

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

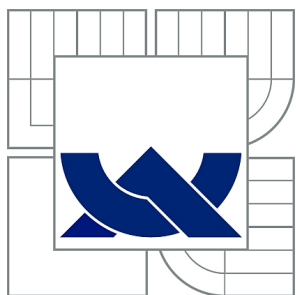
VLIV PŘENOSOVÝCH PARAMETRŮ NA SPOTŘEBU ELEKTRICKÉ
ENERGIE ZIGBEE ZAŘÍZENÍ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

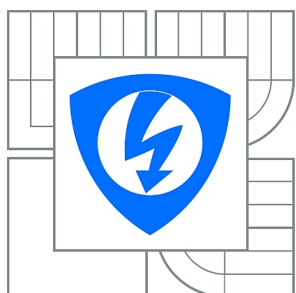
Bc. JAN POPELKA

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VLIV PŘENOSOVÝCH PARAMETRŮ NA SPOTŘEBU ELEKTRICKÉ ENERGIE ZIGBEE ZAŘÍZENÍ

IMPACT OF COMMUNICATION PARAMETERS ON ENERGY CONSUMPTION OF ZIGBEE
DEVICES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN POPELKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MILAN ŠIMEK, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jan Popelka

ID: 106726

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Vliv přenosových parametrů na spotřebu elektrické energie Zigbee zařízení

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce provést výzkum spotřeby elektrické energie při bezdrátové komunikaci mezi dvěma bezdrátovými jednotkami. Jedním z částí úkolu je realizovat testovací scénář, kdy student naprogramuje vysílací uzel tak, aby u něj dokázal měnit parametry, jako jsou velikost paketu, interval odesílání, vysílací výkon a další. Výzkum vlivu těchto parametrů na celkovou spotřebu el. energie zařízení je hlavním cílem diplomové práce. Součástí práce je i návrh na automatizaci měření a zpracování výsledků a dále také detailní studie vybíjecích charakteristik použitých akumulátorů.

DOPORUČENÁ LITERATURA:

- [1] Stojmenovic I., Handbook of Sensor Networks, Wiley, ISBN:13 978-0-471-68472-5, 2005.
- [2] FARAHANI, Shahin. Zigbee Wireless Networks and Transceivers. [s.l.] : Elsevier, 2008. 329 s. ISBN

Termín zadání: 11.2.2013

Termín odevzdání: 29.5.2013

Vedoucí práce: Ing. Milan Šimek, Ph.D.

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato diplomová práce se zabývá výzkumem spotřeby energie a návrhem jednoduché bezdrátové sensorové sítě. K přenosu dat využívají moduly bezdrátové komunikační technologie ZigBee. Hlavní vlastností bezdrátových sensorových sítí je minimální spotřeba energie zařízení při nízké ceně a maximální spolehlivosti přenosu dat. Dále se v práci popisuje vytvořený firmware pro komunikující uzly s možností měnit parametry přenosu dat a tím ovlivnit spotřebu energie uzlu. Výpočty spotřeby elektrické energie, porovnání s naměřenými výsledky a vybíjecí charakteristiky napájecích akumulátorů jsou obsaženy v závěrečných kapitolách práce.

KLÍČOVÁ SLOVA

Bezdrátová sensorová síť, mikrokontrolér, rádiový čip, bezdrátové sítě osobního dosahu, standard IEEE, standard ZigBee, fyzická vrstva, linková vrstva, síťová vrstva, aplikační vrstva, koordinátor, směrovač, koncové zařízení, bezdrátová komunikační jednotka, AVR studio, ZigBee BitCloud.

ABSTRACT

This thesis deals with the energy research and design a simple wireless sensor networks. The data modules with ZigBee wireless communication technology. The main feature of wireless sensor networks is the minimal energy devices at low cost and maximum data reliability. Furthermore, the work describes firmware for communicating nodes with the ability to change parameters of data transmission and thus affect the energy consumption node. Calculations of energy consumption, compared with measured results and discharge characteristics of the battery packs are included in the final chapters of this work.

KEYWORDS

Wireless Sensor Network, Microcontroller, Radio Chip, Wireless Personal Area Networks, standard IEEE, standard ZigBee, Physical layer, Link layer, Network layer, Application layer, Coordinator, Router, End Device, Wireless Communications Unit, AVR studio, ZigBee BitCloud.

POPELKA, Jan *Vliv přenosových parametrů na spotřebu elektrické energie Zigbee zařízení*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 72 s. Vedoucí práce byl Ing. Milan Šimek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Vliv přenosových parametrů na spotřebu elektrické energie Zigbee zařízení“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval panu Ing. Milanu Šimkovi Ph.D. za vedení při této diplomové práci, ochotu, konzultace a praktické rady. Dále bych chtěl poděkovat své rodině, přítelkyni a známým za podporu během celého studiu.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

| | |
|--|-----------|
| Úvod | 12 |
| 1 Bezdrátové senzorové sítě | 13 |
| 1.1 Technologie v bezdrátové senzorové síti | 13 |
| 1.2 Standardy v bezdrátové technologii | 14 |
| 1.3 Standard 802.15.4 | 15 |
| 1.3.1 Fyzická a linková vrstva | 15 |
| 1.3.2 Topologie bezdrátové sítě | 16 |
| 2 Standard ZigBee | 18 |
| 2.1 Typy zařízení standardu ZigBee | 19 |
| 2.2 Adresace v síti PAN | 19 |
| 2.3 Síťová vrstva | 20 |
| 2.3.1 Komunikace broadcast ve standardu ZigBee | 21 |
| 2.4 Aplikační vrstva | 22 |
| 2.4.1 Aplikační profily | 23 |
| 2.4.2 Aplikační podvrstva APS | 24 |
| 3 Sériový přenos dat | 26 |
| 3.1 Sériový asynchronní přenos | 26 |
| 3.2 Sériový synchronní přenos | 26 |
| 3.3 USART | 27 |
| 4 ZigBee BitCloud | 29 |
| 4.1 Architektura BitCloud | 29 |
| 5 Použitý hardware | 31 |
| 5.1 Rádiový čip AT86RF230 | 31 |
| 5.2 Napájení | 33 |
| 6 Energetická náročnost | 37 |
| 6.1 Energie v bezdrátových sítích | 37 |
| 6.2 Spotřeba energie v bezdrátových sítích | 37 |
| 6.2.1 Spotřeba energie přijímací části | 38 |
| 6.2.2 Spotřeba energie vysílací části | 38 |
| 6.3 Energetický model při zpracování zprávy | 39 |

| | | |
|-----------|--|-----------|
| 7 | Praktické řešení práce | 41 |
| 7.1 | Použité zařízení a software | 41 |
| 7.2 | Realizace | 44 |
| 7.2.1 | Blokové schéma | 44 |
| 7.2.2 | Programování modulu pomocí BitCloud | 45 |
| 7.3 | Řešení ve vývojovém prostředí | 48 |
| 7.3.1 | Koordinátor | 48 |
| 7.3.2 | Router | 49 |
| 7.3.3 | Nastavení parametrů | 49 |
| 7.3.4 | Start sítě | 51 |
| 7.3.5 | Universal Synchronous Receiver and Transmitter | 51 |
| 7.4 | Vytvoření sítě a přenos dat | 52 |
| 8 | Výpočty doby přenosu rámce | 54 |
| 8.1 | Naměřené doby přenosu | 55 |
| 9 | Výpočty spotřeby energie | 60 |
| 9.1 | Režim vysílání | 60 |
| 9.2 | Režim příjmu | 62 |
| 9.3 | Výdrž akumulátorů | 63 |
| 9.3.1 | Výpočet doby provozu | 64 |
| 10 | Závěr | 65 |
| | Literatura | 67 |
| | Seznam symbolů, veličin a zkratk | 70 |
| A | Přílohy | 72 |
| A.1 | Příložené DVD | 72 |

SEZNAM OBRÁZKŮ

| | | |
|------|---|----|
| 1.1 | Struktura uzlu WSN sítě [3] | 14 |
| 1.2 | Topologie sítě typu hvězda [3] | 17 |
| 1.3 | Topologie sítě typu peer-to-peer | 17 |
| 2.1 | OSI model standardu ZigBee | 18 |
| 2.2 | Adresové pole MAC rámce [8] | 19 |
| 2.3 | Síťová vrstva ZigBee [1] | 20 |
| 2.4 | Struktura aplikační vrstvy standardu ZigBee | 22 |
| 2.5 | Struktura APS rámce [1] | 25 |
| 3.1 | Komunikace po USART sběrnici [11] | 28 |
| 4.1 | Architektura BitCloud (převzato z [12]) | 29 |
| 5.1 | Popis pinů modulu AT86RF230, (převzato z [20]) | 31 |
| 5.2 | Vybíjecí průběhy (použito z [23]) | 34 |
| 5.3 | Okamžitá spotřeba proudu zařízení (převzato z [24]) | 35 |
| 5.4 | Impulzní odběh z akumulátoru (převzato z [24]) | 35 |
| 6.1 | Údaje o velikosti zprávy <i>Data Request</i> [15] | 39 |
| 7.1 | Vývojové prostředí AVR Studia 5.1 | 41 |
| 7.2 | Programátor a debugger ATMEL AVR JTAGICE mkII | 42 |
| 7.3 | Programovací jednotka MIB520 (převzato z [18]) | 42 |
| 7.4 | Blokové schéma MIB520CB [18] | 43 |
| 7.5 | Bezdrátová komunikační jednotka IRIS (převzato z [19]) | 43 |
| 7.6 | Blokové schéma XM2110CA [19] | 44 |
| 7.7 | Blokové schéma | 45 |
| 7.8 | Diagram počáteční síťové sekvence (převzato z [12]) | 46 |
| 7.9 | Formát ASDU pole [12] | 46 |
| 7.10 | Přenos dat v síti (převzato z [12]) | 47 |
| 7.11 | Přenos dat pomocí USART - callback režim (převzato z [12]) | 48 |
| 7.12 | Ukázka komunikace v ZigBee | 52 |
| 7.13 | Ukázka paketu | 53 |
| 7.14 | Příjem dat po sériovém portu | 53 |
| 8.1 | Struktura zapouzdření vrstev (převzato z [24]) | 54 |
| 8.2 | Průběh zasílání paketů s intervalem 1s | 56 |
| 8.3 | Doba přenosu zprávy 1B při vysílacím výkonu: a)+3dBm b)0dBm | 56 |
| 8.4 | Doba přenosu zprávy 10B při vysílacím výkonu: a)+3dBm b)0dBm | 57 |
| 8.5 | Doby přenosu zprávy 50B při vysílacích výkonech: a)+3dBm b)+1,6dBm c)0dBm d)-3,2dBm e)-7,2dBm f)-17,2dBm | 58 |
| 8.6 | Doba přenosu zprávy 100B při vysílacím výkonu: a)+3dBm b)0dBm | 59 |
| 9.1 | Závislost spotřeby energie na výkonu a velikosti přenášených dat | 62 |

| | | |
|-----|---|----|
| 9.2 | Vybíjecí charakteristika 2xNiMH akumulátorů | 64 |
|-----|---|----|

SEZNAM TABULEK

| | | |
|-----|--|----|
| 1.1 | Tabulka frekvenčních pásem a typy fyzické vrstvy [6] | 16 |
| 2.1 | Struktura BTT tabulky [1] | 21 |
| 5.1 | Tabulka nastavení výstupního výkonu | 32 |
| 5.2 | Spotřeba proudu (převzato z [20]) | 33 |

ÚVOD

Tato diplomová práce se věnuje výzkumu spotřeby elektrické energie při komunikaci mezi dvěma bezdrátovými zařízeními a vlivu změny nastavení vysílacích parametrů na tuto spotřebu.

V posledních letech dochází ke stále většímu využívání bezdrátových sensorových sítí. Jedná se skupinu malých zařízení umístěných v určitém prostoru, které mezi sebou vzájemně komunikují pomocí bezdrátového připojení. Obsahují prvky které jsou malé, jednoduché, levné a mají malou spotřebu energie. Uplatnění nachází například k účelu monitorování, kdy dochází ke sběru dat ze sensorů a následně jsou tato data odeslána do nadřazeného systému ke zpracování a vyhodnocení.

První dvě kapitoly popisují standardy IEEE 802.15.4 a ZigBee používající se v bezdrátové technologii, popis jejich vrstev a topologie. U standardu ZigBee jsou blíže popsány tři typy zařízení: koordinátor, směrovač a koncové zařízení. Dále je rozebrána síťová a aplikační vrstva standardu ZigBee včetně broadcast komunikace a aplikačních profilů.

Další kapitoly se zaměřují na sériový přenos dat. Sériový synchronní přenos je používán u bitově orientovaných protokolů. Je využíván pro datovou komunikaci k přenosu většího objemu dat. Následuje rozbor sériového rozhraní USART používající se pro přenos dat po datové lince. Následuje popis softwaru ZigBee BitCloud, který je určený pro vývoj firmware pro bezdrátové aplikace běžící od firmy Atmel, a jeho architektura.

Následující části se zabírají energetickou náročností v bezdrátových sítích. Nejdříve jsou popsány vlastnosti použitého hardwaru při realizaci praktické části a jejich napájení akumulátory. Poté je vysvětlen energetický model při zpracování zpráv.

Praktická část této práce popisuje použité zařízení a software pro realizaci vytvoření komunikace mezi dvěma bezdrátovými komunikačními jednotkami. Následuje popis praktického řešení zaměřený na zvolený typ zařízení, nastavení parametrů přenosu dat mezi zařízeními a vytvoření sítě. Jsou zde obsaženy části kódu a praktické ukázky realizace.

V posledních kapitoly praktické části obsahují výpočtu dobu přenosu rámce a následné porovnání s naměřenými výsledky, a výpočty spotřeby energie při vysílání a příjmu. Poté je zjištěna výdrž akumulátoru a naměřena vybíjecí charakteristika, a výpočet doby provozu při použití těchto akumulátorů.

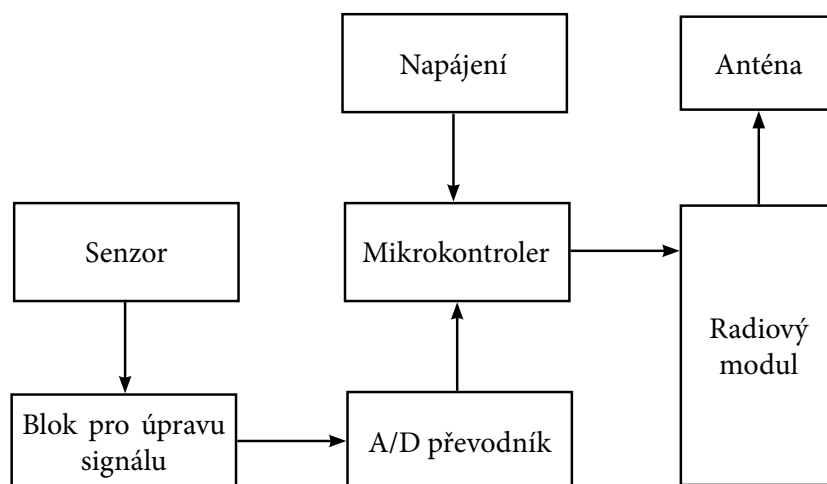
1 BEZDRÁTOVÉ SENZOROVÉ SÍTĚ

Bezdrátová senzorová síť (WSN - Wireless Sensor Network) započala svoji existenci na konci devadesátých let. WSN představuje skupinu malých zařízení, které jsou rozmístěna v určitém prostoru a mohou mezi sebou vzájemně komunikovat pomocí bezdrátového připojení. Mohou sloužit například za účelem monitorování, kdy dochází ke sběru dat ze senzorů a následné odeslání zjištěných dat do nadřazeného systému, který data zpracuje, analyzuje a případně vyhodnotí. Bezdrátovou senzorovou síť může být například radiový systém pro městskou hromadnou dopravu, který zasílá informace o poloze vozidel. Dále to může být meteorologická stanice nacházející se mimo běžně dostupnou oblast, která komunikuje s centrální stanicí pomocí GPRS (General Packet Radio Service). Dále také bezdrátové systémy pro inteligentní domácnosti umožňující ovládat vytápění domu, osvětlení a bezpečnostní zařízení, široké uplatnění ve zdravotnictví, léčebnách, armádě, také jako monitorovací zařízení veřejného osvětlení a mnoho dalších. Všeobecně WSN obsahují prvky, které jsou malé, jednoduché, levné a s malou spotřebou energie [1], [4].

1.1 Technologie v bezdrátové senzorové síti

Bezdrátová senzorová síť je tvořena uzly, které také nesou název *nody* (z anglického node). Uzel bývá složen ze senzorové a výpočetní části a dále napájecího a radiového modulu. Důležitými vlastnostmi pro WSN jsou úspora energie a vysoká flexibilita. Velké množství síťových uzlů se po většinu času nachází v režimu spánku a musí být natolik přizpůsobené, aby právě v tomto režimu byly co nejméně energeticky náročné. Tyto sítě mohou být tvořeny několika uzly až po teoreticky neomezené množství. Vytvořená síť nemá pevně danou infrastrukturu, která by definovala, které dva uzly budou vzájemně spojeny. Každý uzel má prioritu stejnou. Pokud dojde k přidání nového uzlu, může tento uzel navázat spojení s libovolným okolním uzlem a dynamicky se změni topologie sítě [1],[2].

Blokové schéma struktury uzlu WSN sítě je zobrazen na obrázku 1.1. Na vstupu tohoto uzlu je senzor, který snímá vybrané veličiny. Převážně se jedná o integrované obvody s digitálním vstupem. Následují bloky pro úpravu vstupního signálu ze senzoru, dále A/D převodník a hlavní blok celého uzlu mikrokontrolér. Ten bývá obvykle 8-bit pracující při nízkém napětí a obsahuje potřebné komunikační sběrnice I2C (Inter-Integrated Circuit) , SPI (Serial Peripheral Interface), UART (Universal Asynchronous Receiver and Transmitter). Výstupními bloky jsou radiový modul s anténou. Při výběru těchto prvků bývá kladen důraz na dosah vysílání, rozměry a celkové spotřeby zařízení [1].



Obr. 1.1: Struktura uzlu WSN sítě [3]

1.2 Standardy v bezdrátové technologii

V posledních letech dochází ke stále většímu využívání bezdrátové technologie jako komunikačního kanálu pro telekomunikaci, telemetrii a sensorové systémy. Bezdrátové systémy v některých případech přinášejí pro uživatele spoustu výhod oproti běžné sběrníkové komunikaci. Těmito výhodami je efektivnější instalace, velká škála sensorových zařízení a v neposlední řadě také mobilita těchto zařízení. Také je však potřeba si uvědomit i nevýhody, které tyto systémy přinášejí. Jedná se o dobu životnosti zařízení, která bývá omezena z důvodu nutnosti napájení z omezených zdrojů energie a také spolehlivost datového přenosu, jež je ovlivněna rušením z cizích zdrojů v použitém přenosovém pásmu. Avšak i tyto problémy se dají řešit například použitím nezávislých zdrojů elektrické energie (solární panely) a optimalizací radiových spojů v bezdrátové technologii. V neposlední řadě dochází ke srovnání, zda je výhodné používat bezdrátovou technologii nebo kabelové spojení. Výhodami kabelového řešení jsou nízké náklady, spolehlivost a spolehlivý výkon. Bohužel tyto přednosti jsou ocenitelné pouze v případě statické instalace. V případech kdy je zapotřebí přenášení dat z pohybujících se objektů, efektivní řešení přináší použití bezdrátové technologie. Výhodami jsou také nižší náklady na pořízení, kratší prostoje a levnější údržbu zařízení [4].

Bezdrátové sítě obsahující komunikační jednotky s dosahem do několika desítek metrů se nazývají *bezdrátové sítě osobního dosahu*. Jsou označeny anglickou zkratkou WPAN (Wireless Personal Area Networks) [4]. Tyto sítě jsou určeny pro komunikaci přenosových a mobilních zařízení jako jsou například malé osobní počítače, PDA zařízení, mobilní telefony, pagery, různá periferní zařízení a spotřební elektronika. Technologie WPAN je určena pomocí standardu IEEE 802.15, který je součástí standardizačního výboru č. 802 pro normalizaci lokální a metropolitní sítě a je řízen institutem IEEE (Institute of Electrical and Electronics Engineers) [5].

1.3 Standard 802.15.4

Standard IEEE 802.15.4 definuje fyzickou a linkovou vrstvu komunikačního modelu pro rádiové bezdrátové sítě krátkého dosahu. Je určen pro komunikaci malou rychlostí, několik desítek až stovek bitů za sekundu. Hlavními vlastnostmi pro které je standard určen je nízká cena, malá spotřeba energie a jednoduchá implementace [6].

1.3.1 Fyzická a linková vrstva

Fyzická vrstva definuje fyzické vlastnosti jako je komunikace ve frekvenčních pásmech, počet kanálů určené pro přenos, vysílací výkon, druh použité modulace, komunikační rychlost a jiné. Komunikace probíhá v pásmech 868 MHz, 915 MHz a 2,4 GHz. V tomto komunikačním modelu používají vysílače a přijímače techniku rozprostřeného spektra (DSSS - Direct Sequence Spread Spectrum a PSSS - Parallel Sequence Spread Spectrum) a také tři druhy modulace (QPSK - Quadrature Phase Shift Keying, BPSK - Binary Phase Shift Keying, ASK - Amplitude Shift Keying). Bezdrátová zařízení mohou dosahovat maximální přenosové rychlosti 250 kb/s při maximální velikosti datové části rámce na fyzické vrstvě 127 B. Nejrozšířenější je komunikace v pásmu 2,4 GHz. Na rozdíl od ostatních pásem je zde umožněn přenos po šestnácti kanálech, oproti jednomu v pásmu 868 MHz, a vyšší komunikační rychlost. Nevýhodami v tomto ohledu je velké využívání pásma 2,4 GHz (WiFi, Bluetooth, atd.), horší šíření signálu uvnitř budov a menší povolené vysílací výkony. Přehled fyzických vrstev definovaných standardem IEEE 802.15.4 je v tabulce 1.1 [6].

Hlavním úkolem linkové neboli také MAC vrstvy je řízení přenosu dat mezi jednotlivými uzly v síti. Podle standardu IEEE 802.15.4 je založena na metodě náhodného přístupu k médiu CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance). Princip této metody spočívá v tom, že zařízení které chce vysílat po náhodné prodlevě ověří, zda je kanál volný a vyšle paket. Zda je kanál volný lze zjistit pomocí měření úrovně energie na kanálu nebo detekcí nosné frekvence. Také je možné použít obě metody současně. Tato metoda však nemůže zaručit že data

| Pásmo (MHz) | Přenosová rychlost | Modulace | Počet kanálů | Region |
|-------------|--------------------|--------------|--------------|----------|
| 868 | 20 | DSSS, BPSK | 1 | Evropa |
| 868 | 100 | DSSS, O-QPSK | 1 | Evropa |
| 868 | 250 | PSSS, ASK | 1 | Evropa |
| 915 | 40 | DSSS, BPSK | 10 | USA |
| 915 | 250 | PSSS, ASK | 10 | USA |
| 915 | 250 | DSSS, O-QPSK | 10 | USA |
| 2450 | 250 | DSSS, O-QPSK | 16 | Globální |

Tab. 1.1: Tabulka frekvenčních pásem a typy fyzické vrstvy [6]

budou úspěšně doručena, lze pro zvýšení pravděpodobnosti doručení paketu použít automatické přeposílání ARQ (Automatic Repeat reQuest). Pokud za určení čas nedojde k potvrzení doručení paketu ACK (Acknowledgement), dochází automaticky k opakovanému přeposílání. Metoda náhodného přístupu má výhodu ve snadné implementaci a krátké době latence přístupu k médiu. Je zde možné také použít jednodušší hardware a delší dobu spánku zařízení, protože není nutnost synchronizace zařízení. Nevýhodou představuje malá účinnost při velkém zatížení kanálu [6].

1.3.2 Topologie bezdrátové sítě

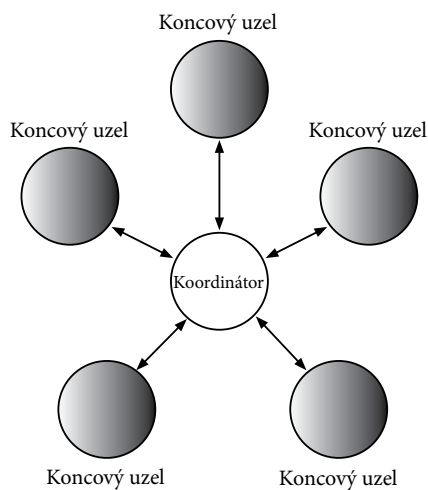
Existuje celá řada topologií pro radiové komunikační sítě. Sít ve standardu IEEE 802.15.4 může mít topologii hvězda nebo peer-to-peer topologii označovanou také jako mesh topologie. Každá síť musí mít alespoň jedno FFD zařízení, které pracuje v režimu koordinátora [3].

Topologie hvězda

V topologii typu hvězda probíhá komunikace přes jeden centrální uzel sítě, který dále přijímá nebo posílá zprávy na koncové uzly. Tento centrální uzel se nazývá koordinátor. Všechny vzdálené uzly mohou tedy posílat nebo přijímat zprávy pouze přes koordinátora, nemohou si zprávy posílat mezi sebou navzájem. Výhodou tohoto typu topologie je její jednoduchost, schopnost dosáhnout nízké spotřeby energie pro koncové uzly a nízká latence mezi vzdáleným uzlem a koordinátorem [3]. Topologie typu hvězda je zobrazena na obrázku 1.2.

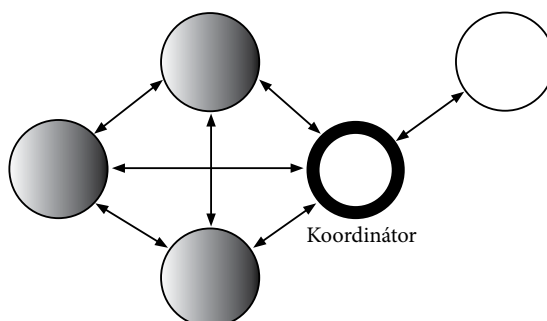
Topologie peer-to-peer

V topologii peer-to-peer je možné přenášet zprávu z jakéhokoliv uzlu v síti na libovolný uzel který je v dosahu sítě aniž by bylo potřeba využívat koordinátora.



Obr. 1.2: Topologie sítě typu hvězda [3]

Tato komunikace je známá jako multihop. Vnitřní uzly sítě slouží jako směrovače při zasílání zpráv z jednoho koncového uzlu do druhého. Koordinátor slouží pouze k řízení sítě. Pokud nastane situace kdy se některý uzel odpojí, koncový uzel může komunikovat s jiným uzlem v jeho dosahu a umožnit tak poslat zprávu na požadované místo. Nevýhodou tohoto typu sítě je ve spotřebě energie pro uzly, které jsou použity jako směrovače. To má za následek kratší životnost baterie. Také čím větší je počet uzlů od mezi koncovými body, tím se také prodlužuje trvání zaslání zpráv. Standard IEEE 802.15.4 není přímo standardem pro směrování, protože nedefinuje síťovou vrstvu. Pro směrování je tedy potřeba využít vyšších vrstev [3].

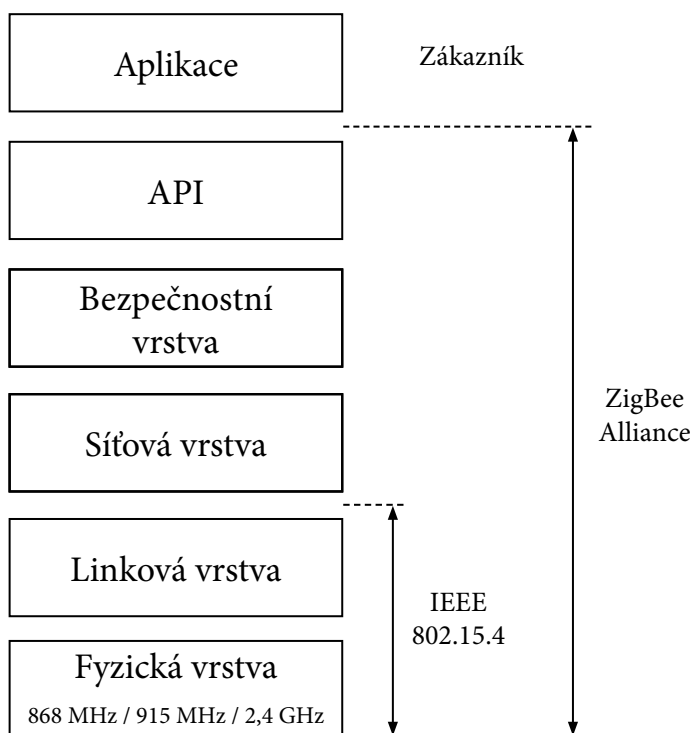


Obr. 1.3: Topologie sítě typu peer-to-peer

2 STANDARD ZIGBEE

ZigBee je standardem pro bezdrátovou komunikaci, který byl navržen převážně pro senzorové a kontrolní sítě. Protokol ZigBee je postaven na standardu IEEE 802.15.4 rozšiřující tento standard o vrstvu síťovou a vrstvu aplikační a služby pro zabezpečení komunikace. Hlavními požadavky jsou jednoduchost implementace, nízká cena a spotřeba energie [4].

Protokol ZigBee rozšiřuje jedno skokovou komunikaci definovanou standardem IEEE 802.15.4 tak, aby byla možná komunikace dvou zařízení, které nejsou ve vzájemném radiovém dosahu, pomocí multihop komunikace. Dále zabezpečují komunikaci pomocí šifrování, definování aplikačních profilů pro různé typy aplikací a jiné. V ZigBee komunikují všechna zařízení v síti na stejném kanále a využívá se zde plně asynchronní přenos [4], [1]. Na obrázku 2.1 je zobrazen OSI model protokolu ZigBee.



Obr. 2.1: OSI model standardu ZigBee

2.1 Typy zařízení standardu ZigBee

V protokolu ZigBee se používají tři typy zařízení: koordinátor, směrovač a koncové zařízení.

ZigBee koordinátor (IEEE 802.15.4 PAN Koordinátor), typ FFD - slouží k vytváření a kontrole sítě, shromažďuje informace o konfiguraci sítě, směruje pakety a může připojovat další zařízení. Je zde potřeba trvalého napájení a nemůže přejít do režimu spánku.

ZigBee router (IEEE 802.15.4 PAN Koordinátor), typ FFD (Full Function Device) - je účastníkem sítě sloužící jako přístupový bod do sítě a přeposílá data ostatním uzlům v síti a rozšiřuje tak pokrytí sítě. Může sloužit i jako zdroj dat. Stejně jako koordinátor je potřeba trvalého napájení.

ZigBee koncové zařízení (IEEE 802.15.4 Device), typ FFD nebo RFD (Reduced Function Device) - je nejjednodušší účastník sítě. Slouží pouze pro příjem nebo odeslání dat. Nepřipojuje žádné jiné zařízení ani nepřeposílá data ostatním uzlům v síti. Koncové zařízení je připojeno do sítě pomocí koordinátora nebo směrovače. Je nejméně náročný na paměť, umožňuje režim spánku a tím šetřit energii, proto může být napájen z baterie [1].

2.2 Adresace v síti PAN

Pro adresaci jednotlivých zařízení v síti je možné použít binární adresovací kódy dlouhé (64 bit) nebo zkrácené (16 bit). Každá síť bývá určena pomocí 16bitového identifikátoru PAN ID, který je použit v případě, kdy v jednom prostoru dochází k provozu více sítí podle standardu IEEE 802.15.4. Každá síť obsahující jedinečný PAN ID je řízena koordinátorem [8]. Složení adresového pole je na obrázku 2.2.

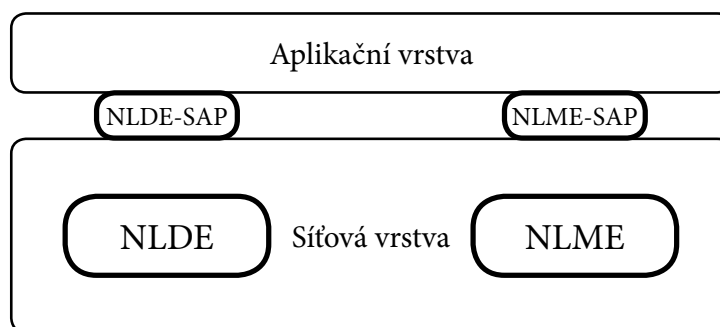
| | | | |
|---------------------------|-----------------------------|-------------------------|---------------------------|
| PAN ID zdroje 0/2 byte | Adresa zdroje 0/2/8 byte | PAN ID cíle 0/2 byte | Adresa cíle 0/2/8 byte |
| Adresové pole | | | |

Obr. 2.2: Adresové pole MAC rámce [8]

2.3 Síťová vrstva

Úkolem síťové vrstvy je připojování k síti a odpojování od ní, zabezpečení a směrování rámců, a umožňuje najít zařízení v rámci jednoho přeskočku. Jako mechanismus pro základní zabezpečení se používá 64bitový nebo 128bitový kryptografický algoritmus AES. Další úkolem je vytváření a udržování směrovacích tabulek sousedních uzlů, a poskytuje informace o vzniklých cestách mezi uzly. V případě koordinátora sítě odpovídá síťová vrstva za start sítě a přiřazuje adresy novým zařízením [7].

Síťová vrstva standardu ZigBee podporuje síťovou topologii typu hvězda (star), síť (mesh) a strom (tree). Síťová vrstva dále poskytuje dva typy služeb. Jedná se o službu *datovou* a službu *řídící*. Obrázek 2.3 popisuje jednotlivé podvrstvy a přístupové body síťové vrstvy. Úkolem podvrstvy NLDE (Network Layer Data Entity) je přenos dat a podvrstva NLME (Network Layer Management Entity) má na starosti řízení směrování a přenosu. Pomocí přístupových bodů NLDE-SAP a NLME-SAP jsou vyšší aplikační vrstvě zpřístupněny služby síťové vrstvy [4].



Obr. 2.3: Síťová vrstva ZigBee [1]

U standardu ZigBee jsou definovány tři způsoby vysílání. Prvním typem komunikace je **unicast** (bod-bod), kdy dochází k vysílání na základě síťové nebo MAC adresy od jednoho zařízení ke konkrétnímu druhému zařízení. Dalším typem je komunikace **multicast** (jeden skupině uzlů), kde je jeden zdroj který vysílá pro definovanou skupinu příjemců v rámci skupiny. Poslední možností komunikace je **broadcast** (jeden všem), jedná se všesměrové vysílání od jednoho zařízení ke všem ostatním. V sítích WSN se nejčastěji používají typy komunikace unicast a broadcast. V případě komunikace unicast tento typ, narozdíl od broadcast komunikace, nepřináší velká implementační úskalí [1], [4].

2.3.1 Komunikace broadcast ve standardu ZigBee

Tato komunikace představuje všesměrové vysílání - odesílání dat ke všem účastníkům sítě. Zařízení koordinátor a router při každém přijetí broadcast rámce zkontroluje zda je jeho adresátem. Pokud zjistí že ne, posílá tento rámec všesměrově dál. Koncové zařízení (End device) nemůže broadcasty vysílat ani přeposílat [1].

Standardy IEEE 802.15.4 a ZigBee definují následující broadcast adresy (používá se krátká MAC adresa):

0xFFFF komunikace broadcast ke všem zařízením.

0xFFFFD komunikace broadcast ke všem aktivním zařízením.

0xFFFFC komunikace broadcast pouze k zařízením typu router a koordinátor [1].

Při vysílání broadcast je adresa cílového uzlu nastavena na 0xFFFF (65535 decimálně). Ve chvíli kdy dojde k přijetí paketu s touto adresou je paket zařízením kterého přijalo uložen do paměti, dojde ke zpracování a po náhodně dlouhé době je vyslán zpátky do sítě. Tato doba se označuje jako *broadcast jitter* a vkládá se před vysláním, aby se snížil počet kolizí. Pokud by nastala situace kdy by náhodné zpoždění nebylo použito, všechny sousední uzly by přeposlaly daný broadcast paket ve stejný čas a tím by docházelo k častým kolizím. Každý paket obsahuje ve své paměti sekvenční číslo. V případě kopie daného broadcast paketu se paket zahodí na základě kontroly sekvenčního čísla. Broadcast komunikace nepoužívá kvůli velké režii a možného zahlcení sítě potvrzování pomocí ACK zpráv. v ZigBee je definováno tzv. *pasivní potvrzování*, které funguje na principu kontroly, zda sousední uzel paket přijal. Zařízení které broadcast paket vyslalo se ihned po odeslání přepne do režimu přijímání a čeká zda přijme paket se stejným sekvenčním číslem. Pokud tato situace nastane, broadcast paket byl úspěšně zaslán. Zařízení koordinátor a router, které můžou vysílat všesměrově, udržují broadcastové záznamy BTR (Broadcast Transaction Record) v broadcastové tabulce BTT (Broadcast Transaction Table). Doba záznamu BTR je časově omezena [4], [1]. Struktura BTT tabulky je zobrazena na obrázku 2.1.

| | | |
|----------------------|----------------------|------------------------|
| Zdrojová adresa (2B) | Sekvenční číslo (1B) | Expirační časovač (1B) |
|----------------------|----------------------|------------------------|

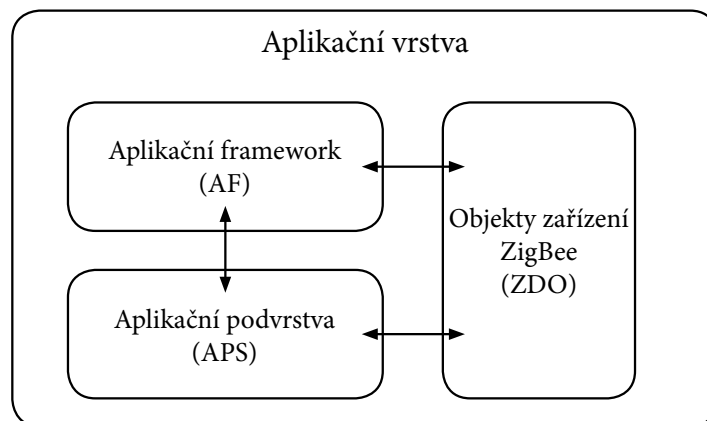
Tab. 2.1: Struktura BTT tabulky [1]

Tato tabulka slouží jako ochrana proti zacyklení broadcastu. Zařízení porovná každý přijatý broadcast s BTT. Pokud se přijatý broadcast shoduje s nějakým záznamem z BTT, tak zařízení tento broadcast zahodí. V opačném případě, zda se broadcast v tabulce nenachází, zařízení si vytvoří BTR záznam shodný s přijatým

broadcastem a tento broadcast přepošle dále. V situaci kdy nebyl broadcast doručen se vysílání opakuje. Záznam BTR je z tabulky BTT odstaněn až po uplynutí definovaného času BTR. Když je broadcast tabulka již plná, všechny příchozí broadcasty se zahazují [1].

2.4 Aplikační vrstva

Jedná se o nejvýše definovanou vrstvu ve standardu ZigBee. Je složena z aplikační podvrstvy - APS (Application Support Sub-layer), z objektů zařízení ZigBee - ZDO (ZigBee Device Object) a aplikačních objektů definovaných výrobcí - AF (Application Framework). Aplikační podvrstva udržuje vazební tabulky aby bylo možné propojit dvě zařízení na základě jejich služeb a potřeb, a přeposílá zprávy mezi zařízeními vzájemně sebou vázanými. Objekt ZigBee slouží pro definici, zda zařízení v síti je koordinátor nebo koncové zařízení. Dále vytváří spojení nebo odpovídá na žádosti a zřizuje zabezpečené spojení zařízeními v síti. Umožňuje také hledat zařízení v síti a zjišťovat jimi poskytované služby [8]. Složení aplikační vrstvy znázorňuje obrázek 2.4.



Obr. 2.4: Struktura aplikační vrstvy standardu ZigBee

Úkolem podvrstvy APS je vytvoření rozhraní mezi síťovou a aplikační vrstvou. V další podvrstvě AF jsou uloženy aplikační profily zařízení ZigBee. Na jedno síťové zařízení může být adresováno až 240 aplikací. Každé aplikaci je přiřazeno identifikační číslo v rozsahu 1 - 240. Tyto identifikátory se nazývají *endpointy* (přístupové body). Pro vrstvu ZDO je vyhrazena adresa endpoint 0. Ve standardu je možné aby

jedno zařízení komunikovalo s libovolným zařízením definovaným podle aplikačního profilu za použití stejného frekvenčního kanálu. Podvrstva ZDO slouží k vytvoření rozhraní mezi bloky APS a AF a je určena pro konfiguraci zařízení ZigBee koordinátor, router nebo koncové zařízení. K adresování individuálních aplikací na zařízení se používají adresy Endpoint, Profile ID a Cluster, které jsou definovány na aplikační vrstvě. Samotné zařízení se adresuje pomocí linkové a síťové adresy [4].

V ZigBee se používají adresy:

PAN ID - jedná se o 16bitovou adresu PAN sítě

MAC Adrr - rozšířená 64bitová adresa na linkové vrstvě

NWK Adrr - 16bitová adresa na fyzické vrstvě

Endpoint - 8bitový identifikátor aplikace na aplikační podvrstvě APS

Profile ID - 16bitový identifikátor aplikačního profilu

Cluster - objekt obsahující množinu atributů a příkazů. Velikost clusteru 16 bitů [4].

2.4.1 Aplikační profily

Aplikační profily se využívají pro sjednocení přenosu dat a umožňují aby zařízení mezi sebou mohla vzájemně operovat na aplikační vrstvě. Profily jsou navrženy podle konkrétního aplikačního nasazení a každý profil slouží k definici množiny zařízení a povelů, které se mohou na dané zařízení aplikovat. Také je určen k definici formátu aplikačních zpráv a ostatní potřebné vlastnosti v rámci aplikačního prostoru. Hlavním úkolem je jednoduchý a sjednocený vývoj aplikací. Přesněji řečeno to znamená, že zařízení které bylo vyrobeno jedním výrobcem může ovládat zařízení od jiného výrobce za předpokladu, že obě tyto zařízení mají stejný aplikační profil. Každý aplikační profil je identifikován 16bitovým číslem (profile ID) a je složen ze dvou základních částí - *Cluster* a *Deskriptor* [1].

Cluster

Cluster obsahuje množinu příkazy a atributy. Každý cluster má svůj vlastní 16 bitový identifikátor Cluster ID. V aplikačním profilu se může nacházet několik clusterů. I když aplikační profily nemají uložené samotné clustery, ale pouze jejich čísla. **Atributy** jsou určeny pro ukládání dat a stavů. Například naměřená hodnota z čidla se uloží do některé z atributů. Také atributy jsou označeny svým unikátním 16bitovým identifikátorem Attribute ID. **Command** je označení pro příkaz v rámci clusteru vykonávající danou operaci a obsahující 8bitový identifikátor Command ID [1], [4].

Deskriptory

Používají se pro informační popis konfigurace zařízení. Poskytují informace o typu zařízení (koordinátor, router, koncové zařízení), pracovní frekvenci zařízení a zbývající kapacitě baterie. Zařízení si pomocí deskriptorů mezi sebou vyměňují podporované funkce a díky těmto informacím se následně spárovat a zjednodušit tak celý proces konfigurace sítě. Výměna deskriptorů se provádí pomocí ZDO. Deskriptory mají opět identifikátory s vlastním 16bitovým číslem a obsahují 5 informačních bloků [1], [4].

Node descriptor (Popis uzlu) - nese informace zda je zařízení typu koordinátor, router nebo koncové zařízení. Také informuje o použití Complex a User deskriptoru, použitém frekvenčním pásmu, napájení, kód výrobce a další možnosti.

Simple descriptor (Jednoduchý popis uzlu) - popisuje možnosti Endpointu a spojuje identifikátory Endpoint ID, Profile ID, Cluster ID a Device ID do společného popisovače. Tento deskriptor neobsahuje žádný cluster, ale pouze identifikátory použitých clusterů. Dochází zde k předpokladu, že zařízení které zjišťuje popis daného endpointu dokáže podle identifikátoru zjistit o jaký cluster se jedná. Při výměně informací o endpointu pomocí identifikátoru není potřeba posílat cluster samostatně, ale stačí pouze Cluster ID.

Power descriptor (Popis napájecího uzlu) - obsahuje informace o bateriovém napájení, zbývající kapacitě, a jiné.

User descriptor (Uživatelským definovaný popis) - představuje volitelné doplňkové informace uživatele. Umožňuje velikost až 16 ASCII znaků.

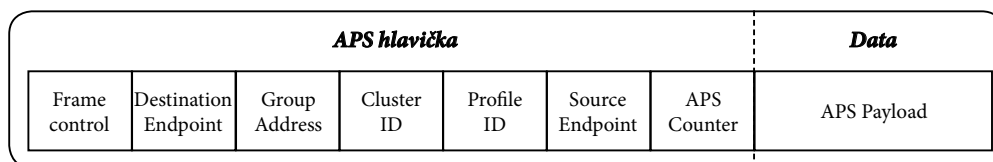
Complex descriptor (Úplný popis uzlu) - informuje o volitelné položce, sériovém čísle zařízení a jeho názvu.

Použití Power, User a Complex deskriptoru není příliš časté a je možné je nahradit použitím Basic clusteru [1], [4].

2.4.2 Aplikační podvrstva APS

Poskytuje datové služby pro podvrstvy AF a ZDO. Kromě typu komunikace unicast, multicast a broadcast podvrstva APS definuje další komunikační model *nepřímé adresování*. Tento typ komunikace se využívá v situaci kdy vysílací zařízení nezná adresu příjemce, zašle data koordinátorovi a ten následně data odešle na adresu příjemce. V tomto případě je však potřeba aby odesílatel byl s příjemcem spárován

v párovací tabulce. Zde jsou uloženy informace o síťové adrese příjemce, Endpoint adresa a číslo clusteru [4]. Struktura APS rámcí je zobrazena na obrázku 2.5.



Obr. 2.5: Struktura APS rámcí [1]

Pole *Frame Control* definuje typ rámcí. Může se jednat o typ datový, příkazový nebo potvrzovací. Následují pole *Destination Endpoint* a *Group Address*, které se však nevyskytují v rámci zároveň. Pokud je nastaveno pole *Group Address*, tak je rámcí doručen ke všem prvkům obsahující Endpoint adresy a patřící do dané skupiny. Pole *Cluster ID* obsahující číslo clusteru, které se použije pro spárování dvou zařízení. V dalším poli je obsaženo číslo aplikačního profilu. Poslední pole *APS Counter* se používá pro identifikaci duplicitních rámcí. Po každém přenosu je počítadlo inkrementováno [4].

3 SÉRIOVÝ PŘENOS DAT

Jsou-li data (bajty) přenášena postupně bit po bitu, jedná se **sériový přenos dat**. Ve většině sítí se používá právě tento typ komunikace. Nejmenší položka, která je sériově přenášena se označuje jako znak (character) a jeho rozsah je 7 nebo 8 bitů. Znak vyjádřený ve formě posloupnosti dvojkových bitů, které se přenášejí je označen jako značka. Pro sériovou komunikaci jedním směrem je dostačující použití pouze jednoho vodiče, a umožňuje přenos dat na větší vzdálenosti. Pokud dochází k přenosu celého bajtu najednou, jedná se **přenos paralelní**. Při této komunikaci je potřeba aby byl zajištěn dostatečný počet souběžných (paralelních) vodičů. V tomto ohledu se paralelní přenos dat využívá spíše pro krátké vzdálenosti. Typickým paralelním přenosem dat je propojení počítače s tiskárnou přes tzv. sériové rozhraní [10].

Pro oba typy komunikace (sériovou a paralelní) je potřeba dodržet, aby přijímač správně vyhodnotil okamžik platnosti jednotlivých značek, které byly vygenerovány vysílačem. Proto je potřeba, aby mezi vysílačem a přijímačem byla nějakým způsobem zajištěna časová synchronizace. Podle druhu synchronizace se sériový přenos dat dále rozděluje na synchronní a asynchronní [9].

3.1 Sériový asynchronní přenos

Při tomto přenosu mohou být jednotlivé značky přenášeny s libovolným časovým odstupem mezi sebou. Příjemce však dopředu netuší kdy má očekávat další znak a proto musí nějakým způsobem být schopen tuto situaci rozpoznat. Pro takový případ se využívá tzv. start-bit, kterým začíná každý asynchronně přenášený znak. Příjemce po obdržení start-bitu ví kdy má očekávat příjem znaků, a také mu poslouží pro správné nastavení časové základny. To je potřeba, aby příjemce správně určil časové okamžiky, kdy má vyhodnotit stavy jednotlivých datových bitů, které po start-bitu následují [10].

Za datovými bity může následovat paritní bit a nakonec stop-bit, který ovšem v sobě žádné informace nenese. Jeho smyslem je zajištění minimálního odstavu mezi jednotlivými datovými znaky. Vyslání dalšího znaku může proběhnout až po odvyslání celého předchozího znaku, včetně stop-bitu [10].

3.2 Sériový synchronní přenos

U synchronního přenosu se přenášejí celé bloky znaky. Datové bity jednotlivých znaků následují těsně za sebou, nedochází zde k žádným časovým odstupům a také

nejsou použity žádné start a stop bity. Avšak začátek bloku bývá označen speciálním synchronizačním znakem (znak SYN). Úkolem tohoto synchronizačního znaku je zajištění potřebné časové synchronizace odesílatele i příjemce. To umožní příjemci přesně určit časové okamžiky, ve kterých má jednotlivé datové bity vyhodnocovat. Na konci bloku znaků se opět nachází synchronizační znaky. Ty mohou být nepřetržitě vysílány až do začátku dalšího datového bloku [10].

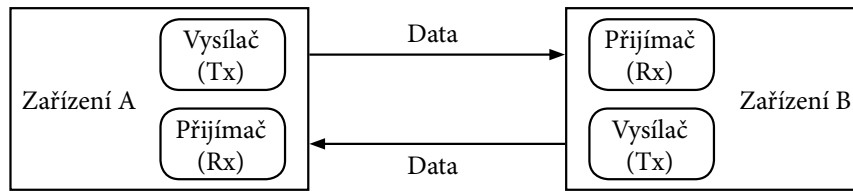
Synchronní přenos je ve srovnání s asynchronním přenosem rychlejší, protože není zatížen režii start a stop bitů. Synchronní přenos se často používá u bitově orientovaných protokolů, kde dochází k seskupení informací do rámců. U datové komunikace se využívá převážně pro přenos větších objemů dat [10], [9].

3.3 USART

Jedná se o zařízení určené pro sériovou komunikaci. Může být nastaveno pro asynchronní nebo synchronní režim. Zařízení které komunikuje po USART (Universal Synchronous and Asynchronous Receiver and Transmitter) sběrnici nemusí používat synchronizační linku, ale zařízení se navzájem synchronizují zasíláním start a stop bitů, sloužící k ohraničení každého rámce. Proto je potřeba k přenosu jednoho bajtu po datové lince přenést celkem 10 bitů. Takovýto princip se používá u asynchronní sběrnice UART, kde nedochází k časové synchronizaci. U synchronního přenosu pomocí USART může být použit externí zdroj taktu pro synchronizování vysílače s přijímačem. Pokud zařízení umožňuje použití USART sběrnice, lze se rozhodnout zda budou data zasílána asynchronně nebo synchronně pomocí synchronizační linky. USART pro každý směr přenosu používá jednu datovou linku, tento model přenosu je zobrazen na obrázku 3.1. V případě asynchronního módu je umožněn přenos dat v duplexním režimu, tzn. je umožněno současně vysílat i přijímat data. Pokud je zvolen režim synchronní, je možno pracovat jako Master nebo Slave [4], [11].

Protože od přijímač neví jak dlouho od příjmu start bitu má data načítat, jsou pro USART přenos definovány přenosové rychlosti (Baud Rate). Jakmile má přijímač nastavenou přenosovou rychlost, ví jak dlouho trvá přenos jednoho bitu a dokáže tak zjistit kdy bude obdržén stop-bit pro ukončení přenosu. Rychlost přenosu dat může být vyjádřena v bitech za sekundu (b/s) nebo v Baudech. Pokud je tedy nastavena přenosová rychlost 4800 baudů, lze po datové lince přenést 4800 bitů za sekundu, přenos jednoho bitu bude trvat $1/4800 = 208 \mu\text{s}$. K přenosu dat po USART se používá protokol RS232 pracující s úrovněmi HIGH = -12 V a LOW = +12 V [4].

V případě vysílacího zařízení vypadá proces přenosu následovně. Pokud před samotným přenosem ještě zařízení nevysílá žádný signál, je na datové lince signál



Obr. 3.1: Komunikace po USART sběrnici [11]

s úrovní HIGH. Tudíž jsou na lince stop-bity. V případě kdy jsou již data připravena k vyslání, je vyslán start bit, jehož hodnota odpovídá signálu s úrovní LOW. Pro zvolenou rychlost 4800 baud trvá tento start bit $208 \mu\text{s}$. Jakmile byl start-bit vyslán, zařízení vysílá jednotlivé bity se šířkou $208 \mu\text{s}$. Po vyslání osmého bitu dochází k nastavení úrovně signálu HIGH minimálně na $208 \mu\text{s}$ a tím dojde k oznámení konce jednoho bajtu. Pokud přenos dále pokračuje, signál je nastaven na úroveň LOW po dobu $208 \mu\text{s}$, čímž dochází k vygenerování nového start-bitu a pokračování přenosu dat. Na straně přijímače je postup následující. Přijímač detekuje signál s úrovní LOW na vedení. Pokud je signál doba signálu kratší než polovina trvání bitu, tzn. $104 \mu\text{s}$, je signál považován za šum a zařízení na něj nereaguje. V opačném případě zařízení po uplynutí doby $104 \mu\text{s}$ spustí časovač a následně každých $208 \mu\text{s}$ načte úroveň signálu. Hodnoty se tedy načítají vždy z poloviny signálu. Příjem jednoho bajtu je u konce až po načtení 8 bitů, v tomto případě po $1664 \mu\text{s}$ [4].

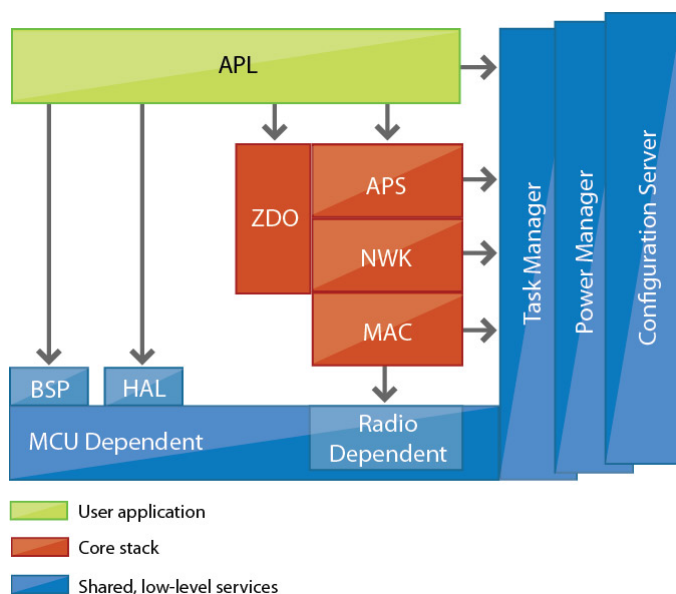
Pro kontrolu rámce při přijetí je možné nastavit paritní bit na obou stranách. Paritní bit je přidán na konec rámce a jeho funkcí je zaručit, že počet bitů s hodnotou 1 je buď lichý nebo sudý. Podle tohoto požadavku se na obou stranách komunikace nastaví buď lichá nebo sudá parita. Pokud je počet bitů v rámci s hodnotou 1 lichý a USART je nastaven na sudou paritu, pak je na konec rámce přidán bit s hodnotou 1, aby byla dodržena podmínka počtu jedničkových bitů. V případě kdy je parita dodržena, je přidán na konec paritní bit s hodnotou 0. Přijímací zařízení ví zda má očekávat lichý nebo sudý počet bitů s hodnotou 1. Jakmile se počet jedničkových bitů rovná nastavené hodnotě parity, pak přijímací zařízení potvrdí správně přijatá data [4].

4 ZIGBEE BITCLOUD

Jedná se o softwarový stack určený pro vývoj firmware pro spolehlivé, škálovatelné a bezpečné bezdrátové aplikace běžících od firmy Atmel. BitCloud je navržen pro podporu uživatelských aplikací splňující různé požadavky a umožňuje plnou škálu přizpůsobení. Tento software je určen jak pro domácí, tak i průmyslové účely. BitCloud je plně kompatibilní se ZigBee PRO a ZigBee standard, a poskytuje možnost rozšíření rozhraní API a tím využít rozšířené funkce [12].

4.1 Architektura BitCloud

Vnitřní architektura BitCloud navazuje na síťové a logické vrstvy, které se nachází ve standardu IEEE 802.15.4 a standardu ZigBee. Kromě hlavního zásobníku obsahující vykonávací protokoly obsahuje další vrstvy provádějící sdílené služby. Jedná se o správce úloh, službu bezpečnosti a nebo správce napájení [12].



Obr. 4.1: Architektura BitCloud (převzato z [12])

Celá architektura sítě je zobrazena na obrázku a je rozdělena na 3 části. Zeleně oddělená část je uživatelská aplikační vrstva. Zde se vytvářejí kompletní uživatelské obslužné kódy, např. *změř teplotu*.

Oranžová barva označuje vrstvu, kterou definuje standard ZigBee. V této vrstvě se nachází 4 části, kde každá část přidá vlastní informaci k datům která budou odeslána.

Popis jednotlivých vrstev v BitCloud:

Vrstva APL - uživatelem vytvořená aplikace.

Vrstva APS - podpora podvrstvy aplikací.

Vrstva ZDO - objekt zařízení ZigBee.

Vrstva MAC - linková vrstva.

Vrstva NWK - síťová vrstva.

Task Manager (správce úloh) - jedná se o plánovač a časovač, zprostředkovává použití MCU a uživatelské aplikace.

Power Manager - odpovídá za správu napájení, je odpovědný za vypnutí všech periférií při přechodu do úsporného režimu a jejich opětovné obnovení při aktivním režimu.

Configuration Manager - slouží pro definici funkcí pro nastavení základních parametrů bezdrátového přenosu. Jedná se o adresu zařízení, PAN ID, typ zařízení, atd.

Vrstva HAL - vrstva obsahující soubor funkcí pro ovládání základních hardwarových zařízení jako je EEPROM paměť, USART, SPI, IRQ, apod.

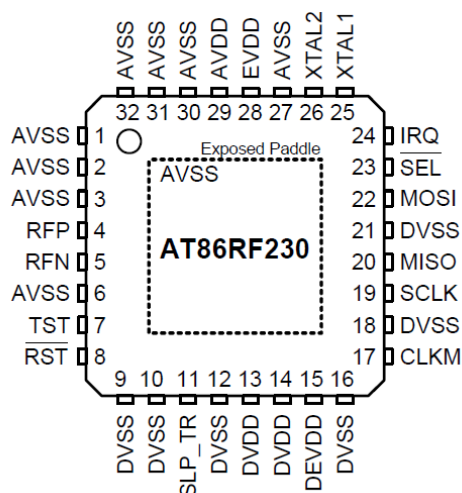
Vrstva BSP - obsahuje ovladače periférií připojených na mikrokontrolér jako jsou tlačítka, senzory, čipy, diody, a jiné [12].

5 POUŽITÝ HARDWARE

5.1 Rádiový čip AT86RF230

V následující kapitole je popsán rádiový čip AT86RF230 od společnosti Atmel. Tento čip bývá jedním z nejčastějších rádiových čipů pro nízkoenergetické bezdrátové systémy a také je použit pro praktické řešení této diplomové práce [20].

Na obrázku 5.1 je graficky znázorněno schéma rádiového čipu AT86RF230 s popisem pinů.



Obr. 5.1: Popis pinů modulu AT86RF230, (převzato z [20])

Čip AT86RF230 patří do skupiny nízkoenergetických rádiových přijímačů pracujících v pásmu 2,4 GHz a je navržen pro sítě WPAN a aplikace ZigBee / IEEE802.15.4. Přes sběrnici SPI je propojen s mikrokontrolérem a umožňuje rádiové rozhraní mezi anténou a mikrokontrolérem. V sobě má obsažen analogový přijímač, digitální demodulátor O-QPSK s časovou a frekvenční synchronizací. Dále obsahuje paměť RAM s velikostí 128 B, která slouží jako úložiště dat pro přijatá nebo odesílaná data. Tento typ paměti má nevýhodu, při odpojení napájení dochází ke ztrátě dat. K čipu je možné připojení externí antény, krystalového oscilátoru a oddělovací kondenzátory. U zmíněného čipu je možné nastavit vysílací výkon v 16-ti různých úrovních od 3 dBm do -17,2 dBm [20]. Všechny tyto úrovně jsou zobrazeny v tabulce 5.1.

Tab. 5.1: Tabulka nastavení výstupního výkonu

| Úroveň | Hodnota [3:0] | Výstupní výkon [dBm] |
|--------|---------------|----------------------|
| 0 | 0x0 | +3.0 |
| 1 | 0x1 | +2.6 |
| 2 | 0x2 | +2.1 |
| 3 | 0x3 | +1.6 |
| 4 | 0x4 | +1.1 |
| 5 | 0x5 | +0.5 |
| 6 | 0x6 | -0.2 |
| 7 | 0x7 | -1.2 |
| 8 | 0x8 | -2.2 |
| 9 | 0x9 | -3.2 |
| 10 | 0xA | -4.2 |
| 11 | 0xB | -5.2 |
| 12 | 0xC | -7.2 |
| 13 | 0xD | -9.2 |
| 14 | 0xE | -12.2 |
| 15 | 0xF | -17.2 |

Vlastnosti rádiového čipu (podle [20]):

Nastavení výstupního výkonu – od -17,2 dBm do +3 dBm

Nízká spotřeba

– režim spánku: 20 nA

– režim příjmu: 15,5 mA

– režim vysílání: 16,5 mA (při maximálním výkonu 3 dBm)

Napájecí napětí – 1,8 V

Paměť SRAM – 128 B pro data vyrovnávací paměti

Následující tabulka 5.2 uvádí hodnoty odběru proudu při vykonávání daných úkonů. Tyto teoretické údaje o spotřebě jsou zjištěny pro při napájecím napětí $U_{DD} = 3V$, frekvenci $f = 2,45$ GHz a teplotě prostředí $T = 25$ °C.

Tab. 5.2: Spotřeba proudu (převzato z [20])

| Parametr | Hodnota [mA] | Situace |
|----------------------------------|--------------|--------------------|
| Napájecí proud v režimu vysílání | 16,5 | $P_{TX} = 3$ dBm |
| | 14,5 | $P_{TX} = 1$ dBm |
| | 12,5 | $P_{TX} = -3$ dBm |
| | 9,5 | $P_{TX} = -17$ dBm |

Při každé úrovni výkonu se může spotřeba proudu snížit o cca. 2 mA, klesne-li napětí U_{DD} na hodnotu 1,8V.

5.2 Napájení

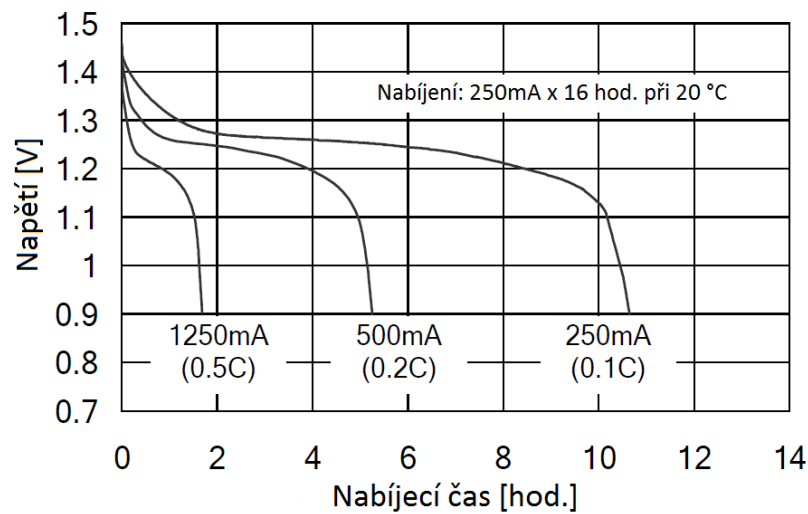
Důležitým aspektem z hlediska celkové výdrže síťového uzlu je efektivita napájení. Patří mezi klíčové parametry pro zařízení WSN. Platforma hardwaru je napájena z běžně dostupných akumulátorů. Pro tuto práci byly zvoleny akumulátory typu NiMH od společnosti GP Batteries. Pro uzly WSN nejsou ani tak důležité vlastnosti baterií jako je počet nabíjecích cyklů, rychlost nabíjení a maximální vybíjecí proud, ale důležitými parametry je kapacita, nominální napětí, samovybíjení a závislost na okolní teplotě [22].

Pokles napětí je závislý na zátěži, tzn. na velikosti proudu, který dané zařízení odebírá. Tento parametr jsem však v aplikacích WSN zanedbatelný. Ve většině literatury o WSN se běžně dělají odhady kolik měsíců, popř. roků, dané zařízení může být napájeno z běžného akumulátoru. V tomto případě je brát také v úvahu problém, že každý akumulátor je samovybíjen. Tento parametr je natolik velký, že se nedá zanedbat. Akumulátory NiMH se prakticky sami vybijí během jednoho roku. Nové typy NiMH akumulátorů mají samovybíjení na hranici 6 % za měsíc. Samovybíjení není lineární a při vybíjení akumulátoru se zmenšuje [22].

Vlastnosti akumulátoru GP270AAHC typu NiMH (podle [23]):

- Nominální napětí 1,2 V
- Stav nabití: 250 mA na 16 hodin při teplotě 20 °C
- Závislost na teplotě
- Nízká cena

– Maximální nabíjecí napětí: 1,5V 250mA při nabíjení

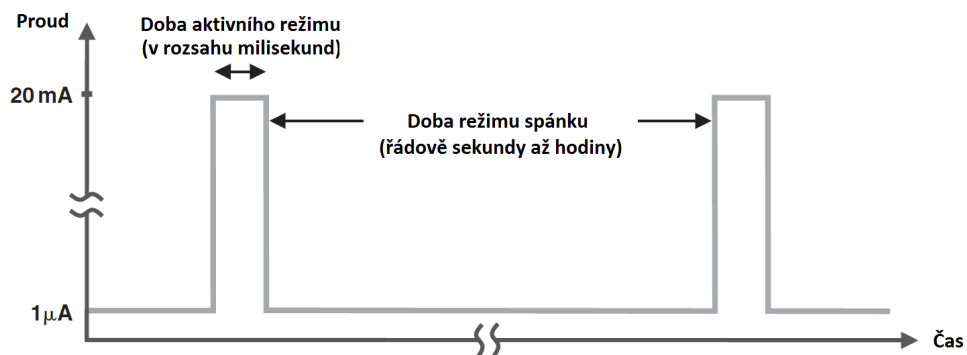


Obr. 5.2: Vybíjecí průběhy (použito z [23])

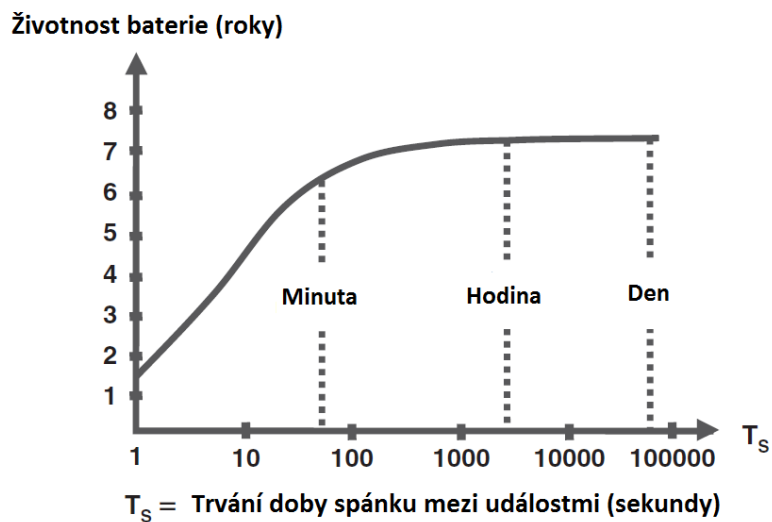
Pro zařízení v tomto případě bude špičkový odběrový proud přibližně 28 mA. Výsledný proud je dán spotřebou dílčích částí zařízení: mikrokontrolér ATmega1281 - 8 mA, RF modul AT86RF230 - 16,5 mA při zaslání dat s výkonem +3dBm a 15,5 mA v režimu příjmu, ostatní periférie - 2 mA.

Ve většině bezdrátových sensorových sítích krátkého dosahu je průměrná spotřeba proudu zařízení nízká, avšak může také nastat okamžik kdy je okamžitý odběr proudu vysoký. Jak je uvedeno na obrázku 5.3, vysílač má v režimu spánku odběr proudu $1\mu\text{A}$. Ovšem každé 2 sekundy se vysílač dostává do aktivního ve kterém zůstává po dobu 5 ms. Při tomto aktivním režimu stoupne maximální činný proud na hodnotu 20 mA. Průměrná spotřeba proudu vysílače je pouhých $50\mu\text{A}$, ale špičkový proud 20 mA může mít nepříznivý vliv na skutečnou kapacitu baterie, a to zejména v případě, kdy nemá baterie dostatek času na odpočinek a zotavení mezi obdobími vysokého vybíjecího proudu [24].

Skutečnou kapacitu akumulátoru pro konkrétní scénáře použití lze experimentálně stanovit. Výrobci baterií jsou schopni poskytnout skutečný odhad kapacity baterie v případě pokud je k dispozici aktuální profil aplikace (např. obr. 5.3) Poměr skutečné schopnosti akumulátoru k jeho jmenovité kapacitě se nazývá *battery efficiency factor*. Jedním ze způsobů jak se vyhnout období s vysokým vybíjecím proudem je použití dostatečně velkého kondenzátoru, který dodává proud vysílači v aktivním režimu. Když je zařízení v režimu spánku, baterie nabíjí kondenzátor a jakmile je zařízení aktivní tento proud dodá poskytovaný proud do zařízení. Tímto způsobem se zajistí delší vybíjecí doba baterie a zvýšení účinnosti [24].



Obr. 5.3: Okamžitá spotřeba proudu zařízení (převzato z [24])



Obr. 5.4: Impulzní odběh z akumulátoru (převzato z [24])

Na obrázku 5.4 je znázorněn vliv impulzního odběru na životnost baterie. Z toho vyplývá, že čím větší je délka režimu spánku mezi událostmi, tím se zvětšuje životnost baterie. Jakmile je čas mezi událostmi větší než jedna minuta, začíná být potlačen negativní vliv špičkového odběru [24].

6 ENERGETICKÁ NÁROČNOST

Protože bezdrátové sensorové uzly mají omezené možnosti na použité energické zdroje, je velký požadavek kladen na spotřebu energie při procesech. Proto je důležitá analýza spotřeby elektrické energie a tím zjištění jakou má uzel životnost v rámci sítě. Spotřebu energie nelze vypočítat pouze z katalogového listu, kde jsou udány hodnoty odběru proudu při různých délkách vysílaného paketu, ale i jakými intervaly jsou pakety zasilány a klidový režim mezi jednotlivými přenosy. Následující část je zaměřena na analýzu spotřeby energie při RF komunikaci využívající standart 802.15.4. Pro zlepšení úspory energie mohou být použity různé varianty v celém protokolu. Taková technika obsahuje režim spánku, síťové směrovací protokoly, lokalizace, zabezpečení a jiné energetické služby a funkce [13].

6.1 Energie v bezdrátových sítích

Stejně jako je pro bezdrátové sensorové sítě důležité přenosové pásmo a výpočetní výkon, tak i energie je zde důležitým prostředkem. Důležitým aspektem který je třeba si uvědomit je, že bezdrátové uzly mají k dispozici pouze energii dodávanou z bateriového zdroje. Pokud dojde k vybití, uzel se vypne a přeruší se komunikace. Navíc některé uzly neslouží pouze jako koncové zařízení ale jsou použity jako směrovač, kde dochází k většímu odběru energie a tím jsou nároky na zabezpečení dostatečné energie větší. Tyto uzly mohou být napájeny i pomocí elektrické sítě, ale vždy je tato možnost realizovatelná, např. v případech kdy se uzly nachází v těžko dostupných místech. Další variantou zajištění energie pro uzly mohou představovat solární panely [13].

6.2 Spotřeba energie v bezdrátových sítích

Spotřeba energie každého uzlu WSN je složena ze třech hlavních úkolů provádějících se uvnitř uzlu [14]:

Výpočetní výkon mikrokontroléru - je odvozena od spotřeby mikrokontroléru, který obsahuje mikroprocesor, paměť a A/D převodník. Tato výpočetní jednotka pracuje při každé činnosti kterou uzel provádí. Jedná se hlavně o sběr dat a jejich zpracování.

Komunikace RF - spotřeba při komunikaci je složena ze spotřeby při přijímání paketů (R_x power) a při odesílání paketů (T_x power). Velký vliv představuje maximální dosah přijímací a vysílací části.

Snímání dat - v tomto případě záleží na typu senzoru, na snímané veličině a také na způsobu jakým je snímána. Sensory jsou aktivovány pro snímání a pro sběr dat z okolního prostředí. Každý typ senzorů spotřebovává různé množství energie. Množství energie se může pohybovat v rozmezí od jednotek mA až po stovky mA.

Po sečtení všech těchto úkolů dostaneme výraz 6.1 celkové spotřeby energie:

$$E = E_{\mu p} + E_{RF} + E_{sensor}, \quad (6.1)$$

kde $E_{\mu p}$ je spotřeba energie mikroprocesoru, E_{RF} je spotřeba při komunikaci a E_{sensor} spotřeba energie všech používaných čidel. Největší spotřebu má uzel při komunikaci, kdy nasbíraná data přijímá a vysílá. Energetická náročnost výpočetního bloku je narozdíl od komunikace výrazně nižší. Spotřeba energie se převážně řeší na fyzické a linkové vrstvě [13], [14].

6.2.1 Spotřeba energie přijímací části

Spotřeba v přijímací části je závislá na spotřebě obvodů, která přijímaný signál zpracovávají, a na počtu přijatým bitů. Výpočet energetické náročnosti přijímací části se provádí pomocí rovnice 6.2.

$$E_{rx} = E_d \cdot n, [J] \quad (6.2)$$

kde E_{rx} je energie při přijímání paketů o n bitech a E_d spotřeba obvodů při zpracování signálu na 1 bit [14].

6.2.2 Spotřeba energie vysílací části

Spotřeba energie ve vysílací části je ovlivněna několika parametry. Závisí na velikosti odesílaného paketu, na vzájemné vzdálenosti komunikujících uzlů, způsobu použité modulace a vysílacím výkonu zesilovače. Podle rovnice 6.3 se provádí výpočet energetické náročnosti vysílací části.

$$E_{tx} = E_m \cdot n + E_z \cdot n \cdot d^2, [J] \quad (6.3)$$

kde E_{tx} představuje spotřebovanou energii při vysílání paketů o n bitech na vzdálenost d metrů, E_m energie při modulaci jednoho bitu a E_z energie kterou spotřebovává zesilovač [14].

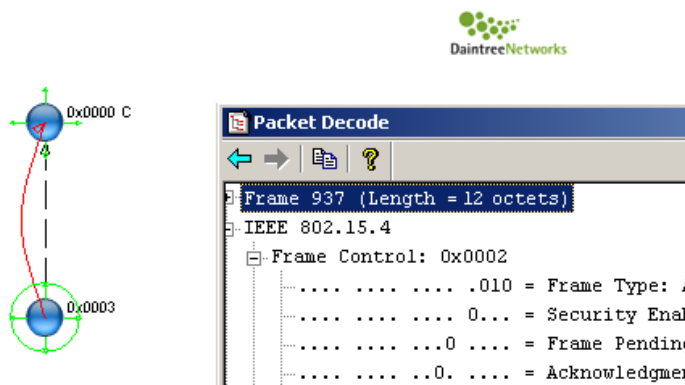
6.3 Energetický model při zpracování zprávy

Výpočet spotřeby energie při komunikaci mezi zařízeními závisí také na tom, po jaký čas byly zařízení v určitém režimu komunikace. Jak bylo popsáno v kapitole 5.1 rádiový čip AT86RF230 může být v režimu vysílání, přijímání, nečinnosti nebo spánku. Při režimu vysílání je odběr proudu tohoto čipu 16,5 mA při maximálním vysílacím výkonu 3 dBm, v režimu příjmu 15,5 mA, v nečinném režimu 8 mA a v režimu spánku 8 μ A. Spotřeba energie je tedy přímo úměrná času po který bylo zařízení v daném režimu. Postup komunikace probíhá následovně. Jakmile chce uzel vysílat zprávy, přepne se do aktivního režimu. Následně čeká náhodný čas než dojde k připojení s bezdrátovým zařízením. Tento náhodný čas představuje číslo z intervalu $\langle 0; 2^{BE} - 1 \rangle$, kde hodnota BE se zvýší s každým neúspěšným pokusem o připojení. Poté se provádí CCA (Clear Channel Assessment), v překladu odhad volného kanálu [15]. Funkce CCA vyhodnotí zda je kanál volný nebo obsazený, a tedy zda je možnost vysílání. Díky této funkci se dokáže automaticky předcházet konfliktům.

Doba přístupu k bezdrátovému zařízení se vypočítá podle vzorce 6.4. Zkratka PDO představuje *počáteční dobu odstupu* a JDO *jednotku doby odstupu*.

$$PDO + CCA = (2^{BE=3} - 1) \cdot JDO + CCA = 7 \cdot 320\mu s + 128\mu s = 2,368ms \quad (6.4)$$

Jakmile dojde k úspěšnému připojení s médiem, odesílá se nejdříve zpráva *Data Request*, která obsahuje identifikátor souseda, který o komunikaci požádal. Po dobu kdy je zařízení v režimu vysílání nebo přijímání závisí na velikosti posílané zprávy a rychlosti datového toku rádiového čipu [15]. Rádiový čip AT86RF230 umožňuje maximální bitovou rychlost $b = 250$ kb/s. Jak je zobrazeno na obrázku 6.1, velikost zprávy *Data Request* je $k = 12$ bajtů.



Obr. 6.1: Údaje o velikosti zprávy *Data Request* [15]

Doba přenosu zprávy *Data Request* o velikosti k je tedy rovna $t_k = (k \cdot 8)/(b \cdot 1000) = 0,38$ ms. Energie zpracovaná během přenosu E_{Tx} se rovná: $U[V] \cdot I[A] \cdot t[s] = 2,4V \cdot 16,5mA \cdot 0,38ms = 15\mu J$. Parametr U představuje napětí a I proud, který je odebírán při vysílacím režimu.

Po obdržení zprávy *Data Request* přijímačem, přechází zařízení z režimu příjmu na režim vysílání a zasílá potvrzovací zprávu ACK. Poté co zařízení na druhé straně tuto zprávu obdrželo dojde k vysílání požadovaných dat. Velikost přenášených dat je rovna velikosti datového pole (*Payload*) a záhlaví, které je k datovému poli přidáno. Ke zprávě přidá vrstva aplikační a síťová každá 8 bajtů záhlaví a vrstva linková společně s fyzickou rozšíří zprávu o další 9 a 2 bajty. Celková velikost přenášené zprávy s velikostí samotného datového pole 10 bajtů bude mít velikost 37 bajtů. [15]. Po dosazení do předchozího rovnice se spotřeba energie při přenosu zprávy 37 bajtů rovná: $U[V] \cdot I[A] \cdot t[s] = 2,4V \cdot 16,5mA \cdot 1,184ms = 46,9\mu J$.

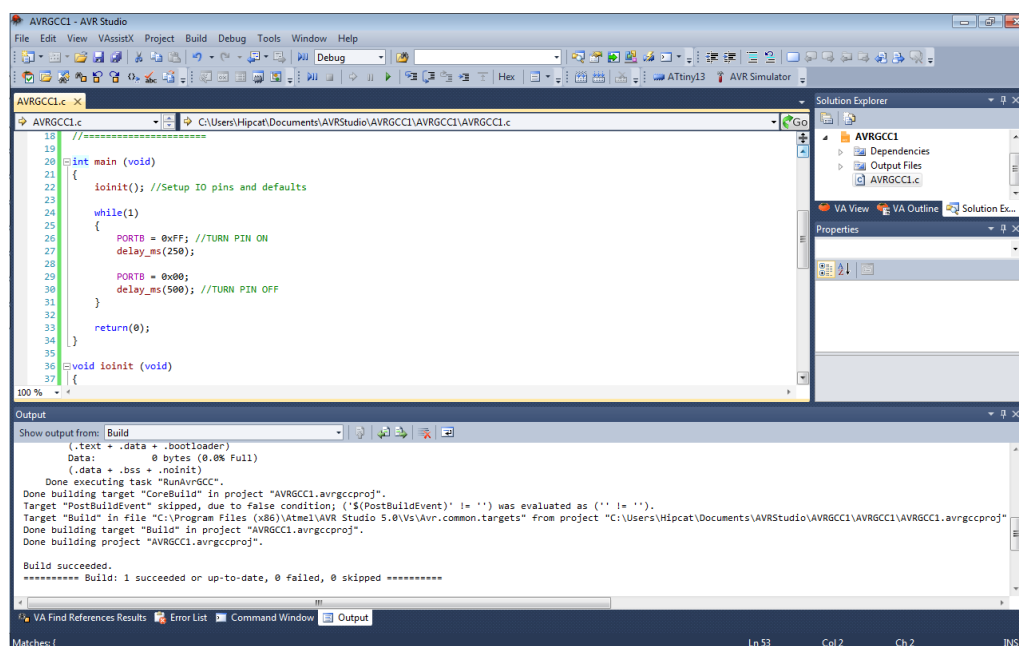
7 PRAKTICKÉ ŘEŠENÍ PRÁCE

Tato kapitola popisuje komunikaci mezi dvěma bezdrátovými komunikačními jednotkami IRIS. Jeden z uzlů bude typem zařízení *koordinátor* a druhý představuje zařízení typu *Router*. Vytvořená komunikace probíhá pomocí standardu ZigBee. Pro programování mikrokontroléru ATmega1281 je použit programovací jazyk C.

7.1 Použité zařízení a software

Pro realizaci praktického řešení budou použity následující zařízení a software:

AVR Studio 5.1 - jedná se o vývojové prostředí pro programování mikrokontrolérů Atmel. Umožňuje psaní a ladění programu v jazycích C a ASSEMBLER. Program napsaný v jazyce C se při sestavování nejprve přeloží do assembleru a poté se kompiluje do strojového byte kódu [16].



Obr. 7.1: Vývojové prostředí AVR Studia 5.1

Programátor a debugger ATMEL AVR JTAGICE mkII - jde o výkonný nástroj na programování 8-bitových procesorů (v tomto případě ATmega1281). Může být k počítači připojen buď přes rozhraní RS232 nebo přes USB port. V našem případě bylo zvoleno připojení přes USB port. Tato možnost připojení také umožňuje přímo napájení aplikace z portu USB. K následnému propojení k aplikaci je použit 10-pinový JTAG konektor [17].



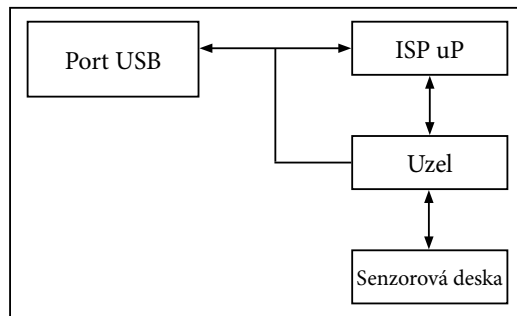
Obr. 7.2: Programátor a debugger ATMEL AVR JTAGICE mkII

MIB520 - základnová stanice pro bezdrátové senzorové sítě sloužící pro programování uzlů IRIS, MICA a MICA2 přes USB port. Kromě přenosu dat MIB520CB poskytuje také programové rozhraní USB a nabízí dva samostatné uzly: jeden slouží pro programování uzlů a druhý pro datovou komunikaci přes USB port. MIB520CB má také procesor, který je možno programovat pomocí programu *Mote Processor Radio Boards*. USB port lze použít také jako zdroj napájení [18].



Obr. 7.3: Programovací jednotka MIB520 (převzato z [18])

Rychlost USB rozhraní je 57,6 baudů. Rozhraní Uzel obsahuje 51-pinový konektor a indikátor pomocí třech LED diod pro indikaci stavů. JTAG rozhraní má konektor s 10-ti piny a napájení přes USB [18]. Blokové schéma základnové stanice MIB520 je zobrazeno na obrázku 7.4.



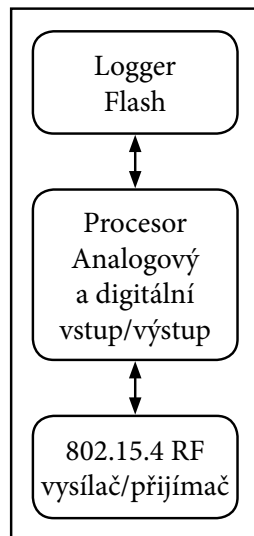
Obr. 7.4: Blokové schéma MIB520CB [18]

Bezdrátová komunikační jednotka IRIS - je nízko-výkonový modul pro bezdrátové senzorové sítě pracující v pásmu 2,4 GHz. Používá standard IEEE 802.15.4 a dosahuje přenosové rychlosti 250 kb/s. Může být použit pro monitorování budov a zabezpečení. Obsahuje konektor pro rozšíření umožňující měřit teplotu, svítivost, barometrický tlak a jiné. Je napájen pomocí dvou AA baterií [19].



Obr. 7.5: Bezdrátová komunikační jednotka IRIS (převzato z [19])

Komunikační uzel XM2110CA obsahuje nízko-výkonový procesor Atmel ATmega1281. Ten spouští aplikace ze své vnitřní paměti. Jednoprocesorová deska může zpracovávat aplikace a současně bezdrátovou komunikaci. Uzel IRIS má 51-pinový konektor podporující analogové vstupy, digitální vstup/výstup, I2C, SPI a UART rozhraní. Tyto rozhraní umožňují připojení k široké škále externích periférií [19]. Obrázek 7.6 znázorňuje blokové schéma XM2110CA.



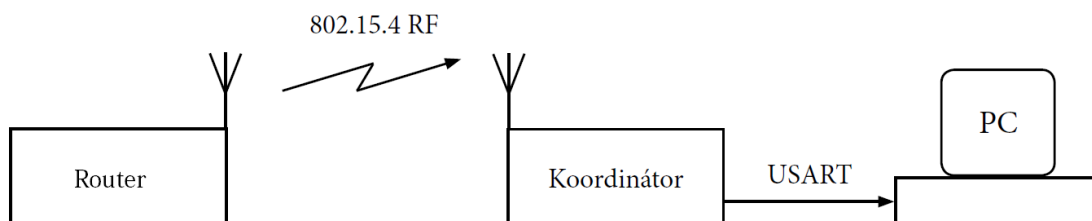
Obr. 7.6: Blokové schéma XM2110CA [19]

7.2 Realizace

Cílem je vytvoření firmwaru pro mikrokontrolér ATmega1281, který umožňuje bezdrátovou komunikaci mezi dvěma bezdrátovými zařízeními podle standardu IEEE 802.15.4 a následné zasílání dat do počítače a jejich zpracování. Dalším cílem práce je programování vysílacího uzlu tak, aby bylo možno u něj měnit parametry, jako je velikost paketu, interval odesílání, vysílací výkon a jiné. Tyto parametry mají vliv na celkové spotřebě energie zařízení.

7.2.1 Blokové schéma

Na obrázku 7.7 je zobrazeno blokové schéma bezdrátového přenosu mezi dvěma uzly a následné zasílání přijatých dat do počítače. Na začátku přenosu je bezdrátová komunikační jednotka IRIS, která je nastavená jako typ komunikačního zařízení *Router*. Router odesílá data pomocí standardu IEEE 802.15.4 do uzlu typu *Koordinátor*. Zde jsou data přijata a vyslána po sériovém rozhraní USART do počítače, který přijatá data následně zpracuje.



Obr. 7.7: Blokové schéma

7.2.2 Programování modulu pomocí BitCloud

Sestavení sítě a spojení

V BitCloudu je typ zařízení definován parametrem `CS_DEVICE_TYPE` typu `DeviceType_t` a je možné jej nastavit na jednu následujících hodnot:

Koordinátor: `DEVICE_TYPE_COORDINATOR` (nebo `0x00`),

Směrovač: `DEVICE_TYPE_ROUTER` (nebo `0x01`),

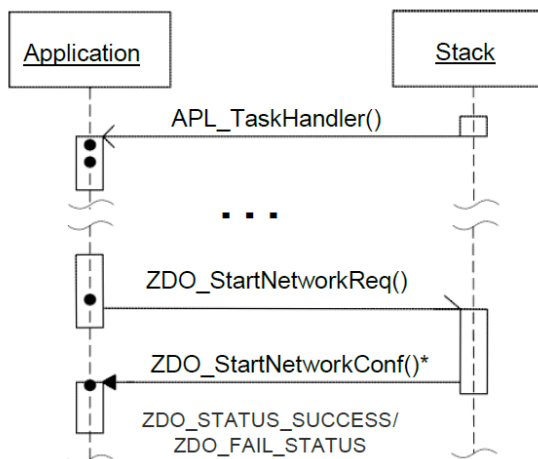
Koncové zařízení: `DEVICE_TYPE_END_DEVICE` (nebo `0x02`).

V souladu se standardem ZigBee musí být každé zařízení identifikováno pomocí rozšířené adresy, označované také MAC nebo IEEE adresa. V BitCloudu se tato adresa určuje podle parametru `CS_UID`. Uživatel je zodpovědný za přidělování a každé zařízení musí mít přidělenou unikátní adresu. Pro připojení k ZigBee síti je také potřeba nastavení tzv. krátké adresy neboli označované jako síťová adresa. Nastavení síťové adresy se provádí pomocí parametru `CS_NWK_ADDR`. V tomto případě musí být dále nastavena v parametru `CS_NWK_UNIQUE_ADDR` hodnota na `true`, aby byla adresa přidělena. Pokud by byla zvolena hodnota `false`, došlo by k dynamickému adresování a adresa by byla nastavena automaticky [12].

Před zahájením sítě je uzel zodpovědný za nastavení parametrů charakterizující buď síť kterou si přeje vytvořit (pro koordinátora), nebo síť do které se chce připojit (pro směrovač a koncové zařízení). Těmito parametry jsou: podporované modulační schéma (`CS_CHANNEL_PAGE`), podporovaný kanál frekvence zadávaný přes 32-bit kanál masky (`CS_CHANNEL_MASK`) a 64-bit rozšířená PAN ID (`CS_EXT_PANID`) [12].

Jakmile jsou předchozí síťové parametry správně nastaveny je vyslána žádost pro start sítě prováděná asynchronním voláním `ZDO_StartNetworkReq()`. Po dokončení startu/připojení do sítě ZDO informuje o uplatnění výsledku zpětným voláním s argumentem typu `ZDO_StartNetworkConf_t`. Tato struktura obsahuje stav změn

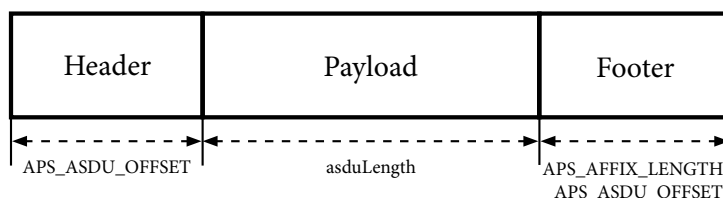
a informaci o síti (např. síťovou adresu pro uzel). Stav `ZDO_SUCCESS_STATUS` je přijat v případě, že postup byl proveden správně. Jiný status by znamenal, že start sítě selhal [12]. Veškeré funkce a datové typy obsahuje knihovna `zdo.h`.



Obr. 7.8: Diagram počáteční síťové sekvence (převzato z [12])

Přenos dat

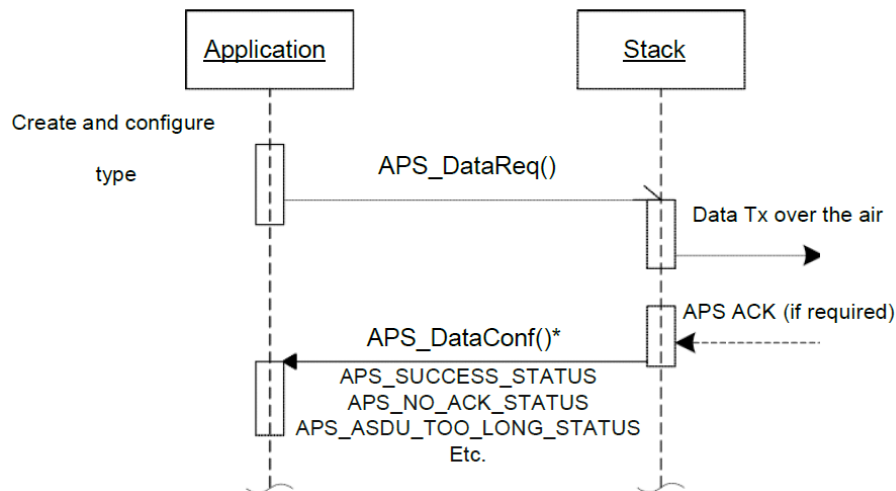
Další důležitou funkcí je přenos dat mezi uzly. Aby mohla být data přenesena je potřeba nastavení parametru uzlu. Tyto parametry se nastavují ve funkci žádosti o přenos dat `APS_DataReq()`. Zde je nastavena cílová adresa, koncový bod ID, aplikační profil ID, klastr ID, definice bufferu ASDU (obsahující ASDU a `asduLength`) a jiné. Funkce `APS_DataInd` určuje umístění zpětného volání funkce pro přijetí dat a funkce `APS_DataConf` umístění zpětného volání pro odeslání dat.



Obr. 7.9: Formát ASDU pole [12]

Maximální povolená velikost přenášených dat je pro nezabezpečený přenos 95 bajtů a pro zabezpečený přenos 77 bajtů [12].

Jakmile jsou data odeslána k cílovému uzlu, najde se nejspolehlivější cesta k uzlu. Tyto cesty jsou průběžně aktualizovány. Lze nastavit maximální počet skoků pro přenos dat. To se nastavuje pomocí položky *radius* v poli žádosti o přenos dat `APS_DataReq_t` [12]. Protože v tomto případě se jedná o komunikaci mezi dvěma uzly je hodnota *radius* nastavena na 0. Všechny funkce a datové typy jsou obsaženy v knihovně `aps.h`.



Obr. 7.10: Přenos dat v síti (převzato z [12])

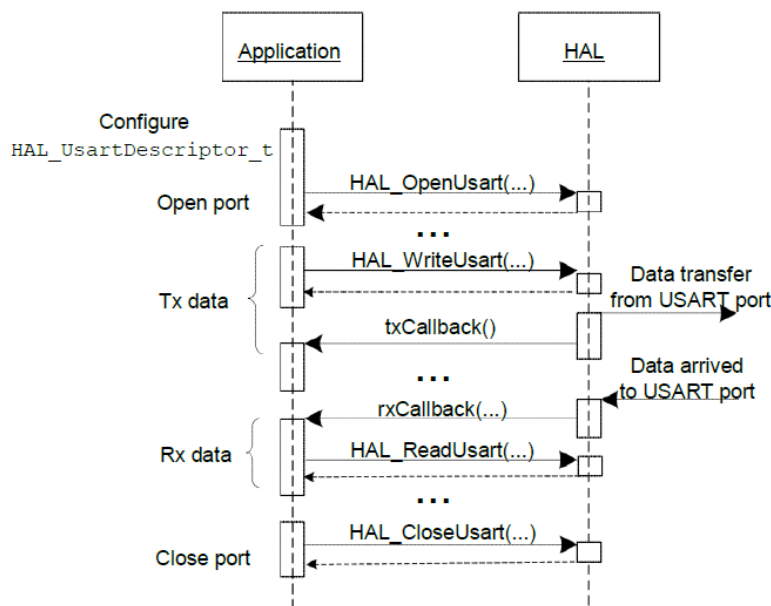
Přenos dat po sériovém rozhraní USART

Aby mohla být data přenášena pomocí sériového rozhraní USART je potřeba nastavit jeho parametry v globální proměnné typu `HAL_UsartDescriptor_t`. Zde se nastavují všechny běžné parametry jako je synchronní/asynchronní režim, přenosová rychlost, řízení toku, režim parity, atd. Pro použití USART kanálu je nutné použít funkci `HAL_OpenUsart()` s argumentem globální proměnné typu `HAL_UsartDescriptor_t`. Následně je vrácena hodnota, která označuje zda je port úspěšně otevřen a mohou být přenesena data. Pro uzavření kanálu je použita funkce `HAL_CloseUsart()` [12].

Pro praktické řešení komunikace po sériovém rozhraní USART bude nastaven **režim Callback**.

Pro přenos dat je volána funkce `HAL_WriteUsart()` s ukazatelem na argumenty pro délku dat a vyrovnávací paměti. Pokud vrácená hodnota je větší než 0, bude provedena funkce definována jako *txCallback*. USART je schopen přijímat data pokud v polích *rxBuffer* a *rxBufferLength* není nastaveno NULL (hodnota 0 je respektována). Pokud data dorazila na *rxBuffer* je zavolán zpětný režim *rxCallback*. Aby byla přijatá data z *rxBufferu* přenesena do vyrovnávací paměti, je použita funkce

HAL_ReadUsart(), která následně vrátí data a délku dat [12]. Všechny funkce a datové typy pro přenos po sériovém rozhraní obsahuje knihovna usart.h.



Obr. 7.11: Přenos dat pomocí USART - callback režim (převzato z [12])

7.3 Řešení ve vývojovém prostředí

Praktické řešení vytvořené komunikace mezi dvěma koncovými uzly představuje projekt s názvem *Blank*. Tento projekt obsahuje vytvořený firmware pro koordinátora, koncové zařízení a firmware pro rozhraní USART.

7.3.1 Koordinátor

Aby koordinátor pracoval správně, musí být vytvořena všechna potřebná nastavení. Za nejdůležitější část lze považovat nastavení typu zařízení. Typ zařízení je dán parametrem `CS_DEVICE_TYPE` typu `DeviceType_t`. Dále je potřeba nastavení síťové adresy. To je provedeno pomocí parametru `CS_NWK_ADDR`. Pro zařízení koordinátor je zvolena adresa `0x0000`. Pro zvolení statického přidělení adres je parametr `CS_NWK_UNIQUE_ADDR` nastaven na hodnotu `true`. Rozšířená adresa neboli MAC adresa určena podle parametru `CS_UID`, je pro koordinátora nastavena na `0x0099`. Dále dochází k nastavení dalších parametrů jako je modulační schéma, 64-bitová rozšířená PAN ID a také kanál na kterém bude komunikace probíhat. Pro tento účel slouží parametr `CS_CHANNEL_MASK` kde je nastaven 13. kanál. Všechna tato nastavení jsou obsažena v souboru *configuration.h*.

```

/* Editace nastaveni uzlu */
#define CS_NWK_ADDR 0x0000 /* sitova adresa*/
#define CS_UID      0x0099 /* MAC adresa*/
#define CS_CHANNEL_MASK 0x2000 /* nastaveni 13.kanal*/
#define CS_NWK_UNIQUE_ADDR true

/* zvoleny typ zarizeni */
#define CS_DEVICE_TYPE_COORDINATOR

```

7.3.2 Router

Obdobným způsobem jakým je nastaveno zařízení typu koordinátor se postupuje také v případě směrovače, neboli routeru. Rozdíly jsou pouze v nastavení typu zařízení, síťové adresy a MAC adresy. Síťová adresa je pro router nastavena na 0x0003 a MAC adresa 0x0039. Ostatní parametry jsou nastaveny stejně.

```

/* Editace nastaveni uzlu */
#define CS_NWK_ADDR 0x0003 /* sitova adresa*/
#define CS_UID      0x0039 /* MAC adresa*/
#define CS_CHANNEL_MASK 0x2000 /* nastaveni 13.kanal*/
#define CS_NWK_UNIQUE_ADDR true

/* zvoleny typ zarizeni */
#define CS_DEVICE_TYPE_END_DEVICE

```

7.3.3 Nastavení parametrů

Následně je potřeba nastavení parametrů vysílání pro bezdrátové zařízení. Ve stejném souboru *configuration.h* kde dochází k nastavení uzlu a typu zařízení je možnost nastavení vysílacího výkonu daného zařízení. Radiový čip AT86RF230 umožňuje nastavení 16-ti různých úrovní vysílacího výkonu v rozmezí od +3dBm po -17,2dBm. Jednotlivé úrovně výkonu jsou popsány v dřívější kapitole. Tento parametr je nastaven pomocí příkazu `CS_RF_TX_POWER` a zvolený výkon je +3dBm.

```

#define CS_RF_TX_POWER 3 /* nastaveni vykonu +3dBm */

```

Dalšími parametry ovlivňující spotřebu energie je velikost zasílaného paketu a interval zasílání. Parametry nastavení velikosti zasílaného paketu lze editovat v souboru *blank.h*. Zde dochází nejen k nastavení velikosti zprávy, ale také vytvoření struktury ASDU pole a definice bufferu. Velikost datové části je nastavována ve vytvořeném poli struktury s názvem *payload*. Hodnota v závorce určuje délku pole v bajtech. Text zasílané zprávy lze následně editovat v souboru *blank.c*.

Ukázka části zdrojového kódu v souboru *blank.h*

```
typedef struct
{
    char payload[50]; /* Velikost datove casti */
}AsduMessage_t;

//Application message buffer structure
BEGIN_PACK
typedef struct
{
    uint8_t header[APS_ASDU_OFFSET]; //APS header
    uint8_t data[MAX_ASDU_SIZE]; //Application data - the payload
    uint8_t footer[APS_AFFIX_LENGTH - APS_ASDU_OFFSET]; // Footer
} PACK AppMessageBuffer_t;
END_PACK
```

Nastavení zasílané zprávy v souboru *blank.c*

```
char msg[]="Text vysilane zpravy";

memcpy(AsduMessage.payload, msg, strlen(msg));
```

Interval s jakým budou jednotlivé zprávy vysílány určuje časovač vytvořený v souboru *blank.c*. Tento časovač s názvem *mainTimer* je nastaven na hodnotu 1000, kdy tato hodnota určuje interval v jednotkách ms. Také je nastaven v opakujícím se režimu.

Ukázka části zdrojového kódu v souboru *blank.c*

```
/* deklarace callbacku pro hlavni casovac */
void mainTimerFired(void);
```

```

/* definice hlavního časovace mainTimer */
static HAL_AppTimer_t mainTimer = {
    .interval = 1000,
    .mode = TIMER_REPEAT_MODE,
    .callback = mainTimerFired,
};

```

7.3.4 Start sítě

Dále následují instrukce pro registraci požadavku na start sítě. Nejdříve je provedena registrace pointeru na funkci, která se zavolá až bude požadavek vykonán. Poté je vrstvě ZDO předán požadavek na start sítě. Tento firmware se nachází v souboru *blank.c*.

```

static ZDO_StartNetworkReq_t startNetwork = {
    .ZDO_StartNetworkConf = ZDO_StartNetworkConf
};

```

7.3.5 Universal Synchronous Receiver and Transmitter

V souboru *boardabstraction.c* je implementováno rozhraní USART pro výpis přijatých hodnot z koordinátoru. Nastavení parametrů pro přenos se provádí v *usartDescriptor*, který musí být nastaven správně. *UsartDescriptor* představuje strukturu nastavení. Tento strukturovaný datový typ je definován v BitCloudu jako *HAL_UsartDescriptor_t*.

```

static HAL_UsartDescriptor_t usartDescriptor = {
    .tty                = APP_USART_CHANNEL,
    .mode               = USART_MODE_ASYNC,
    .baudrate           = APP_USART_BAUDRATE,
    .dataLength         = USART_DATA8,
    .parity              = USART_PARITY_NONE,
    .stopbits           = USART_STOPBIT_1,
    .rxBuffer           = NULL,
    .rxBufferLength     = 0,
    .txBuffer           = usartTxBuffer,
    .txBufferLength     = USART_TX_BUFFER_SIZE,
    .rxCallback         = NULL,
    .txCallback         = NULL,
    .flowControl        = USART_FLOW_CONTROL_NONE,
};

```

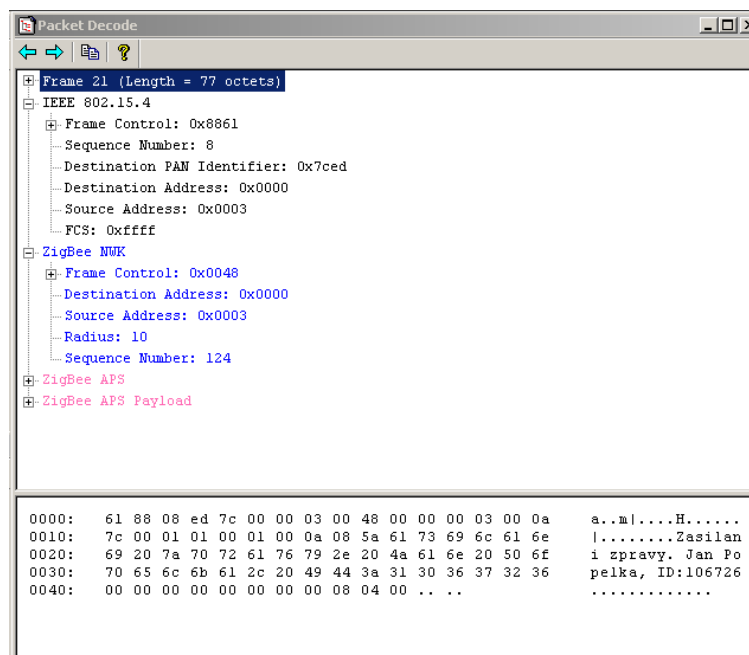
7.4 Vytvoření sítě a přenos dat

Vytváření nové sítě provádí pouze zařízení které je typu koordinátor a není v daný okamžik již součástí jiné sítě. Před vytvořením nové sítě proměří koordinátor energii v kanálu na kterém je síť zakládána. Poté koordinátor vyšle dotaz na počet sítí na daném kanále. Jako dotaz na vysílán *Beacon Request* rámeček. Síť je pak přiděleno unikátní PAN identifikátor sítě a síť je vytvořena. Tento identifikátor nesmí být shodný se žádnou jinou sítí. Poté se mohou zařízení připojit. Existují dva způsoby pro zjišťování dostupných zařízení, požadavek MAC adresy a požadavek síťové adresy. Pokud je síťová adresa známa, odesílá se požadavek MAC adresy přímo k danému zařízení. Požadavek síťové adresy je vyslán všem zařízením a MAC adresa obsažena v užitečných datech daného paketu. Směrovač před svým připojením do již existující sítě vysílá na zvoleném kanálu *Beacon Request* a tím aktivně hledá dostupného koordinátora. Pokud dojde k vytvoření sítě na zvoleném kanálu na kterém jsou zařízení připojena, obdrží směrovač *Beacon* rámeček. Poté již můžou zařízení spolu komunikovat. Na obrázku 7.12 je zobrazena komunikace mezi směrovačem a koordinátorem. Pro monitorování provozu byl použit software Sensor Network Analyzer.

| Seq No / | Channel | Time | Time Delta | MAC Src | MAC Dest | NWK Src | NWK Dest | Protocol | Payload | Packet Type |
|----------|---------|--------------|---------------|---------|-----------|---------|----------|---------------|-----------------|---|
| 1 | 13 | 16:37:48.863 | | 0x0000 | Broadca.. | 0x0000 | 0xffff | ZigBee NWK | | NWK Command: Link Status |
| 2 | 13 | 16:38:01.512 | +00:00:12.649 | | Broadca.. | | | IEEE 802.15.4 | | Command: Beacon Request |
| 3 | 13 | 16:38:01.515 | +00:00:00.003 | 0x0000 | | | | ZigBee NWK | 5a:61:73:69:6.. | Beacon: BD: 15, SD: 15, PC: 1, AP: 1, NWK PC: |
| 4 | 13 | 16:38:02.022 | +00:00:00.508 | 0x0003 | | 0x0003 | 0x0000 | ZigBee NWK | | NWK Command: Rejoin Request |
| 5 | 13 | 16:38:02.024 | +00:00:00.001 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 6 | 13 | 16:38:02.026 | +00:00:00.002 | 0x0000 | 0x0003 | 0x0000 | 0x0003 | ZigBee NWK | | NWK Command: Rejoin Response |
| 7 | 13 | 16:38:02.027 | +00:00:00.002 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 8 | 13 | 16:38:02.110 | +00:00:00.082 | 0x0003 | Broadca.. | 0x0003 | 0xffffd | ZigBee ZDP | | ZDP: EndDeviceAnnce |
| 9 | 13 | 16:38:02.151 | +00:00:00.042 | 0x0000 | Broadca.. | 0x0003 | 0xffffd | ZigBee ZDP | | ZDP: EndDeviceAnnce |
| 10 | 13 | 16:38:02.188 | +00:00:00.037 | 0x0003 | Broadca.. | 0x0003 | 0xffffd | ZigBee ZDP | | ZDP: EndDeviceAnnce |
| 11 | 13 | 16:38:02.193 | +00:00:00.004 | 0x0000 | Broadca.. | 0x0003 | 0xffffd | ZigBee ZDP | | ZDP: EndDeviceAnnce |
| 12 | 13 | 16:38:02.229 | +00:00:00.037 | 0x0003 | Broadca.. | 0x0003 | 0xffffd | ZigBee ZDP | | ZDP: EndDeviceAnnce |
| 13 | 13 | 16:38:02.233 | +00:00:00.004 | 0x0000 | Broadca.. | 0x0003 | 0xffffd | ZigBee ZDP | | ZDP: EndDeviceAnnce |
| 14 | 13 | 16:38:03.228 | +00:00:00.995 | 0x0003 | 0x0000 | 0x0003 | 0x0000 | ZigBee APS | 5a:61:73:69:6.. | 0x0001: 0x0001 |
| 15 | 13 | 16:38:03.231 | +00:00:00.003 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 16 | 13 | 16:38:03.924 | +00:00:00.693 | 0x0000 | Broadca.. | 0x0000 | 0xffffc | ZigBee NWK | | NWK Command: Link Status |
| 17 | 13 | 16:38:04.227 | +00:00:00.303 | 0x0003 | 0x0000 | 0x0003 | 0x0000 | ZigBee APS | 5a:61:73:69:6.. | 0x0001: 0x0001 |
| 18 | 13 | 16:38:04.230 | +00:00:00.003 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 19 | 13 | 16:38:05.224 | +00:00:00.994 | 0x0003 | 0x0000 | 0x0003 | 0x0000 | ZigBee APS | 5a:61:73:69:6.. | 0x0001: 0x0001 |
| 20 | 13 | 16:38:05.227 | +00:00:00.003 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 21 | 13 | 16:38:06.220 | +00:00:00.999 | 0x0003 | 0x0000 | 0x0003 | 0x0000 | ZigBee APS | 5a:61:73:69:6.. | 0x0001: 0x0001 |
| 22 | 13 | 16:38:06.223 | +00:00:00.003 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 23 | 13 | 16:38:07.219 | +00:00:00.996 | 0x0003 | 0x0000 | 0x0003 | 0x0000 | ZigBee APS | 5a:61:73:69:6.. | 0x0001: 0x0001 |
| 24 | 13 | 16:38:07.222 | +00:00:00.003 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 25 | 13 | 16:38:08.217 | +00:00:00.995 | 0x0003 | 0x0000 | 0x0003 | 0x0000 | ZigBee APS | 5a:61:73:69:6.. | 0x0001: 0x0001 |
| 26 | 13 | 16:38:08.219 | +00:00:00.003 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 27 | 13 | 16:38:09.215 | +00:00:00.996 | 0x0003 | 0x0000 | 0x0003 | 0x0000 | ZigBee APS | 5a:61:73:69:6.. | 0x0001: 0x0001 |
| 28 | 13 | 16:38:09.218 | +00:00:00.003 | | | | | IEEE 802.15.4 | | Acknowledgment |
| 29 | 13 | 16:38:10.213 | +00:00:00.995 | 0x0003 | 0x0000 | 0x0003 | 0x0000 | ZigBee APS | 5a:61:73:69:6.. | 0x0001: 0x0001 |
| 30 | 13 | 16:38:10.216 | +00:00:00.003 | | | | | IEEE 802.15.4 | | Acknowledgment |

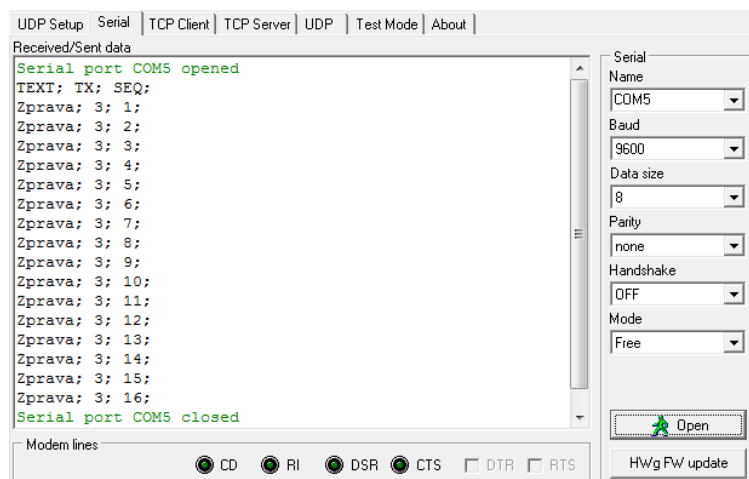
Obr. 7.12: Ukázka komunikace v ZigBee

Na následujícím obrázku 7.13 je detailnější popis přenášeného paketu včetně přenášené zprávy. V tomto případě byla velikost datové části *Payload* nastavena na 50 bajtů.



Obr. 7.13: Ukázka paketu

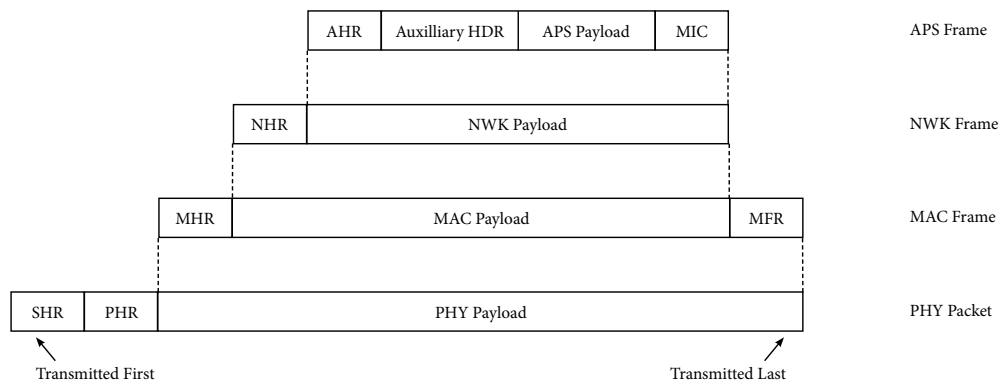
Jakmile jsou data přijata koordinátorem, posílají se po sběrnici USART do USB portu počítače. V praktickém řešení diplomové práce zasílá router koordinátorovi data obsahující přenášenou zprávu, nastavený vysílací výkon na zařízení a sekvenční číslo paketu. K následnému zobrazení obdržených informací a případné zpracování byl použit software Herkules. Ukázka přijatých dat přes sběrnici USART je zobrazena na obrázku 7.14.



Obr. 7.14: Příjem dat po sériovém portu

8 VÝPOČTY DOBY PŘENOSU RÁMCE

Následující kapitola je zaměřena na výpočty dobu přenosu rámce při různých velikostech samotné datové části. Při řešení praktické části kdy je úkolem realizovat komunikaci mezi dvěma uzly, je použit softwarový stack Bitcloud. Při takto vytvořené komunikaci mezi dvěma zařízeními jsou data posílána z aplikační vrstvy. Následující obrázek 8.1 zobrazuje strukturu aplikačního rámce.



Obr. 8.1: Struktura zapouzdření vrstev (převzato z [24])

Celková velikost přenášených dat je rovna součtu velikosti datového pole aplikační vrstvy *APS Payload* a záhlaví, které je k datovému poli přidáno. Na každé z vrstev aplikační a síťové je přidáno záhlaví 8 bajtů, vrstva linková rozšíří zprávu o 9 bajtů a záhlaví a vrstva fyzická pak o 2 bajty. V případě kdy velikost samotného datového pole je 10 bajtů bude celková velikost přenášeného rámce 37 bajtů. Velikost doby přenosu rámce se vypočítá podle vzorce 8.1:

$$\frac{(27 + d) \times 8}{b \times 1000}, \quad (8.1)$$

kde d je velikost přenášené zprávy v bajtech a hodnota b je bitová rychlost. Při výpočtu je potřeba si uvědomit převod z bajtů na bity, proto se celý horní řádek násobí hodnotou 8.

V dalším kroku jsou vypočítány doby přenosu pro konkrétní zvolené velikosti datové části **1B**, **10B**, **50B** a **100B**. Dále bude uvažován přenos dat v ideálním prostředí kdy bitová rychlost dosahuje 250 kb/s.

Doba přenosu rámce s datovou částí o velikosti **1B** je tedy:

$$\frac{(27 + 1) \times 8}{250 \times 1000} = \frac{28 \times 8}{250 \times 1000} = 0,896ms \quad (8.2)$$

Při přenosu zprávy s velikostí **10B** je doba přenosu:

$$\frac{(27 + 10) \times 8}{250 \times 1000} = \frac{37 \times 8}{250 \times 1000} = 1,184ms \quad (8.3)$$

Pro velikost rámce s datovou částí o velikosti **50B** je doba přenosu:

$$\frac{(27 + 50) \times 8}{250 \times 1000} = \frac{77 \times 8}{250 \times 1000} = 2,464ms \quad (8.4)$$

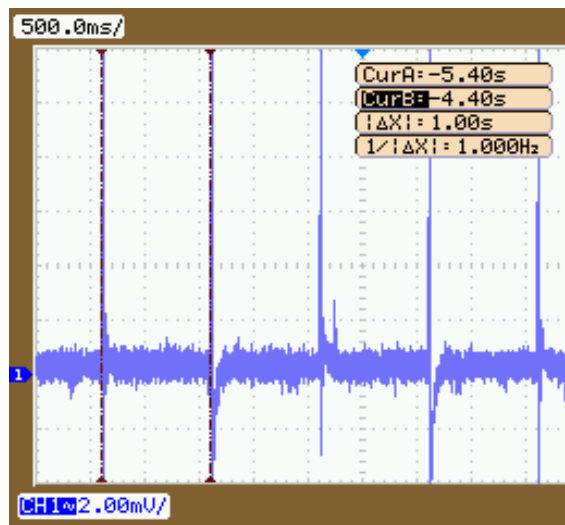
A doba přenosu rámce s datovou částí o velikosti **100B** je:

$$\frac{(27 + 100) \times 8}{250 \times 1000} = \frac{127 \times 8}{250 \times 1000} = 4,064ms \quad (8.5)$$

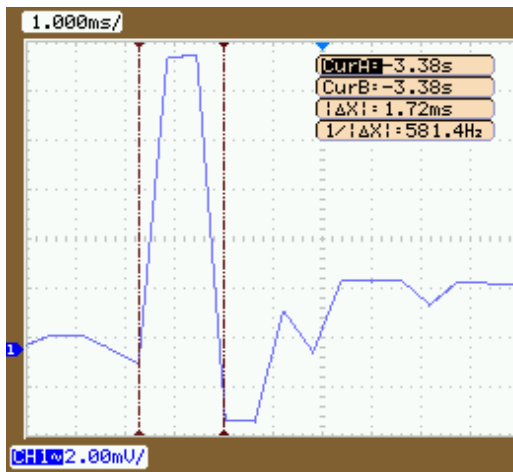
8.1 Naměřené doby přenosu

V dalším postupu diplomové práce následuje porovnání teoretických hodnot výpočtu doby přenosu rámce skutečnými hodnotami. Výstupem tohoto měření jsou grafy zobrazující průběh závislosti napětí na bočníku R_B pro odlišné parametry přenosu rámce. Velikost napájecího napětí uzlu U_{CC} byla v rozmezí 2,4V - 2,5V a odpor bočníku $R_B = 1 \Omega$. Hodnota napětí není přesně stanovena z důvodu, že vysílací uzel byl při každém měření napájen stejnými akumulátory, které se odebíraným proudem zařízení postupně vybíjely. Přenos dat mezi oběma zařízeními byl realizován ve vzdálenosti 3,5 m. Interval zasílání paketů, který byl při všech měření nastaven na 1 s je zobrazen na obrázku 8.2. Délka intervalu skutečně odpovídá nastavené hodnotě na vysílacím uzlu.

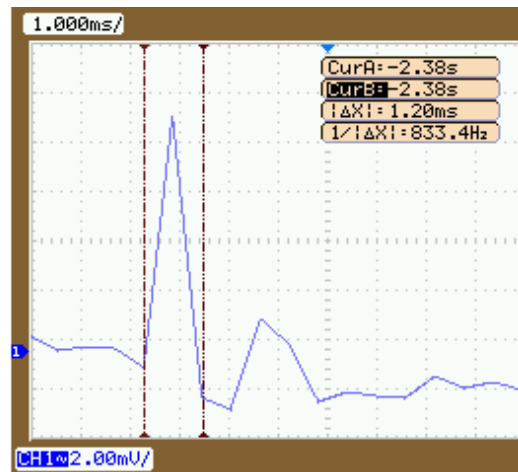
V průběhu dalších měření byly postupně měněny parametry velikosti zasílaných dat. Následující obrázek zobrazuje skutečnou dobu odesílání zprávy o velikosti **1B**, při vysílacích výkonech +3dBm a +1,6dBm. Při této nastavené velikosti paketu docházelo k rozdílům mezi skutečnými a vypočtenými hodnotami. Největší rozdíl vznikl při vysílacím výkonu +3dBm, při kterém byla očekávána odchylka minimální. Při výkonu 0dBm se skutečná hodnota již více přibližuje hodnotě naměřené. Obě doby přenosu jsou zobrazeny na obrázku 8.3.



Obr. 8.2: Průběh zasílání paketů s intervalem 1s



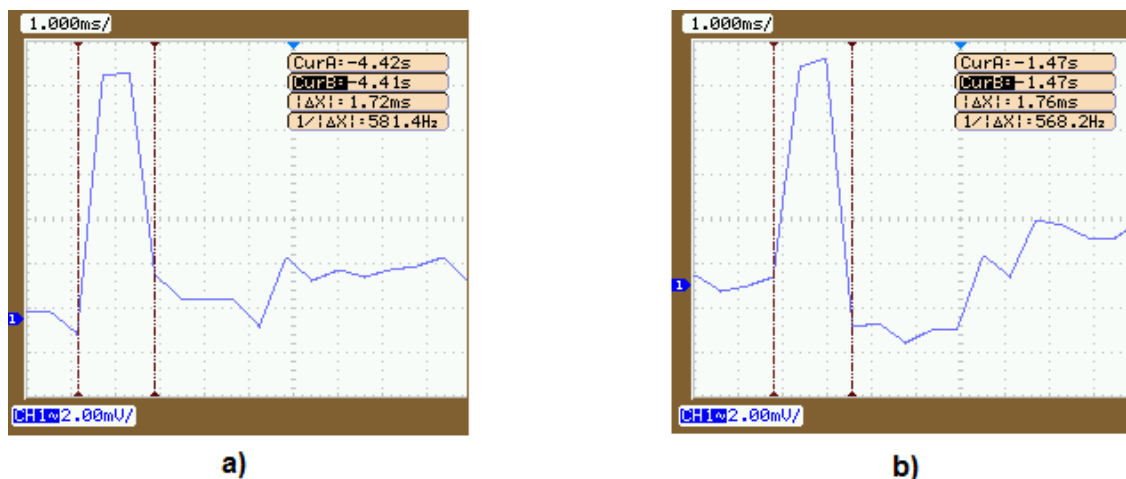
a)



b)

Obr. 8.3: Doba přenosu zprávy 1B při vysílacím výkonu: a) +3dBm b) 0dBm

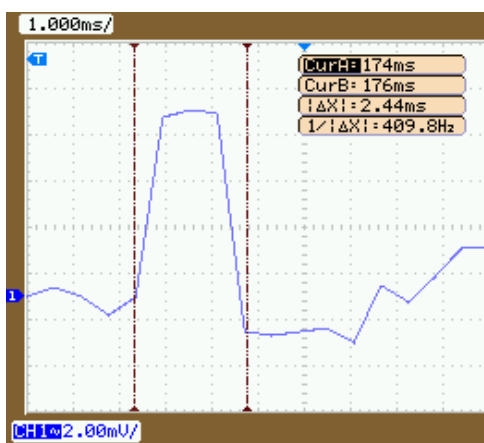
Při dalším měření byla velikost datového pole rámce změněna na **10B**. Vysílací výkony zůstaly i při tomto měření na hodnotách +3dBm a 0dBm. Také v tomto měření docházelo k rozdílu mezi vypočtenou a reálnou dobou přenosu. Naměřené doby přenosu zobrazuje obrázek 8.4.



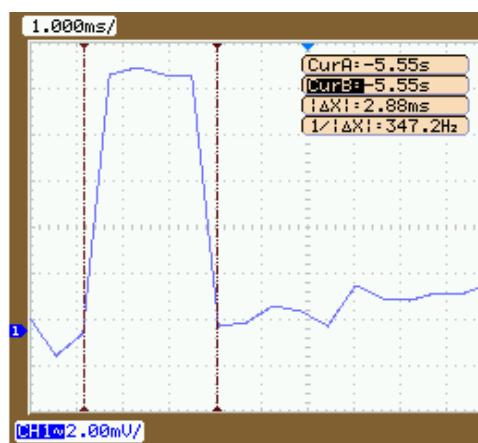
Obr. 8.4: Doba přenosu zprávy 10B při vysílacím výkonu: a)+3dBm b)0dBm

V následujícím kroku byly měřeny doby přenosu pro datovou část rámce **50B**. Měření bylo provedeno pro více úrovní vysílacího výkonu. Z obrázků 8.5 lze vidět, že při výkonu +3dBm se naměřená doba přenosu shoduje s výpočtem. Se snižujícím se vysílacím výkonem se prodlužuje doba přenosu rámce.

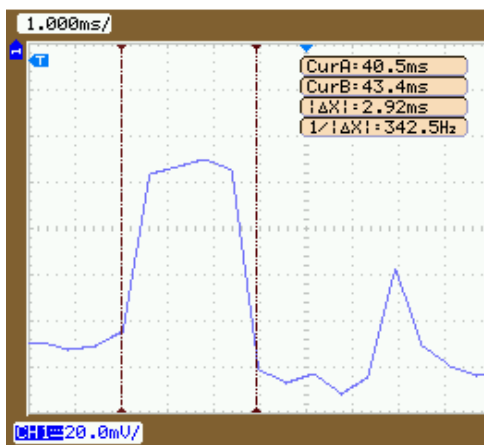
Poslední měření bylo prováděno pro rámec s datovou částí **100B**. Vysílací výkony byly v tomto případě +3dBm a 0dBm. V obou případech se doby přenosu shodují s teoretickými výpočty. Změřené doby přenosu rámce jsou zobrazeny na obr. 8.6.



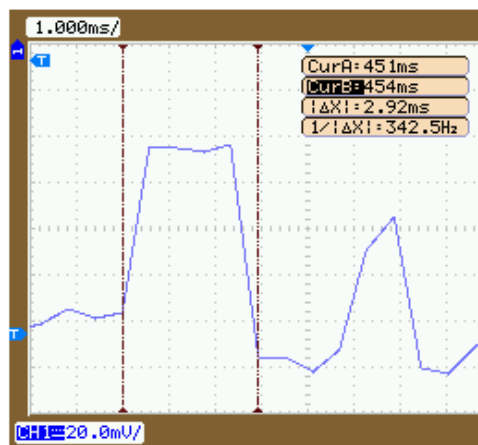
a)



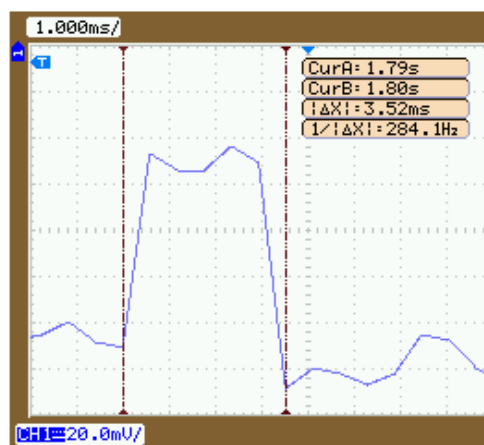
b)



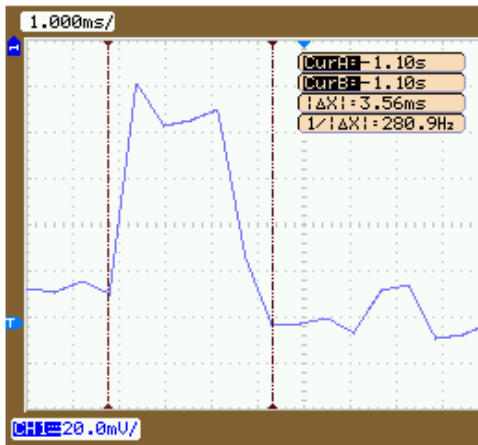
c)



d)

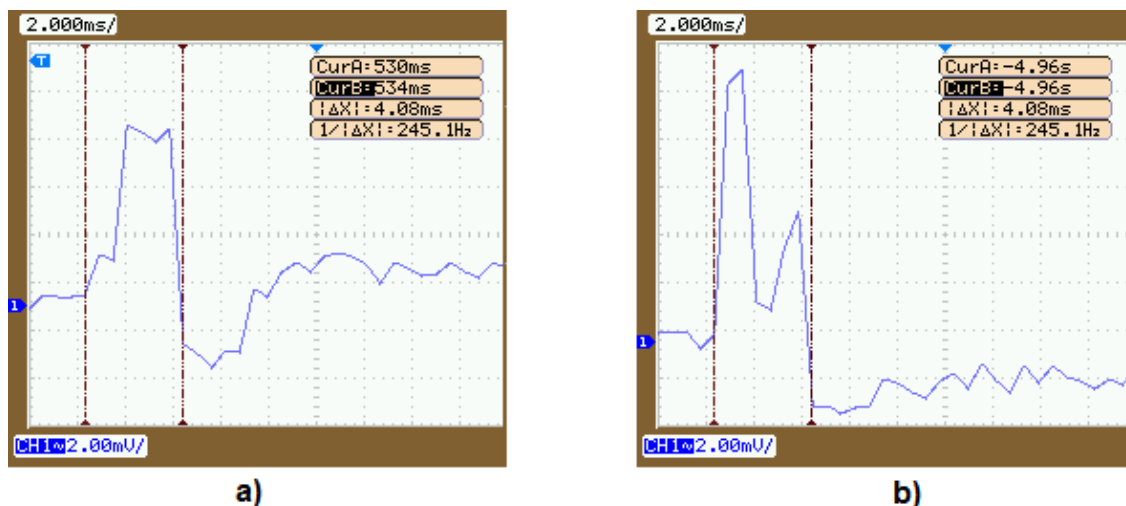


e)



f)

Obr. 8.5: Doby přenosu zprávy 50B při vysílacích výkonech: a)+3dBm b)+1,6dBm c)0dBm d)-3,2dBm e)-7,2dBm f)-17,2dBm



Obr. 8.6: Doba přenosu zprávy 100B při vysílacím výkonu: a) +3dBm b) 0dBm

U naměřených výsledků doby přenosy rámce, při kterých se naměřené výsledky lišily od teoretických výpočtů, mohou být tyto rozdíly způsobeny poklesem přenosové rychlosti a tím prodloužení doby přenosu. Přenosová rychlost závisí na mnoha faktorech kterými je ovlivňována. Jedná se o vlivy způsobené odrazem od stěn a okolních objektů, a především vlivy okolního rušení.

9 VÝPOČTY SPOTŘEBY ENERGIE

Tato část se zabývá výpočtem spotřeby energie bezdrátové komunikační jednotky IRIS. Protože tato jednotka používá pro své napájení dva akumulátory typu AA, je tento parametr významný pro určení délky životnosti těchto akumulátorů a tím i doby po kterou je schopen uzel vysílat data v síti do které je připojen. Pro určení spotřeby energie je potřeba znát velikost napájecího napětí, odebíraný proud, vysílací výkon a také čas po který je uzel v režimu vysílání nebo příjmu. V tomto případě obsahuje jednotka IRIS rádiový čip AT86RF230, který pracuje v napěťovém rozsahu 1,8V až 3,6V. Zvolené 2xNiMH akumulátory GP270AAHC umožňují napěťový rozsah napájení 1,8V až 2,7V. Při výpočtech je zvolena průměrná hodnota vybíjení za celou dobu 2,25V. V režimu příjmu je velikost odebíraného proudu 16 mA. Při režimu vysílání závisí velikost odebíraného proudu na vysílacím výkonu. V našem případě jsou počítány spotřeby energie pro 3 úrovně vysílacího výkonu. Hodnoty vysílacího výkonu jsou zvoleny **+3dBm**, **-3,2dBm** a **-17,2dBm**. Při výkonu **+3dBm** je velikost odebíraného proudu 17 mA, při výkonu **-3,2dBm** je velikost proudu 13 mA a pro výkon **-17,2dBm** je hodnota proudu 10 mA. Časy vysílání a příjmu závisí na velikosti odesílaného nebo přijímaného paketu a jsou známy z předchozích výpočtů. Výpočet energie se provede podle následujícího vzorce 9.1:

$$\text{Energie} = U[\text{V}] \times I[\text{A}] \times t[\text{s}] \quad (9.1)$$

9.1 Režim vysílání

Nejdříve budou vypočteny hodnoty pro režim vysílání při výkonu **+3dBm**. Do předchozího vzorce bude dosazeno napětí $U = 2,25\text{V}$, odebíraný proud vysílání $I = 17\text{mA}$ a pro odeslání dat o velikosti **1B** je čas přenosu $t = 0,896\text{ms}$. Výsledná spotřeba energie pak bude rovna:

$$E_{Tx} = 2,25 \cdot 17^{-3} \cdot 0,896^{-3} = 34,27\mu\text{J} \quad (9.2)$$

Spotřeba energie při odeslání samotných dat o velikosti **10B**, kdy přenos celého rámce trvá 1,184 ms.

$$E_{Tx} = 2,25 \cdot 17^{-3} \cdot 1,184^{-3} = 45,29\mu\text{J} \quad (9.3)$$

Spotřeba energie při odesílání dat s velikostí **50B**, s délkou přenosu rámce 2,464 ms.

$$E_{Tx} = 2,25 \cdot 17^{-3} \cdot 2,464^{-3} = 94,25\mu J \quad (9.4)$$

Výpočet pro paket s velikostí dat **100B** při délce trvání přenosu 4,064 ms.

$$E_{Tx} = 2,25 \cdot 17^{-3} \cdot 4,064^{-3} = 155,45\mu J \quad (9.5)$$

V dalším kroku je zvolen vysílací výkon **-3,2dBm** s velikostí odebíraného proudu 13 mA. Pro přenos zprávy o velikosti **1B** je spotřeba energie rovna:

$$E_{Tx} = 2,25 \cdot 13^{-3} \cdot 0,896^{-3} = 26,21\mu J \quad (9.6)$$

Při přenosu zprávy o velikosti **10B** je spotřeba energie:

$$E_{Tx} = 2,25 \cdot 13^{-3} \cdot 1,184^{-3} = 34,63\mu J \quad (9.7)$$

Spotřeba energie při přenosu zprávy o velikosti **50B** je rovna:

$$E_{Tx} = 2,25 \cdot 13^{-3} \cdot 2,464^{-3} = 72,07\mu J \quad (9.8)$$

Spotřeba energie pro přenos zprávy o délce **100B** je:

$$E_{Tx} = 2,25 \cdot 13^{-3} \cdot 4,064^{-3} = 118,87\mu J \quad (9.9)$$

V posledním kroku je velikost vysílacího výkonu **-17,2dBm**. Při této hodnotě výkonu odebírá uzel proud 10 mA. Pro přenos zprávy o velikosti **1B** je spotřeba energie rovna:

$$E_{Tx} = 2,25 \cdot 10^{-3} \cdot 0,896^{-3} = 20,16\mu J \quad (9.10)$$

Při přenosu zprávy o velikosti **10B** je spotřeba energie:

$$E_{Tx} = 2,25 \cdot 10^{-3} \cdot 1,184^{-3} = 26,64\mu J \quad (9.11)$$

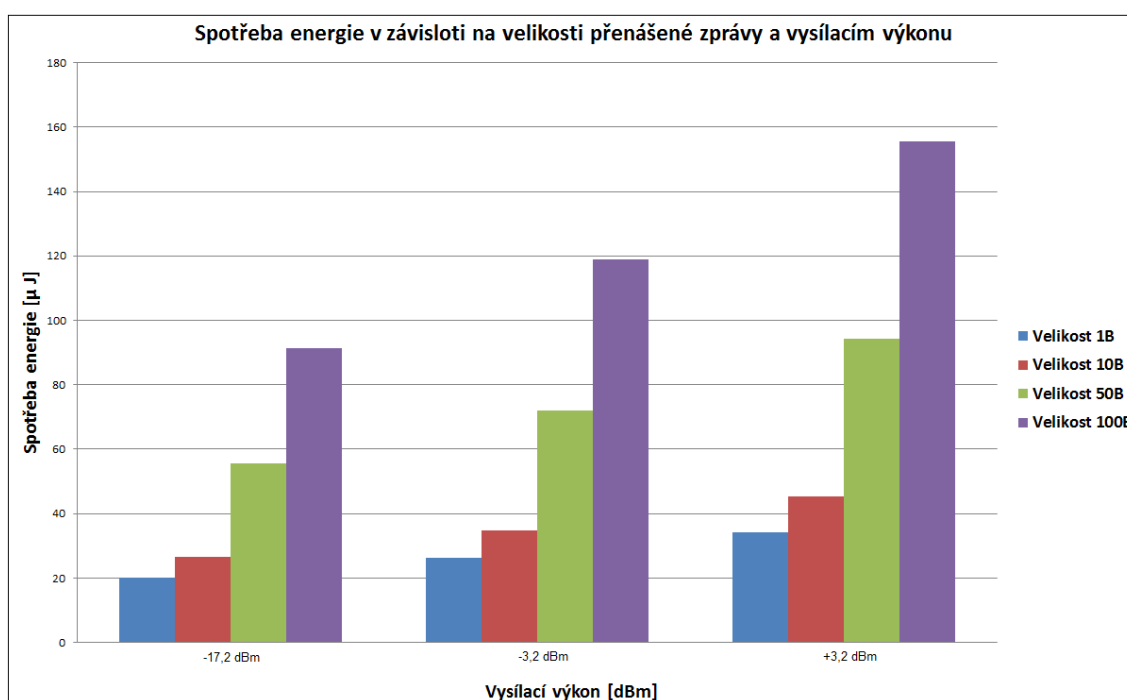
Spotřeba energie při přenosu zprávy o velikosti **50B** je rovna:

$$E_{Tx} = 2,25 \cdot 10^{-3} \cdot 2,464^{-3} = 55,44\mu J \quad (9.12)$$

Spotřeba energie pro přenos zprávy o délce **100B** je:

$$E_{Tx} = 2,25 \cdot 10^{-3} \cdot 4,064^{-3} = 91,44\mu J \quad (9.13)$$

Na obrázku je grafické znázornění spotřeby energie při různém nastavení vysílacího výkonu a velikosti přenášených dat.



Obr. 9.1: Závislost spotřeby energie na výkonu a velikosti přenášených dat

9.2 Režim příjmu

V tomto případě již neplatí změna odběru proudu při různém nastavení výkonu jako tomu bylo u režimu vysílání. Velikost odebíraného proudu je při jakémkoliv příjmu **15,5 mA**. Ostatní parametry zůstávají stejné. Při příjmu **1B** zprávy s dobou přenosu 0,896 ms je spotřeba energie:

$$E_{Rx} = 2,25 \cdot 15,5^{-3} \cdot 0,896^{-3} = 31,25\mu J \quad (9.14)$$

Při příjmu dat o velikosti **10B** s dobou přenosu 1,184 ms je spotřeba energie:

$$E_{Rx} = 2,25 \cdot 15,5^{-3} \cdot 1,184^{-3} = 41,29\mu J \quad (9.15)$$

Při příjmu dat o velikosti **50B** s dobou přenosu 2,464 ms je spotřeba energie:

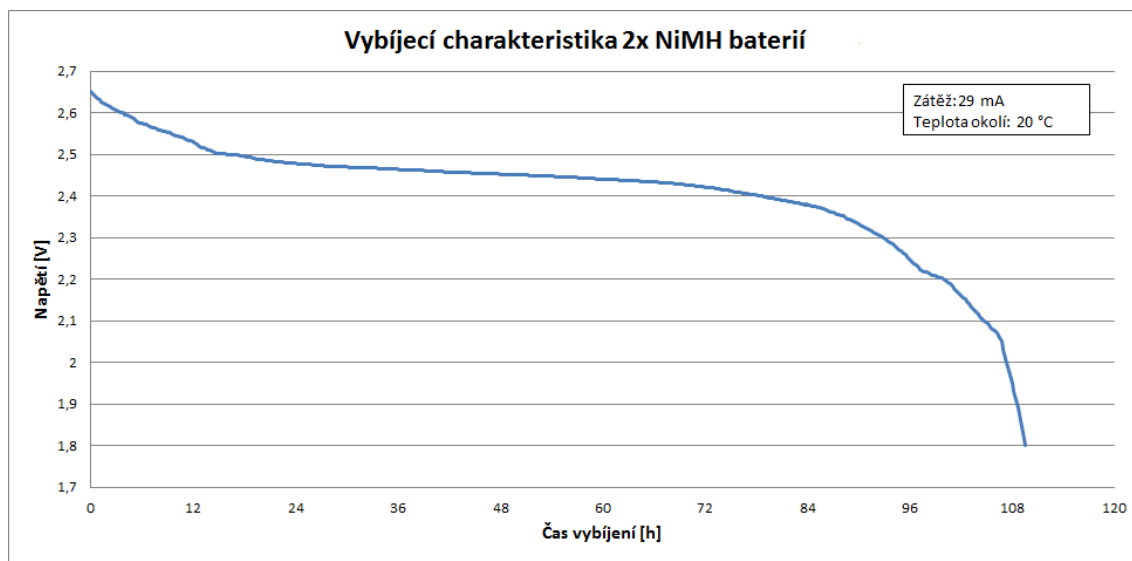
$$E_{Rx} = 2,25 \cdot 15,5^{-3} \cdot 2,464^{-3} = 85,93\mu J \quad (9.16)$$

A pro příjem dat s velikostí **100B** a dobou přenosu 4,064 ms se spotřeba rovná:

$$E_{Rx} = 2,25 \cdot 15,5^{-3} \cdot 4,064^{-3} = 141,73\mu J \quad (9.17)$$

9.3 Výdrž akumulátorů

Hlavním zdrojem energie u bezdrátových jednotek sensorových uzlů jsou dva akumulátory. Proto se klade velký důraz na spotřebu těchto zařízení. Spotřeba zařízení a tím výdrž akumulátorů je ovlivněna režimem v jakém zařízení pracuje, po jakou dobu se v jednotlivém režimu nachází, velikostí přenášených dat a vysílacím výkonem. Největší odběr proudu má komunikační uzel v režimu vysílání při maximálním výkonu +3 dBm, velikost proudu je 17 mA. Odběr proudu mikrokontroléru ATmega1281 je 10 mA. Při praktickém měření byl odebíraný proud bezdrátovou komunikační jednotkou při maximálním vysílacím výkonu 28 mA. Tato hodnota pak byla použita pro zjištění vybíjecí charakteristiky zvolených akumulátorů a tím zjištěný jakou maximální dobu provozu jsou schopny komunikačním jednotkám poskytnout. Jako zátěž byl zvolen odpor 75 Ω , který se ze všech dostupných odporů nejvíce přiblížil skutečné hodnotě. Připojením tohoto odporu do série s 2xNiMH akumulátory byl průměrný vybíjecí proud 29 mA. Na obrázku je zobrazena vybíjecí charakteristika akumulátorů. Vybíjecí charakteristika byla měřena od maximálního nabití akumulátorů 2,652V až po hodnotu 1,8V. Do této hodnoty napětí je bezdrátová komunikační jednotka schopna pracovat. Maximální doba provozu jakou mohou baterie poskytnout při neustále plném vytížení je 109 h a 30 min.



Obr. 9.2: Vybíjecí charakteristika 2xNiMH akumulátorů

9.3.1 Výpočet doby provozu

Bezdrátové komunikační jednotce dodávají energii dva akumulátory NiMH velikosti AA, které jsou zapojeny do série. Nominální hodnota napětí NiMH akumulátorů je udávána 1,4V. Z toho vyplývá, že celkové napětí zdroje bude 2,8V a kapacita bude stejná jako kapacita jednoho akumulátoru. Během praktického měření byly použity vlastní akumulátory GP270AAHC s kapacitou 2700 mAh, a tato hodnota bude také použita v teoretickém výpočtu doby provozu.

Z předchozím informací má zdroj bezdrátové komunikační jednotky celkové napětí 2,8V s nominální kapacitou 2700 mAh. Celková energie zdroje je proto 7,56 Wh. Pokud se při výpočtu příkonu zanedbají uzlu zanedbají malé odchylky proudu vznikající s přechodem do různých stavů a při vysílání, je možné vypočítat příkon komunikační jednotky. Pokud bude uvažováno že komunikační jednotka je neustále v režimu vysílání s maximálním výkonem, pak celkový maximální příkon činí 64,4 mW. Tento celkový příkon je tvořen součtem dílčích příkonů radiového čipu, mikrokontroléru a ostatních periférií. Maximální doba provozu s plně nabitými akumulátory je pak rovna podílu $7,56 \text{ Wh} / 0,0644 \text{ W} = 117,39 \text{ h}$. V porovnání s praktickými výsledky si lze všimnout, že doba přenosu se liší. Tento rozdíl je způsoben vyšším odběrem proudu při praktickém měření a také reálnou maximální hodnotou napětí akumulátoru.

10 ZÁVĚR

Cílem této diplomové práce bylo provést výzkum spotřeby elektrické energie při vytvořené bezdrátové komunikaci mezi dvěma bezdrátovými jednotkami a vliv změny nastavení vysílacích parametrů na této spotřebě.

V teoretické části bylo všeobecně shrnuto a popsáno využití bezdrátových senzorových sítí a typy zařízení, které se při této komunikaci používají. Následně byly popsány standardy používající se v bezdrátové technologii. Jedná se o standardy IEEE 802.15.4 a standard ZigBee, který předchází standard rozšiřuje o síťovou a aplikační vrstvu. Následoval popis topologie, jednotlivé vrstvy komunikačního modelu a tři typy ZigBee zařízení: koordinátor, směrovač a koncové zařízení. Poté byl rozebrán sériový přenos dat s popisem sériového synchronního přenosu využívající se u bitově orientovaných protokolů a pro datovou komunikaci využívající se pro přenos většího objemu dat.

Následující kapitoly praktické části popisují software ZigBee BitCloud, určený pro vývoj firmwaru pro bezdrátové aplikace. Tento software byl použit při praktickém řešení práce. Dále je popsán použitý rádiový čip AT86RF230 společně s tabulkou úrovní vysílacích výkonů a jeho popisem spotřeby energie při jednotlivých režimech. S těmito hodnotami pak bylo počítáno v praktické části. Poté bylo rozebráno napájení a životnost zařízení napájených pomocí vybraných akumulátorů. Poslední dvě kapitoly teoretické části jsou zaměřeny na spotřebu energie v bezdrátových sítích a energetickým modelem při zpracování zprávy.

V praktické části bylo nejdříve realizováno vytvoření dvou bezdrátových komunikačních jednotek. Jedno zařízení je typu koordinátor a druhé typu směrovač, který v pravidelných intervalech zasílá koordinátorovy síti přenášená data. Přijaté pakety jsou následně po sběrnici USART zasílána do počítače. V praktické ukázce jsou v terminálu zobrazeny informace o textu přenášené zprávy, vysílacím výkonu a sekvenčním číslem přijatého paketu. Dále měla být přenášena informace o aktuální hodnotě napětí akumulátoru vysílacího zařízení umožňující sledování spotřeby energie, to se však nepodařilo zrealizovat. U zařízení byly měněny parametry ovlivňující dobu zasílání paketů a spotřebu elektrické energie zařízení. Pakety byly zasílány ve čtyřech velikostech a s různými úrovněmi vysílacího výkonu. Čím je velikost paketu větší, tím je delší doba přenosu a větší spotřeba energie zařízení. Protože jsou data zasílána z aplikační vrstvy, každá z nižších vrstev přidá k datové části záhlaví o určité délce. Dále platí, že s vyšší úrovní vysílacího výkonu a častějších intervalech odesílání roste spotřeba energie.

Dále byly teoretické výpočty doby přenosu porovnány s naměřenými výsledky. Při měření byla měněna velikost přenášené zprávy a úrovně vysílacího výkonu. Ve většině případů byly naměřeny doby přenosu větší než teoretické hodnoty. To mohlo

být způsobeno poklesem přenosové rychlosti, která je ovlivněna vlivy způsobené odrazem od okolních objektů a především vlivy okolního rušení. Měření pro velikosti datové části rámce o velikosti 1B, 10B a 100B bylo provedeno pouze pro dvě úrovně vysílání. Pro více úrovní vysílacího výkonu nebylo z časových důvodů již uskutečněno více měření. Pokud by byla obě komunikující zařízení v blízkém dosahu s co možná nejmenším vlivem rušení, je možné nastavit menší úroveň vysílacího výkonu a tím snížit spotřebu energie. Na větší vzdálenosti a v okolí se zhoršenými přenosovými podmínkami je vhodné úroveň vysílacího výkonu zvýšit. V poslední části byla změřena vybíjecí charakteristika použitých akumulátorů a výpočet doby provozu při neustále maximální velikosti odebíraného proudu zařízení.

LITERATURA

- [1] MRÁZ, Ľ. *AMINA - Wireless sensor network project* [online]. poslední aktualizace 2011 [cit. 1. 12. 2012]. Dostupné z URL: <<http://www.aminawsn.org/>>.
- [2] KREIBICH, O; NEUŽIL, J. *Bezdrátové senzorové sítě* [online]. 2012 [cit. 1. 12. 2012]. Dostupné z URL: <<http://www.mmspektrum.com/clanek/bezdratove-senzorove-site.html>>.
- [3] TOWNSEND, CH; ARMS, S. *Wireless Sensor Networks: Principles and Applications* [online]. 2004, s. 439-450. [cit. 1. 12. 2012]. Dostupné z URL: <<http://web.cecs.pdx.edu/~hieu2/capstone/WSN02.pdf>>.
- [4] ŠIMEK, M. Skripta *Bezdrátové senzorové sítě* Brno: Vysoké učení technické v Brně, 2012. 165 s.
- [5] Institute IEEE. *IEEE 802.15 Working Group for Wireless Personal Area Networks* [online]. 2001, poslední aktualizace 29. 9. 2005 [cit. 2. 12. 2012]. Dostupné z URL: <<http://www.ieee802.org/15/about.html>>.
- [6] HYNČICA, O; PAVLATA, K. *Bezdrátové komunikační systémy založené na IEEE 802.15.4 v procesní automatizaci* [online]. 2011, [cit. 2. 12. 2012]. Dostupné z URL: <<http://www.odbornecasopisy.cz/res/pdf/43411.pdf>>.
- [7] VOJÁČEK, A. *ZigBee - novinka na poli bezdrátové komunikace* [online]. poslední aktualizace 7. 6. 2005 [cit. 2. 12. 2012]. Dostupné z URL: <<http://www.hw.cz/Rozhrani/ART1299-ZigBee---novinka-na-poli-bezdratove-komunikace.html>>.
- [8] KOTON, J; ČÍKA, P; KŘIVÁNEK, V. *Standard nízkorychlostní bezdrátové komunikace ZigBee* [online]. 2006, [cit. 3. 12. 2012]. Dostupné z URL: <<http://access.feld.cvut.cz/view.php?cisloclanku=2006032001>>. ISSN 1214-9675
- [9] CS.VSB.cz. *Sériový přenos dat* [online]. [cit. 6. 12. 2012]. Dostupné z URL: <<http://www.cs.vsb.cz/grygarek/LAN/sem/sercomm.html>>.
- [10] PC-SITE.OWEBU.cz. *Základní formy přenosů* [online]. [cit. 6. 12. 2012]. Dostupné z URL: <<http://pc-site.owebu.cz/?page=PPre>>.
- [11] FRÝZA, T. *Řízení sériové komunikace* [online]. 2012, [cit. 7. 12. 2012]. Dostupné z URL: <http://www.urel.feec.vutbr.cz/~fryza/downloads/mpt_pred_07.pdf>.

- [12] ATMEL. *Atmel BitCloud Developer Guide* [online]. 2011, [cit. 7. 12. 2012]. Dostupné z URL: <<http://www.atmel.com/Images/doc8199.pdf>>.
- [13] MORÁVEK, P; KOMOSNÝ, D; ŠIMEK, M; MRÁZ, L. *Energy Demands of 802.15.4/ZigBee. Communication with IRIS Sensor Motes* 2011, ISBN 978-1-4577-1411-5.
- [14] CHIANG, MU-HUAN. *Energy Optimization in Wireless Sensor Networks*. North Carolina State University, 2007, USA : VDM Verlag Dr. Müller e.K., 100 s. ISBN 978-3-8364-6132-0.
- [15] ŠIMEK, M; MORÁVEK, P; SILVA, J. *Wireless Sensor Networking in Matlab: Step-by-Step*.
- [16] ATMEL. *Atmel Studio 6 - The Integrated Development Platform* [online]. 2012, [cit. 9. 12. 2012]. Dostupné z URL: <http://www.atmel.com/Microsite/atmel_studio6/default.aspx>.
- [17] ATMEL. *Connection to a target board with the AVR JTAGICE mkII* [online]. 2006, [cit. 9. 12. 2012]. Dostupné z URL: <<http://www.atmel.com/Images/doc2562.pdf>>.
- [18] MIB520. *Datasheet* San Jose, California: Crossbow Technology, Document Part Number: 6020-0091-04 Rev A
- [19] IRIS. *Datasheet* San Jose, California: Crossbow Technology, Document Part Number: 6020-0124-01 Rev A
- [20] ATMEL. *AT86RF230 Datasheet* Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM Applications [online]. Dostupné z URL: <<http://www.atmel.com/Images/doc5131.pdf>>.
- [21] ATMEL. *ATmega1281 Datasheet* 8-bit AVR Microcontroller with 256K Bytes In-System Programmable Flash [online]. Dostupné z URL: <http://www.datasheetcatalog.com/datasheets_pdf/A/T/M/E/ATMEGA1281.shtml>.
- [22] BUCHMANN, I. *Batteries in a Portable World: A Handbook on Rechargeable Batteries for Non-Engineers*. Cadex Electronics Inc., 2001, 292 s. ISBN 0968211828.
- [23] GP Batteries. *GP270AAHC Datasheet* [online]. Dostupné z URL: <<http://www.uni-kl.de/elektronik-lager/481011>>.

- [24] FARAHANI, S. *ZigBee Wireless Networks and Transceivers*. Newnws., 2008, 360 s. ISBN 978-0-7506-8393-7.
- [25] BOLDIŠ, P. *Bibliografické citace dokumentů podle ČSN ISO 690 a ČSN ISO 690-2* [online]. 2001, poslední aktualizace 11. 11. 2004 [cit. 17. 2. 2005]. Dostupné z URL: <<http://www.boldis.cz/citace/citace.html>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

- ACK Acknowledgement – Potvrzení
- AES Advanced Encryption Standard – Pokročilý šifrovací standard
- AF Arithmetic Logic Unit – Aritmetická logická jednotka
- ALU Application Framework – Aplikační rámec
- API Application Programming Interface – Rozhraní pro programování aplikací
- APS Application Support Sub-layer – Pomocná aplikační podvrstva
- ARQ Automatic Repeat reQuest – Automatický požadavek na opakování
- ASK Amplitude Shift Keying – Amplitudové digitální modulace
- BPSK Binary Phase Shift Keying – Binární fázová modulace
- BSP Board Support Package – Podpůrný balík desky
- BTT Broadcast Transaction Table – Tabulka přenesených transakcí
- BTR Broadcast Transaction Record – Zápis přenesených transakcí
- CCA Clear Channel Assessment
- CSMA-CA Carrier Sense Multiple Access with Collision Avoidance – Princip detekování nosné s předcházení kolizím
- DSSS Direct Sequence Spread Spectrum – Technika přímého rozprostřeného spektra
- FFD Full Function Device – Zařízení s plnou funkcionalitou
- GPRS General Packet Radio Service – Obecný paketový rádiový systém
- HAL Hardware Abstraction Layer – Abstraktní hardwarová vrstva
- IEEE Institute of Electrical and Electronics Engineers – Institut pro elektrotechnické a elektronické inženýrství
- I2C Inter-Integrated Circuit – Dvojvodičové spojení
- LIFS Long Interframe Space
- MAC Media Access Control – Kontrola přístupu k médiu

MIPS Million Instruction Per Second – Počet instrukcí za sekundu

MPDU MAC Protocol Data Unit

MSDU MAC Service Data Unit

NLDE Network Layer Data Entity – Datová jednotka síťové vrstvy

NLME Network Layer Management Entity – Manager jednotky síťové vrstvy

O-QPSK Offset Quadrature Phase Shift Keying

PAN Personal Area Network – Osobní síť

PHR PHY Header – Hlavička fyzické vrstvy

PSSS Parallel Sequence Spread Spectrum – Modulace s rozptřením spektra signálu

QPSK Quadrature Phase Shift Keying – Kvadrurní fázová modulace

RF Radio Frequency – Rádiová frekvence

RFD Reduced Function Device – Zařízení s redukovanou funkcí

SHR Serial of Header – Sériové hlavičky

SIFS Short Interframe Space

SPI Serial Peripheral Interface – Sériové periferní rozhraní

UART Universal Asynchronous Receiver and Transmitter – Univerzální asynchronní přijímač a vysílač

USART Universal Synchronous and Asynchronous Receiver and Transmitter – Univerzální synchronní a asynchronní přijímač a vysílač

USB Universal Serial Bus – Univerzální sériová sběrnice

WPAN Wireless Personal Area Networks – Bezdrátové osobní sítě

WSN Wireless sensor network – Bezdrátová senzorová síť

ZDO ZigBee Device Object – ZigBee objekt zařízení

A PŘÍLOHY

A.1 Přiložené DVD

Na přiloženém DVD je uložen prakticky řešený firmware pro oba typy komunikační jednotky. Dále je zde software AVR Studio 5.1 ve kterém byly programy vytvořeny, dále software WinAVR a potřebné ovladače a doplňky. DVD také obsahuje elektronickou podobu diplomové práce a původní zdrojový kód v programu \LaTeX .