



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**LOKALIZACE OBJEKTŮ V PROSTŘEDÍ BEZDRÁTOVÉ
SENZOROVÉ SÍTĚ**

LOCALIZATION OF OBJECTS IN WIRELESS SENSOR NETWORK ENVIRONMENT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ VÍCHA

VEDOUcí PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2021

Zadání diplomové práce



Student: **Vícha Tomáš, Bc.**
Program: Informační technologie
Obor: Počítačové a vestavěné systémy
Název: **Lokalizace objektů v prostředí bezdrátové sensorové sítě**
Localization of Objects in Wireless Sensor Network Environment
Kategorie: Vestavěné systémy
Zadání:

1. Prostudujte bezdrátovou komunikaci ve volně dostupném pásmu ISM. Pozornost zaměřte zejména na standard IEEE 802.15.4/ZigBee/Bluetooth Mesh a související.
2. Připravte přehledovou studii o možnostech a technikách lokalizace objektů v bezdrátové sensorové síti.
3. S ohledem na zjištěné skutečnosti zvolte jednu ze zkoumaných technologií, na kterou se následně zaměříte. Dále navrhnete detailní koncepci systému pro určování polohy objektů.
4. Vyberte vhodné komponenty a v návrhovém systému připravte schéma zapojení sensorových lokalizačních modulů. Následně realizujte desky plošných spojů v potřebném množství s ohledem na zvolenou techniku lokalizace.
5. Implementujte firmware nezbytný ke zprovoznění jednotlivých modulů a vytvořte jednoduchou uživatelskou aplikaci pro PC.
6. Proveďte důkladné otestování navrženého systému v reálných podmínkách.
7. Zhodnoťte dosažené výsledky a zvažte možnosti dalšího rozšíření.

Literatura:

- Dle pokynů vedoucích.

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

Vedoucí práce: **Šimek Václav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 19. května 2021
Datum schválení: 30. října 2020

Abstrakt

Tato práce se zabývá využitím metod pro získávání prostorového povědomí v rámci technologií určených pro tvorbu bezdrátových senzorových sítí krátkého dosahu. Práce obsahuje rozbor několika technik, které je možné použít pro získání odhadu pozice objektů v rámci senzorové sítě. Pro praktické řešení byla vybrána metoda založená na měření časových diferencí zasílaných zpráv. Byla realizována obvodová implementace uzlu sítě stavějícího na čipu DW1000, který pracuje na technologii ultra širokopásmového vysílání. Dále byla realizována i senzorová síť s příslušnou uživatelskou aplikací na její obsluhu a zobrazování výsledků lokalizace.

Abstract

This thesis deals with the use of spatial awareness methods within technologies designed for creation of short-range wireless sensor networks. The thesis analyzes several techniques that can be used to estimate position of objects within the sensor network. For a practical solution, a method based on measuring the time differences of the sent messages was chosen. A circuit implementation of a network node based on the DW1000 chip, which works on ultra-wideband transmission technology, was implemented. A sensor network with the appropriate user application for its operation and display of localization results was also implemented.

Klíčová slova

TDoA, UWB, DWM1000, Lokalizace uzlu, KW41Z, ZigBee

Keywords

TDoA, UWB, DWM1000, Node localization, KW41Z, ZigBee

Citace

VÍCHA, Tomáš. *Lokalizace objektů v prostředí bezdrátové senzorové sítě*. Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Lokalizace objektů v prostředí bezdrátové senzorové sítě

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Tomáš Vícha
26. května 2021

Poděkování

Rád bych poděkoval vedoucímu této diplomové práce panu Ing. Václavu Šimkovi za odborné vedení a cenné rady při návrhu a vývoji celého lokalizačního konceptu. Můj dík patří i mé rodině a přátelům, kteří mi byli při tvorbě práce velkou oporou a poskytli svůj čas pro její korekturu.

Obsah

1	Úvod	3
2	Bezdrátové komunikační technologie	4
2.1	Bezdrátová komunikace ve volně dostupném pásmu ISM	4
2.2	Standard IEEE 802.15.4	5
2.2.1	Fyzická vrstva	6
2.2.2	Linková vrstva	6
2.2.3	Typy uzlů	7
2.2.4	Síťová topologie	8
2.3	ZigBee	8
2.3.1	Síťová vrstva	9
2.3.2	Aplikační vrstva	10
2.3.3	Formování sítě	11
2.3.4	Připojení zařízení do sítě	11
2.3.5	Adresace zařízení	11
2.3.6	Zasílání paketů	12
2.4	Bluetooth	12
2.4.1	Zařízení a komunikace	13
2.4.2	Protokolový zásobník Bluetooth Classic	13
2.4.3	Protokolový zásobník Bluetooth Low Energy	14
2.4.4	Komunikace a výměna dat s Bluetooth Low Energy	15
2.5	UWB	16
2.5.1	IEEE 802.15.4-2011	17
3	Lokalizace objektů v bezdrátové senzorové síti	19
3.1	Metody založené na změřeném dosahu	19
3.1.1	Měření zpoždění (ToF)	19
3.1.2	Měření diference zpoždění (TDoA)	21
3.1.3	Měření síly signálu (RSSI)	21
3.1.4	Měření úhlu (AoA)	22
3.2	Metody bez změřeného dosahu	23
3.2.1	APIT	23
3.2.2	DV-Hop	24
3.3	Metody pro určování poloh uzlů v sítích UWB	24
3.3.1	Two-way ranging	24
3.3.2	TDoA	25
4	Návrh systému na určování polohy objektů	27

4.1	Senzorová síť	27
4.2	Komunikace v rámci senzorové sítě	28
4.2.1	Lokalizace a synchronizace času	29
4.2.2	Řízení a předávání dat	29
4.3	Identifikace uzlů v síti	30
4.4	Synchronizace času mezi kotevními uzly	31
4.5	Senzorový uzel	31
4.5.1	MCU Kinetis KW41Z	32
4.5.2	Modul DWM1000	33
4.6	Aplikace	33
5	Obvodová realizace senzorového uzlu	36
5.1	Regulace napětí	36
5.2	Nabíjení akumulátoru	37
5.3	UWB modul	39
5.4	mikrokontrolér a RF část	40
5.5	Výsledná deska plošných spojů a její parametry	42
6	Uživatelská aplikace a obslužný firmware	45
6.1	Firmware senzorového uzlu	45
6.1.1	Obsluha UWB modulu	46
6.1.2	Tag	48
6.1.3	Kotevní uzel	48
6.1.4	Gateway uzel	50
6.2	Uživatelská aplikace	51
6.2.1	Obsluha rozhraní Bluetooth Low Energy	51
6.2.2	Zpracování a tvorba zpráv pro rozhraní Bluetooth Low Energy	52
6.2.3	Hlavní logika pro určování pozic uzlů	52
6.2.4	Zobrazování výsledků	53
7	Závěr	55
	Literatura	57

Kapitola 1

Úvod

Senzorové sítě jsou v dnešním propojeném světě založeném na neustálém toku dat perspektivní oblastí, která nalézá uplatnění snad ve všech směrech lidské činnosti. Využívají se pro sledování prostředí, ve kterém se nacházíme. Pomáhají předpovídat erupce sopek nebo počasí. Sledují změny globálního klimatu nebo monitorují náš domov či pracoviště. Pro většinu případů je vhodné znát umístění sensorových uzlů, abychom mohli naměřené hodnoty správně interpretovat. Pokud jsou uzly stacionární, lze si jejich pozici poznačit. Pokud jsou ovšem senzory v pohybu, nastává problém, který je potřeba řešit. Dobrým příkladem oblasti, ve které může docházet k neustálému pohybu sensorových uzlů je logistika. Využitím malého zařízení připevněného na paletu se zbožím lze nejen monitorovat jeho teplotu a vlhkost, ale také sledovat pohyb palet v rámci skladu.

Tato práce se zabývá právě možnostmi lokalizace uzlů v rámci sensorové sítě. V první části je rozebrána existující technologie pro tvorbu bezdrátových sensorových sítí, a to konkrétně v bezlicenčním pásmu ISM. Podrobně je popsán standard IEEE 802.15.4 a technologie ZigBee, která na tomto standardu staví. Dále je uvedeno i bezdrátové rozhraní Bluetooth nebo technologie ultra širokopásmové komunikace. V další kapitole je věnována pozornost podrobnému rozebrání existujících technik, které lze v rámci bezdrátových komunikačních technologií využít pro lokalizaci sensorových uzlů. Následně je pozornost zaměřena na techniky, které využívají výhod šíření ultra širokopásmové komunikace.

Po teoretickém rozboru následuje kapitola věnovaná návrhu vlastní sítě, ve které bude docházet k určování pozice hledaných uzlů. Jsou zde uvedeny podrobné vlastnosti komunikace v rámci sensorové sítě a také komponenty, ze kterých bude realizován uzel pro danou síť. Následující kapitola se věnuje právě realizaci samotného hardware sensorového uzlu, jeho zapojení a napájení. Tato část je důležitá pro poslední kapitolu, která se věnuje implementaci firmware pro vytvoření sensorový uzel a také tvorbě uživatelské aplikace, která provádí ovládání celé sítě a stará se o výpočty lokalizace sensorových uzlů, jejichž pozice následně graficky reprezentuje.

Kapitola 2

Bezdrátové komunikační technologie

Tato kapitola shrnuje získané poznatky z oblasti bezdrátové komunikace ve volně dostupném pásmu ISM. Kmitočty patřící do tohoto pásma jsou důležité pro vědecké, lékařské a průmyslové aplikace. Mimo to se s nimi setkáváme i v běžném životě, a to jak komerčně, tak i soukromě. Svět je doslova zaplaven rádiovým signálem šířícím se v tomto bezlicenčním pásmu. Můžeme si díky tomu uvařit jídlo nebo pohodlně brouzdat po internetu bez omezujících kabelů.

V první části této kapitoly jsou popsány oblasti použití volně dostupných pásem společně s výčtem jednotlivých frekvenčních rozsahů použitelných na území České republiky. V další části této kapitoly je pozornost věnována technologiím, díky kterým lze vytvářet bezdrátová spojení v rámci sensorových sítí. Podrobně je rozebrán standard IEEE 802.15.4, jež je základním kamenem pro další technologie. Jednou z těchto technologií je ZigBee s níž se můžeme setkávat například v prvcích chytré domácnosti. Další bezdrátovou technologií, která patří do rodiny standardů IEEE 802.15, je Bluetooth. Ten je znám hlavně z prostředí mobilních a chytrých telefonů, ovšem nic nebrání jeho použití i v sensorových sítích. Jako poslední je pospána širokopásmová bezdrátová technologie UWB, již se v poslední době dostává velké pozornosti. Skalním fanouškům produktů společnosti Apple jistě neuniklo nedávné představení AirTagu, který demonstruje potenciál této technologie.

Všechny uvedené standardy pro bezdrátovou komunikaci mohou sloužit jak pro tvorbu sensorových sítí, tak i pro určování přibližné polohy jejich uzlů. Tyto dva případy použití se u každé jednotlivé technologie vzájemně nevyklučují, naopak určování polohy uzlů může být čistě jenom doplňkem komunikace v rámci sítě.

2.1 Bezdrátová komunikace ve volně dostupném pásmu ISM

ISM (z anglického industrial, scientific and medical) je soubor pásem pro bezdrátové vysílání primárně určených pro průmyslové, vědecké a lékařské aplikace. Jedná se o sadu volně dostupných kmitočtů, na kterých lze provozovat bezdrátovou komunikaci bez potřeby vlastnění licence. Protože jsou tato pásma primárně určena pro výše uvedené obory, je nutné na rozdíl od licencovaných pásem počítat s rušením. Interference můžou totiž způsobovat aplikace jako je například radiofrekvenční ohřev či bezdrátový přenos energie. Pro běžného smrtníka je nelepším příkladem rušení mikrovlnná trouba pracující právě v pásmu ISM.

Mikrovlnná trouba operuje na frekvenci 2.45 GHz a její vysílání může tedy interferovat komunikaci standardů Wi-Fi či Bluetooth.

Jednotlivá pásma i s jejich rozsahy jsou definovány Radiokomunikačním úřadem ITU (Mezinárodní telekomunikační unie), který dále upravuje či omezuje použití daných pásem. Díky rozdělení světa na tři regiony jsou některé frekvence dostupné pouze pro určitý region. Příkladem může být frekvenční rozsah 902-928 MHz dostupný pouze pro druhý region (zahrnující Ameriku a Grónsko). Rozdělení frekvenčních pásem ISM dostupných v České republice se řídí všeobecným oprávněním č. VO-R/10/12.2019-9 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu [32]. Rozdělení znázorňuje tabulka 2.1.

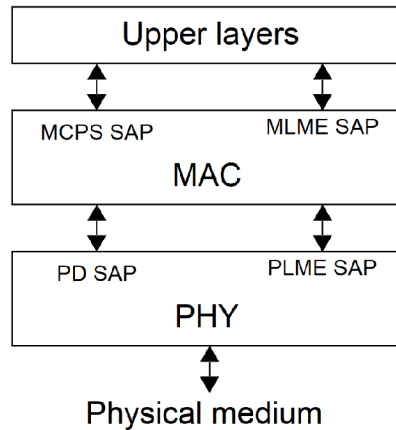
Díky faktu, že na provoz zařízení v pásmech ISM není nutná licence od příslušného telekomunikačního úřadu, stala se tato pásma hodně využívaná i pro tzv. ne ISM aplikace. Mezi tyto aplikace se řadí veškeré bezdrátové přenosy, a to ať už pro komunikaci anebo řízení. Kupříkladu ovládání RC modelů (RC z anglického radio controlled) operujících na frekvencích 27 MHz nebo 40 MHz. Dále to jsou systémy pro radio-frekvenční identifikaci (RFID) využívající kmitočet 13.56 MHz. Poslední a asi největší skupinou jsou zařízení pro komunikaci na krátkou vzdálenost využívající například standardy z rodiny IEEE 802.11. Ty slouží pro tvorbu bezdrátové lokální sítě Wi-Fi, dále Bluetooth nebo standard IEEE 802.15.4, jež je základem pro sítě založené na ZigBee, MiWi či Thread.

Tabulka 2.1: Pásma ISM použitelná v České republice (vychází z [32])

Frekvenční rozsah	Střední frekvence	Šířka pásma
6.765–6.795 MHz	6.78 MHz	30 kHz
13.553–13.567 MHz	13.56 MHz	14 kHz
26.957–27.283 MHz	27.12 MHz	326 kHz
40.66–40.7 MHz	40.68 MHz	40 kHz
433.05–434.79 MHz	433.92 MHz	1.84 MHz
2.4–2.5 GHz	2.45 GHz	100 MHz
5.725–5.875 GHz	5.8 GHz	150 MHz
24–24.25 GHz	24.125 GHz	250 MHz
61–61.5 GHz	61.25 GHz	500 MHz
122–123 GHz	122.5 GHz	1 GHz
244–246 GHz	245 GHz	2 GHz

2.2 Standard IEEE 802.15.4

Obsah této podkapitoly vychází ze standardu IEEE 802.15.4 [10] a také z článku [15]. Tento technický standard je první ze zástupců technologií využívajících bezlicenční pásma ISM. Specifikuje bezdrátové sítě krátkého dosahu s nízkou přenosovou rychlostí a malou spotřebou, tedy LR-WPANs (Low-Rate Wireless Personal Area Networks). Je postaven na OSI modelu a jeho specifikace zahrnuje fyzickou vrstvu (PHY – Physical Layer) a vrstvu linkovou (MAC – Medium Access Control). Vyšší vrstvy z OSI modelu nejsou zahrnuty z důvodů zachování jednoduchosti a univerzálnosti, tak to ilustruje i obrázek 2.1. V případě implementace dalších vrstev již mluvíme o protokolech a sítích ZigBee, MiWi, Thread a dále.



Obrázek 2.1: Vrstvy protokolu 802.15.4 [10]

2.2.1 Fyzická vrstva

Fyzická vrstva, jakožto nejnižší vrstva referenčního modelu, zajišťuje přenos dat prostřednictvím datových rámců a řídí rádiové spoje na fyzické úrovni. Definuje také frekvenční pásma a typy modulace signálu. Standard umožňuje, aby zařízení pracovala na jednom z následujících kmitočtů o daném počtu kanálů:

- 868–868.6 MHz (jeden kanál)
- 902–928 MHz (až deset kanálů)
- 2400–2483.5 MHz (až 16 kanálů)

Maximální propustnost komunikace se poté odvíjí od použité modulace a to od 20 kbps až do 250 kbps. Přenos dat na fyzické vrstvě se následně realizuje pomocí datových rámců PPDU (Physical Packet Data Unit). Každý datový rámec se skládá z následujících tří částí (viz obrázek 2.2):

- SHR (Synchronization Header) – skládající se z Preamble (tvořící první čtyři bajty) sloužící pro synchronizaci komunikace a SFD (pátý bajt) značí konec synchronizační části rámce
- PHR (PHY Header) – obsahuje délku rámce
- PSDU (Physical Service Data Unit) – obsahuje vlastní přenášená data z vyšších vrstev

Důležitou součástí fyzické vrstvy je také detekce energie aktivního kanálu, která se využívá pro výběr přenosového kanálu a také k výpočtu síly přijatého signálu RSSI (Received Signal Strength Indicator). Tuto informaci lze následně využít, jak bude uvedeno v kapitole 3.1.3, pro určování vzdálenosti jednotlivých uzlů mezi sebou a díky tomu k určení polohy jednotlivých uzlů sítě.

2.2.2 Linková vrstva

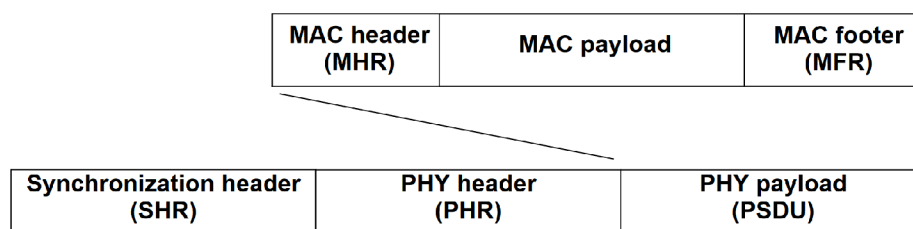
Druhá a také poslední je linková vrstva. Ta zajišťuje přístup ke kanálům, tvorbu a transport rámců linkové vrstvy skrze tyto kanály. Dále se stará o nastavení přenosových parametrů linky, a nakonec poskytuje rozhraní pro vyšší vrstvy stavějících na protokolu 802.15.4.

Rámec linkové vrstvy se nazývá MPDU (Mac Protocol Data Unit) a skládá se z následujících částí (viz obrázek 2.2):

- MHR (MAC Header) – obsahuje kontrolní část specifikující obsah rámce (Frame Control), sekvenční číslo rámce, adresu příjemce a odesílatele rámce, adresu příjemce a odesílatele podsítě a bezpečnostní záhlaví
- MAC Payload – obsahuje data vyšších vrstev (pokud jsou použity), jinak přenášená data
- MFR (MAC Footer) – kontrolní součet pro daný rámec

Rámce linkové vrstvy se dále ještě dělí na pět skupin podle jejich použití na:

- Beacon frame – slouží pro synchronizaci zařízení, která jsou provozována ve stavu nízké spotřeby
- Data frame – používají se pro všechny datové přenosy
- Acknowledgment frame – slouží pro potvrzení úspěšného přijetí rámce
- MAC Command frame – slouží pro nastavování a řízení klientských zařízení v síti
- Multipurpose — pro nesespecifické použití, umožňuje přenos dat s nízkou režii díky možnosti využít pouze jednobajtového pole Frame Control a vynechání ostatních položek z MHR



Obrázek 2.2: Schéma PPDU a MPDU [10]

2.2.3 Typy uzlů

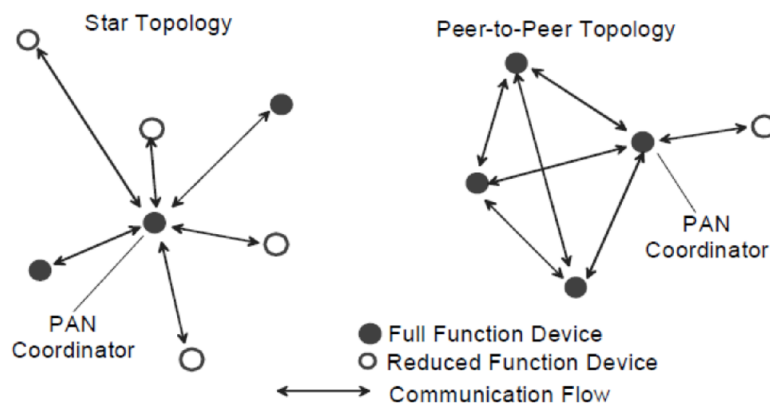
Standard rozděluje uzly na dva typy. První z nich je FFD (full-function device) neboli plně funkční zařízení a druhý typ RFD (reduced-function devices) neboli zařízení s omezenou funkcionalitou. FFD může fungovat ve třech režimech, a to jako koordinátor PAN, koordinátor nebo koncové zařízení. To že je FFD koordinátor znamená, že může komunikovat s ostatními zařízeními v síti, a dokonce může přeposílat mezi jednotlivými zařízeními zprávy. Koordinátor je nutným propojovacím prvkem pro každou část podsítě a může jich být vícero. Naproti tomu, koordinátor PAN může být v rámci sítě pouze jeden a jedná se o hlavní řídicí prvek dané sítě.

Zařízení s omezenou funkcionalitou se v síti chovají jako listy grafu. Musí být vždy propojeny s FFD, bez kterého nejsou schopny mezi sebou komunikovat. Výhodou zařízení s omezenou funkcionalitou je fakt, že nemusí disponovat velkým výpočetním výkonem a mohou na rozdíl od FFD šetřit svými energetickými zdroji. Díky tomu mohou být zařízení levnější a disponovat dlouhou výdrží na baterii.

2.2.4 Síťová topologie

V základu existují pouze dva typy síťových topologií pro standard 802.15.4. První z nich je Star Topology (Hvězda). Jedná se o síť s centralizovanou komunikací přes jeden hlavní bod v síti (PAN koordinátor). Přes tento uzel musí probíhat veškerá komunikace, což vede na velké vytížení daného uzlu.

Druhou topologií je topologie Peer-to-peer (neboli point-to-point), kterou lze rozšířit na obecnou Mesh síť použitím redundantních propojů mezi uzly. Na rozdíl od Star Topology se jedná o decentralizovanou síť obsahující jednoho PAN koordinátora a vícero koordinátorů, kteří slouží jako komunikační mosty mezi ostatními zařízeními. Zprávy už nemusí proudit přes hlavní uzel, ale zátěž se rovnoměrně rozprostře. Nevýhodou této topologie je nutnost použití složitějších směrovacích protokolů. Výše zmíněné topologie jsou ukázány na obrázku 2.4.



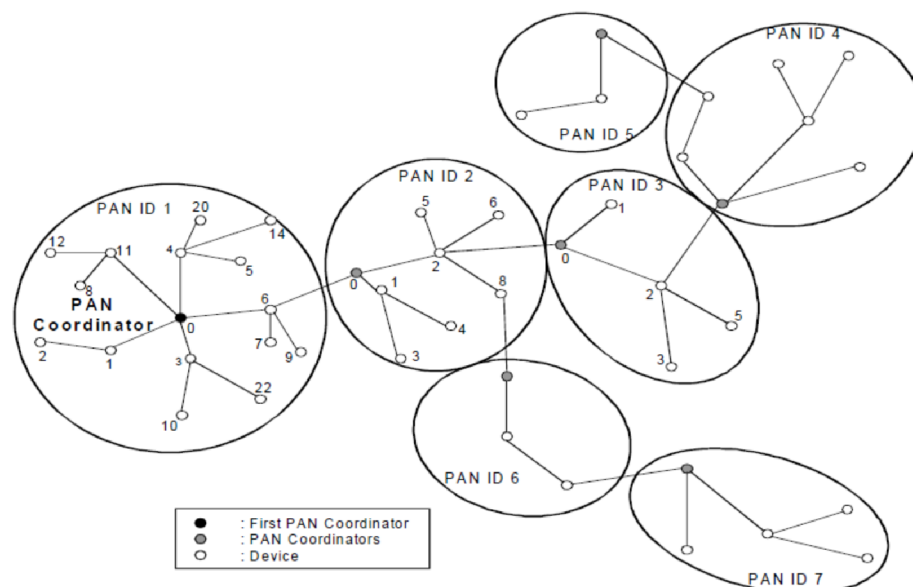
Obrázek 2.3: Ukázka topologie hvězda a peer-to-peer (mesh) [10]

Odvozením od Mesh sítě lze chápat síť s topologií Cluster-Tree (obrázek 2.4). Jedná se o topologii, ve které se nachází převážně zařízení FFD. Jeden hlavní PAN koordinátor pověřuje některá FFD, aby se stala PAN koordinátory pro svou vlastní podsíť. Tímto vznikne několik navzájem nezávislých podsítí, které mezi sebou komunikují pomocí svých PAN koordinátorů. Díky tomu lze vytvářet rozsáhlé sítě, protože se tímto uspořádáním snižuje náročnost směrovacích protokolů. Nevýhodou může být větší latence přenášených zpráv.

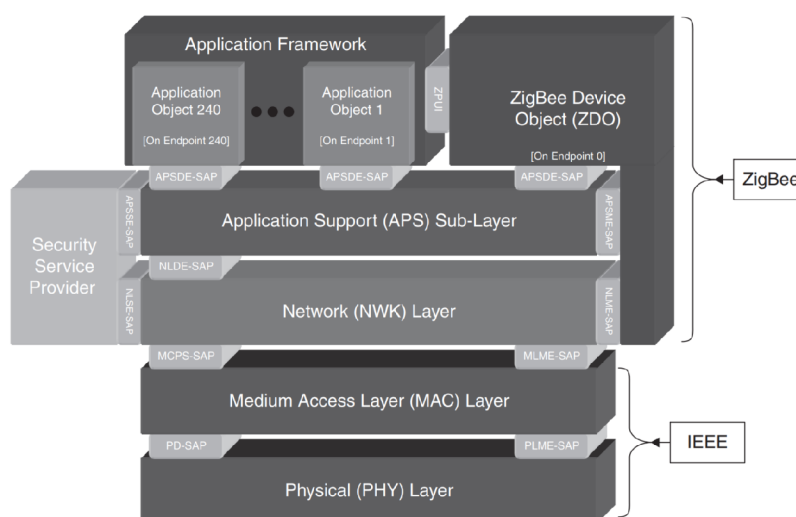
2.3 ZigBee

Popis standardu ZigBee v této podkapitole čerpá z knihy Zigbee wireless networking [11]. Standard je zastřešován a vyvíjen organizací ZigBee Alliance. Je postaven na IEEE 802.15.4 a dále jej rozšiřuje, a to zejména o dvě nové vrstvy. První z nich je vrstva síťová (NWK – Network layer) a druhá je vrstva aplikační (APL – Application layer). Celý protokolový zásobník ilustruje obrázek 2.5.

Zigbee umožňuje tvorbu rozsáhlých mesh sítí, a to až do teoretické velikosti 65 000 uzlů. Zařízení primárně fungují na frekvenci 2.4 GHz, kde mají k dispozici 16 kanálů o šířce 2 MHz. Teoretická propustnost bez režie enkapsulace je pak 250 kbps. Vzdálenost, na kterou mohou zařízení mezi sebou komunikovat, se ve vnitřním prostředí pohybuje od 10 m do 100 m, a to v závislosti na překážkách jako jsou stěny a podobně. V přímém dohledu to



Obrázek 2.4: Ukázka topologie cluster-tree [10]



Obrázek 2.5: Protokolový zásobní pro standard ZigBee [11]

může být i více jak 300 m. Protože je ZigBee postaveno na standardu IEEE 802.15.4, můžou zařízení využívat krom pásma 2.4 GHz i pásmo 868 MHz nebo 915 MHz.

2.3.1 Síťová vrstva

Síťová vrstva zodpovídá za správu sítě, směrování a udržování si směrovacích informací. Dále se stará o zakládání samotné sítě a připojování či odpojování zařízení z nebo do této sítě. Správu zařízení provádí ZigBee koordinátor, známý již z IEEE 802.15.4 jako PAN koordinátor, který přiděluje novým zařízením adresy. Ze standardu IEEE 802.15.4 jsou dále přebrány i další dvě možné variace zařízení a to FFD, které se v ZigBee síti nazývá ZigBee router a ZigBee End Device, které zastupuje RFD tak i FFD.

Pakety síťové vrstvy se označují jako NPDU (Network Protocol Data Unit) a obsahují jak povinnou část, tak i část, která je volitelná. Obecně se paket na síťové vrstvě skládá z následujících položek:

- Frame control – zde se definuje, jestli se jedná o datový nebo příkazový paket, dále tato část obsahuje informace o adresaci a sekvencování
- Destination address a source address – 16bitová adresa zařízení
- Radius – obdoba TTL u IP paketu, hodnota, která se dekrementuje při každém přijetí paketu směřovanému k cílovému zařízení
- Sequence number – určuje pořadí paketu a je inkrementováno s každým novým odeslaným paketem

Mezi nepovinné položky paketu pak patří 64bitová IEEE adresa zdroje a cíle, ta slouží pro detekci konfliktů adres v síti. Další nepovinnou položkou je Multicast control, který udává, zdali má být paket směřován skrze zařízení nepatřící do dané multicastové skupiny. A nakonec může paket ještě obsahovat položku Source route subframe a přenášená data jako Frame payload.

Jak již bylo zmíněno, síťová vrstva se stará i o směřování přenášených paketů v rámci sítě. Výběr směrovacího algoritmu závisí na topologii. Pro stromovou topologii lze použít primitivní Tree routing algoritmus, který spočívá v ustavení sítě ZigBee koordinátorem a zjištěním důležitých parametrů jako je počet směrovačů a hloubka zanoření sítě. Díky tomu se vytvoří mapa sítě a následně mohou být zařízeními přidělovány adresy na základě jejich pozice ve stromě. Samotné směřování vychází ze znalosti polohy cílového uzlu ve stromu. Algoritmus Tree routing již ovšem nelze použít na síť s mesh topologií. U těchto sítí je nutno využít algoritmus Router Discovery. Jedná se o algoritmus typu distance-vector, tudíž přesná topologie sítě není nikdy známá. Funguje na principu všesměrového vysílání příkazových rámců typu Route Request a posílání odpovědí příkazovými rámci typu Route Replay. Jeho výhodou je, že velice rychle reaguje na změny v síti, a to jak na přidání nového uzlu, tak i na odebrání či selhání libovolného uzlu v síti.

2.3.2 Aplikační vrstva

Aplikační vrstva se ve specifikaci ZigBee skládá ze tří hlavních komponent:

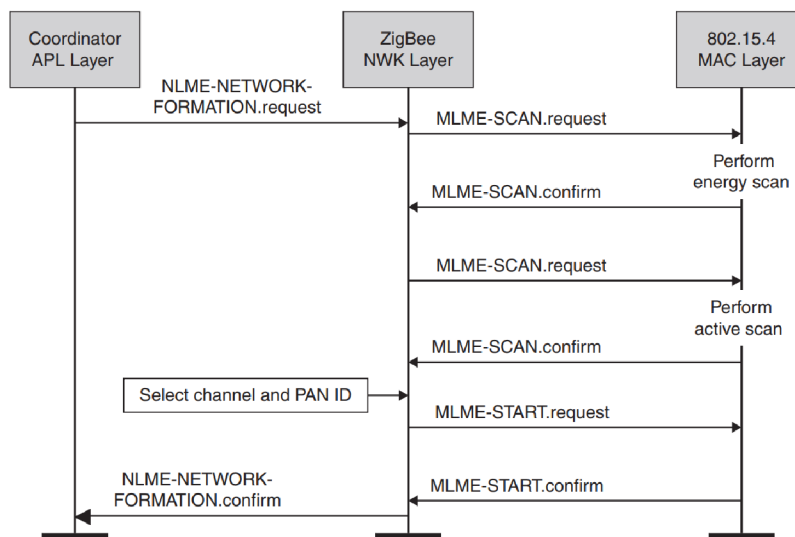
- Aplikační mezivrstva (APS – Application Support Sublayer) – slouží jako rozhraní mezi síťovou vrstvou a samotnou aplikací. Tato mezivrstva má na starosti obecnou práci s pakety jako je filtrování či zasílání potvrzení o přijetí nebo žádosti o znovu zaslání. Dále se stará o udržování adresovacích a vazebních tabulek a předává data jednotlivým aplikačním koncovým bodům
- Aplikační framework (AF – Application Framework) – obsahuje aplikační objekty s aplikačními profily, které mohou provádět síťové operace skrze své koncové body (endpoints), koncové body jsou indexovány od 1 do 240
- ZDO – ZigBee Device Object – jedná se o speciální koncový bod s indexem nula, na kterém běží aplikace starající se o stav zařízení. Tento koncový bod je implicitní v každém ZigBee zařízení a obsluhuje speciální aplikační profil (ZDP – ZigBee Device Profile) s indexem nula

2.3.3 Formování sítě

Proces formování sítě spočívá ve zvolení unikátního identifikátoru pro síť, který se nazývá PAN ID (16bitový identifikátor), a o vybrání jednoho kanálu, obvykle toho, na kterém probíhá nejméně komunikace. Vytvářením sítě je pověřen ZigBee koordinátor, který provede skenování na množství energie na každém kanálu a následně provede i aktivní skenování všech kanálů (active scan) pomocí beacon requestu. Následně je síťovou vrstvou vybrán nejvhodnější kanál a zvolen neduplicitní identifikátor sítě. Tímto dojde ke zformování sítě. Diagram znázorňující proces formování sítě lze vidět na obrázku 2.6.

2.3.4 Připojení zařízení do sítě

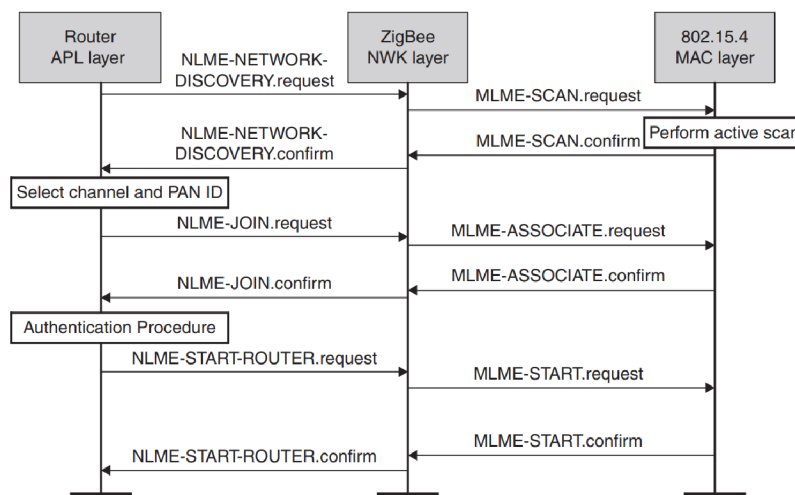
Jedná se o proces zjišťování, jaké sítě a jaké uzly se nacházejí v okolí a o vybrání a připojení se k jedné z nich. Připojovat se do sítě mohou zařízení typu ZigBee Router nebo ZigBee End Device. Proces probíhá obdobně jako u formování sítě. Nejdříve se provede aktivní skenování okolí. Z příchozích beacon rámců jsou vyčteny údaje o dostupných sítích. Následně je zvolena jedna síť a zařízení zašle protistraně (ZigBee Router nebo ZigBee koordinátor) association request. Příjemce si uloží záznam o sousedním uzlu a přiřadí mu 16bitovou síťovou adresu, tuto informaci pošle nazpět. Následně ještě může proběhnout procedura autentizace. Připojování zařízení do ZigBee sítě poté znázorňuje diagram na obrázku 2.7.



Obrázek 2.6: Diagram formování ZigBee sítě [2]

2.3.5 Adresace zařízení

Pro adresaci se používají dva druhy adres. První z nich je adresa MAC také, nazývaná jako dlouhá adresa (IEEE address nebo long address), jedná se o 64bitovou hodnotu přiřazovanou výrobcem zařízení, která je unikátní pro všechna zařízení ze standardu IEEE 802.15.4. Přidělování těchto adres řídí organizace IEEE a stejně jako MAC adresa síťových zařízení v rámci standardu Ethernet i zde je prvních 24 bitů vyhrazeno na identifikaci výrobce a zbývající bity identifikují samotné zařízení. Druhá adresa se nazývá síťová nebo také krátká (short address) a její délka je 16 bitů, přiřazuje se každému zařízení, které bylo přidáno do ZigBee sítě.



Obrázek 2.7: Diagram připojení zařízení k ZigBee síti [2]

2.3.6 Zasílání paketů

Rámce lze zasílat nejen unicastově, ale i broadcastově nebo multicastově. Při broadcastu lze navíc zvolit, jestli mají být rámce zaslány všem zařízením, nebo pouze těm, které nespí nebo pouze zařízením typu ZigBee Router. Lze také využít možnosti omezení dosahu pomocí položky Radius v paketu síťové vrstvy. Pro možnost vysílání rámců multicastově lze využít skupinovou adresu (group address), která je uložena v záhlaví paketu pro aplikační mezivrstvu. Ta následně určí, jestli je paket určen pro konkrétní uzel a případně paket zahodí. [2]

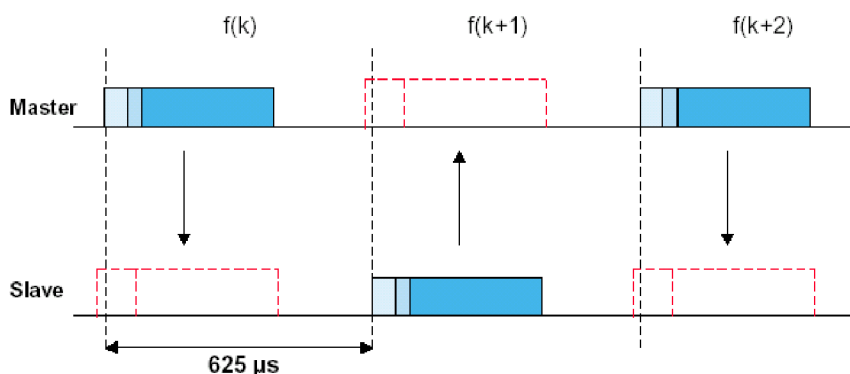
2.4 Bluetooth

Tato podkapitola čerpá z článku [31] a v případě Bluetooth Low Energy (BLE) i z článku [29]. Bluetooth, také nazýváno Bluetooth Classic, je podobně jako ZigBee určen pro vytváření sítí krátkého dosahu, na rozdíl od něj ovšem nestaví na standardu IEEE 802.15.4. Bluetooth je definovaný ve standardu IEEE 802.15.1 a vývoj je zastřešován organizací Bluetooth Special Interest Group (Bluetooth SIG). Jedná se o kompletní protokolový zásobník, který řeší jak přenos dat přes rádiové rozhraní, tak i správu zařízení a jejich komunikaci. Primárním využitím technologie Bluetooth není tvorba senzorových sítí, i když jej lze na tuto oblast také využít, ale propojování dvou a více zařízení komunikujících v režimu master-slave.

Bluetooth za svou existenci od roku 1999, kdy byla vydaná jeho první verze 1.0, prošel řadou vylepšení. Za zmínku jistě stojí verze 2.0, která v roce 2004 zvýšila přenosovou rychlost z původních 721 kbps na 2.1 Mbps díky využití metody Enhanced Data Rate (EDR). Další podstatný milník ve vývoji byla verze 4.0, jejíž specifikace přinesla důležitou novinku ve formě Bluetooth Low Energy. Tato novinka, jak již název napovídá, se zaměřuje na nízkoenergetickou komunikaci a je tak vhodná pro použití s různými senzory, které musí fungovat s omezenými energetickými zdroji. Poslední hlavní verze vydaná v roce 2016 nese číslici 5. Jejím hlavním zaměřením je oblast internetu věcí (IoT) a dále vylepšení specifikace BLE hlavně v segmentu nositelné elektroniky.

2.4.1 Zařízení a komunikace

Jak již bylo zmíněno výše, standard Bluetooth zaobírá jak protokoly vyšších vrstev, tak i samotnou komunikaci na fyzickém rozhraní. Zařízení spolu komunikují v ISM pásmu 2.4 GHz, kde mají k dispozici 79 kanálů o šířce 1 MHz. Komunikace používá techniku FHSS (Frequency Hopping Spread Spectrum), která spočívá v přeskakování mezi jednotlivými kanály a to rychlostí 1600 přeskoků za vteřinu. Využitím FHSS se řeší problém s rušením na velice vytíženém pásmu 2.4 GHz, kdy již obsazené kanály jsou přeskakovány. Díky vysoké rychlosti přeskakování jsou jednotlivé kanály rozděleny na časové úseky (timeslot) o délce 625 μ s. Při komunikaci mezi zařízeními se toho využívá jako TDD (Time-Division Duplexing), kdy zařízení typu master vysílá pakety pouze v sudé časové úseky a zařízení typu slave vysílá svoje pakety v liché časové úseky. Ilustrace tohoto chování je vyobrazena na obrázku 2.8.

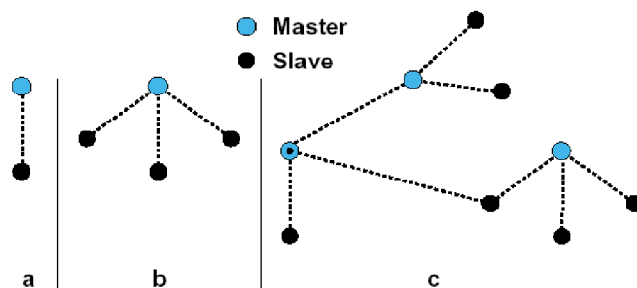


Obrázek 2.8: TDD u přenosu paketů mezi zařízeními typu master a typu slave [27]

Jednotlivá zařízení se propojují do sítí v módu ad-hoc, takovéto sítě se v terminologii Bluetooth označují jako piconet. V síti vždy existuje jedno zařízení s rolí master a jedno až sedm dalších s rolí slave. Roli master získává to, které začne komunikovat jako první, tím také ustanoví novou síť. Role nejsou pevně dané a mohou se měnit vždy s ustavením nové sítě. Pokud se v síti nachází pouze jedno zařízení typu slave, mluvíme o topologii point-to-point. O typu point-to-multipoint mluvíme tehdy, když je k master uzlu připojeno mezi dvěma až sedmi slave uzly. Předávání paketů poté probíhá vždy a výhradně přes master zařízení. Poslední možnou topologií je scatternet, jedná se o propojení dvou a více piconetů do jedné větší sítě. Aby mohlo dojít k propojení, musí vždy jedno zařízení být pro jednu síť v roli master a pro druhou v roli slave. Jednotlivé topologie jsou znázorněny na obrázku 2.9.

2.4.2 Protokolový zásobník Bluetooth Classic

Tak jak je zvykem u každého protokolového zásobníku, tak i ten u Bluetooth Classic se sestává z několika vrstev, které ilustruje obrázek 2.10. První vrstva je Bluetooth radio, které se stará o příjem a vysílání signálů na bezdrátovém rozhraní daného zařízení. Druhou vrstvou je baseband, který zpracovává odesílaná a přijímaná data, která komunikuje s vrstvou předchozí. Dále provádí detekci a opravu chyb v přijímaných datech a také se stará o řízení toku paketů a správu napájení. Adresaci a navazování spojení mezi zařízeními, tj. tvorbu sítí piconet, obstarává třetí vrstva označovaná jako LMP (Link Management Protocol). Spojení, která mohou mezi sebou zařízení na této vrstvě navázat, jsou dvojího typu.



Obrázek 2.9: Možné síťové topologie pro Bluetooth: point-to-point, point-to-multipoint, scatternet [26]

První je SCO (Synchronous Connection-Oriented). Toto spojení je symetrické a využívá se pro propojení point-to-point převážně na přenos hlasu. Druhým typem je ACL (Asynchronous Connectionless links), které slouží primárně pro přenos datových paketů. Asynchronní spojení umožňuje všesměrové vysílání v rámci dané sítě.

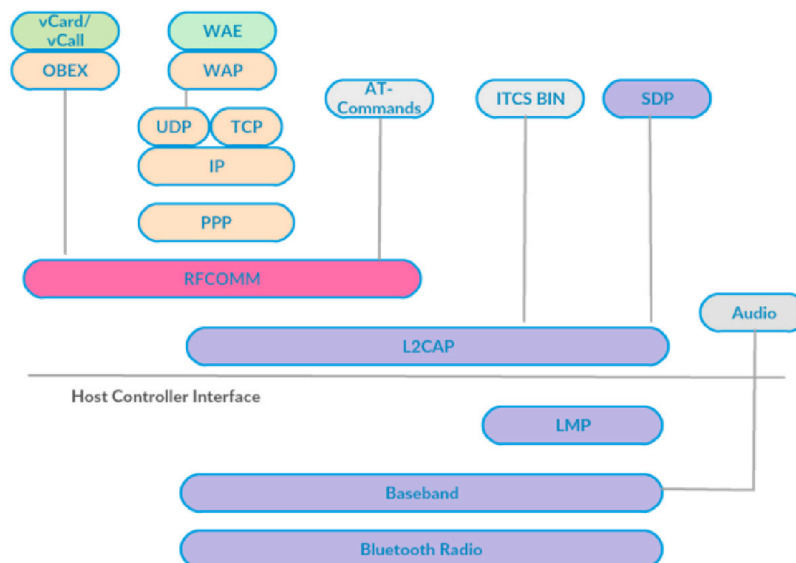
Rozhraní řídicí komunikaci mezi vrstvami řadiče a hostitelskými protokoly se označuje zkratkou HCI (Host Controller Interface). Toto rozhraní umožňuje ovládat vrstvu basebandu a link managementu pomocí přístupu ke kontrolním a stavovým registrům zařízení. Toto rozhraní využívá protokol L2CAP (Logical Link Control and Adaptation Protocol), který multiplexuje jednotlivá logická spojení mezi zařízeními pro protokoly vyšších vrstev, dále provádí segmentaci a znovusestavování paketů. Dalším protokolem vyšších vrstev, který využívá rozhraní HCI je RFCOMM (Radio Frequency Communications). Tento transportní protokol emuluje sériový port RS-232 a poskytuje jednoduché a spolehlivé spojení podobné transportnímu protokolu TCP. Poslední důležitý protokol, který je používán pro zjištění dostupných služeb poskytovaných Bluetooth zařízením je SDP (Service Discovery Protocol).

Ostatní protokoly vyšších vrstev implementují různé funkce daného Bluetooth zařízení. Může se jednat o přenos dat, hlasu, datagramovou komunikaci přes IP protokol a podobně. Každé zařízení poskytuje tzv. Bluetooth profily, které definují, jaké služby a funkce dané zařízení nabízí. Jako příklad lze uvést A2DP (Advanced Audio Distribution Profile), který je využíván pro vysílání zvuků do reproduktorů či sluchátek.

2.4.3 Protokolový zásobník Bluetooth Low Energy

Bluetooth Low Energy zkráceně BLE přináší oproti Bluetooth Classic redukovanou a optimalizovanou fyzickou vrstvu (Bluetooth radio), která snižuje počet kanálů ze 79 na pouhých 32. Každý kanál má šířku 2 MHz oproti 1 MHz u Bluetooth Classic. Díky tomu, že se okolní zařízení hledají na podstatně menším počtu kanálů, dochází ke zvýšení efektivity a k rychlejšímu navázání spojení. Protože BLE klade důraz na nízkou energetickou náročnost je pro udržení spojení mezi zařízeními vyžadována i menší rádiová aktivita.

Protokolový zásobník, obrázek 2.11, se dělí na tři sekce: řadič, hostitel a aplikace. Fyzická a linková vrstva náleží do řadiče. HCI zde zaujímá stejnou úlohu jako u Bluetooth Classic a to zprostředkování komunikace mezi řadičem a hostitelskými protokoly. Hostitelskou sekci okupují následující vrstvy:



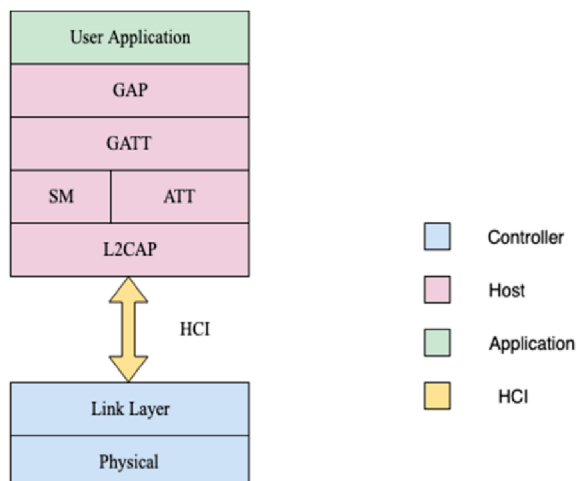
Obrázek 2.10: Protokolový zásobník Bluetooth Classic [31]

- L2CAP (Logical Link Control and Adaptation Protocol) — podobně jako u Bluetooth Classic se stará o segmentaci a znovusestavování paketů
- SM (Security Manager) — zajišťuje párování a distribuci klíčů, autentifikaci a šifrování dat
- ATT (Attribute Protocol) — zajišťuje prezenci a přenos atributů (dat) ze serveru na klienta
- GATT (Generic Attribute Profile) — definuje chování a funkce zařízení ve formě profilů, které zastřešují dostupné atributy daného zařízení
- GAP (Generic Access Profile) — jedná se o rozhraní pro uživatelskou aplikaci, specifikuje role zařízení, módy a procedury

2.4.4 Komunikace a výměna dat s Bluetooth Low Energy

Každé zařízení se chová jako server nebo jako klient. Toto rozdělení obvykle koresponduje s tím, zdali je zařízení v roli master (klient) nebo slave (server). Zařízení se ovšem může chovat také jako klient a server dohromady. Každý server implementuje GATT databázi, která obsahuje různé profily definující chování a dostupné aplikace na BLE zařízení. Profily jsou určeny standardem a zahrnují hierarchicky strukturované sekce zvané služby (services). Každá služba zahrnuje kolekci dat označovaných jako charakteristiky (characteristic). Standardem definované služby a charakteristiky jsou reprezentovány 16bitovou adresou. Mimo to lze vytvářet i vlastní služby, které jsou definovány pomocí 128bitové adresy v UUID formátu.

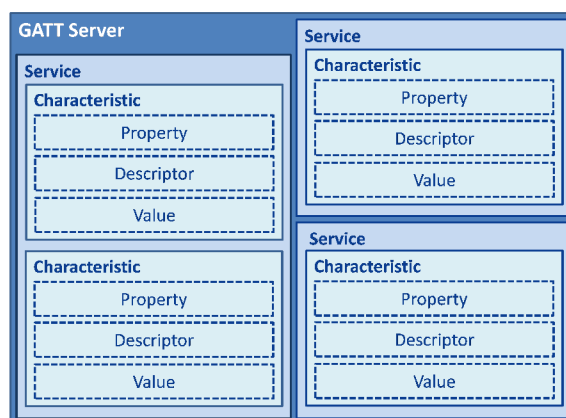
GATT zapouzdřuje ATT, které organizuje svoje data do atributů. Každému atributu je přiřazen jedinečný identifikátor, oprávnění přístupu a samotná hodnota. Služby jsou chápány jako seskupení jistých atributů. Atributy jsou v kontextu služeb brány jako charakteristiky, které obsahují data, deskriptor a vlastnosti. Tato hierarchie GATT serveru je



Obrázek 2.11: Protokolový zásobník Bluetooth Low Energy [31]

ukázána na obrázku 2.12. Vlastnosti charakteristik následně indikují klientovi, jaké operace jsou s danou charakteristikou dovoleny provádět a mohou nabývat následujících typů:

- Broadcast — umožňuje posílání dat pomocí oznamovacích (advertising) paketů
- Readable — klient může pouze číst hodnotu dané charakteristiky
- Writable — klient může nastavit novou hodnotu dané charakteristiky
- Notifiable -- klient si může zaregistrovat přijímání oznámení o změně dané charakteristiky serverem



Obrázek 2.12: Hierarchie dat v GATT serveru [29]

2.5 UWB

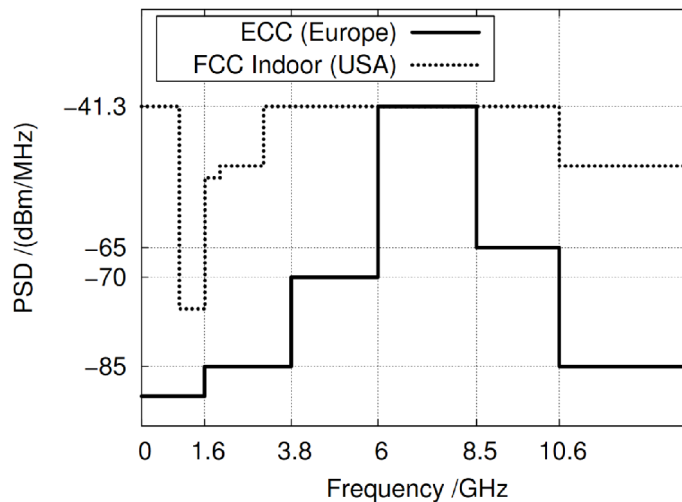
UWB neboli ultra-wide band je technologie širokopásmového vysílání určeného pro komunikaci na krátkou vzdálenost při využití velké šíře rádiového spektra. Jedná se o pulzní

vysílání, které bylo dle Mezinárodní telekomunikační unie (ITU) definováno jako vysílání, jehož frekvenční šířka překračuje hodnotu 500 MHz nebo relativní šířka pásma tohoto vysílání překračuje hodnotu 0.2 podle vztahu 2.1, kde f_{max} je horní frekvence a f_{min} je spodní frekvence pásma. [30]

$$\frac{2 \times (f_{max} - f_{min})}{f_{max} + f_{min}} \quad (2.1)$$

Protože komunikace využívá velkou šířku pásma, nutně musí docházet k překrývání se standardním úzkopásmovým vysíláním. Toto ovšem není problém, protože u širokopásmového vysílání je energie rozprostřena v každém pulzu do velkého spektrálního rozsahu. Díky tomu dochází k ovlivnění úzkopásmového vysílání, které je založeno na modifikaci nosné sinusové vlny, jenom na úrovni šumu. Ultra široké vysílání je proto vysoce odolné vůči interferenci s již existujícími rádiovými systémy. [16]

Za jako první využití ultra širokopásmového vysílání lze považovat Marconiho jiskrovou telegrafii, díky které přenesl informaci přes Atlantik již na začátku dvacátého století. Následně si technologie prošla zkoumáním převážně v armádním prostředí pro radarové účely a na dlouho nebyla světu dostupná. Až v roce 2002 zformulovala americká Federální komise pro komunikaci pravidla, která umožnila obecné využití technologie UWB. Následně se k regulaci připojily i další úřady po celém světě. Dle Radiotelekomunikační unie bylo následně stanoveno doporučení na využívání kmitočtového pásma v rozsahu 3.1-10.6 GHz a byla stanovena obecná kmitočtová maska s maximálními hodnotami spektrální hustoty výkonu (viz obrázek 2.13). [16][25]



Obrázek 2.13: Obecná kmitočtová maska rádiové UWB systémy dle ETSI EN 302065 [25]

2.5.1 IEEE 802.15.4-2011

UWB je pouze jiný způsob jakým se dá informace šířit prostředím. Nedefinuje žádný protokol, který by se dal využít pro komunikaci. Pro tyto potřeby proto vznikla nová revize původního standardu IEEE 802.15.4. Tato revize jej rozšířila o možnost použití dvou nových fyzických vrstev (PHY). První z nich je založena na technologii Chirp Spread Spectrum (CSS) operující na frekvenci 2.4 GHz. Druhá varianta nové fyzické vrstvy umožňuje využití ultra širokopásmové technologie UWB a to na 16 kanálech. Tyto kanály jsou rozděleny

na tři pásma, kde každý kanál má šíři alespoň 500 MHz. První pásmo jako jediné operuje pod hranicí 1 GHz, proto se také nazývá sub-gigahertzové a obsahuje pouze jeden kanál. Další pásmo se nachází v oblasti 3.1-4.8 GHz a nazývá se nízké pásmo, obsahuje čtyři kanály. Poslední pásmo se jmenuje vysoké pásmo a rozprostírá se na frekvencích 6-10.6 GHz a nachází se v něm celkem 11 možných kanálů. Každé zařízení implementující jedno ze tří pásem musí vždy povinně v daném pásmu podporovat jeden konkrétní kanál. Pro sub-gigahertzové pásmo je to kanál 0, pro nízké pásmo kanál 3 a pro vysoké pásmo je to kanál 9. [10] Podrobný rozpis kanálů znázorňuje tabulka 2.2.

Tabulka 2.2: Kanály podporované IEEE 802.15.4-2011 UWB PHY s vyznačenými povinnými kanály pro dané pásma [10]

Číslo kanálu	Střední frekvence [MHz]	Šířka pásma [MHz]	UWB pásmo
0	499.2	499.2	Sub-gigahertzové
1	3494.4	499.2	Nízké pásmo
2	3993.6	499.2	
3	4492.8	499.2	
4	3993.6	1331.2	
5	6489.6	499.2	Vysoké pásmo
6	6988.8	499.2	
7	6489.6	1081.6	
8	7488.0	499.2	
9	7987.2	499.2	
10	9496.4	499.2	
11	7982.2	1331.2	
12	8958.6	499.2	
13	9484.8	499.2	
14	9984.0	499.2	
15	9484.8	1354.97	

Kapitola 3

Lokalizace objektů v bezdrátové senzorové síti

V dnešní době jsou senzorové sítě rychle rostoucí a perspektivní oblastí. V mnoha případech u nich potřebujeme znát polohu jednotlivých uzlů abychom mohli správně interpretovat naměřené hodnoty. Další oblastí, ve které je užitečné znát polohu uzlů, je sledování pohybu osob nebo zboží. Sledované objekty jsou vybaveny speciálním typem uzlu obvykle nazývaným jako tag.

Jednotlivé senzorové sítě se od sebe liší v mnoha ohledech. Uzly mohou být v sítích rozmístěny náhodně nebo můžou být umístěny do pravidelné topologie. Ve výchozím stavu nemusí žádný z uzlů znát svoji polohu. Jakmile uzel začne komunikovat s dalšími uzly v jeho dosahu (sousední uzly), může svou polohu vůči jeho sousedům odhadnout. Díky výměně zpráv v rámci celé sítě mohou jednotlivé uzly získat představu o své celkové poloze v rámci dané sítě. Toto povědomí o své poloze se označuje anglickým termínem location awereness.

Tato kapitola shrnuje nejdůležitější metody používané pro lokalizaci jednotlivých uzlů senzorové sítě. Metody lze rozdělit do kategorií podle různých vlastností, které mají společné. V následujícím textu jsou v první části zmíněny metody využívající změřeného dosahu uzlů (range-based). Další část se zabývá trochu méně využívanými metodami, které neměří vzdálenost mezi uzly, nýbrž provádí aproximaci polohy na základě vlastností sítě (range-free). V poslední části této kapitoly je pozornost následně zaměřena na metody, které nejvíce benefítují na vlastnostech ultra širokopásmové komunikace.

3.1 Metody založené na změřeném dosahu

Poznatky v této podkapitole vycházejí z knih [19], [4], [18] a z článku [24]. Princip těchto metod je založen na měření vzdálenosti mezi dvěma uzly. Pro dostatečný počet změřených vzdáleností je následně možné určit přibližnou pozici lokalizovaného uzlu, a to buď ve dvou nebo tří rozměrném prostoru. Využívá se zde měření času, síly signálu nebo úhlu pod kterým signál přichází.

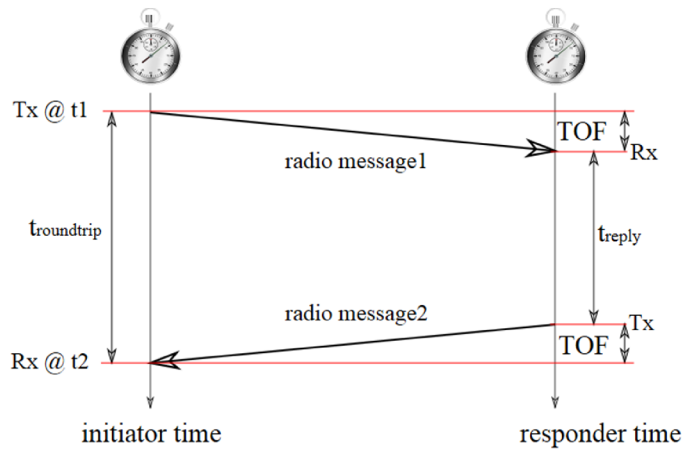
3.1.1 Měření zpoždění (ToF)

Pokud je prostředí dostatečně koherentní pro šíření signálu konstantní rychlostí, ze znalosti této rychlosti a naměřené hodnoty času potřebného k propagaci signálu je možné odvodit vzdálenost, kterou tento signál musel urazit z bodu A do bodu B [24]. Na základě tohoto tvrzení je postavena metoda ToF (Time of Flight), označované také jako ToA (Time of

Arrival) využívající vlastnosti rádiového signálu. Protože rychlost šíření rádiového signálu je velice vysoká (ve vakuu je rovna rychlosti světla), je nutné využít velice přesného měření času. V opačném případě by bylo dosaženo velkých odchylek od reálné vzdálenosti. Kupříkladu pro lokalizaci s přesností jednoho metru je potřeba měřit čas s přesností na 3.3 nanosekundy.

Pro určení doby letu signálu je potřeba znát přesný čas, kdy byl signál vyslán a také přesný čas, kdy byl signál přijat. Protože vysílání a příjem neprovádí jedno, ale dvě separátní zařízení, je nutné, aby mezi těmito zařízeními byl přesně synchronizován čas. Poté lze do přenášené zprávy mezi zařízeními vložit časové razítko doby, kdy byla zpráva odeslána. Příjímač na základě této zprávy může určit časový rozdíl odpovídající době letu signálu. Druhou možností, která již nevyžaduje, aby mezi dvěma uzly byly přesně synchronizované hodiny, je kooperativní metoda. Ta využívá výměnu zpráv, ve které se jedna strana označuje jako iniciátor (initiator) a druhá strana jako odpovídač (responder). Při této metodě si iniciátor v okamžiku odeslání zprávy zaznamená časové razítko (t_1). Odpovídač přijme zprávu a okamžitě na ní odpovídá, čas, který přitom uběhl (t_{reply}) přikládá do odesílané zprávy. Iniciátor přijme zprávu a uloží si druhé časové razítko (t_2) označující čas přijetí zprávy. Iniciátor v tuto chvíli disponuje všemi potřebnými údaji pro určení vzdálenosti mezi sebou a odpovídačem. Čas, reprezentující vzdálenost mezi dvěma uzly, se vypočítá podle vztahu 3.1. Diagram na obrázku 3.1 následně znázorňuje výměnu zpráv v čase mezi iniciátorem a odpovídačem.

$$ToF = \frac{(t_2 - t_1) - t_{reply}}{2} \quad (3.1)$$

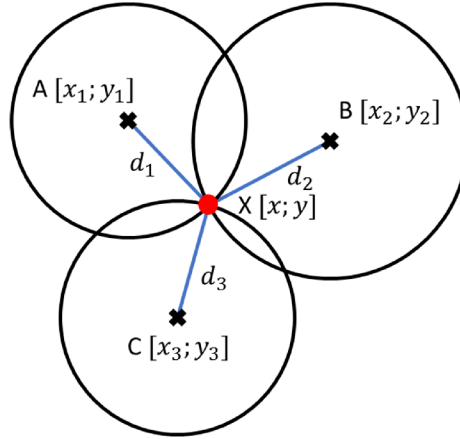


Obrázek 3.1: Určení ToF pomocí kooperativní metody [6]

Díky času, potřebného pro propagaci signálu mezi dvěma uzly, můžeme určit vzdálenost. Znalost pouze této jedné vzdálenosti ovšem nestačí pro určení polohy vysílacího uzlu. Pro potřeby určení polohy ve dvourozměrném prostoru je potřeba využití alespoň třech kotevních uzlů se známou pozicí. Tyto uzly musí být v dosahu vysílajícího uzlu, jehož polohu určujeme. Díky znalosti vzdálenosti od jednotlivých kotevních uzlů (znázorněných na obrázku 3.2 jako kružnice) můžeme pomocí obecného vzorce 3.2 pro multilateraci vypočítat souřadnice hledaného uzlu. Stačí tedy vytvořit soustavu rovnic, do níž se dosadí souřadnice každého kotevního uzlu i $[x_i; y_i]$ a vzdálenost d_i od tohoto kotevního uzlu k uzlu hledanému. Z výsledné soustavy rovnic se následně vypočítají souřadnice hledaného uzlu $[x; y]$.

Pro tvorbu soustavy rovnic se využívá následujícího vztahu pro neznámý uzel $X [x; y]$ a známé pozice uzlu $i [x_i; y_i]$ a jeho vzdálenosti d_i od uzlu X .

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2 \quad (3.2)$$



Obrázek 3.2: Učení souřadnic hledaného uzlu pomocí multilaterace

3.1.2 Měření difference zpoždění (TDoA)

TDoA (Time Difference of Arrival) pracuje na podobném principu jako předchozí metoda zjišťování vzdálenosti, avšak oproti nutnosti přítomnosti informace o přesném okamžiku odeslání signálu vysílačem zjišťuje rozdíl mezi přijetím dvou různých signálů. Odpadá tím nutnost přesných a synchronizovaných hodin na lokalizovaném uzlu. Přesné hodiny jsou potřeba jenom na kotevních uzlech, které měří diferenci.

Existují dva rozdílné přístupy pro lokalizaci uzlu. První z nich je založena na principu vyslání dvou signálů s různou rychlostí propagace daným prostředím v jednom okamžiku. Kotevní uzel po přijetí prvního signálu spustí časovač, kterým měří čas do přijetí druhého signálu. Na základě této difference a ze znalosti rychlosti šíření signálu médiem může být určena vzdálenost od lokalizovaného uzlu. Následně se použije stejně jako v předchozím případě multilaterace pro výpočet koordinátu lokalizovaného uzlu. V tomto případě se využívá elektromagnetického záření ve formě rádiového signálu, který má řádově větší rychlost šíření oproti zvuku, který je využíván ve formě ultrazvukových signálů.

Druhý přístup nevyužívá dva rozdílné signály, naproti tomu využívá faktu, že každý kotevní uzel obdrží signál v jiný čas, a to v závislosti na vzdálenosti kotevních uzlů mezi sebou tak i na vzdálenosti kotevních uzlů od uzlu lokalizovaného. To vyžaduje, aby byly hodiny na kotevních uzlech vzájemně synchronizované. Podrobněji se tomuto přístupu věnuje kapitola 3.3.2.

3.1.3 Měření síly signálu (RSSI)

RSSI (Received Signal Strength Indication) je hodnota síly přijímaného signálu, kterou většina sensorových uzlů, disponujících bezdrátovým komunikačním rozhraním, dokáže poskytnout. Metoda je založena na faktu, že energie rádiového signálu postupně klesá se vzdáleností od zdroje a je ovlivňována útlumem při průchodu prostředím. Vzdálenost lze

poté odhadnou na základě znalosti vysílacího výkonu, síly přijímaného signálu a předem definovaného útlumu prostředí.

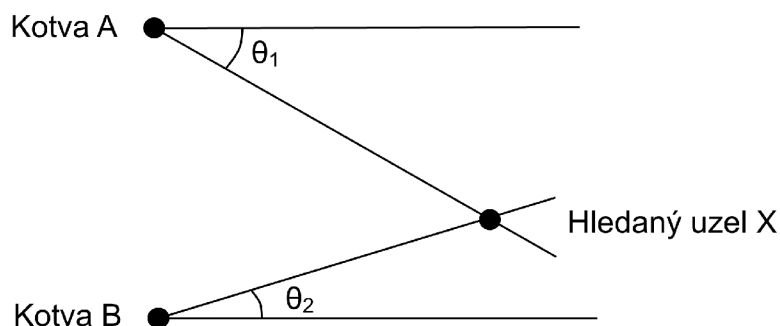
Lokalizace poté může být založena na algoritmu k-nejbližších, díky kterému lze odhadnout polohu neznámého uzlu. Myšlenka je založena na předpokladu, že pokud mají dva uzly stejnou nebo velice podobnou hodnotu RSSI, tak se nacházejí blízko sebe. Pro tento algoritmus je nutné mít v dané oblasti dostatečný počet kotevních uzlů se známou polohou, a to ideálně rovnoměrně rozprostřených po dané oblasti. Jakmile se objeví nový neznámý uzel, je okamžitě porovnávána jeho hodnota RSSI se známými kotevními uzly. Následně se vypočte průměrná poloha uzlu pomocí k-nejbližších kotevních uzlů.

Další možnou technikou lokalizace je využití otisků, tato technika se skládá ze dvou částí. První z nich je fáze trénování, při které je daná oblast rozdělena na sektory, ve kterých se zjišťují hodnoty RSSI pro každý daný sektor. Z takto naměřených hodnot se poté vytvoří databáze. Ve druhé fázi se již provádí pouze porovnání hodnoty RSSI neznámého uzlu s hodnotami v databázi. Díky tomu lze určit, ve kterém sektoru se hledaný uzel nachází.

Nevýhodou metod založených na měření síly přijímaného signálu je fakt, že měřená hodnota je vysoce náchylná na prostředí a může ji ovlivňovat jak teplota, tak i vlhkost vzduchu. Také je potřeba počítat s tím, že síla signálu bude ovlivňována různými překážkami a například pohyb lidí v daném prostředí může způsobovat rozdíly naměřených hodnot od skutečných v průběhu krátkých časových úseků.

3.1.4 Měření úhlu (AoA)

Poslední range-based metodou je AoA (Angle of Arrival), někdy také označována jako DoA (Direction of Arrival). Tato metoda k výpočtu pozice uzlu využívá úhel, který vzniká mezi směrem šíření signálu a udaným referenčním směrem, tak jak ilustruje obrázek 3.3. Metoda klade velké nároky na HW kotevních uzlů, protože si vynucuje využití antén schopných určit směr přijímaného signálu. Tohoto bývá dosaženo použitím několika směrových antén. Pro určení pozice vysílajícího uzlu je následně nutné využití alespoň dvou kotevních uzlů. Výpočet je založen na triangulaci a probíhá tak, že se vytvoří přímky ve směru přicházejícího signálu od všech kotevních uzlů, jejichž průsečík reprezentuje výslednou pozici hledaného uzlu. Tato metoda je velice náchylná na chyby měření úhlu, kdy minimální odchylka může mít obrovský dopad na lokalizaci uzlu.



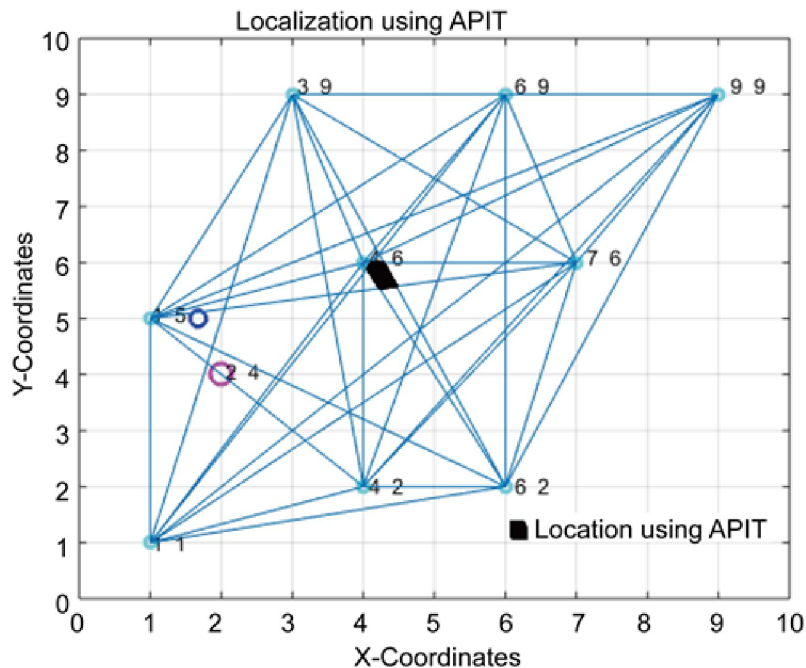
Obrázek 3.3: Triangulace uzlu pomocí dvou kotev

3.2 Metody bez změřeného dosahu

Jedinou informací, kterou range-free metody využívají při sestavení topologie je informace, zda je v dosahu radiového signálu jiný uzel. Na základě tohoto zjištění jsou schopny odhadnout svoji pozici. Nevýhoda těchto metod je, že polohu daného uzlu určují odhadem. Pro ilustraci jsou v následujícím textu uvedeny dvě často používané metody.

3.2.1 APIT

APIT (Approximate Point in Triangulation), jedná se o nelokalizovaný iterační algoritmus využívající vysílání signalizačních zpráv z kotevních uzlů. Využívá přístupu založeného na ploše, kde pro provedení odhadu lokalizace sensorových uzlů je celá oblast sensorové sítě rozdělena na trojúhelníky kotevnými uzly (obrázek 3.4). Přítomnost uzlu uvnitř nebo vně těchto trojúhelníkových regionů umožňuje uzlu zužovat oblast, ve které se může potenciálně nacházet. Využitím různých kombinací kotevních uzlů může následně docházet ke zmenšování velikosti oblasti, ve které se daný uzel nachází. Díky tomu lze zpřesňovat odhad jeho polohy. Metoda používaná pro zužování pravděpodobnosti rozpětí, ve kterém se uzel nachází, se nazývá PIT (point in triangulation). V této metodě si lokalizovaný uzel vybere vždy tři kotevní uzly ze všech kotevních uzlů ležících v jeho blízkosti a propojí je do pomyslného trojúhelníku. Následně zjišťuje, zdali se nachází uvnitř nebo vně tohoto trojúhelníku. Algoritmus APIT opakuje metodu PIT do té doby, dokud nejsou vyčerpány všechny kombinace kotevních uzlů, anebo do té doby, dokud není dosaženo požadované přesnosti. [1]



Obrázek 3.4: Ukázka principu lokalizace algoritmem APIT [1]

3.2.2 DV-Hop

DV-Hop (Distance Vector Hop) lokalizace, jejíž popis vychází z článku [12], využívá mechanismy podobné klasickému směrování pomocí distance vector protokolů. Využívá přitom záplavového šíření signalizačních zpráv v rámci sensorové sítě. Signalizační zprávy vysílá každý kotevní uzel a obsahují jeho pozici a také hop-count parametr, který je na počátku nastaven na hodnotu jedna. Tento parametr se při každém předání zprávy dalším uzlem v síti inkrementuje o jedničku. Každý uzel v síti si následně udržuje tabulku obsahující pozici všech kotev spolu s informací o počtu skoků k dané kotvě. Jakmile uzel obdrží signalizační zprávu, porovná hop-count parametr s obsahem svojí tabulky a pokud je menší, tabulku si aktualizuje a přidá informaci od jakého uzlu tuto zprávu přijal. Výsledkem je, že každý uzel zná nejkratší cestu ke každému kotevnímu uzlu v síti ve formě počtu skoků k této kotvě.

Pro potřebu převodu počtu skoků ke skutečné fyzické vzdálenosti je nutné, aby systém odhadl průměrnou vzdálenost jednoho skoku bez použití range-based metod. Tohoto lze dosáhnout díky faktu, že každý kotevní uzel zná pozice ostatních kotevních uzlů a také počet skoků potřebný pro jejich dosažení. Průměrnou vzdálenost na skok vypočítá jedna z kotev, indexována jako i , pomocí vztahu 3.3.

$$AvgHopSize_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_j} \quad (3.3)$$

Ve vztahu jsou $[x_j; y_j]$ koordináty j -té kotvy a h_j je vzdálenost ve skocích od kotvy i ke kotvě j . Jakmile je průměrná vzdálenost vypočítána, je zpropagována skrze nejbližší uzly do celé sítě. Pokud je uzel schopen vypočítat ze zjištěné průměrné velikosti skoku vzdálenost alespoň ke třem kotevním uzlům, může určit svoji polohu v rámci sítě pomocí multilaterace, která je popsána výše v kapitole 3.1.1.

3.3 Metody pro určování poloh uzlů v sítích UWB

Informace pro tuto podkapitulu jsou čerpány z dokumentů firmy Decawave [6] a [5]. U sensorových sítí založených na technologii UWB lze využít pro lokalizaci objektů všech range-based metod uvedených v kapitole 3.1. Ovšem největší výhodou budou čerpat ty, které jsou založené na měření času. Z charakteristik UWB signálů totiž umí těžit nejvíce. Důvod tkví ve vlastnosti pulzního vysílání, které není oproti běžným rádiovým přenosům tolik náchylné na odrazy a multicestné šíření. Není tedy problém zjistit přesný čas přijetí vyslaného signálu. V následujícím textu se bude pozornost upínat ke dvěma nejpoužívanějším technikám pro lokalizaci založené na měření času v UWB sítích, ve kterých se pro lokalizovaný uzel používá označení tag.

3.3.1 Two-way ranging

Jedná se o vylepšení kooperativní metody ToF, která určuje vzdálenost mezi dvěma body na základě známé rychlosti šíření elektromagnetického záření a doby, po kterou toto záření putuje mezi vysílačem a přijímačem. Vylepšení spočívá v přidání výměny zpráv i opačným směrem, a to z důvodů minimalizace chyb způsobených posuvem hodin nebo frekvence na jednotlivých uzlech. Komunikace mezi iniciátorem a odpovídačem (obrázek 3.5), nazývaná také jako „poll-response-final“, probíhá následovně:

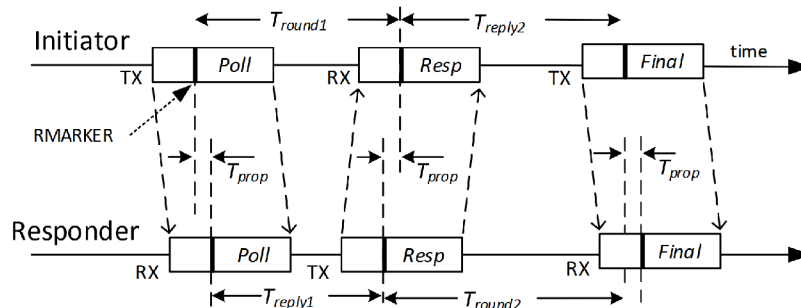
1. Iniciátor odesílá Poll zprávu a při jejím odeslání spouští časovač t_{round1}

2. Odpovídač přijme Poll zprávu, na kterou odešle odpověď ve formě Response, zároveň si ukládá dobu mezi přijetím Poll zprávy a odesláním Response t_{reply1} a také spouští časovač t_{round2}
3. Iniciátor přijme Response zprávu, zastaví časovač t_{round1} , jehož hodnotu odešle zprávou Final společně s dobou, která uběhla mezi přijetím Response a odesláním Final zprávy t_{reply2}
4. Odpovídač přijme Final zprávu, zastaví časovač t_{round2} a ze zjištěných údajů vypočítá čas propagace signálu t_{prop} pomocí vztahu 3.4

$$t_{prop} = \frac{t_{round1} \times t_{round2} - t_{reply1} \times t_{reply2}}{t_{round1} + t_{round2} + t_{reply1} + t_{reply2}} \quad (3.4)$$

V případě metody Two-way ranging se za iniciátora považuje uzel typu tag a za odpovídače kotevní uzel. Z výše popsané komunikace je také patrné, že jediný uzel, který zná vzdálenost iniciátora, respektive tagu, je odpovídač, respektive kotva. Aby se mohl tag vypočtenou vzdálenost dozvědět, přidává se do modelu ještě poslední zpráva typu Report, kterou kotva odešle na konci komunikace tagu. Pro zjištění polohy tagu, musí proběhnout výše popsaná komunikace s alespoň třemi kotvami v dosahu. Na naměřené vzdálenosti se poté aplikuje multilaterace popsaná v kapitole 3.1.1.

Mezi výhody metody Two-way ranging nepochybně patří to, že zde není potřeba mít synchronizované hodiny mezi tagem a kotvou. Na druhou stranu obrovská nevýhoda této metody spočívá v nutnosti výměny zpráv a díky tomu i potřebě tagu být povětšinou času v aktivním stavu. To pro uzel, jež je primárně určen k běhu na omezený energetický zdroj není výhodné.



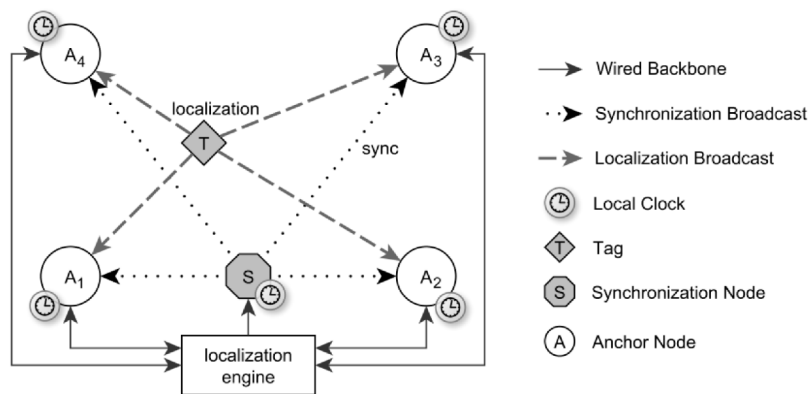
Obrázek 3.5: Časový diagram zasílání zpráv v metodě Two-way ranging [6]

3.3.2 TDoA

Tato metoda byla již částečně popsána v kapitole 3.1.2. Pro případ lokalizace objektů v senzorové síti pomocí UWB zde bude podrobněji ukázána druhá varianta nevyužívající dvou druhů signálů s odlišnou rychlostí propagace prostředím. Tato metoda využívá podobného, avšak reverzovaného principu, na kterém funguje globální družicový polohový systém (GNSS). U něj jsou vysoko nad zemským povrchem umístěny družice obsahující velice přesné atomové hodiny, které jsou napříč celou družicovou sítí synchronizované. Každá družice vysílá signál obsahující čas vyslání dané zprávy. Pozemský přijímač získává tyto zprávy z několika různých družic a každá z těchto zpráv obsahuje jináč časové razítko

v závislosti na vzdálenosti mezi pozemským přijímačem a družicí. Díky této diferenci mezi zprávami lze následně dostatečně přesně určit polohu pozemního zařízení. [28]

U TDoA probíhá komunikace opačným směrem. Tag v pravidelných, předem daných intervalech vysílá signalizační zprávy, které zachycují kotevní uzly v okolí. Každý kotevní uzel, který přijal zprávu si poznačí čas jejího přijetí a tuto informaci odešle spolu se svým identifikátorem a identifikátorem tagu do speciálního výpočetního uzlu. Princip lokalizace tag uzlu spočívá v různě dlouhém časovém okamžiku mezi vysláním signalizační zprávy a jejím přijetím v závislosti na vzdálenosti tag uzlu od každého z kotevních uzlů. Protože kotevní uzly mají přesně synchronizované hodiny, lze jednotlivé časové okamžiky přijetí signalizační zprávy kotvami brát jako difference oproti kotevnímu uzlu, který zprávu přijal jako první. Na základě těchto diferencí následně specializovaný výpočetní uzel provede odhad polohy tagu pomocí multilaterace. Obrázek 3.6 následně ilustruje možnou podobu sítě, ve které kotevní uzly zachytávají zprávy z tag uzlu v různých časových okamžicích. Lze si povšimnout i speciálního uzlu, který slouží pro synchronizaci času mezi kotvami a také výpočetního uzlu, který sbírá veškeré zprávy pro určení polohy tagu.



Obrázek 3.6: Příklad fungování TDoA v síti využívající UWB k lokalizaci uzlu [28]

Kapitola 4

Návrh systému na určování polohy objektů

Pro lokalizaci uzlů v rámci sensorové sítě jsem zvolil technologii ultra širokopásmové komunikace slibující přesnost určování polohy hledaného uzlu v řádech desítek centimetrů [8]. Pro tuto přesnost lze využít jednu z technik zmíněných v předchozí kapitole 3.3.

V tomto projektu jsem se rozhodl zvolit metodu TDoA, protože minimalizuje energetické nároky na straně uzlu typu tag a tím umožňuje jeho umístění na různé předměty nebo osoby bez potřeby častého nabíjení. Celý systém je složený z několika, a to minimálně šesti uzlů, které jsou propojeny v jednu síť. Následný centrální prvek, na kterém běží uživatelská aplikace, provádí zpracování zpráv a výpočty pozic hledaných uzlů.

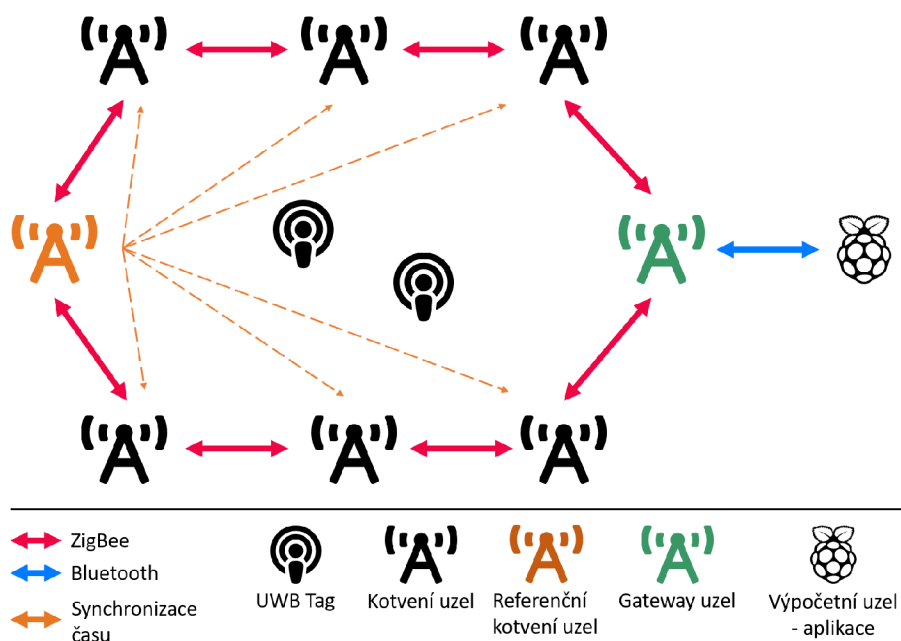
Systém na určování polohy objektů lze pro jednoduchost rozdělit na několik separátních částí, kterým bude věnována pozornost v následujícím textu. Tyto části zahrnují návrh sensorové sítě, návrh přenášených zpráv, komunikace a identifikace uzlů, výběr metody pro synchronizaci času, návrh hardware sensorového uzlu a obslužnou aplikaci běžící na výpočetním zařízení.

4.1 Sensorová síť

Sensorová síť, jejíž schéma ilustruje obrázek 4.1, sestává ze tří typů uzlů, kde každý z nich zastává rozdílnou roli v celém systému. Tyto uzly sdílí stejný hardwarový základ a jejich rozdílná funkcionální je realizována příslušným firmwarem. Prvním typem uzlu jsou tagy, které reprezentují hledané objekty v síti. Předpokladem je, že se tyto uzly budou nacházet uvnitř dané sítě na předem neznámé pozici. Každý aktivní tag vysílá v pravidelných intervalech signalizační zprávy na svém UWB rozhraní. Tyto zprávy jsou následně zachytávány druhým typem uzlů, které jsou označovány podle terminologie jako kotva, respektive kotevní uzly. Kotevních uzlů musí být několik, pro úspěšné určení polohy tagu v trojrozměrném prostoru minimálně čtyři. Každá kotva má své pevné umístění, které je předem známé. Přijetím zprávy dojde k označení časového razítka, kdy byla zpráva zachycena a identifikátoru tagu. Tato informace je následně propagována celou sensorovou sítí, která je založena na standardu ZigBee. ZigBee zde slouží nejen pro předávání informace o přijetí signalizačních zpráv, ale také pro řízení a sledování stavu celé sítě. Každý kotevní uzel přitom odpovídá zařízení typu ZigBee router. Centrálním prvkem, přes který prochází veškerá komunikace je gateway uzel. Ten reprezentuje rozhraní mezi sítí a výpočetním zařízením na kterém běží uživatelská aplikace. Gateway uzel funguje jako ZigBee koordinátor a stará

se o sestavení celé sítě. Pro komunikaci s výpočetním zařízením využívá rozhraní Bluetooth Low Energy.

Protože je gateway uzel vybaven rozhraním Bluetooth Low Energy, je volba zařízení pro provádění výpočtů a běh uživatelské aplikace široká. Pro komunikaci lze totiž využít jakékoliv zařízení, které je vybavené rozhraním Bluetooth 4.0 a disponuje příslušným API pro ovládání Bluetooth Low Energy. Pro potřeby testování zasílání zpráv lze díky tomu využít jakoukoliv aplikaci terminálu pro mobilní telefon, tablet nebo počítač. Pro finální výpočetní zařízení, na kterém bude běžet uživatelská aplikace, jsem zvolil jednodeskový minipočítač Raspberry Pi 4B. Toto zařízení jsem zvolil z důvodu možnosti běhu plnohodnotné linuxové distribuce, v tomto případě výchozího Raspbianu. Tato volba zajišťuje využití široké škály linuxových balíčků a také komunitně vyvíjených knihoven uzpůsobených přímo pro minipočítač Raspberry Pi.



Obrázek 4.1: Schéma navrhované sensorové sítě pro určování polohy objektů

4.2 Komunikace v rámci sensorové sítě

Jak již bylo zmíněno, celá sensorová síť se zakládá na třech různých komunikačních technologiích. Každá komunikační technologie má své specifické vlastnosti, které jsou vhodné na konkrétní účely. UWB rozhraní každého uzlu neimplementuje protokoly vyšších vrstev a veškerá komunikace je řešena pouze na úrovni linkové vrstvy standardu IEEE 802.15.4. Toto rozhraní je v rámci práce určeno pouze pro potřeby lokalizace a synchronizace času v rámci sensorové sítě. Naproti tomu ZigBee zajišťuje primární a spolehlivou komunikaci mezi uzly, řeší sestavování sítě a směrování zpráv. Zprávy zde použité slouží k řízení a předávání výsledků. Bluetooth Low Energy následně vykonává pouze roli prostředníka mezi uživatelskou aplikací a sensorovou sítí, zprávy jsou tedy pouze kopírovány mezi ZigBee sítí a rozhraním uživatelské aplikace.

4.2.1 Lokalizace a synchronizace času

Pro potřeby lokalizace sensorových uzlů není nutné přenášet téměř žádná data. Samotný princip použité metody časových diferencí totiž závisí pouze na znalosti přesného času příjmu signalizační zprávy kotevním uzlem. Jediné, co je potřeba znát, je zdroj této zprávy a také její sekvenční číslo. Stejná situace platí i pro zprávy používané k synchronizaci času.

Protože jsou oba používané typy zpráv určeny pouze ke zjištění času příjmu na kotevním uzlu, a kromě sekvenčního čísla, informaci o odesilateli a identifikátoru typu zprávy neobsahují žádná data, zvolil jsem pro jejich reprezentaci na linkové vrstvě UWB rohnaní rámec typu Multipurpose. Tento rámec dovoluje minimalizovat velikost přenášených dat a díky tomu i režii spojenou s odesláním a příjmem. Informace o odesilateli a sekvenční číslo jsou součástí formátu použitého rámce, informace o typu zprávy je vložena jako funkční kód v podobně jednoho bajtu dat. Na obrázku 4.2 je vidět struktura použitého rámce s popiskem jednotlivých částí.

1B	1B	2B	1B	2B
MP Frame Control	Sequence number	Source Address	FX CODE	FCS
MHR			MAC Payload	MFR

Obrázek 4.2: Rámec typu Multipurpose linkové vrstvy, který slouží pro přenášení signalizačních a synchronizačních zpráv na UWB rozhraní sensorového uzlu. MP Frame Control indikuje rámec typu Multipurpose a dále obsažené hodnoty rámce. Sequence number slouží pro odlišení jednotlivých zpráv stejného typu a od stejného odesilatele. Source address, adresa odesilatele, obsahuje MAC adresu UWB modulu. FX CODE je funkční kód odlišující jednotlivé typy zpráv. FCS obsahuje kontrolní součet rámce.

4.2.2 Řízení a předávání dat

Podobně jako zprávy pro lokalizaci a synchronizaci času mezi kotevními uzly, dodržují i zprávy pro řízení a předávání dat jistý formát. Ve směru z uživatelské aplikace jsou zprávy chápány jako příkazy, které mají oproti datovým zprávám jednodušší strukturu, viz obrázek 4.3. V každé zprávě se jako první vyskytuje jednobajtová hodnota funkčního kódu, která determinuje význam zprávy, případný formát a délku přenášených dat. Zpráva nemusí obsahovat žádná data a může sestávat pouze z funkčního kódu. Není zde potřeba určovat příjemce zprávy, protože se jedná buďto o příkaz pro gateway uzel nebo o příkaz pro všechny kotevní uzly šířené záplavově. Seznam všech potřebných příkazů je následující:

- createNetwork – příkaz pro vytvoření ZigBee sítě předaný z uživatelské aplikace do gateway uzlu
- permitJoin – příkaz všem kotevním uzlům a gateway uzlu sloužící pro umožnění připojení dalšího kotevního uzlu do ZigBee sítě
- startTDOA – příkaz spuštění příjmu signalizačních zpráv z tag uzlů na UWB rozhraní všech kotevních uzlů

- broadcastSync – příkaz, který určí jeden z kotevních uzlů jako referenční a odešle z jeho UWB rozhraní synchronizační zprávu, ostatní kotevní uzly čekají na příjem této zprávy
- stopAll – příkaz pro ukončení veškeré činnosti na UWB rozhraní kotevních uzlů

Zprávy, které pocházejí ze ZigBee sítě a mají se skrz rozhraní Bluetooth Low Energy dostat do uživatelské aplikace, jsou chápány jako datové. Jejich struktura zahrnuje jako první prvek stejný formát funkčního kódu jako u zpráv příkazových. Dále následuje jeden bajt sekvenčního čísla pro potřeby kontroly souvislosti přijatých zpráv. Třetím prvkem je identifikátor odesílatele, za kterým následuje prvek nesoucí informaci o velikosti dat. Jako poslední se ve zprávě nacházejí samotná data o maximální možné délce 255 bajtů. Následující seznam zahrnuje veškeré potřebné datové zprávy pro tuto práci:

- selfIdentification – samo ohlášení nově připojeného uzlu do ZigBee sítě
- TDOATimestamp – časové razítko přijaté signalizační zprávy z kotevního uzlu, kterou odeslal tag uzlu
- syncDataTx – časové razítko odeslání synchronizační zprávy z referenčního kotevního uzlu
- syncDataRx – časové razítko přijetí synchronizační zprávy z ostatních kotevních uzlů, které zprávu přijaly

Command message				
1B	Variable			
Message type	Data payload			
Data message				
1B	1B	1B	1B	Variable
Message type	Sequence number	Source ID	Data length	Data payload

Obrázek 4.3: Formát dvou typů zpráv, které se používají pro komunikaci v rámci senzorové sítě

4.3 Identifikace uzlů v síti

Protože celý systém na určování pozice uzlů využívá třech různých komunikačních technologií, bylo potřeba vymyslet způsob, jak provést jednotnou identifikaci každého jednotlivého uzlu sítě. Tato identifikace musí být jednoznačná a měla by přetrvávat i po průchodu zpráv přes různá komunikační rozhraní sítě. Pro tento účel jsem zavedl systém číslování pomocí jednobajtového identifikátoru uzlu. Tento identifikátor je součástí zasílaných zpráv, tak jak je uvedeno v předcházejícím textu. V případě zpráv pro lokalizaci a synchronizaci času je tato jednobajtová hodnota součástí spodních osmi bitů adresy zdroje a vrchních osm

bitů je doplněno nulami. V případě zpráv pro řízení a předávání dat je tento identifikátor součástí pouze datové zprávy, která vždy pochází z konkrétního uzlu ZigBee sítě. U příkazových zpráv není zdroj potřeba identifikovat, protože tento typ zpráv může vysílat pouze uživatelská aplikace. Číslování se následně řídí konvencí, kdy se začíná od hodnoty 0x41 odpovídající v tabulce ASCII znaku „A“. Tato konkrétní hodnota je přiřazena pro gateway uzlu. Pro kotevní uzly je určen rozsah 0x42 až 0x4B, který odpovídá znakům „B“ až „K“. Nakonec pro uzly typu tag je určen rozsah 0x4C až 0x4F odpovídající znakům „L“ až „O“.

4.4 Synchronizace času mezi kotevními uzly

Pro správné určování poloh uzlů je u metody TDoA klíčové, aby byly hodiny na kotvách správně synchronizované. Pro řešení tohoto problému jsem zvolil metodu lineární extrapolace vycházející z článku [17]. Tato metoda vychází z obecného modelu hodinového signálu, který popisuje vztah mezi dvěma hodinovými systémy. Čas startu čítačů hodinového signálu se na obou systémech liší, tato skutečnost se nazývá offset. Frekvence hodinového signálu se mezi systémy může také mírně lišit z důvodů fyzických vlastností daných použitým hardware, tomuto jevu se říká drift hodin. Poslední jev zvaný clock skew spočívá v odlišnosti hodnot čítačů v závislosti na vzdálenosti mezi hodinovými systémy.

$$t_j = \omega_j t_i + \phi_j \quad (4.1)$$

Metoda lineární extrapolace zanedbává vliv driftu hodin a zaměřuje se primárně na kompenzaci offsetu a clock skew. Podle vztahu 4.1 se provádí korekce času kotevního uzlu j na základě času referenčního kotevního uzlu t_i . ω_j je clock skew pro kotevní uzlu j a ϕ_j je offset mezi časem na kotevním uzlu j a časem na referenčním kotevním uzlu i . Clock skew ω_j se vypočítá pomocí vztahu 4.2 a offset ϕ_j pomocí vztahu 4.3.

$$\omega_j = \frac{T_i^{(k)} - T_i^{(k-1)}}{R_j^{(k)} - R_j^{(k-1)}} \quad (4.2)$$

$$\phi_j = T_i^{(k)} - R_j^{(k)} \quad (4.3)$$

Ve výše uvedených vztazích $T^{(k)}$ znamená čas vyslání k -té synchronizační zprávy referenčním kotevním uzlem i a $R^{(k)}$ je čas přijetí k -té vyslané synchronizační zprávy na kotevním uzlu j . Po získání offsetu a clock skew pro každý nereferenční kotevní uzlu j je možné následně začít provádět korekce časů přijímaných lokalizačních zpráv (TAO – time of arrival) podle vztahu 4.4.

$$newTAO = (TAO - R_j^{(k)})\omega_j + T_i^{(k)} \quad (4.4)$$

4.5 Senzorový uzlu

Jak již bylo zmíněno v předchozí kapitole, jsou všechny typy uzlů sensorové sítě založeny na stejném hardware. Toto řešení jsem zvolil z důvodů zjednodušení návrhu celého systému, kdy je rozdílná funkcionality realizována čistě softwarově. Tento přístup mimo jiné přináší flexibilitu při vývoji a testování, kdy lze využívat jedno zařízení k vícero účelům.

Jako srdce každé jednotlivé desky sensorového uzlu jsem zvolil mikrokontrolér Kinetis KW41Z od firmy NXP. Mikrokontrolér disponuje bezdrátovým rozhraním pro IEEE 802.15.4

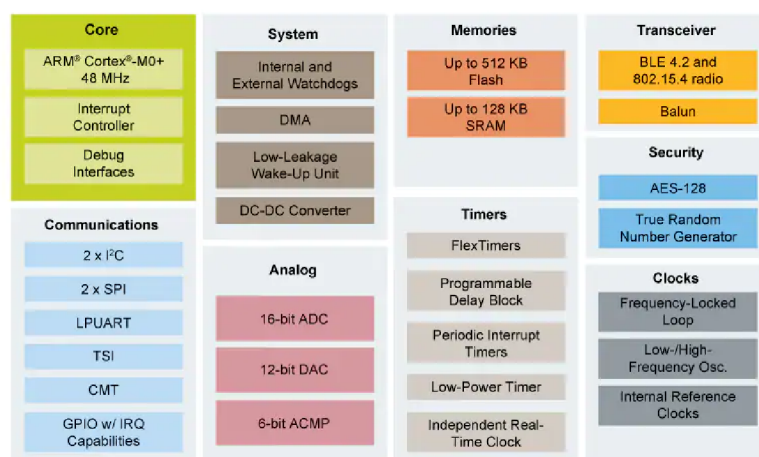
s podporou technologie ZigBee a také rozhraním Bluetooth Low Energy. Pro využití technologie UWB k realizaci lokalizace jsem zvolil modul DWM1000 od společnosti Decawave. Modul je dodáván jako hotové řešení obsahující samotný komunikační čip, napájecí obvody a keramickou anténu. Komunikace s tímto modulem je realizována pomocí rozhraní SPI.

Napájení sensorových uzlů je realizováno pomocí lithium-polymerových akumulátorů s kapacitou 1050 mAh a jmenovitým napětím 3.7 V. Nabíjení akumulátorů je řešeno přes rozhraní USB napětím 5 V. Hlavní napájecí větev, ze které jsou napájeny mikrokontrolér a UWB modul, vyžaduje napětí 3.3 V. Tento nesoulad bylo nutné řešit vhodným regulátorem napětí, pro tento účel jsem zvolil regulátor napětí s nízkým úbytkem LP5912-3.3DRVT od výrobce Texas Instruments. Od stejného výrobce jsem zvolil i čip pro řízení nabíjení lithium-polymerových akumulátorů BQ24075TRGTR.

Programování mikrokontroléru Kinetis KW41z je řešeno pomocí vyvedeného rozhraní SWD. Toto rozhraní slouží i pro ladění vyvíjeného firmware. Pro komunikaci s okolím je vyvedena i asynchronní sériová sběrnice UART. Pro indikaci stavů každého sensorového uzlu je na jeho desce umístěno i několik svítivých diod. Tyto diody indikují například stav nabíjení, průběh komunikace přes technologii UWB a také vnitřní stavy mikrokontroléru.

4.5.1 MCU Kinetis KW41Z

Mikrokontrolér Kinetis KW41Z, jehož blokový diagram obsahuje obrázek 4.4, je založen na procesoru Arm Cortex-M0+ taktovaném na frekvenci 48 MHz. Pro sensorový uzel jsem zvolil model MKW41Z512VHT4 obsahující 512 kB flash paměti a 128 kB operační paměti. Čip je uložen v QFN pouzdře se 48 postranními vývody a jednou zemnicí ploškou. Z bezdrátové konektivity mikrokontrolér obsahuje integrovanou podporu pro standard IEEE 802.15.4, díky kterému umožňuje použití s bezdrátovými sítěmi ZigBee a Thread. Dále obsahuje bezdrátové rozhraní Bluetooth Low Energy 4.2. SDK dodávané k tomuto mikrokontroléru integruje podporu pro protokolové zásobníky všech tří zmíněných síťových technologií. MCU je svými parametry zaměřen na low-power použití a lze jej provozovat v devíti různých nízko příkonových módech. Pro komunikaci s okolními periferiemi lze využít dvou sériových rozhraní SPI, dvou rozhraní I²C a rozhraní UART. Napájet jej lze napětím v rozsahu od 0.9 V do 4.2 V. [21]

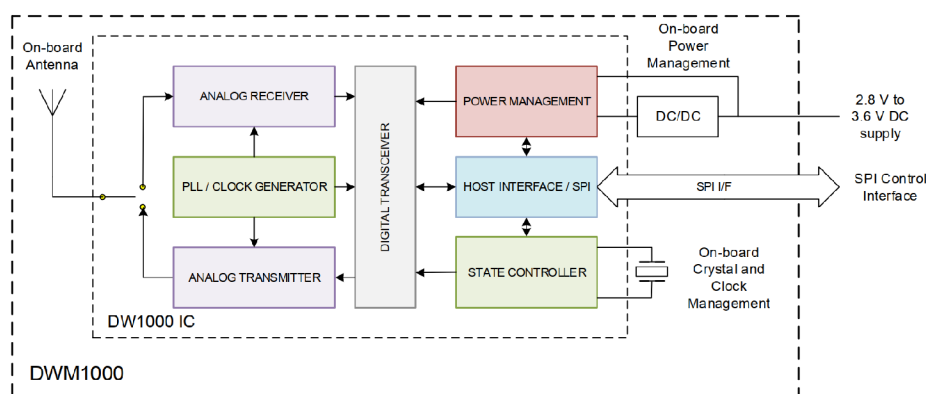


Obrázek 4.4: Blokový diagram mikrokontroléru Kinetis KW41Z [22]

4.5.2 Modul DWM1000

Modul DWM1000 je založen na integrovaném obvodu DW1000, který implementuje standard IEEE 802.15.4-2011 pro bezdrátovou komunikaci v ultra širokopásmovém pulzním vysílání (UWB). Podporuje komunikaci na kanálech 1, 2, 3, 4, 5 a 7 ve frekvenčním rozsahu od 3.5 GHz do 6.5 GHz, a to až na vzdálenost 300 m. Lze jej využívat kromě komunikace i pro lokalizační účely kde podporuje jak metodu ToF tak i TDoA. [8]

Modul, jehož vnitřní diagram se nachází na obrázku 4.5, kromě samotného integrovaného obvodu obsahuje také integrované hodiny, správu napájení a vlastní anténu. Se svým okolím komunikuje přes vyvedené sériové rozhraní SPI. Pro napájení je potřeba napětí 3.3 V s tolerovaným rozsahem od 2.8 V do 3.6 V. Modul má také připravené vývody pro GPIO, některé z nich sdílejí i specifickou funkcionalitu. Příkladem může být vývod na řízení 4 svítivých diod indikujících datové přenosy na bezdrátovém rozhraní. [8]



Obrázek 4.5: Diagram propojení komponent uvnitř modulu DWM1000 [8]

4.6 Aplikace

Aplikace běžící na jednodeskovém minipočítači Raspberry Pi má dva podstatné cíle. Prvním z nich je interakce s uživatelem, zobrazování informací o stavu sítě a poloze hledaných uzlů. Druhý cíl je řízení sítě, zpracování příchozích zpráv a výpočet polohy hledaných uzlů na základě časových diferencí z kotevních uzlů.

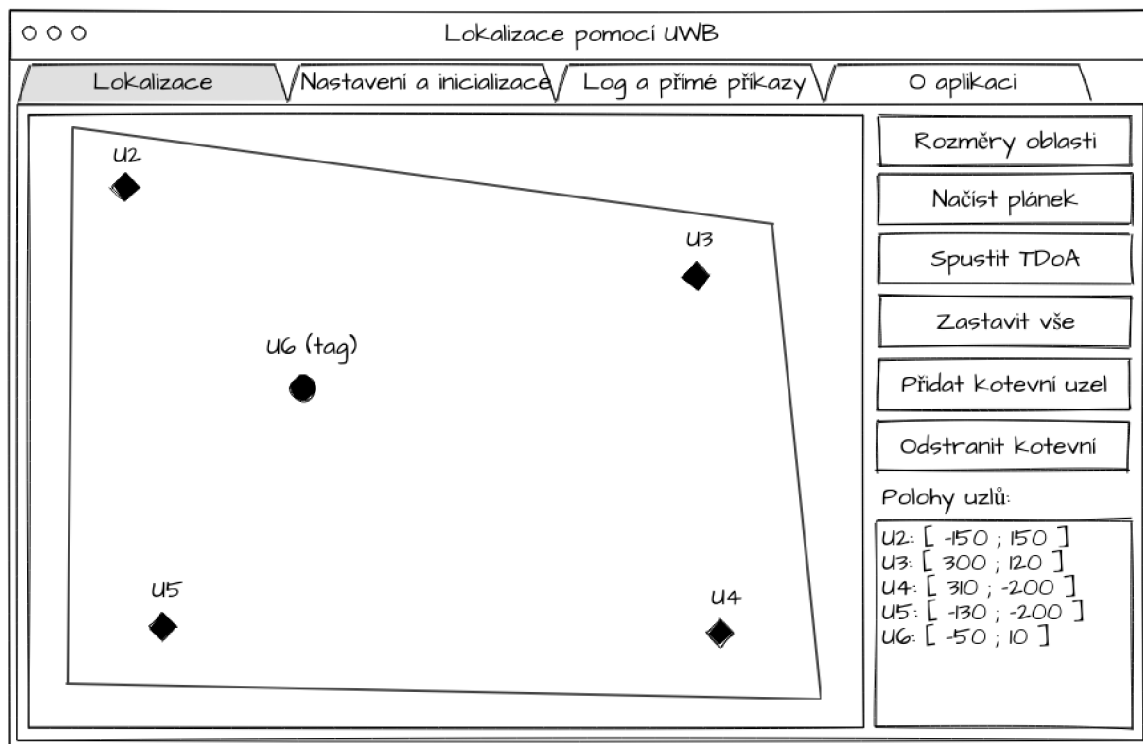
Uživatelské prostředí aplikace je rozděleno na čtyři jednotlivé sekce pomocí prvku panel. Tímto rozdělením vznikne pro každou sekci samostatná stránka obsahující potřebné formulářové prvky. Mezi stránkami lze následně přepínat, okno aplikace díky tomu zůstává relativně malé. Velikost okna aplikace byl důležitý parametr pro návrh, protože jsem počítal s využitím přenosného displeje o rozlišení 1366×768 pixelů.

První stránka, kterou ilustruje skica na obrázku 4.6, představuje hlavní gró celého konceptu. Nachází se zde totiž grafická reprezentace celé sensorové sítě a průběhu určování polohy tag uzlů. Hlavní část okna je věnovaná prvku, do kterého jsou zakresleny pozice všech kotevních uzlů a následně i všech úspěšně vyhledaných tag uzlů. Jako podklad může uživatel použít obrázek reprezentující plánec oblasti s rozmístěnými sensorovými uzly. Díky pláncu může uživatel získat lepší představu o rozmístění všech uzlů v rámci sítě. Zbylá část stránky obsahuje ovládací tlačítka pro přidání rozměrů oblasti diagramu, přidání pláncu, spuštění operace vyhledávání poloh zájmových uzlů, zastavení operace vyhledávání a následně tlačítka pro přidání a odebrání kotevních uzlů do, respektive z diagramu sítě. Po

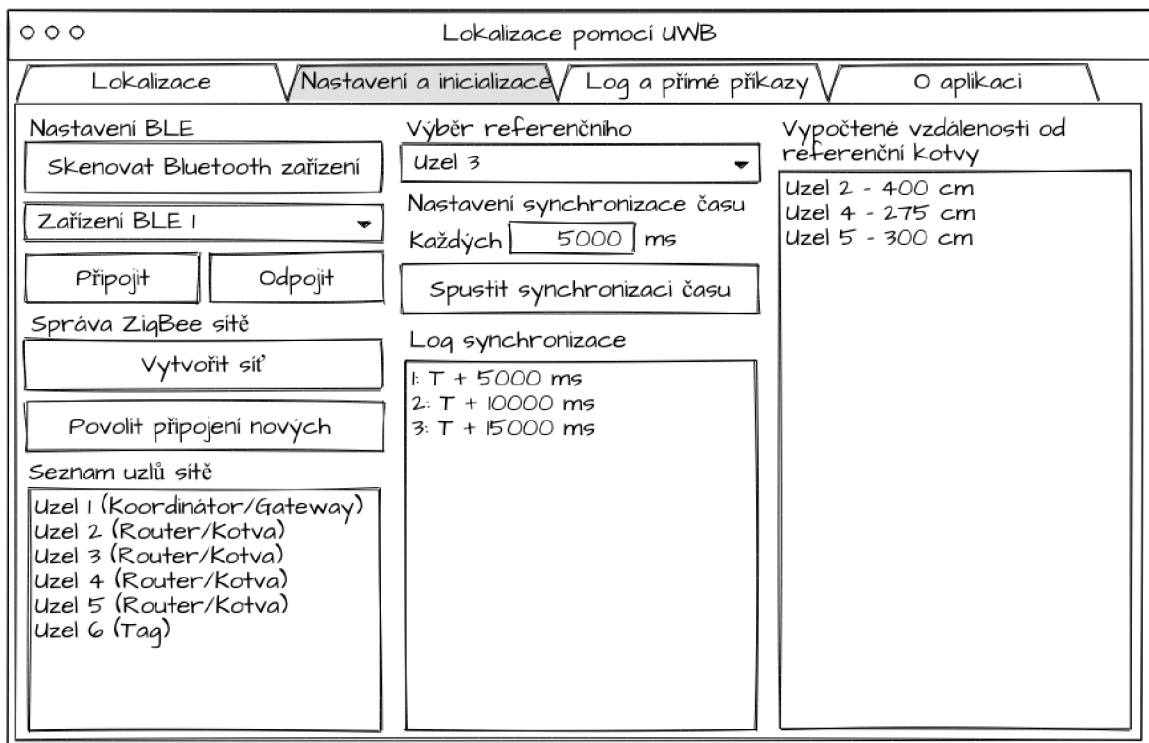
sledním prvkem na stránce je textové pole, do kterého jsou programově vepisovány pozice jednotlivých zobrazených uzlů vztažených ke středu diagramu.

Druhá stránka, jejíž grafické ztvárnění prezentuje obrázek 4.7, obsahuje všechny potřebné ovládací prvky. Tato stránka je pojmenována jako „Nastavení a inicializace“. Aby se všechny prvky na stránku vměstnaly, je využito rozložení do třech sloupců. První sloupec obsahuje ovládací prvky pro připojení přes rozhraní Bluetooth ke gateway uzlu. Další tlačítka obsluhují inicializaci ZigBee sítě s možností dodatečného připojení zařízení po vypršení poinitializačního intervalu. Posledním prvkem tohoto sloupce je textový seznam všech aktivních uzlů sítě. Druhý sloupec je věnován obsluze důležité části, kterou je distribuce a synchronizace času mezi kotevními uzly. V první části se nachází prvek selekce referenčního kotevního uzlu ze všech aktuálně dostupných. Následuje výběr intervalu zaslání synchronizačních zpráv a tlačítko pro spuštění či pozastavení synchronizace. Jako poslední se zde nachází textové pole s logem všech uskutečněných synchronizací. Poslední sloupec obsahuje pouze textové pole, do kterého jsou programově vypočítány vzdálenosti od referenčního kotevního uzlu ke všem zbývajícím kotevním uzlům.

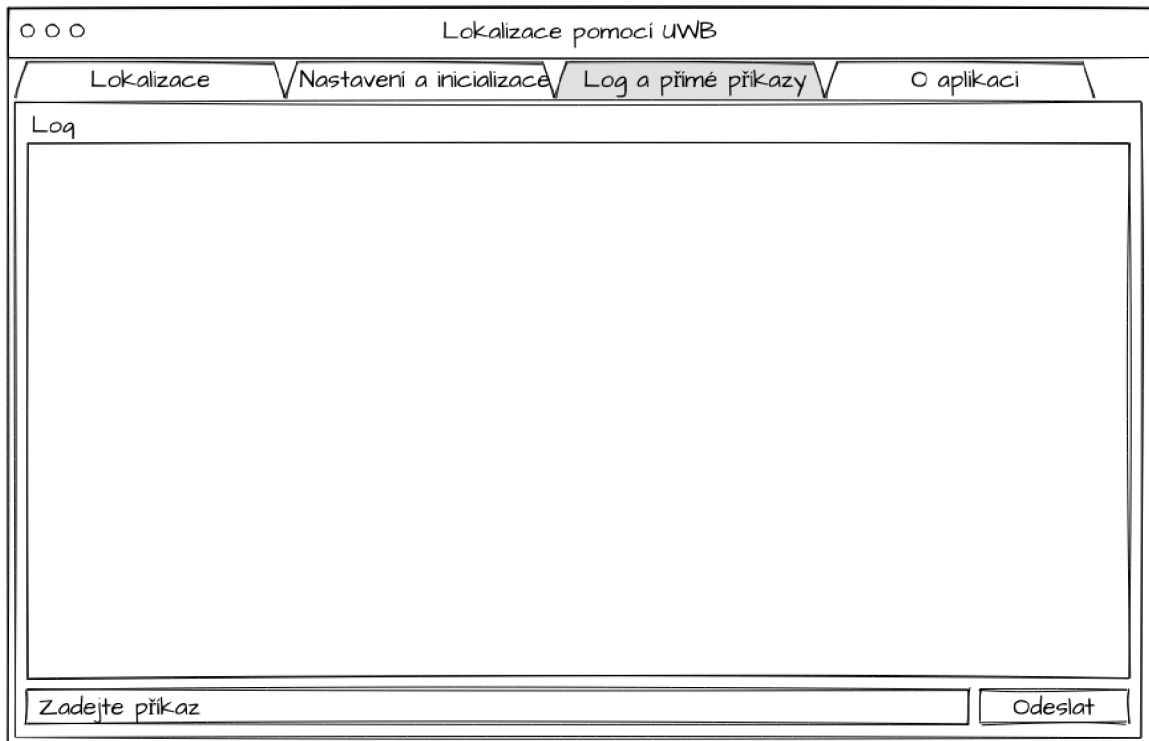
Třetí stránka obsahuje pouze velké textové pole, které je určené pro logování všech akcí v čase a také pole s tlačítkem pro zaslání příkazu do sítě přes rozhraní Bluetooth. Skicu, která reprezentuje rozložení této stránky zobrazuje obrázek 4.8. Poslední stránka, pro kterou již neexistuje skicový návrh obsahuje pouze informace o samotné aplikaci a jejím autorovi.



Obrázek 4.6: Skica stránky pro ovládání lokalizace zobrazení výsledků



Obrázek 4.7: Skica návrh podoby stránky s inicializací a nastavením



Obrázek 4.8: Skica s návrhem logovací stránky

Kapitola 5

Obvodová realizace sensorového uzlu

V této podkapitole je podrobně popsána realizace desky sensorového uzlu. Jsou zde uvedeny parametry a požadavky hlavních komponent. V první řadě se dostává pozornost části, bez které by deska vůbec nefungovala, touto částí je napájení. Deska je napájena z lithium-polymerového akumulátoru, který je nabíjen přes microUSB port. Pro zásobování ostatních komponent elektrickou energií ovšem uvedené zdroje nejsou vhodné a tuto situaci je potřeba vyřešit.

Další část textu se zabývá hlavními komponentami, jež obě využívají bezdrátové rozhraní. Jednou z těchto komponent je samotné srdce celé desky, mikrokontrolér Kinetis KW41Z, u něž je na rozdíl od modulu DWM1000 nutné realizovat bezdrátovou část přímo na desce plošných spojů. Návrh radiofrekvenční části je složitá disciplína, naštěstí jsou k dispozici referenční designy, které mnoho věcí usnadní. Poslední část textu se následně zabývá parametry desky a popisuje její jednotlivé vrstvy.

5.1 Regulace napětí

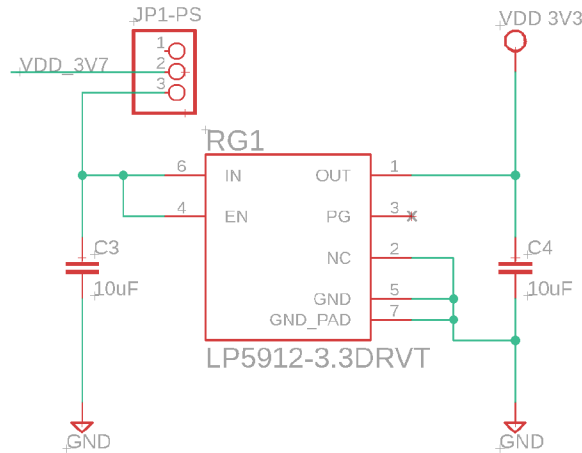
Napájení komponent na desce sensorového uzlu je řešeno větví o napětí 3.3 V. Jako primární napájecí zdroj je ovšem zvolen lithium-polymerový akumulátor, jehož napětí v průběhu používání kolísá z plně nabitého stavu o hodnotě 4.2 V až do stavu vybití na hodnotu 3 V. Takto velký rozsah je pro provoz sensorového uzlu nepřijatelný, a to i z důvodů omezených provozních rozsahů komponent, kdy například UWB modul DWM1000 toleruje napájecí napětí maximálně do hodnoty 3.6 V.

Problém, jak udržet napájecí napětí na přijatelné úrovni lze snadno vyřešit několika způsoby. Jedním z nich je použití regulátoru napětí (LDO – low-dropout regulator), který udržuje napětí na předem výrobcem udané hodnotě. Samotná komponenta, ve verzi určené pro povrchovou montáž, nabývá velice kompaktních rozměrů a je proto vhodnou volbou pro návrh desek s omezenými rozměry. Regulátor pro svojí správnou funkci následně potřebuje pouze dva kondenzátory, které slouží pro filtraci a stabilizaci na vstupu a výstupu.

Jak již bylo uvedeno v kapitole věnované návrhu, byl použit regulátor napětí s nízkým úbytkem LP5912-3.3DRVT od výrobce Texas Instruments. Tato komponenta je v provedení low-noise, tedy s nízkým až nulovým výstupním šumem. Přítomnost šumu v napěťové kaskádě by totiž mohla mít významný vliv na fungování UWB rádia. Mohlo by totiž do-

cházet ke zvýšení chybovosti příjmu a snížení přesnosti určování časového okamžiku přijetí zprávy. To vše by následně vedlo ke zhoršení přesnosti určování přibližné polohy uzlů.

Maximální výstupní proud poskytovaný regulátorem dosahuje hodnoty 500 mA, to odpovídá součtu špičkového odběru všech ostatních komponent s drobnou rezervou. Pro filtraci na vstupu a výstupu regulátoru jsou použity dva tantalové kondenzátory o hodnotě 10 μF . Vstup regulátoru je spojen pomocí mechanické propojky (jumper), sloužící jako spínač, s obvodem řízení nabíjení lithium-polymerových akumulátorů. Schéma zapojení regulátoru napětí s patřičnými filtračními kondenzátory a spínací mechanickou propojkou se nachází na obrázku 5.1.



Obrázek 5.1: Zapojení regulátoru napětí

5.2 Nabíjení akumulátoru

Použitý lithium-polymerový akumulátor již obsahuje vestavěnou ochranu proti přebíjení, hlubokému vybití a zkratu, proto již není potřeba na sensorový uzel umísťovat žádné další ochranné obvody. Pro řízení nabíjení akumulátoru jsem zvolil integrovaný obvod BQ24075TRGTR, který jako zdroj používá napětí o hodnotě 5 V z microUSB konektoru.

Aby se nabíjený akumulátor neopotřeboval rychleji, než je nezbytně nutné, umožňuje obvod pro řízení nabíjení nastavit maximální nabíjecí proud I_{CHG} . Jeho nastavení je řešené pomocí vhodně zvoleného rezistoru R_{ISET} , jehož hodnotu lze vypočítat podle vzorce 5.1 [13]. K_{ISET} reprezentuje konstantu z manuálu nabíjecího čipu o hodnotě 890 $\text{A}\Omega$. R_{ISET} odpovídá rezistoru R12 ze schématu zapojení 5.3.

Podle manuálu k akumulátoru je doporučená velikost konstantního nabíjecího proudu rovna hodnotě $0.2C$ [3]. C v tomto případě značí kapacitu měřenou v mAh. Protože použitý akumulátor má kapacitu rovnou 1050 mAh, je maximální doporučený konstantní nabíjecí proud roven 210 mA. Dosazením do vzorce 5.1 vychází hodnota 4238 Ω , které odpovídá nejbližší katalogová hodnota rezistoru 4.7 $\text{k}\Omega$.

$$R_{ISET} = \frac{K_{ISET}}{I_{CHG}} = \frac{890}{0.210} = 4238\Omega \quad (5.1)$$

Další užitečná funkce související se zdravím akumulátor, kterou nabíjecí obvod nabízí, je možnost nastavení interního časovače pro nabíjení. Jedná se o bezpečnostní mechanismus, který ukončí nabíjení v případě, že dojde k překročení časová kvanta určeného rezistorem

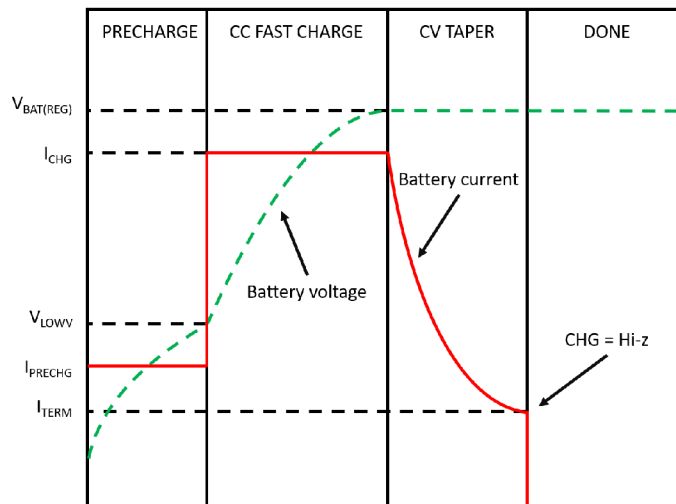
R_{TMR} . To, že se akumulátor do jisté doby plně nenabije může značit, že je defektní nebo na konci své životnosti. Pokud by se i nadále pokračovalo v dodávkách energie, mohlo by dojít k poškození jak akumulátoru, tak hlavně samotného senzorového uzlu. Pro účely ochrany obvod obsahuje rovnou dva časovače, a to pro fázi přednabíjení (pre-charge) tak i pro samotné hlavní nabíjení (fast-charge), které trvá desetkrát déle. [13]

Z manuálu akumulátoru lze vyčíst, že standardní doba nabíjení nepřesahuje 8 hodin [3]. Díky vztahu 5.2 lze vypočítat hodnotu rezistoru R_{TMR} , který odpovídá rezistoru R11 ve schématu zapojení. Konstanta K_{TMR} z manuálu nabíjecího obvodu nabývá hodnoty 45 s/k Ω . Pro dosažení je ještě nutné převést časový interval z hodin na vteřiny, tj. 8h = 28800 s. Pro výsledek o hodnotě 64 k Ω odpovídá nejbližší vyšší katalogová rezistivita odporu velikosti 64.9 k Ω .

$$R_{TMR} = \frac{t_{MAXCHG}}{10 \times K_{TMR}} = \frac{28800}{450} = 64k\Omega \quad (5.2)$$

Celý průběh nabíjecího cyklu ilustruje obrázek 5.2. V něm si lze povšimnout rozdělení celého cyklu na čtyři fáze. V první fázi je akumulátor nabíjen omezeným proudem I_{PRECHG} , který je vyjádřen vztahem 5.3, kde K_{PRECHG} je rovno hodnotě 88 a R_{ISET} je již vypočítaná hodnota rezistoru. Pomalé nabíjení trvá, dokud napětí na akumulátoru nepřesáhne hodnotu $V_{LOWV} = 3V$. Jakmile je napětí vyšší, vnitřní stav přepne na nabíjení maximálním proudem I_{CHG} . Fáze rychlého nabíjení trvá do doby, dokud je napětí akumulátoru nižší než $V_{BAT(REG)} = 4.2V$. Třetí fáze poté udržuje napětí na konstantní úrovni 4.2 V a pomalu snižuje nabíjecí proud. Pokud proud klesne pod úroveň $I_{TERM} = 0.1 \times I_{CHG}$ je akumulátor plně nabit a proces nabíjení končí.

$$I_{PRECHG} = \frac{K_{PRECHG}}{R_{ISET}} = \frac{88}{4700} = 19mA \quad (5.3)$$



Obrázek 5.2: Průběh nabíjení akumulátoru pomocí integrovaného obvodu BQ24075TRGTR [13]

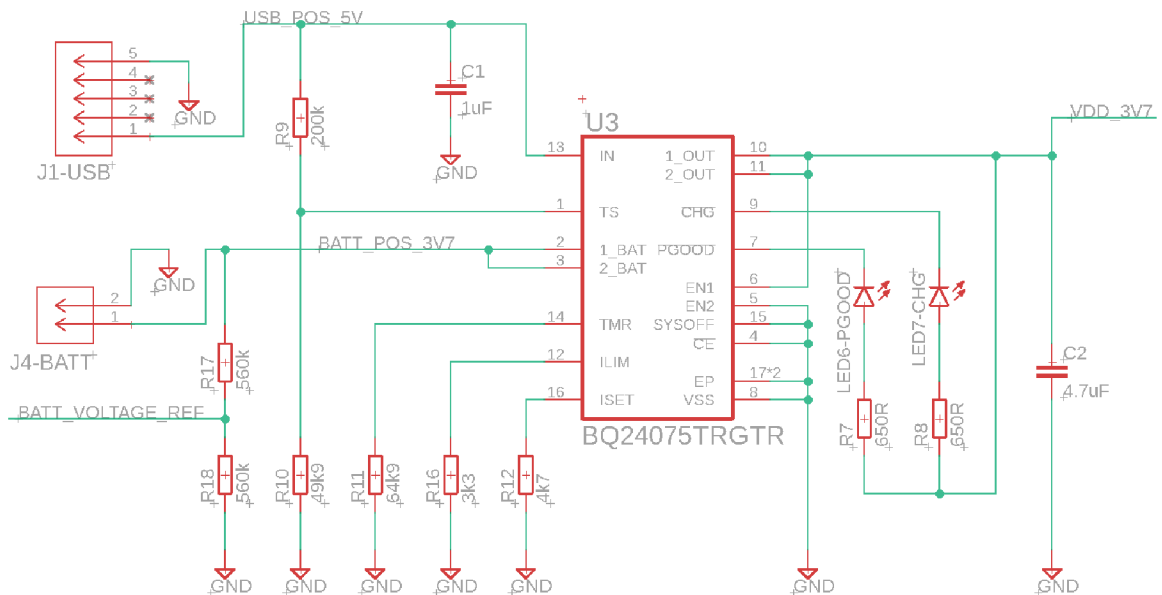
Poslední nastavitelnou funkcí je omezování maximálního vstupního proudu do nabíječky. Jedná se o maximální proud, který může být odebírán z microUSB portu pro potřeby nabíjení a vlastní spotřebu systému. Protože plánované použití senzorového uzlu počítá s provozem buďto na akumulátor nebo na napájení z USB portu, zvolil jsem pro maximální

proud hodnotu 500 mA. Pro tuto hodnotu je nastaven i rezistor R_{TMR} odpovídající součástce R16 ve schématu zapojení. Ve vzorci 5.4 je konstanta K_{ILIM} rovna hodnotě 1600 AΩ [13]. Po dosazení je hodnota rezistoru R_{TMR} rovna 3200 Ω a po převedení na nejbližší vyšší katalogovou hodnotu 3.3 kΩ.

$$R_{ILIM} = \frac{K_{ILIM}}{I_{IN-MAX}} = \frac{1600}{0.5} = 3200\Omega \quad (5.4)$$

Protože integrovaný obvod BQ24075TRGTR má možnost ovlivňovat proces nabíjení na základě teploty akumulátoru, je nutné buďto připojit termistor nebo využít předdefinované hodnoty rezistorů pro děličku napětí. Tato dělička symbolizuje chování termistoru s konstantní teplotou. Doporučené hodnoty rezistoru jsou následující R10 = 49.9 kΩ a R9 = 200 kΩ. [13]

Aby bylo možné uživatelsky určit průběh nabíjení, přidal jsem na příslušnou část desky sensorového uzlu indikační diody. LED označená jako PGOOD indikuje přítomnost napájení z microUSB konektoru, které je v přijatelných hodnotách pro nabíjení akumulátoru. LED s označením CHG svítí v průběhu nabíjení akumulátoru a zhasne po jeho plném nabití. Pokud tato svítivá dioda bliká, značí to překročení vymezeného časové kvanta pro jeden z časovačů nabíjení.

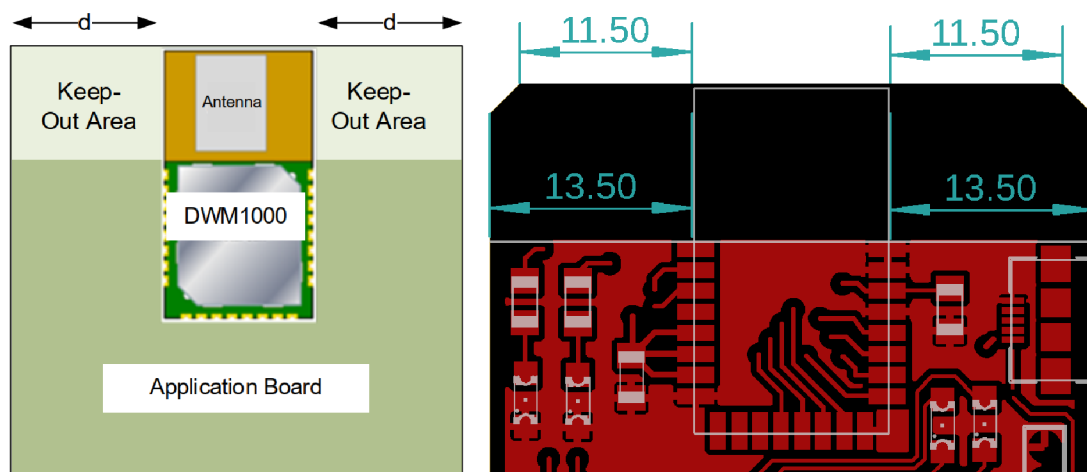


Obrázek 5.3: Zapojení obvodu pro nabíjení akumulátoru

5.3 UWB modul

Využitím již hotového modulu namísto návrhu designu radiofrekvenční části pro ultra širokopásmovou komunikaci jsem dosáhl podstatného zjednodušení celého procesu tvorby desky sensorového uzlu. Díky tomu jsem snížil riziko vzniku chyb, které by mohly mít neblahé následky na kvalitu signálu a následného určování polohy. Podle dokumentace k modulu DWM1000 bylo nutné dodržet pouze určitá doporučení týkajících se umístění modulu na výslednou desku plošných spojů a také parametrů některých signálových cest.

Pro finální umístění modulu na desku platí tři jasná pravidla. Prvním z nich je, že oblast okolo antény nesmí obsahovat žádný materiál, který není transparentní pro vyzařované elektromagnetické záření. Tato oblast je na obrázku 5.4 uvedena jako „Keep-Out Area“ a platí jak pro desku, tak i pro jakékoliv další objekty v podobě vodičů nebo součástí obalu sensorového uzlu. Druhé pravidlo udává optimální horizontální umístění modulu, který by měl být od okrajů hlavní desky vzdálen minimálně 10 mm. Tato hodnota je optimální pro vertikální polarizaci vyzařovaného signálu, jehož úhel vyzařování se snižuje se vzrůstající vzdáleností d . Poslední pravidlo je neméně důležité pro kvalitu radiofrekvenčního návrhu a říká, že veškeré ostatní nevyužité oblasti hlavní desky musí být pokryté zemnicí plochou.



Obrázek 5.4: Vlevo doporučené umístění modulu DWM1000 dle manuálu [9], vpravo výsledné umístění modulu na desce plošných spojů (rozměry jsou udávány v mm)

Modul komunikuje s mikrokontrolérem přes vyvedené rozhraní SPI. Pokud není toto sériové rozhraní sdílené mezi dalšími periferiemi, není na něj z pohledu modulu kladen žádný speciální požadavek. V případě desky sensorového uzlu se jedná o přímé propojení modulu s mikrokontrolérem. Jako přímý propoj je řešen i signál RST, sloužící k obnově výchozího stavu modulu, a také signál WAKEPU, který je určen pro probuzení modulu ze stavu s nízkou spotřebou. Posledním signálem, který již vyžaduje ošetření pomocí pull-down rezistoru, je signál IRQ. Tento signál je spojen se zemí pomocí rezistoru R_6 o hodnotě 100 k Ω a slouží k přerušování běhu mikrokontroléru a spuštění obslužné rutiny reagující na nastalou událost.

Mezi dalšími vývody UWB modulu jsou i piny pro obecné požití. Čtyři z nich mohou sloužit jako indikace událostí na bezdrátovém rozhraní. Pro tuto funkci jsem je osadil čtyřmi svítivými diodami, které indikují příchozí a odchozí vysílání, správně přijatý datový rámec a také sekvenci oddělující jednotlivé příchozí rámce. Vizualní indikace takovýchto událostí je vysoce užitečná při vývoji a ladění firmware sensorového uzlu.

5.4 mikrokontrolér a RF část

Technická dokumentace k mikrokontroléru Kinetis KW41Z důrazně doporučuje držet se striktně referenčních návrhů pro umístování komponent v přímé blízkosti pouzdra čipu [20]. Z tohoto pohledu je nejdůležitější správně umístit vysokofrekvenční součástky jako jsou krystaly a dále také dodržet navržený design radiofrekvenční části. Jedině tak je totiž

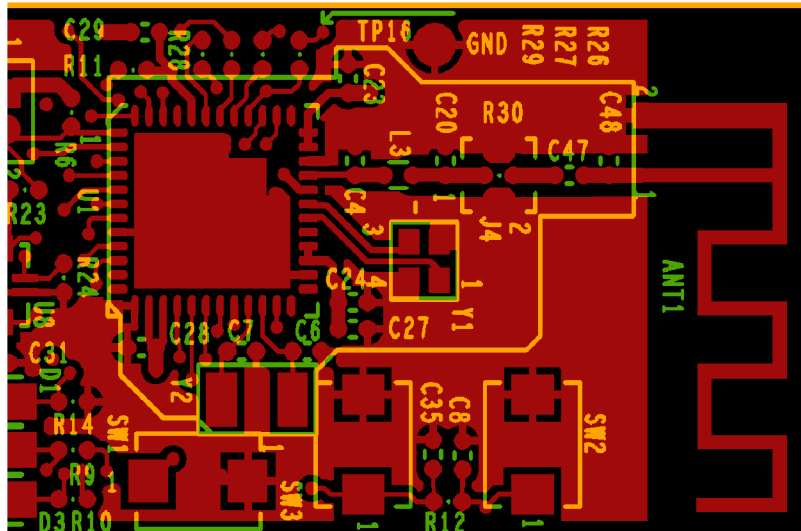
možné zajisti správné fungování bezdrátového rozhraní pro standard 802.15.4 a technologii Bluetooth Low Energy.

Na umístění krystalů se vztahují dva požadavky. První z nich říká, že délka cesty od příslušných vývodů čipu mikrokontroléru až po samotnou plošku krystalu musí být co možná nejkratší [20]. Druhý požadavek se vztahuje k nepřítomnosti vysokofrekvenčních cest v blízkosti krystalu, které by mohly interferovat s hodinovým signálem jím generovaným. Mezi takové cesty patří jak signálové spoje s rychlým přepínáním úrovní, tak i samotná přenosová linka antény. Kvůli výše popsaným důvodům jsem se rozhodl umístit 32 MHz krystal přesně podle referenčního designu.

Podle referenčního designu jsem postupoval i v případě návrhu radiofrekvenční části. Zde jsem se ovšem rozhodl neumisťovat volitelné kondenzátory C20 a C48, jež jsou vidět na obrázku 5.5, a které nejsou osazeny ani na referenční desce, kterou jsem měl k dispozici. Další změna, kterou jsem provedl byla úprava šířky mikropásku sloužícího jako přenosová linka signálu jdoucího k anténě. Protože jsem oproti referenčnímu řešení provedl změny ve vlastnostech desky plošných spojů, které budou podrobně popsány v dalším textu, musel jsem šířku upravit pomocí rovnice 5.5 abych zachoval impedanci mikropásku na hodnotě 50 Ω [14].

$$w = \frac{7.48 \times h}{e^{(Z_0 \frac{\sqrt{\epsilon_r + 1.41}}{87})}} - 1.25 \times t \quad (5.5)$$

Po dosažení do výše uvedeného vztahu, kde Z_0 je požadovaná impedance 50 Ω, t je výška mikropásku rovna 0.018 mm, h je výška dielektrika rovna 0.254 mm a ϵ_r je relativní permitivita dielektrika o velikosti 4.4, jsem získal novou šířku mikropásku o hodnotě 0.453 mm.

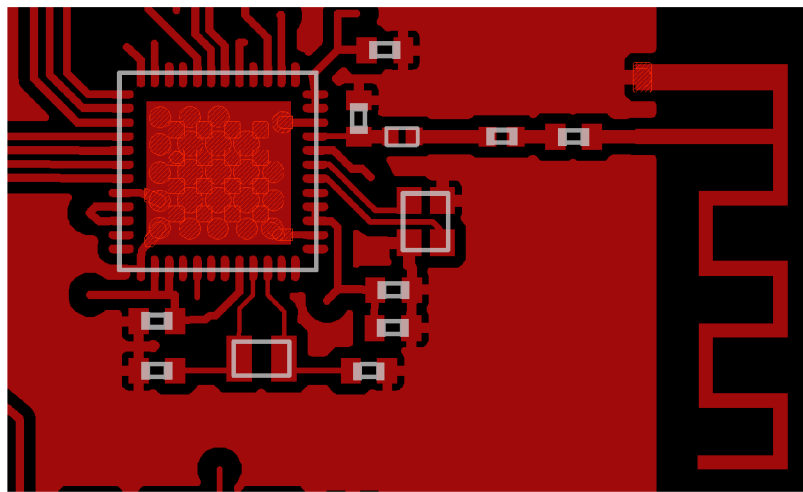


Obrázek 5.5: Výřez vrchní vrstvy referenčního návrhu pro mikrokontrolér Kinetis KW41Z obsahující polygonem vyznačenou kritickou část obvodu (zdroj: referenční materiály pro desku USB-KW41Z²)

²Adresa URL pro stažení souborů referenčního designu pro desku USB-KW41Z : <https://www.nxp.com/webapp/Download?colCode=USB-KW41Z-DESIGNFILES>

Pro anténu jsem zvolil kompaktní design ve formě meandrové antény, jejíž rozměry vychází z doporučení výrobce NXP pro tvorbu planárních antén v pásmu 2.4 GHz [23]. Ke správně navrženému designu radiofrekvenční části patří, stejně jako je zmiňováno u modulu DWM1000, využití zemnicí plochy všude tam, kde nejsou vedeny žádné signály. Dalším důležitým prvkem podle technické dokumentace je využití velkého množství průchodek zemnicími plochami, a to v části, kterou vede přenosová linka k anténě, tak i v blízkosti antény samotné.

Pro své napájení vyžaduje mikrokontrolér využití několika lokálních blokovacích kondenzátorů, které mu dodávají potřebné stabilní napětí i při špičkových odběrech a také filtrují případné rušení z okolí. Tyto kondenzátory se na obrázku 5.6 nacházejí v bezprostředním okolí pouzdra čipu a jsou propojeny s jednotlivými skupinami napájecích vstupů. Jednou ze skupin je napájení RF části, která je od zbytku napájecích vstupů oddělena právě kvůli potřebné stabilitě.



Obrázek 5.6: Výřez vrchní vrstvy designu desky sensorového uzlu na kterém lze vidět podobnost s referenčním designem rozložení komponent a návrhu RF části

Z mikrokontroléru jsou vyvedeny signály rozhraní SPI, které slouží pro komunikaci s modulem DWM100. Dále je vyvedeno rozhraní UART pro potřeby ladění a také pro budoucí možné využití při výměně dat s periferiemi. Pro programování vnitřní paměti slouží rozhraní SWD, přes které probíhá hlavní ladění vyvíjeného firmware. Nakonec jsem ještě k mikrokontroléru připojil dvě stavové svítivé diody.

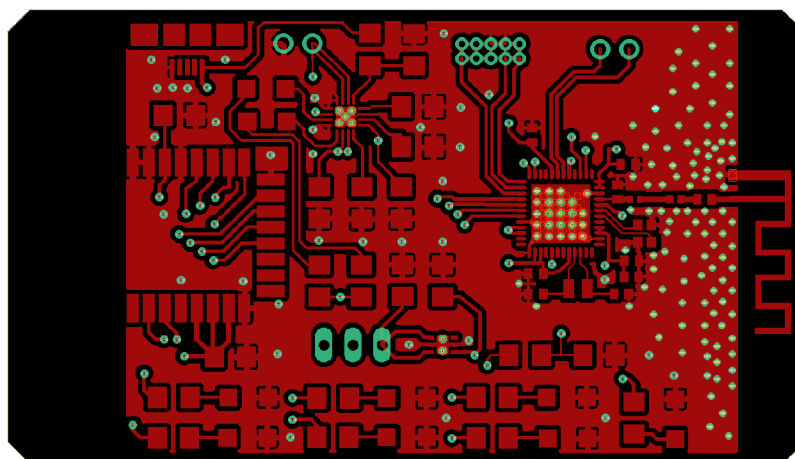
5.5 Výsledná deska plošných spojů a její parametry

Ač samotná deska není příliš složitá z pohledu návrhu, byla pro její tvorbu zvolena technologie čtyřvrstvého DSP. Důvod pro toto rozhodnutí pramení z doporučení výrobce NXP, kdy čtyřvrstvá deska plošných spojů nabízí neoptimálnější charakteristiky pro bezdrátové rozhraní [20]. Materiál, ze kterého se deska skládá je standardní epoxidový laminát se skleněnou výztuží FR4. DSP dále obsahuje množství prokovených průchodek skrz celou desku. Využití slepých či pohřbených průchodek nebylo potřeba a jejich použití by zbytečně prodražilo výrobu. Výrobní tloušťky jednotlivých vrstev mědi a dielektrika znázorňuje obrázek 5.7 a vychází přesně z doporučení návrhu pro obvody mikrokontrolérem Kinetis KW41Z. Deska má rozměry 70×40 mm a její tloušťka je 1.58 mm.



Obrázek 5.7: Parametry výsledné desky plošných spojů

Na první vrstvě desky plošných spojů (obrázek 5.8) jsou osazeny všechny použité komponenty. Je zde rozvedena většina signálových cest a také se zde nachází přenosová linka pro rádiové rozhraní a samotná leptaná meandrová anténa. Zbylá nevyužitá místa mimo části určené pro antény jsou pokryta zemnicím polygonem.

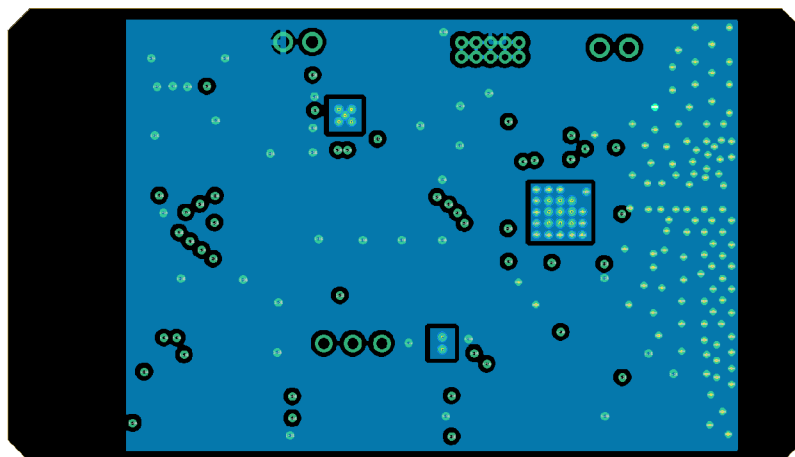


Obrázek 5.8: První vrstva DPS: TOP

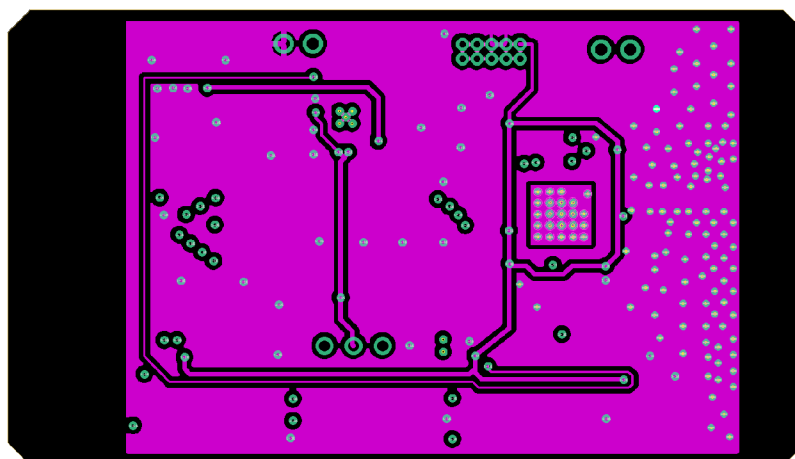
Druhá vrstva (obrázek 5.9) obsahuje primární polygon zemnicí plochy, který rozvádí signál GND po celé ploše desky.

Třetí vrstva (obrázek 5.10) slouží pro rozvody napájení. Nachází se zde hlavní napájecí větev 3.3 V, mimo ní také rozvod napájení z baterie a microUSB portu. Zbytek plochy zabírá sekundární zemnicí polygon.

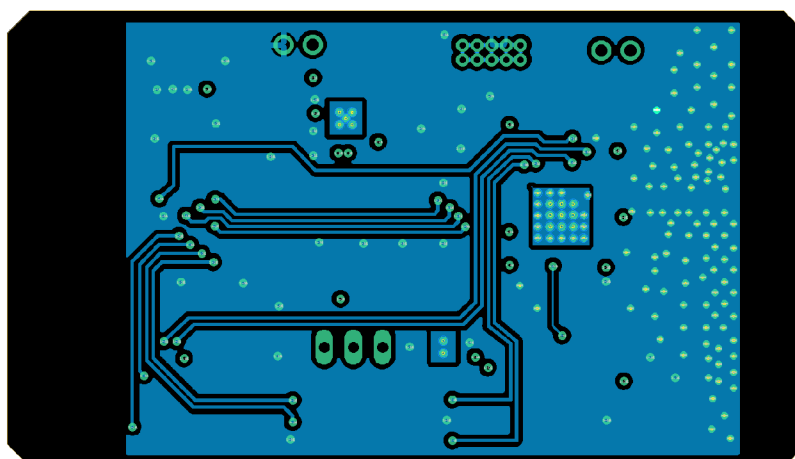
Poslední spodní vrstva (obrázek 5.11) rozvádí zbylé signály, které nebylo možné nebo jednoduché vést ve vrchní vrstvě. Stejně jako u ostatních vrstev i zde je nevyužitá část plochy určená pro rozvod zemnicího polygonu.



Obrázek 5.9: Druhá vrstva DPS: GND



Obrázek 5.10: Třetí vrstva DPS: POWER



Obrázek 5.11: Poslední vrstva DPS: BOTTOM

Kapitola 6

Uživatelská aplikace a obslužný firmware

V této kapitole je podrobně rozebrána implementace firmaware jednotlivých typů uzlů použitých pro tvorbu sensorové sítě. Část textu je vyhrazena pro popis obsluhy UWB modulu DWM1000, který je používán jak na kotevních uzlech, tak i na uzlech typu tag. Následně je podrobně popsána i uživatelská aplikace, která kromě zobrazování výsledků provádí i výpočty synchronizace hodinových signálů a výpočty pozic uzlů typu tag v rámci sítě.

6.1 Firmware sensorového uzlu

Implementace firmwaru pro všechny tři typy uzlů probíhala v integrovaném vývojovém prostředí MCUXpresso od společnosti NXP. Pro toto vývojové prostředí a mikrokontrolér Kinetis KW41Z jsem využil dostupnou sadu vývojových nástrojů (SDK) ve verzi 2.2.2¹. Toto SDK obsahuje soubor modulů a funkcí, které abstrahují přímou manipulaci s registry zařízení. Krom toho obsahují vývojové nástroje i plnou implementaci pro protokolové zásobníky technologií ZigBee a Bluetooth Low Energy. Dalším nástrojem, který jsem využil pro implementaci firmware kotevního a gateway uzlu, byl operační systém reálného času FreeRTOS ve verzi 9.0.0. Operační systém reálného času (RTOS) umožňuje tvorbu, běh a plánování procesů, dále provádí správu paměti a poskytuje prostředky pro použití front, zásobníků a časovačů. V případě kotevního a gateway uzlu bylo použití RTOS z důvodů velké komplexnosti manipulace s protokolovými zásobníky nevyhnutelné.

Protože všechny tři implementované verze firmware běží na stejném návrhu desky sensorového uzlu, použil jsem pro ně i stejné nastavení taktovací frekvence a nastavení pinů mikrokontroléru. Taktovací frekvence byla zvolena maximální možná a to 48 MHz. Nastavení pinů vychází z návrhu desky sensorového uzlu a zahrnuje aktivaci sériového rozhraní SPI, vstupní a výstupní piny pro obsluhu modulu DWM1000 (DW_RST, DW_WAKE, DW_IRQ) a také dva výstupní piny pro vizuální signalizaci pomocí LED (STATUS_LED_A, STATUS_LED_B).

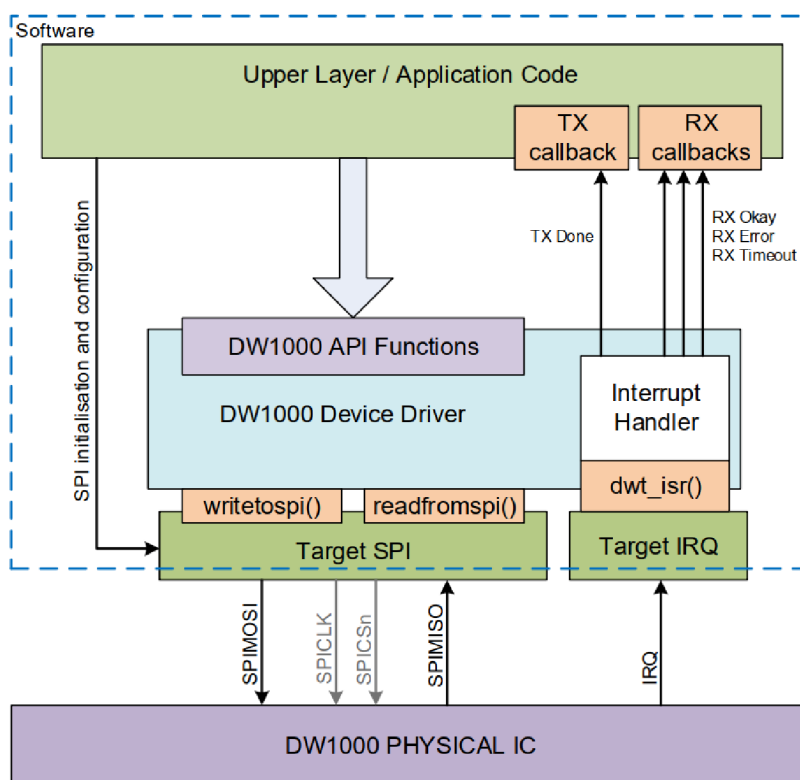
¹SDK pro mikrokontrolér Kinets KW41Z dostupné ze stránek společnosti NXP po bezplatné registraci <https://mcuxpresso.nxp.com/en/builder?hw=MKW41Z512xxx4>

6.1.1 Obsluha UWB modulu

Protože se modul DWM1000 používá jak na uzlech typu tag, tak i na kotevních uzlech rozhodl jsem se tuto část textu vyhradit jeho použití. Až na drobné odlišnosti totiž oba uzly sdílejí stejnou část kódu. Odlišnosti mezi nimi spočívajících například ve využití aktivního čekání na tagu oproti suspendaci procesu na kotevním uzlu při využití funkce `sleepMS`. Jedná se ovšem o detaily, které nezpůsobují rozdílné chování mezi implementacemi.

Aplikační framework

Jak již bylo zmíněno v předchozích kapitolách, komunikace s modulem DWM1000 probíhá pomocí sériového rozhraní SPI. Toto rozhraní se používá pro přímý přístup k registrům zařízení, kdy lze pomocí zápisu provádět konfiguraci a pomocí čtení získávat příslušné hodnoty. Při nastalé události jako je například příjem nového rámce je vygenerováno přerušení pomocí příslušného výstupního pinu, které je potřeba na mikrokontroléru zpracovat. Pro pohodlnější práci s modulem DWM1000 je ovšem vhodnější použít existující framework od společnosti Decawave². Tento framework, použitý ve verzi 5.1.0, poskytuje aplikační rozhraní (API), které abstrahuje přímou práci s rozhraním SPI a umožňuje díky tomu podstatně urychlit vývoj software.



Obrázek 6.1: Softwarový framework pro práci s modulem DWM1000 [7]

Framework pro modul DWM1000 se skládá ze třech podstatných částí, tak jak lze vidět na obrázku 6.1. První část tvoří samotné aplikační rozhraní, které je dostupné pro použití v hlavní aplikaci běžící na mikrokontroléru. Další část frameworku tvoří obsluha přerušení,

²https://www.decawave.com/wp-content/uploads/2020/04/DWS1000_ExampleCode_v1_0_1.zip

která se stará o registraci a spuštění obslužných funkcí pro události typu příjem nebo odeslání rámce na UWB rozhraní. Spuštění těchto procedur se děje na základě volání funkce `dwt_isr`, které je řešeno prostřednictvím třetí podstatné části frameworku, kterou je ovladač zařízení. Ovladač zařízení pro modul DWM1000 je soubor platformě závislých funkcí, které nejsou přímou součástí frameworku a které je nutné na cílový mikrokontrolér implementovat. Tato součást rovněž zahrnuje implementaci funkcí pro komunikaci s rozhraním SPI, kterými jsou `writetospi` a `readfromspi`. Implementace ovladače zařízení pro mikrokontrolér Kinetis KW41Z je v rámci této práce řešena v několika souborech, které se shodně nacházejí v adresáři `decaplatform` v příslušných projektech.

Inicializace modulu

Inicializace modulu probíhá v posloupnosti několika kroků a je zastřešována funkcí `startUWBRadio`. Tato funkce jako své povinné argumenty přebírá ukazatele na funkce obsluhy událostí úspěšného přijetí rámce, úspěšného odeslání rámce, přijetí chybných dat a také vypršení časovače čekání na přijetí plánovaného rámce. Posledním parametrem je příznak odložení startu přijímání dat na UWB rozhraní.

Prvním krokem inicializace je nastavení obsluhy přerušování tak, aby byla funkce `dwt_isr` volána vždy, když nastane přerušování od UWB modulu. Následně je pomocí funkce `init_DW1000_SPI` provedena inicializace sériového rozhraní SPI. SPI je nastaveno tak, aby vzorkovalo data na náběžné hraně hodinového signálu s klidovou hodnotou rovnou logické nule. Pro každý rámec je nastaven počet přenášených bitů na osm s tím, že první bit v pořadí je nejvýznamnější bit (MSB).

Dalším krokem inicializace je konfigurace pinu sloužícího k restartu modulu DWM1000 s následným zápisem logické nuly po dobu 1 ms. Tímto nastavením signálu RST dojde k provedení správné sekvence startu čipu DW1000 uvnitř modulu. Následně je ještě provedena konfigurace pinu IRQ a povolení přerušování pro tento pin.

Dalším krokem v inicializaci je volání funkce aplikačního rozhraní frameworku `dwt_initialise`, která provede přes rozhraní SPI kontrolu funkčnosti modulu a spustí v něm podprogram, který se stará o přesnost měření času přijatých a odeslaných rámců. Dále je ještě provedeno nastavení identifikátoru sítě (PAN ID) na předem zvolenou hodnotu `0xABCD`.

Před případným spuštěním příjmu rámců, je ještě nutné nastavit parametry UWB rozhraní. To se provádí pomocí funkce aplikačního rozhraní `dwt_configure`, která jako svůj argument přebírá ukazatel na konfigurační strukturu. Tato konfigurační struktura obsahuje například volby pro nastavení kanálu, módu hlavičky fyzické vrstvy nebo délku preamble. Pro účely této práce jsem hodnoty konfigurační struktury ponechal na doporučených parametrech.

V předposlední části inicializace jsem se rozhodl využít dvojité vyrovnávací paměti pro příjem rámců na UWB rozhraní pomocí funkce `dwt_setdblrxbuffmode`. Dvojitá vyrovnávací paměť totiž umožňuje přijmout druhý rámec v okamžiku mezi přijetím prvního rámce a potvrzením jeho zpracování. Díky tomu je snížena pravděpodobnost nezachycení další zprávy o 50 %.

Poslední část inicializace provádí zaevidování argumentů funkce `startUWBRadio` pro obsluhu jednotlivých událostí týkajících se správného příjmu a odeslání rámce, vypršení časovače nebo příchodu chybného rámce. Pokud je některý z argumentů nastaven na hodnotu `NULL`, nebude obsluha dané události využita. Pokud u funkce `startUWBRadio` nebyl příznak odloženého příjmu nastaven, je na konci inicializace spuštěno přijímání příchozích rámců.

Funkce pro práci s rámcí

Protože aplikační framework dodávaný k modulu DWM1000 neobsahuje rozhraní pro práci s rámcí linkové vrstvy, vytvořil jsem si pro tyto účely sadu funkcí, jež se nacházejí v souboru `uwb_instance.c`. Tyto funkce slouží jako dodatečné API umožňují operace skládání a parsování se všemi rámcí linkové vrstvy. V kontextu této práce jsou podstatné funkce `sendBlinkMsg`, `sendTimeSyncMsg` a `parseRxFrameMAC`.

Funkce `sendBlinkMsg` a `sendTimeSyncMsg` se chovají obdobně, obě slouží pro skládání rámce typu Multipurpose a liší se pouze ve funkčním kódu vkládaném do datové části rámce, tak jak je popsáno v návrhu komunikace 4.2.1. O sestavení příslušného rámce linkové vrstvy se stará funkce `setTxMacHeaderMultipurpose`, které generuje rámce obsahující pouze položky Frame Control, Sequence Number, Source Address, FX Code a FCS. Frame Control zde udává, že se jedná o rámeček typu Multipurpose, který obsahuje pouze jednobajtovou položku sekvenčního čísla (Sequence Number) a mimo dat ještě dvoubajtovou zdrojovou adresu (Source Address). Sekvenční číslo se doplňuje ze statické proměnné, která je inkrementována s každým odeslaným rámcem. Položka FCS, sloužící jako kontrolní součet, se v rámci této funkce nevyplňuje, doplní ji totiž automaticky firmware UWB modulu v průběhu odesílání rámce. Výsledný rámeček, obsahující v datové části jednobajtový funkční kód určující typ zprávy, nabývá celkové velikosti 7 bajtů. O odeslání se následně stará funkce `sendTXFrame`, která provede kopírování do vyrovnávací paměti UWB modulu, nastavení délky dat a spustí bezodkladné odesílání.

Funkce `parseRxFrameMAC`, jak je patrné již podle názvu, provádí parsování příchozích rámců. Funkce pomocí prvního bajtu, který odpovídá položce Frame Control, zjistí, o jaký rámeček se jedná. Podle zjištěného typu předává ukazatel na strukturu s přijatými daty dále funkci `parseMultipurposeRxFrame` nebo `parseStandardRxFrame`. Data rámce jsou uložena ve struktuře `uwb_recvFrameInfoStruct`, ta obsahuje i položky pro uložení zparsovaných dat. Výhodou tohoto řešení je, že od příjmu dat až po jejich zpracování se pracuje pouze s jednou strukturou.

6.1.2 Tag

Firmware uzlu typu tag byl z pohledu implementace tím nejjednodušším. Byl pro něj založen projekt stavějící na šabloně „baremetal“, která nevyužívá služeb operačního systému reálného času. Po inicializaci mikrokontroléru je řízení předáno funkci `uwbapp_start`, která obsahuje nekonečnou smyčku. Součástí této funkce je i inicializace modulu DWM1000, která je podrobně popsána v předcházející kapitole. Součástí funkce `uwbapp_start` je také nastavení MAC adresy UWB modulu podle popisu v kapitole 4.3 a spuštění vizuální indikace odesílaných rámců pomocí LED.

Jedinou náplní uzlů typu tag je odesílání signalizačních zpráv pomocí funkce `sendBlinkMsg` na rozhraní UWB v pravidelných intervalech. To se děje v rámci nekonečné smyčky, ve které se vždy po odeslání zprávy čeká 1000 ms. Z toho vyplývá, že poloha každého tagu nacházejícího se uvnitř senzorové sítě, může být aktualizována každou vteřinu.

6.1.3 Kotevní uzel

Implementace firmware pro kotevní uzel vychází z ukázkového projektu Router obsaženého v sekci Zigbee 3.0 dodávaného v rámci SDK k mikrokontroléru Kinetis KW41Z. Projekt typu router byl zvolen právě proto, že se každý kotevní uzel v ZigBee síti chová přesně jako směrovač přeposílající zprávy mezi ostatními kotevními uzly a gateway uzlem. Router se

podle definice může taky chovat jako koncové zařízení, čehož je zde využito. V následujícím textu nebude popisována funkce kotevního uzlu jako routeru, protože se jedná o převzatou část z ukázkového projektu. Bude popsána čistě funkce koncového zařízení implementující funkcionalitu kotevního uzlu.

Protože je projekt Router založen na operačním systému reálného času FreeRTOS, je vstupní funkce `main` určena nejenom pro inicializaci hardware, ale také pro inicializace samotného operačního systému. V této části je vytvořena úloha `main_task`, která představuje hlavní prováděcí smyčku. Před prvním během hlavní prováděcí smyčky dojde ještě k inicializaci ostatních částí včetně UWB modulu, dále je provedena inicializace protokolového zásobníku ZigBee a jsou vytvořeny všechny potřebné fronty pro ukládání událostí a zpráv.

Jakmile je inicializační část dokončena, nastává samotný běh nekonečné prováděcí smyčky, ve které se spouští jednotlivé podúlohy pro zpracování událostí souvisejících s protokolovým zásobníkem ZigBee a také podúlohy samotné části koncového zařízení. Celý proces provádění hlavní smyčky je hlídán pomocí časovače `watchdog`, který musí být na konci každého cyklu obnoven. Pokud by časovač nebyl obnoven, znamenalo by to, že software nepracuje správně a je nutné jeho běh resetovat.

První úloha spuštěná v části koncového zařízení je `APP_ZB_DP_zbAppTask`. Tato úloha vyčítá z fronty `APP_uwbDataQ` zprávy, které se mají odeslat přes ZigBee síť skrze gateway uzlu do uživatelské aplikace. Data mají podobu struktury `APP_zbNetworkDataStruct`, která odpovídá datové zprávě popsané v kapitole o návrhu komunikace 4.2.2. Ještě, než jsou data předána k odeslání, je doplněna identifikace aktuálního kotevního uzlu jako položka zdrojové adresy zprávy. O odeslání se stará funkce `sendRawPayload`, která jako své argumenty požaduje kromě ukazatele na strukturu, také její délku a adresu cíle. V tomto případě je cílem gateway uzlu s adresou `0x0`.

Pro přijímání rámců ze ZigBee sítě určené pro koncové zařízení daného kotevního uzlu je připravena funkce `APP_ZB_DP_vEventHandler`. Ta je nepřímo volána z úlohy `bdb_taskBDB`, která obsluhuje události na vrstvě Base Device, při výskytu události `BDB_EVENT_ZPSAF`. Tato událost značí, že byla přijata data patřící tomuto uzlu. Data jsou následně pomocí funkce `processIncomingData` extrahována z rámce linkové vrstvy a předána funkci `processDataFromZBNetwork` ke zpracování. Protože z kapitoly 4.2.2 vyplývá, že data přijímaná v rámci ZigBee sítě kotevními uzly jsou vždy zprávy typu příkaz, jsou zpracovávána jako odpovídající struktura `APP_zbCommandDataStruct`. V případě, že je přijat příkaz `permitJoin`, je volána funkce `BDB_eNsStartNwkSteering`, která zajistí umožnění připojení dalšího kotevního uzlu do sítě například právě přes aktuální kotevní uzlu. Ostatní typy příkazů jsou předávány do části obsluhující UWB rozhraní pomocí funkce `setCommand`.

Funkce `setCommand` slouží pro nastavení aktuálně prováděné operace na základě přijatého příkazu. Pokud byl přijat příkaz `startTDOA`, dojde k okamžitému povolení příjmu rámců na UWB rozhraní a k nastavení příznaku provádění příjmu lokalizačních zpráv. Pokud byl přijat příkaz `broadcastSync`, tak záleží na jeho jediném parametru, kterým je číslo identifikující referenční kotevní uzlu. Pokud se toto číslo shoduje s identifikací aktuálního kotevní uzlu, stává se tento uzlu uzlem referenčním a nastaví svůj vnitřní stav na příslušnou hodnotu. Pokud se číslo s identifikátorem uzlu neshoduje, dojde podobně jako u příkazu `startTDOA` k povolení příjmu rámců z UWB rozhraní. Posledním možným příkazem je `stopAll`, který vypne příjem rámců a nastaví vnitřní stav na výchozí hodnotu.

Druhá úloha spuštěná jako součást koncového zařízení z hlavní prováděcí smyčky je `uwbapp_mainTask`. Tato úloha má na starosti celou logiku obsluhy UWB modulu na kotevním uzlu, stará se o zpracování příchozích rámců z UWB rozhraní a také o běh stavového automatu. Jak již bylo zmíněno v části věnované obsluze UWB modulu 6.1.1, jsou příchozí

rámce ukládány do struktury `uwb_recvFrameInfoStruct`. Protože jsem očekával, že každý kotevní uzel může v krátkém časovém okamžiku přijmout velké množství rámců na svém UWB rozhraní, implementoval jsem nad polem struktur `uwb_recvFrameInfoStruct` cyklickou frontu. Důvod, proč jsem v tomto případě nevyužil nabízeného aplikačního rozhraní operačního systému pro tvorbu front a raději jsem implementoval vlastní řešení, spočíval ve faktu, že kopírování rámců z UWB modulu je realizováno prostřednictvím přerušení. Toto způsobovalo chybu při pokusu o exkluzivní přístup při manipulaci s frontou.

Jak je již zřejmé, při vyvolaném přerušení o příchodu nového rámce je tento rámeček přenesen z UWB modulu pomocí funkce pro obsluhu správně přijatých rámců. Rámeček je uložen na místo v paměti, které náleží aktuálnímu konci fronty. Pokud je fronta plná, je rámeček zahozen. V obou případech je ovšem obnovena možnost přijetí dalšího rámce na UWB rozhraní.

Úloha `uwbapp_mainTask` zpracovává došlé rámce, dokud není fronta prázdná. Jakmile je fronta přichodících rámců prázdná, předává se řízení funkci `mainStateMachine` reprezentující stavový automat obsluhy UWB rozhraní. Stavový automat implementuje logiku synchronizace času na straně referenčního kotevního uzlu. Protože je informace o času odeslání rámce dostupná až po jeho skutečném odvysílání na UWB rozhraní modulu DWM1000, má stavový automat tři stavy. První stav reprezentuje neaktivní synchronizaci a neprovádí proto žádnou akci. Pokud je vnitřní stav změněn funkcí `setCommand` na začátek vysílání synchronizační zprávy, je tato zpráva odeslána pomocí funkce `sendTimeSyncMsg`. Následně je nastaven vnitřní stav na a druhou část synchronizace, ve které se provede uložení času odeslání rámce a toto časové razítko je následně odesláno skrz ZigBee síť a gateway uzel do uživatelské aplikace.

6.1.4 Gateway uzel

Podobně jako firmware pro kotevní uzel, i firmware určený pro gateway uzel vychází z ukázkového projektu obsaženého v SDK, které je dodávané k mikrokontroléru Kinetis KW41Z. V tomto případě se jedná o projekt Hybrid Coordinator, který spojuje funkci ZigBee koordinátora s Bluetooth Low Energy serverem. Stejně jako v případě kotevního uzlu, ani zde se nebudu pouštět do popisu převzatých částí a zaměřím se pouze na části vlastní implementace.

Tento projekt je z důvodů své komplexnosti taktéž založen na operačním systému reálného času FreeRTOS. I zde platí stejný princip o úloze `main_task`, která provádí inicializaci a obsahuje hlavní prováděcí smyčku. Úloha `main_task` je zde ovšem rozšířena o inicializaci protokolového zásobníku pro Bluetooth Low Energy (BLE) a její hlavní smyčka dále obsahuje podúlohu pro práci s událostmi generovanými rozhraním BLE.

Pro potřeby odesílání dat do ZigBee sítě je zde implementována stejná metoda, jako v případě firmware kotevního uzlu, kterou je `APP_ZB_DP_zbAppTask`. Jediným rozdílem v její implementaci je, že příkazové zprávy získávané z fronty `APP_msgBLEtoZB` odesílá nikoliv na jeden uzel, ale záplavově do celé sítě pomocí broadcastové adresy `0xFFFF`. Tímto je zajištěno, že se příkazy pro provedení konkrétní akce dostanou do všech kotevní uzlů sítě.

Příjem rámců určených gateway uzlu je zde kompletně převzatý z firmware kotevního uzlu. Rozdílné je ovšem zpracování jednotlivých datových zpráv pomocí funkce `processDataFromZBNetwork`. Na rozdíl od kotevních uzlů, kterým jsou určeny příkazové zprávy, přijímá gateway uzel ze ZigBee sítě pouze zprávy datové, které obsahují nějaký výsledek operace na kotevních uzlech. Tyto zprávy, reprezentované strukturou `APP_zbNetworkDataStruct` jsou přímo předávány do fronty `APP_msgZBtoBLE`.

Zpracováním zpráv z fronty `APP_msgZBtoBLE` se věnuje podúloha hlavní prováděcí smyčky nazvaná `APP_BLE_FX_bleAppTask`. Tato podúloha vybírá z fronty v každém cyklu hlavní smyčky všechny datové zprávy, které postupně předává pomocí funkce `Upl_RecordUplinkData` na zásobník rozhraní BLE.

Funkce `Upl_RecordUplinkData` je součástí obsluhy GATT serveru, který implementuje službu `service_wireless_comlink`. Tato služba obsahuje dvě charakteristiky, první z nich je `char_downlink_stream`, která slouží pro přenos zpráv z klienta, tedy uživatelské aplikace, do gateway uzlu. Druhá charakteristika `char_uplink_stream` slouží pro přenos dat právě do uživatelské aplikace. Tato charakteristika implementuje také notifikační službu, která informuje klienta o nově zapsaném atributu. Klient tedy nemusí kontrolovat dostupnost nové hodnoty neustálým dotazováním. Příkazové zprávy, které uživatelská aplikace vkládá do atributu charakteristiky `char_downlink_stream`, jsou zachycené pomocí funkce `BleApp_GattServerCallback` a dál propagovány do funkce `APP_BLE_FX_processDataFromBLE`. Zde jsou příkazové zprávy buďto provedeny, případ příkazu `createNetwork` sloužícího pro vytvoření nové ZigBee sítě, nebo předány do fronty `APP_msgBLEtoZB` pro odeslání dále do senzorové sítě.

6.2 Uživatelská aplikace

Pro tvorbu uživatelské aplikace jsem si zvolil jazyk Python ve verzi 3.7. Důvodem pro tuto volbu je má předchozí zkušenost s tímto jazykem, a hlavně s tvorbou uživatelského rozhraní pomocí modulu `wxPython`. Nespornou výhodou pro vývoj je i fakt, že je výsledná aplikace multiplatformní. Vývoj a ladění mohlo tedy probíhat na podstatně výkonnějším počítači, nežli je cílové Raspberry Pi 4B.

Uživatelské rozhraní aplikace vychází přesně z návrhu uvedeného v kapitole 4.6. Pro jeho tvorbu byl využit open-source designer `wxFormBuilder`, který výsledný projekt umí vyexportovat do kódu v jazyce Python právě s využitím modulu `wxPython`. Protože standardní verze modulu `wxPython` neumí dobře pracovat s asynchronními úlohami, byl pro tento účel do projektu dodán ještě modul `wxasync`. Ten je použit pro spuštění samotné aplikace, nad kterou běží okno uživatelského rozhraní.

Aplikace je logicky rozdělena do čtyřech částí, které spolu navzájem komunikují. Každá z těchto částí představuje modul implementující odlišnou funkcionalitu. Mezi tyto moduly tak patří správa a obsluha uživatelského rozhraní, komunikace na rozhraní Bluetooth Low Energy, parsování a tvorba zpráv předávaných přes toto rozhraní, a nakonec samotná logika implementující zpracování příchozích dat a řízení senzorové sítě.

6.2.1 Obsluha rozhraní Bluetooth Low Energy

Pro potřeby obsluhy rozhraní Bluetooth Low Energy byl použit modul `bleak`, který implementuje funkce pro vyhledávání, připojení a komunikaci se zařízením. Jeho provoz je plně asynchronní k hlavnímu běhu programu. Pro potřeby aplikace jsem vytvořil třídu `bleCom`, která zastřešuje veškeré akce s modulem `bleak` a poskytuje rozhraní pro zbytek aplikace. Pro vytvoření objektu této třídy jsou nutné čtyři parametry. První tři musí být typu `callable` a slouží pro předávání výsledků asynchronních operací. Posledním parametrem je fronta zpráv určených k odeslání.

Pro vyhledávání zařízení v okolí slouží metoda `runBLEScan`, která provádí prostřednictvím třídy `BleakScanner` skenování všech dostupných zařízení. Výsledek v podobě seznamu dostupných zařízení propaguje prostřednictvím zpětného volání, v tomto případě metody

`bleScanResult`. Tato metoda, která je součástí hlavní logiky aplikace slouží mimo jiné pro naplnění výběrového prvku uživatelského rozhraní. Pokud si uživatel vybere zařízení z nabízeného seznamu a klikne na tlačítko „Připojit“, dojde k volání metody `runBLEMainLoop`, která provede pokus o připojení k zařízení. Pokud je připojení úspěšné, dojde k zaregistrování odběru notifikace o změně atributu charakteristiky `char_uplink_stream` a následně je spuštěna smyčka pro odesílání zpráv na zařízení pomocí charakteristiky `char_downlink_stream`. Odesílané zprávy jsou vybírány z fronty zpráv a pokud je fronta prázdná, tak smyčka čeká. Pokud dojde k notifikaci o změně atributu charakteristiky `char_uplink_stream`, je tato událost obsloužena zpětným voláním, zde metodu `incommingNotification` ze třídy pro zpracování zpráv. Této metodě jsou předány přijatá data a také informace o odesílateli.

6.2.2 Zpracování a tvorba zpráv pro rozhraní Bluetooth Low Energy

Další částí aplikace, která přímo navazuje na komunikaci přes rozhraní Bluetooth Low Energy je zpracování zpráv pomocí třídy `messageProcessing`. Tato třída, jak již bylo zmíněno, provádí zpracování příchozích zpráv pomocí metody `incommingNotification`. Zprávy přijaté z rozhraní Bluetooth Low Energy mají formát datové zprávy a nabývají typů popsaných v kapitole 4.2.2. V téže kapitole jsou popsány i příkazové zprávy, které jsou z aplikace odesílány.

Metoda `incommingNotification` při zpracování jednotlivých zpráv vytváří objekty třídy `node`, které reprezentují základní atributy uzlů, ze kterých byla zpráva přijata. Na základě zjištěného typu zprávy je následně datová část společně s objektem třídy `node` předána pro další zpracování příslušným metodám. Výsledkem je rozdělení a uložení přijatých dat ve formě objektů třídy `TDoAMessage` nebo `TimeSyncMessage` do příslušných front. V případě zprávy typu `selfIdentification` je informována třída hlavní logiky aplikace prostřednictvím metody `newNodeOccure` o výskytu nového uzlu v síti

Kromě zpracování přijatých zpráv poskytuje třída `messageProcessing` i metody pro odeslání zpráv typu `createNetwork`, `permitJoin`, `startTDOA`, `stopAll` a `broadcastSync`. Poslední jmenovaná jako jediná přenáší ve svém těle i identifikátor určující, který kotevní uzel se bude chovat jako referenční.

6.2.3 Hlavní logika pro určování pozic uzlů

Předposlední část využívá zpracovaných datových zpráv pro realizaci určování poloh uzlů typů `tag`. Logika je implementována třídou `mainLogic`, která obsahuje slovník všech kotevních uzlů a uzlů typu `tag`. Slovník `nodesDict` slouží pro rychlý přístup k objektům `advancedNode` pomocí čísla identifikátoru jednotlivých uzlů. Třída `advancedNode` rozšiřuje třídu `node` o pro lokalizaci relevantní atributy. Dalším důležitým slovníkem pro použití při lokalizaci je slovník `tagsTdoaDict`. Ten obsahuje v konečné hierarchii seznam všech přijatých lokalizačních zpráv ve formátu dvojice (časové razítko a identifikátor kotevního uzlu). Posledním používaným slovníkem k lokalizaci je `tagCurrentSeq`, který udržuje pro každý `tag` informaci o sekvenci aktuálně přijímaných lokalizačních zpráv.

Před samotným procesem lokalizace musí být spuštěn proces synchronizace časů mezi jednotlivými kotevními uzly a referenčním kotevním uzlem. Toto předpokládá přítomnost všech žádaných kotevních uzlů v ZigBee síti. Každý kotevní uzel musí mít určenou svou polohu a jeden z nich musí být zvolen za referenční kotevní uzel. Následně je v zadaných intervalech do sítě posílána zpráva `broadcastSync`, která udává identifikátorem, který uzel má

vysílat synchronizační zprávy na UWB rozhraní. Ostatní kotevní uzly tyto zprávy přijímají a směřují je zpět do aplikace.

Proces synchronizace časů je realizován úlohou `timeSyncMainProcess`, která v sobě obsahuje nekonečnou smyčku pro zpracování. Smyčka čeká na přijetí objektu typu `TimeSyncMessage` s časem odeslání aktuální synchronizační zprávy. Jakmile je objekt k dispozici je vyjmut z fronty. Dále se pokračuje čekáním na druhé frontě, která poskytuje objekty stejného typu s informací o času přijetí synchronizační zprávy na ostatních nereferenčních kotevních uzlech. Jakmile je tato informace k dispozici, je opět vyjmuta a je pro daný nereferenční kotevní uzel z této a předchozí hodnoty příjmu synchronizační zprávy a z času odeslání synchronizační zprávy referenčním uzlem vypočítána hodnota offsetu a clock skew podle vztahů z kapitoly 4.4. Jakmile jsou tyto nové hodnoty k dispozici pro všechny nereferenční kotevní uzly v rámci aktuální synchronizační relace, jsou tyto hodnoty uloženy k příslušným objektům kotevních uzlů ve slovníku `nodesDict`.

Jakmile jsou časy kotevních uzlů synchronizovány, může dojít i ke spuštění fáze lokalizace. To se provádí vysláním příkazu `startTDOA` do ZigBee sítě. Tento příkaz způsobí spuštění příjmu lokalizačních zpráv tagů na jednotlivých kotevních uzlech. Výsledné zprávy typu `TDOATimestamp` jsou následně zasílány do aplikace.

Proces lokalizace sestává ze dvou částí. První část realizovaná úlohou `tdoaMainProcess` provádí příjem objektů typu `TDoAMessage` z příslušné fronty v rámci nekonečné smyčky. Na základě identifikátoru kotevného uzlu, který lokalizační zprávu přijal, je provedena korekce času přijetí zprávy podle vztahu z kapitoly 4.4. Následně je informace o kotevním uzlu a korektní čas přijetí lokalizační zprávy uložen do slovníku `tagsTdoaDict` na pozici definovanou identifikátorem tagu a sekvenčním číslem vyslané lokalizační zprávy. Pokud se již objevil pro příslušný tag objekt typu `TDoAMessage` s jinou hodnotou sekvenčního čísla, nežli je aktuálně používána ve slovníku `tagCurrentSeq`, je původní hodnota sekvenčního čísla spolu s identifikátorem vložena jako dvojice do fronty `tdoaComputeQueue`.

Druhá část, která je realizovaná úlohou `tdoaComputeProcess` vybírá v rámci nekonečné smyčky dvojice z fronty `tdoaComputeQueue`. Tyto dvojice značí ukončený příjem lokalizačních zpráv pro danou sekvenci a identifikátor tagu. Pokud se ve slovníku `tagsTdoaDict` nachází pro danou sekvenci a identifikátor tagu alespoň čtyři lokalizační dvojice (identifikátor kotevního uzlu s časem přijetí lokalizační zprávy), je možné provést určení pozice tag uzlu pomocí TDoA solveru. Tento solver byl převzat z projektu MultilaterationTDOA dostupného na platformě GitHub pod MIT licenci³. Solver vyžaduje pro svůj vstup známé pozice kotevních uzlů a také diferenci, nikoliv ve formě času, ale ve formě vzdálenosti. Pro účel výpočtu difference vzdálenosti ze dvou časových razítek slouží metoda `getDistanceFromTimestamps`. Tato metoda na základě vlastnosti časového razítka generovaného UWB modulem DWM1000 a ze znalosti rychlosti šíření elektromagnetického záření ve vzduchu dovede vypočítat vzdálenostní diferenci dvou časů přijetí lokalizační zprávy.

6.2.4 Zobrazování výsledků

Jak již bylo uvedeno, uživatelské rozhraní aplikace odpovídá návrhům v kapitole 4.6. Každý lokalizovaný uzel je zobrazen na plánku oblasti společně s kotevními uzly, jejichž poloha je zadávána během jejich přidávání do plánku. Pokud není kotevní uzel zaznačen v plánku, nemůže se ani účastnit procesu lokalizace, protože jeho pozice není známá. Další formou prezentace výsledků je i textové pole, do kterého jsou vepisovány pozice všech uzlů. Je zde

³<https://github.com/AlexisTM/MultilaterationTDOA>

dobré zmínit, že souřadný systém, ve kterém se v rámci aplikace nacházejí jednotlivé uzly, má počátek ve středu plánku.

Kapitola 7

Závěr

Cílem této práce bylo navrhnout a zrealizovat systém na určování polohy uzlů uvnitř sensorové sítě. V rámci řešení jsem prostudoval bezdrátovou komunikaci ve volně dostupném pásmu ISM. Pozornost jsem zaměřil na standard IEEE 802.15.4 a technologie ZigBee, které na něm staví. Dále jsem prozkoumal i rozšíření standardu IEEE 802.15.4 o fyzickou vrstvu využívající ultra širokopásmovou komunikaci. Jako poslední ze standardů využívajících pásem ISM jsem zkoumal rozhraní Bluetooth i s jeho variantou Bluetooth Low Energy. Následně jsem provedl přehledovou studii možných metod využívajících bezdrátová rozhraní sensorových sítí pro lokalizaci jednotlivých uzlů. Metody jsem rozdělil na dvě skupiny podle toho, zda k určení polohy uzlů používají měření vzdálenosti nebo ne.

Pro realizaci lokalizace jsem zvolil metodu časových diferencí, provedl jsem návrh celého konceptu sensorové sítě včetně systému předávaných zpráv a identifikace uzlů. Sensorovou síť jsem založil na technologii ZigBee a lokalizaci na ultra širokopásmové komunikace. Pro synchronizaci hodinového signálu mezi kotevními uzly jsem zvolil metodu lineární extrapolace. Dále jsem vybral hlavní komponenty pro tvorbu desek uzlů ve formě mikrokontroléru Kinetis KW41Z a UWB modulu DWM1000. Pro napájení sensorových uzlů jsem se rozhodl využít lithium-polymerových akumulátorů. Nakonec jsem provedl návrh podoby uživatelské aplikace s podrobným popisem jejího fungování. Pro její běh a výpočty jsem zvolil jednodeskový minipočítač Raspberry Pi 4B, který komunikuje s navrženou sensorovou sítí pomocí standardu Bluetooth Low Energy.

Následně jsem provedl realizaci jednotlivých desek sensorového uzlu osazených zmíněnými komponentami. Pro tyto desky jsem implementoval tři odlišné verze firmware představující uzly typu tag, kotva a gateway. Nakonec jsem podle návrhu vytvořil uživatelské prostředí aplikace společně s logikou pro synchronizování hodinových signálů kotevních uzlů a výpočet polohy uzlů typů tag.

Z důvodů pro mě neuspokojivých výsledků dosahovaných s aktuální implementací metody časových diferencí a metody pro synchronizaci času mezi kotevními uzly jsem nebyl schopen provést důkladné otestování celého navrženého konceptu v reálných podmínkách. Díky tomu jsem nebyl schopen potvrdit správnost navrženého řešení.

Pro budoucí vývoj bych vyzkoušel použití jiného konceptu synchronizace času mezi kotevními uzly a dále bych zvážil i implementaci pokročilejší numerické metody pro určování polohy uzlů typů tag na základě časových diferencí. Z dlouhodobějšího hlediska a pro reálné nasazení v praxi by bylo vhodné implementovat výpočetní uzel ve formě webového serveru a uživatelskou aplikaci, která by zobrazovala skutečné polohy uzlů v reálním čase, umístit do prostředí webového prohlížeče. Dalším krokem by mohlo být i automatické zjištění poloh ostatních kotevních uzlů například s využitím metody two-way ranging.

Literatura

- [1] ADEMUWAGUN, A. a FABIO, V. Reach Centroid Localization Algorithm. *Wireless Sensor Network*. 1. vyd. Aberdeen, UK: Scientific Research Publishing. Únor 2017, sv. 9, č. 2, s. 87–101. ISSN 1945-3078.
- [2] ALLIANCE, Z. *ZigBee Specification: ZigBee Document 05-3474-21* [online]. Davis (California): ZigBee Alliance, srpen 2015 [cit. 2021-04-28]. Dostupné z: <https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf>.
- [3] BATTERIES, C. *Specification Approval Sheet* [online]. 1. vyd. CELLEVIA BATTERIES, 2018, 8. 3. 2018. Dostupné z: <https://www.tme.eu/Document/dc66e5813eb66a792d6d087893c5df5b/ce10117.pdf>.
- [4] BENSKY, A. Chapter 14 - Technologies and applications. In: BENSKY, A., ed. *Short-range Wireless Communication (Third Edition)*. Third Edition. Newnes, Srpen 2019, s. 387–430. ISBN 978-0-12-815405-2.
- [5] DECAWAVE. *APS003 APPLICATION NOTE: REAL TIME LOCATION SYSTEMS* [online]. 1.1. Dublin (Ireland): Decawave, 2014, 10. 2018 [cit. 2021-04-28]. Dostupné z: https://www.decawave.com/wp-content/uploads/2018/10/APS003_DW1000-RTLS-Introduction_v1.1.pdf.
- [6] DECAWAVE. *APS013 APPLICATION NOTE: The implementation of two-way ranging with the DW1000* [online]. 2.3. Dublin (Ireland): Decawave, 2015, 10. 2018 [cit. 2021-04-28]. Dostupné z: https://www.decawave.com/wp-content/uploads/2018/10/APS013_The-Implementation-of-Two-Way-Ranging-with-the-DW1000_v2.3.pdf.
- [7] DECAWAVE. *DW1000 DEVICE DRIVER APPLICATION PROGRAMMING INTERFACE (API) GUIDE* [online]. 2.6. Dublin (Ireland): Decawave, 2016, 28. 11. 2017 [cit. 2021-04-28]. Dostupné z: <https://decaforum.decawave.com/uploads/default/original/1X/8b220e1e26fea4ebd83f0b0e5ef42eb9a251310d.pdf>.
- [8] DECAWAVE. *DWM1000 Product Brief* [online]. 1.2. Dublin (Ireland): Decawave, 2018, 9. 2018. Dostupné z: https://www.decawave.com/wp-content/uploads/2018/09/dwm1000_brief.pdf.
- [9] DECAWAVE. *DWM1000 Datasheet: DWM1000IEEE 802.15.4-2011 UWB Transceiver Module* [online]. 1.7. Dublin (Ireland): Decawave, 2020, 9. 2020 [cit. 2021-04-28]. Dostupné z: <https://www.decawave.com/wp-content/uploads/2020/09/DWM1000-Datasheet.pdf>.

- [10] ELECTRICAL, T. I. of a ELECTRONICS ENGINEERS, I. *802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks*. 1. vyd. New York (USA): IEEE, duben 2016. 1–709 s. ISBN 9781504408455.
- [11] GISLASON, D. *Zigbee wireless networking*. 1. vyd. Oxford (UK): Newnes, srpen 2008. ISBN 978-0-7506-8597-9.
- [12] HE, T., HUANG, C. et al. Range-free localization schemes for large scale sensor networks. *Proceedings of the 9th annual international conference on Mobile computing and networking - MobiCom '03*. 1. vyd. New York (USA): Association for Computing Machinery. Zář 2003, č. 7, s. 81–95. MobiCom '03.
- [13] INSTRUMENTS, T. *BQ2407xTStandalone1-Cell 1.5-A LinearBatteryChargerwith PowerPath and Voltage-BasedTS* [online]. 2019. vyd. Texas Instruments, 2009, 12. 2019. Dostupné z: <https://www.ti.com/lit/gpn/BQ24075T>.
- [14] IPC. *IPC-2141A: Design Guide for High-Speed Controlled Impedance Circuit Boards*. 1. vyd. Northbrook (USA): IPC, březen 2004.
- [15] KOUBAA, A., ALVES, M. a TOVAR, E. *IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview* [online]. Porto (Portugal): Polytechnic Institute of Porto, červenec 2005 [cit. 2021-04-28]. Dostupné z: http://www.cister.isep.ipp.pt/docs/ieee_802_15_4_for_wireless_sensor_networks__a_technical_overview/222/attach.pdf.
- [16] KRAMOSIL, J. *Ultra – širokopásmové systémy (UWB)* [online]. Pardubice, zář 2005 [cit. 2021-04-28]. Dostupné z: http://www.ok2kkw.com/00000104/uwb/uwb_pardubice_clanek.pdf.
- [17] LEUGNER, S., PELKA, M. a HELLBRÜCK, H. Comparison of wired and wireless synchronization with clock drift compensation suited for U-TDoA localization. In: IEEE. *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*. 1. vyd. New Orleans (USA: IEEE, říjen 2016, sv. 13, s. 1–4. ISBN 978-1-5090-5440-4.
- [18] LU, X. ruo, CHEN, J. xin et al. A Distance Ratio-Based Algorithm for Indoor Localization in Wireless Sensor Networks. In: CAI, Z., WANG, C. et al., ed. *Wireless Algorithms, Systems, and Applications*. 1. vyd. Cham (Germany): Springer International Publishing, červen 2014, sv. 8491, s. 401–411. Theoretical Computer Science and General Issues. ISBN 978-3-319-07781-9.
- [19] MAO, G. a FIDAN, B. *Localization algorithms and strategies for wireless sensor networks*. 1. vyd. Hershey (USA): Information Science Reference, březen 2009. ISBN 978-1-60566-396-8.
- [20] NXP. *Hardware Design Considerations for MKW41Z/31Z/21Z BLE and IEEE 802.15.4 Device* [online]. 2. vyd. Eindhoven (Netherlands): NXP, listopad 2016, 12. 2017. Dostupné z: https://www.nxp.com/files-static/32bit/doc/app_note/AN5377.pdf.
- [21] NXP. *MKW41Z/31Z/21Z Data Sheet* [online]. 4. vyd. Eindhoven (Netherlands): NXP, 2016, 3. 2018. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/MKW41Z512.pdf>.

- [22] NXP. *Kinetis® W Series KW40Z MCUs Block Diagram* [online]. Eindhoven (Netherlands): [b.n.], 2018 [cit. 2021-05-12]. Dostupné z: <https://www.nxp.com/assets/images/en/block-diagrams/KINETIS-KW41Z-BD.jpg>.
- [23] NXP. *Compact Planar Antennas for 2.4 GHz Communication* [online]. 4. vyd. Eindhoven (Netherlands): NXP, únor 2020, 11. 2017. Dostupné z: <https://www.nxp.com/docs/en/application-note/AN2731.pdf>.
- [24] PAUL, A. K. a SATO, T. Localization in Wireless Sensor Networks: A Survey on Algorithms, Measurement Techniques, Applications and Challenges. *Journal of Sensor and Actuator Networks*. 1. vyd. Basel (Switzerland): MDPI. Říjen 2017, sv. 6, č. 4. ISSN 2224-2708.
- [25] SCHLEICHER, B. a SCHUMACHER, H. Impulse Generator Targeting the European UWB Mask. In: IEEE. *2010 Topical Meeting on Silicon Monolithic Integrated Circuits in RF Systems (SiRF)*. 1. vyd. New Orleans (USA: IEEE, únor 2010, s. 21 – 24. ISBN 978-1-4244-5459-4.
- [26] STEIN, P. *Bluetooth network topology* [online]. Rio de Janeiro (Brasil): [b.n.], 2003 [cit. 2021-05-09]. Dostupné z: https://www.gta.ufrj.br/seminarios/semin2003_1/stein/figura2.gif.
- [27] STEIN, P. *Time Division Duplex* [online]. Rio de Janeiro (Brasil): [b.n.], 2003 [cit. 2021-05-09]. Dostupné z: https://www.gta.ufrj.br/seminarios/semin2003_1/stein/figura3.gif.
- [28] TIEMANN, J., ECKERMAN, F. a WIETFELD, C. ATLAS - an open-source TDOA-based Ultra-wideband localization system. In: IEEE, ed. *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 1. vyd. New York (USA): IEEE, Listopad 2016, s. 1–6. International Conference on Indoor Positioning and Indoor Navigation. ISBN 978-1-5090-2426-1.
- [29] TOSI, J., TAFFONI, F. et al. Performance Evaluation of Bluetooth Low Energy: A Systematic Review. *Sensors*. 1. vyd. Basel (Switzerland): MDPI. Prosinec 2017, sv. 17, č. 12. ISSN 1424-8220.
- [30] WORLD, I. C. to connecting the. *Recommendation SM.1755-0 (05/2006): Characteristics of ultra-wideband technology* [online]. Geneva (Switzerland): ITU: Committed to connecting the world, 2006 [cit. 2021-04-28]. Dostupné z: http://www.itu.int/dms_pubrec/itu-r/rec/sm/R-REC-SM.1755-0-200605-I!!PDF-E.pdf.
- [31] ZEADALLY, S., SIDDIQUI, F. a BAIG, Z. 25 Years of Bluetooth Technology. *Future Internet*. 1. vyd. Basel (Switzerland): MDPI. Zář 2019, sv. 11, č. 9. ISSN 1999-5903.
- [32] ÚŘAD Český telekomunikační. *Všeobecné oprávnění č. VO-R/10/12.2019-9 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu* [online]. Praha: Český telekomunikační úřad, prosinec 2019 [cit. 2021-04-28]. Dostupné z: <https://www.ctu.cz/sites/default/files/obsah/ctu/vseobecne-opravneni-c.vo-r/10/12.2019-9/obrazky/vo-r10-122019-9.pdf>.