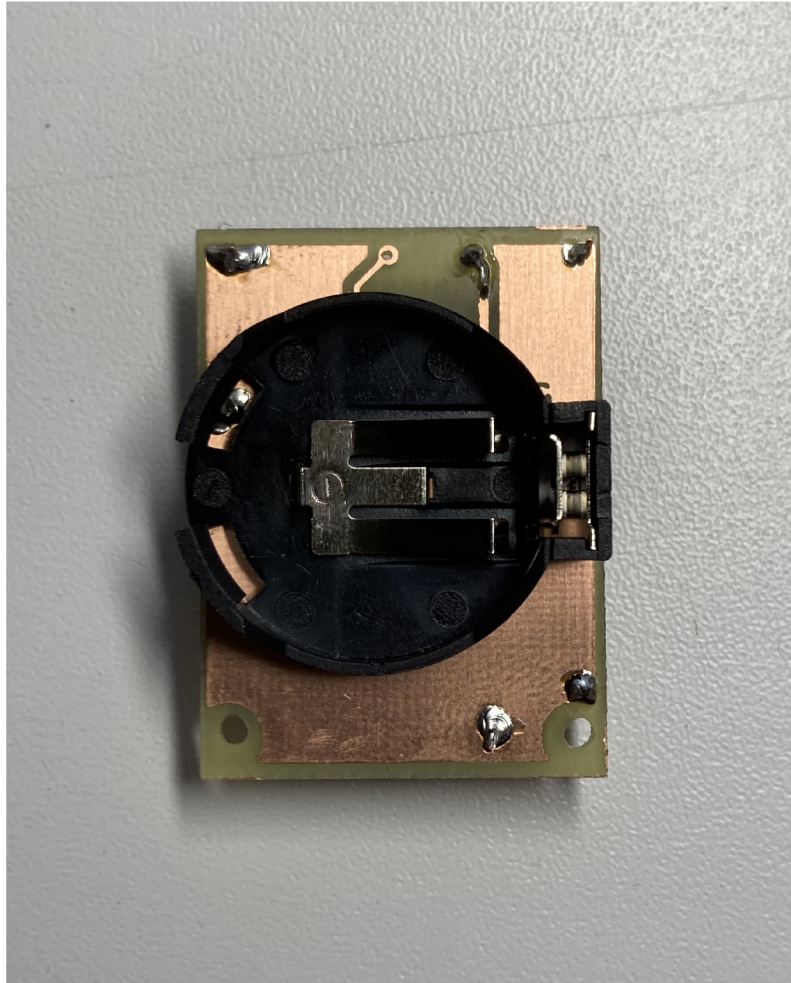


Obr. A.8: Spodní strana desky plošných spojů pomocné desky pro měření.

## A.5.2 Fotografie



Obr. A.9: Fotografie osazené pomocné desky - vrchní strana.



Obr. A.10: Fotografie osazené pomocné desky - vrchní strana.

## B Zdrojový kód zařízení

Výpis B.1: Firmware zařízení - část *main.c*

```
***** MAIN.C ***** 1
#include "sl_component_catalog.h" 2
#include "sl_system_init.h" 3
#include "app.h" 4
#include "sl_power_manager.h" 5

6
int main(void) 7
{ 8
    // Inicializovat Silicon Lab zařízení, systém 9
    // obsluhy a stack protokol 10
    sl_system_init(); 11
12
    // Inicializovat aplikaci 13
    app_init(); 14
15
    while (1) { 16
        // Komponenty Silicon Labs zpracovávají 17
        // akci volanou zevnitř smyčky - app.c 18
        sl_system_process_action(); 19
20
        // Aplikační procesy 21
        app_process_action(); 22
23
        // Umožnit CPU přechod do spánku, pokud 24
        // to systém dovolí 25
        sl_power_manager_sleep(); 26
    } 27
} 28
```

Výpis B.2: Firmware zařízení - 1. část *app.c*

```

***** APP.C *****
#include "em_common.h"
#include "app_assert.h"
#include "sl_bluetooth.h"
#include "gatt_db.h"
#include "app.h"

// Advertising set handle přidělený z Bluetooth
// stacku - maximální počet advertising setů
static uint8_t advertising_set_handle = 0xff;

// Inicializace aplikace
SL_WEAK void app_init(void){}

// Akce procesu aplikace
SL_WEAK void app_process_action(void){}

// Handler události na Bluetooth stacku
void sl_bt_on_event(sl_bt_msg_t *evt)
{
    sl_status_t sc;
    bd_addr address;
    uint8_t address_type;
    uint8_t system_id[8];

    // Zařízení je nastartováno a radio je připraveno
    switch (SL_BT_MSG_ID(evt->header)) {
        case sl_bt_evt_system_boot_id:

            // Získání ID z BT adresy
            sc = sl_bt_system_get_identity_address(
                &address,
                &address_type
            );
            app_assert_status(sc);
    }
}

```

Výpis B.3: Firmware zařízení - 2. část *app.c*

```

// Obrácení ID pro získání systémového ID
system_id[0] = address.addr[5];
system_id[1] = address.addr[4];
system_id[2] = address.addr[3];
system_id[3] = 0xFF;
system_id[4] = 0xFE;
system_id[5] = address.addr[2];
system_id[6] = address.addr[1];
system_id[7] = address.addr[0];

sc = sl_bt_gatt_server_write ...
    ... _attribute_value(gattdb_system_id,
                        0,
                        sizeof(system_id),
                        system_id);
app_assert_status(sc);

// Vytvoření advertising setu
sc = sl_bt_advertiser_create_set(
    &advertising_set_handle);
app_assert_status(sc);

// ---- Redukce spotřeby baterie ----
// Nastavit vysílání pouze na kanál 38
sl_bt_advertiser_set_channel_map(0, 2);

// Advertising interval nastavit na 2500 ms
// -----
sc = sl_bt_advertiser_set_timing(
    advertising_set_handle,
    4000, // min. adv. interval (millisekundy * 1.6)
    4000, // max. adv. interval (millisekundy * 1.6)
    0,    // adv. průběh
    0);  // max. num. adv. událostí
app_assert_status(sc);
sc = sl_bt_advertiser_start(
    advertising_set_handle,
    sl_bt_advertiser_general_discoverable,
    sl_bt_advertiser_scannable_non_connectable);

```

Výpis B.4: Firmware zařízení - 3. část *app.c*

<code>    // Začít vysílat a umožnit zařízení být skenované</code>	74
<code>    app_assert_status(sc);</code>	75
<code>    break;</code>	76
<code>    // -----</code>	77
<code>    // Událost otevření připojení</code>	78
<code>    case sl_bt_evt_connection_opened_id:</code>	79
<code>        break;</code>	80
<code>    // -----</code>	81
<code>    // Událost uzavření připojení - restart advertisingu</code>	82
<code>    case sl_bt_evt_connection_closed_id:</code>	83
<code>        sc = sl_bt_advertiser_start(</code>	84
<code>            advertising_set_handle,</code>	85
<code>            sl_bt_advertiser_general_discoverable,</code>	86
<code>            sl_bt_advertiser_scannable_non_connectable);</code>	87
<code>        app_assert_status(sc);</code>	88
<code>        break;</code>	89
<code>    default:</code>	90
<code>        break;</code>	91
<code>    }</code>	92
<code>  }</code>	93
<code>}</code>	94
<code>}</code>	95
<code>}</code>	96





# C Zdrojový kód aplikace

## Výpis C.1: Zdrojový kód aplikace - 1. část *ActivityMain*

```
import android.animation.ObjectAnimator;
import android.bluetooth.BluetoothAdapter;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.support.annotation.Nullable;
import android.support.design.widget.AppBarLayout;
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.os.Bundle;
import android.support.design.widget.TabLayout;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.content.ContextCompat;
import android.support.v4.view.ViewPager;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.TextView;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class ActivityMain extends ActivityDrawer implements FragmentScanner.OnScanner
    ActionDelegate {

    private static final Logger sLogger = LoggerFactory.getLogger(ActivityMain.class);

    enum Feature {broadcast, scan}

    private static final String EXTRA_FEATURE = "EXTRA_FEATURE";

    private ViewPager mViewPager;
    private ViewPagerAdapter mViewPagerAdapter;
    private ObjectAnimator mFabAnimator;

    private FloatingActionButton mSharedFab;
    private FloatingActionButton fndBTN;
    private ImageView redCircle, greenCircle, blueCircle, beacPic1, beacPic2, beacPic3;
    public AppBarLayout mainBar;
    public TextView mDescriptionView, beac1TxtRssi, beac2TxtRssi, beac3TxtRssi; // Beac text

    public int beaconsNeeded = 2; // Beacons needed for localization
    public int numberOfClicks = 1; // Mouse clicks

    public int[] rssiBeaconsValues = new int[3];
    public double[] distance = new double[3];

    static double powerBG22 = -40.0; // EFR32BG22 has 6 dBm
    static double constEnvir = 4.0; // n constant <2-4>

    private final BroadcastReceiver mBroadcastReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            switch (intent.getAction()) {
                case BluetoothAdapter.ACTION_STATE_CHANGED: {
                    final int btState = intent.getIntExtra(BluetoothAdapter.EXTRA_STATE, -1);
                    switch (btState) {
                        case BluetoothAdapter.STATE_TURNING_OFF:
                            final FragmentScanner fragmentScanner = mViewPagerAdapter.
                                getFragmentScanner();
                            if (fragmentScanner != null) {
                                fragmentScanner.stopBeaconScan();
                            }
                            break;
                    }
                }
            }
        }
    }
}
```

## Výpis C.2: Zdrojový kód aplikace - 2. část *ActivityMain*

```

    }
};

/* MY FNCT */
public void setBeaconsCompON(){
    redCircle.setVisibility(View.VISIBLE);
    greenCircle.setVisibility(View.VISIBLE);
    blueCircle.setVisibility(View.VISIBLE);

    beacPic1.setVisibility(View.VISIBLE);
    beacPic2.setVisibility(View.VISIBLE);
    beacPic3.setVisibility(View.VISIBLE);

    beac1TxtRssi.setVisibility(View.VISIBLE);
    beac2TxtRssi.setVisibility(View.VISIBLE);
    beac3TxtRssi.setVisibility(View.VISIBLE);

    mainBar.setVisibility(View.INVISIBLE);
    mSharedFab.setVisibility(View.INVISIBLE);
}

public void setBeaconsCompOFF(){
    redCircle.setVisibility(View.INVISIBLE);
    greenCircle.setVisibility(View.INVISIBLE);
    blueCircle.setVisibility(View.INVISIBLE);

    beacPic1.setVisibility(View.INVISIBLE);
    beacPic2.setVisibility(View.INVISIBLE);
    beacPic3.setVisibility(View.INVISIBLE);

    beac1TxtRssi.setVisibility(View.INVISIBLE);
    beac2TxtRssi.setVisibility(View.INVISIBLE);
    beac3TxtRssi.setVisibility(View.INVISIBLE);

    mainBar.setVisibility(View.VISIBLE);
    mSharedFab.setVisibility(View.VISIBLE);
}

public double[] calculateDistance(int[] vectorValues){ // Distance calc (meter) from RSSI
    int a;
    double[] calculated = new double[4];
    for(a=0; a<beaconsNeeded; a++){
        // 10^( (MeasuredPower - RSSI) / (10 * constant{ <2,4> }) + round to TENTHS
        calculated[a] = Math.round(Math.pow(10.0, (powerBG22 -
            vectorValues[a] / (10.0 * constEnvir) ) * 10.0) / 10.0);
    }
    return calculated;
}

// Returns dp from meters - must be whole number
// 1dp == 0.025m
// 1dp == 0.1m
public int getDpFromMeters(double metersValue){
    return (int) Math.round(metersValue / 0.1);
}

// Converting pxl to dp, by density
public float convertPxToDp(Context context, int px) {
    return Math.round(px * context.getResources().getDisplayMetrics().density);
}

public void setBeaconSize(int beaconNumber, int beaconSize){
    int xLeft = 0, yTop = 0;
    switch (beaconNumber){
        case 1: //1 = redCircle; DEFAULT POINT: x=80, y=140
            // Adjust the adjusted center coordinates
            xLeft = (int) convertPxToDp(this, 80 - Math.round(beaconSize / 2));
            // with px to dp conversion
            yTop = (int) convertPxToDp(this, 140 - Math.round(beaconSize / 2));
            ImageView RC = findViewById(R.id.circle_1R);
            // Specific beacon
            FrameLayout.LayoutParams layParRedC = new FrameLayout.LayoutParams(
                RC.getLayoutParams());
            // Find LayoutParam
            // Conversion of px to dp and setting parameter
            layParRedC.setMargins(xLeft, yTop, 0, 0);
    }
}

```

75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150

### Výpis C.3: Zdrojový kód aplikace - 3. část *ActivityMain*

```

RC.setLayoutParams (layParRedC);
layParRedC.width = (int) convertPxToDp (this , beaconSize);
layParRedC.height = (int) convertPxToDp (this , beaconSize);
break;

case 2: //2 = greenCircle; DEFAULT POINT: x=280, y=140
xLeft = (int) convertPxToDp (this ,280 - Math.round (beaconSize /2));
yTop = (int) convertPxToDp (this ,140 - Math.round (beaconSize /2));
ImageView GC = findViewById (R.id.circle_2G);
FrameLayout.LayoutParams layParGreenC = new FrameLayout.LayoutParams(
GC.getLayoutParams ());

layParGreenC.setMargins ( xLeft , yTop , 0 , 0);

GC.setLayoutParams (layParGreenC);
layParGreenC.width = (int) convertPxToDp (this , beaconSize);
layParGreenC.height = (int) convertPxToDp (this , beaconSize);
break;

case 3: //3 = blueCircle; DEFAULT POINT: x=180, y=273
xLeft = (int) convertPxToDp (this ,180 - Math.round (beaconSize /2));
yTop = (int) convertPxToDp (this ,273 - Math.round (beaconSize /2));
ImageView BC = findViewById (R.id.circle_3B);
FrameLayout.LayoutParams layParBlueC = new FrameLayout.LayoutParams (
BC.getLayoutParams ());

layParBlueC.setMargins ( xLeft , yTop , 0 , 0);

BC.setLayoutParams (layParBlueC);
layParBlueC.width = (int) convertPxToDp (this , beaconSize);
layParBlueC.height = (int) convertPxToDp (this , beaconSize);
break;

default:
throw new IndexOutOfBoundsException (String.format ("Wrong_number. "));
}
}

public void setBeaconTextValue (int beaconTxtNumber , double beaconTxtMeters ,
double beaconRssiValue){
switch (beaconTxtNumber){
case 1:
beac1TxtRssi.setText ("Beac1:" + beaconRssiValue + "dBm;□~" +
+ beaconTxtMeters + "μm");
break;
case 2:
beac2TxtRssi.setText ("Beac2:" + beaconRssiValue + "dBm;□~" +
+ beaconTxtMeters + "μm");
break;
case 3:
beac3TxtRssi.setText ("Beac3:" + beaconRssiValue + "dBm;□~" +
+ beaconTxtMeters + "μm");
break;
default:
throw new IndexOutOfBoundsException (String.format ("No_number. "));
}
}
/* ----- */

// Startup functions
public static void displayActivityFeature (Context context , Feature feature) {
final Intent activityIntent = new Intent (context , ActivityMain.class);
activityIntent.putExtra (EXTRA_FEATURE , feature);
context.startActivity (activityIntent);
}

@Override
protected void onNewIntent (Intent intent) {
if (intent == null || intent.getExtras () == null) {
return;
}
Bundle extras = intent.getExtras ();
Feature feature = (Feature)extras.getSerializable (EXTRA_FEATURE);
if (feature == null) {
return;
}
mViewPager.setCurrentItem (ViewPagerAdapter.PAGE_INDEX_SCANNER);
}

```

151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228

## Výpis C.4: Zdrojový kód aplikace - 4. část *ActivityMain*

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    sLogger.debug("onCreate()");

    setContentView(R.layout.activity_main);

    mSharedFab = (FloatingActionButton) findViewById(R.id.main_fab_shared);
    mSharedFab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            switch (mViewPager.getCurrentItem()) {
                case ViewPagerAdapter.PAGE_INDEX_SCANNER:
                    final FragmentScanner fragmentScanner = mViewPagerAdapter.
                        getFragmentScanner();

                    if (fragmentScanner != null) {
                        fragmentScanner.actionScanToggle();
                        numberOfClicks = 1; // Is Scanning running check
                    }
                    break;
                default:
                    throw new IndexOutOfBoundsException(String.format
                        ("Cannot have more than 1 item"));
            }
        }
    });

    // FUNCTIONS OF FIND LOC BTN
    // Setting up beacons
    redCircle = (ImageView) findViewById(R.id.circle_1R);
    greenCircle = (ImageView) findViewById(R.id.circle_2G);
    blueCircle = (ImageView) findViewById(R.id.circle_3B);

    beacPic1 = (ImageView) findViewById(R.id.beaconPic1);
    beacPic2 = (ImageView) findViewById(R.id.beaconPic2);
    beacPic3 = (ImageView) findViewById(R.id.beaconPic3);

    beac1TxtRssi = (TextView) findViewById(R.id.txt01);
    beac2TxtRssi = (TextView) findViewById(R.id.txt02);
    beac3TxtRssi = (TextView) findViewById(R.id.txt03);

    // Setting up interface
    mainBar = (AppBarLayout) findViewById(R.id.mainBarLayout);
    mDescriptionView = (TextView) findViewById(R.id.scanner_textview_description);

    // Setting up btn
    fndBTN = (FloatingActionButton) findViewById(R.id.findLocBTN);
    fndBTN.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            final FragmentScanner fragmentScanner = mViewPagerAdapter.getFragmentScanner();
            // Is it possible to start calculating
            if(fragmentScanner.howManyBeacons() == beaconsNeeded){
                if(numberOfClicks % 2 == 1) {
                    // Stopping scanning
                    fragmentScanner.stopBeaconScan();
                    // Changing picture on btn
                    fndBTN.setImageResource(R.drawable.ic_menu_broadcast_off);
                    setBeaconsCompON(); // Setting up beacon circles
                    fragmentScanner.hideAllFoundBeacons(); // Hide all beacons
                    numberOfClicks = 2; // Btn was clicked
                    // Set to 0, for not to return value
                    fragmentScanner.countBG22 = 0;

                    // Get RSSI values and store for later use
                    rssiBeaconsValues = fragmentScanner.returnRSSIValue();

                    // Calculate distance from the formula
                    distance = calculateDistance(rssiBeaconsValues);

                    // Adjust the values for display
                    // Change beacons sizes and setup the right position
                    setBeaconSize(1, getDpFromMeters(distance[0])); // 1=RC,2=GC,3=BC
                    setBeaconSize(2, getDpFromMeters(distance[1]));
                    //setBeaconSize(3, getDpFromMeters(distance[2]));
                }
            }
        }
    });
}

```

## Výpis C.5: Zdrojový kód aplikace - 5. část *ActivityMain*

```

        // Setup the text for lower part of the screen: RSSI, distance
        setBeaconTextValue(1, distance [0], rssiBeaconsValues [0]);
        setBeaconTextValue(2, distance [1], rssiBeaconsValues [1]);
        //setBeaconTextValue(3, distance [2], rssiBeaconsValues [2]);
    }
    else{
        setBeaconsCompOFF (); // Hide beacons
        // Change btn picture
        findBTN.setImageResource(R.drawable.ic_menu_broadcast_on);
        fragmentScanner.clearAllBeacons (); // Setup the default view
    }
} else{
    // When not enough beacons, show how many
    fragmentScanner.notEnoughBeacons ();
}
});

Toolbar toolbar = (Toolbar) findViewById(R.id.main_toolbar);
setSupportActionBar(toolbar);

mViewPager = (ViewPager) findViewById(R.id.main_viewpager);
mViewPagerAdapter = new ViewPagerAdapter(getSupportFragmentManager(), mViewPager, this);

mViewPager.setAdapter(mViewPagerAdapter);
mViewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
    @Override
    public void onPageScrolled(int position, float positionOffset,
                               int positionOffsetPixels) { }
    @Override
    public void onPageSelected(int position) {
        final FragmentScanner fragmentScanner = mViewPagerAdapter.getFragmentScanner();
        if (fragmentScanner == null) {
            return;
        }
        switch (position) {
            case ViewPagerAdapter.PAGE_INDEX_SCANNER:
                selectNavigationItem (ITEM_SCAN);
                break;
            case -1:
            default:
                selectNavigationItem (ITEM_NONE);
        }
    }
}
@Override
public void onPageScrollStateChanged(int state) {
    switch (state) {
        case ViewPager.SCROLL_STATE_DRAGGING:
            mSharedFab.hide ();
            break;
        case ViewPager.SCROLL_STATE_IDLE:
            switch (mViewPager.getCurrentItem()) {
                case ViewPagerAdapter.PAGE_INDEX_SCANNER:
                    break;
                default:
                    mSharedFab.setImageResource(R.drawable.
                                                    ic_menu_search);
                    break;
            }
            mSharedFab.show ();
            break;
    }
}
});

TabLayout tabLayout = (TabLayout) findViewById(R.id.main_tablayout);
tabLayout.setupWithViewPager(mViewPager);
tabLayout.getTabAt(ViewPagerAdapter.PAGE_INDEX_SCANNER).setIcon(R.drawable.
                                                                    ic_menu_search);

initNavigationDrawer(findViewById(android.R.id.content), toolbar);
onNewIntent(getIntent());
}

```

## Výpis C.6: Zdrojový kód aplikace - 6. část *ActivityMain*

```

@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    // This method ensure, that when internal change is made, the view is loaded the right way
    mSharedFab.setImageResource(R.drawable.ic_menu_search);
    final FragmentScanner fragmentScanner = mViewPagerAdapter.getFragmentScanner();
    if (fragmentScanner != null) {
        mViewPagerAdapter().setPrimaryItem(null, mViewPagerAdapter.PAGE_INDEX_SCANNER,
            fragmentScanner);
    }
}

@Override
protected void onStart() {
    super.onStart();
    sLogger.debug("onStart()");
    registerReceiver(mBroadcastReceiver, new IntentFilter(BluetoothAdapter.
        ACTION_STATE_CHANGED));
}

@Override
public void onResume() {
    super.onResume();
    sLogger.debug("onResume()");
}

@Override
protected void onStop() {
    super.onStop();
    sLogger.debug("onStop()");
    unregisterReceiver(mBroadcastReceiver);
}

@Override
public void onBackPressed() {
    super.onBackPressed();
}

public void onScanStatusUpdate(boolean isScanning) {
    if (mViewPagerAdapter.getCurrentItem() != mViewPagerAdapter.PAGE_INDEX_SCANNER) {
        return;
    }
    if (isScanning) { // Show the button as blinking - for visual reference of scanning
        if (mSharedFab != null) mSharedFab.setImageResource(R.drawable.ic_menu_pause);
        int whiteColor = 0xFFFFFFFF;
        int accentColor = ContextCompat.getColor(this, R.color.colorAccent);
        mFabAnimator = ObjectAnimator.ofArgb(mSharedFab.getDrawable(), "tint", accentColor,
            whiteColor);
        mFabAnimator.setDuration(500);
        mFabAnimator.setRepeatCount(ObjectAnimator.INFINITE);
        mFabAnimator.setRepeatMode(ObjectAnimator.REVERSE);
        mFabAnimator.start();
    }
    else {
        if (mSharedFab != null) mSharedFab.setImageResource(R.drawable.ic_menu_search);
        if (mFabAnimator != null) {
            mFabAnimator.setRepeatCount(0);
            mFabAnimator = null;
        }
    }
}

// Creater implemented for better handling
// http://stackoverflow.com/questions/8785221/retrieve-a-fragment-from-a-viewpager
// Trick of retrieving fragment by name to be sure it is directly retrieved from
// the fragment manager. Usefull to be used with the trick in onRestoreInstanceState and
// avoid the call to setUserVisibleHint. It we override intantiateItem, the trick to
// avoid the call to setUserVisibleHint does not work anymore.
private static class ViewPagerAdapter extends FragmentPagerAdapter {

    private final static int PAGE_COUNT = 1;
    private final static int PAGE_INDEX_SCANNER = 0;

    private final int _containerId;
    private final Context _context;
    private final FragmentManager _fragmentManager;

```

## Výpis C.7: Zdrojový kód aplikace - 7. část *ActivityMain*

```

public ViewPagerAdapter(FragmentManager manager, ViewPager container, Context context) { 453
    super(manager); 454
    _containerId = container.getId(); 455
    _context = context; 456
    _fragmentManager = manager; 457
} 458

@Nullable 459
public Fragment getRegisteredFragment(int position) { 460
    return _fragmentManager.findFragmentByTag(getFragmentTag(_containerId, position)); 461
} 462

@Nullable 463
public FragmentScanner getFragmentScanner() { 464
    return (FragmentScanner) getRegisteredFragment(PAGE_INDEX_SCANNER); 465
} 466

@Override 467
public Fragment getItem(int position) { 468
    switch (position) { 469
        case PAGE_INDEX_SCANNER: 470
            return new FragmentScanner(); 471
        default: 472
            throw new IndexOutOfBoundsException(String.format 473
                ("Cannot have more than 1 item")); 474
    } 475
} 476

@Override 477
public int getCount() { 478
    return PAGE_COUNT; 479
} 480

@Override 481
public CharSequence getPageTitle(int position) { 482
    switch (position) { 483
        case PAGE_INDEX_SCANNER: 484
            return _context.getString(R.string.main_tab_scanner); 485
        default: 486
            throw new IndexOutOfBoundsException(String.format 487
                ("Cannot have more than 1 item")); 488
    } 489
} 490

private String getFragmentTag(int viewPagerId, int fragmentPosition) { 491
    return "android:switcher:" + viewPagerId + ":" + fragmentPosition; 492
} 493
} 494
} 495
} 496
} 497
} 498
} 499
} 500
} 501
} 502
} 503

```