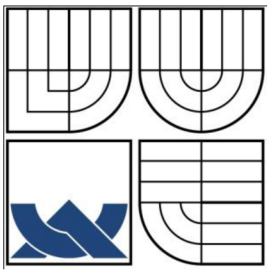
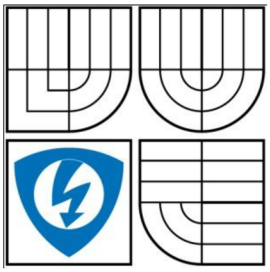


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH PAKETOVÉHO ANALYZÁTORU PRO BEZDRÁTOVÉ SENZOROVÉ SÍTĚ ZALOŽENÉ NA STANDARDU IEEE 802.15.4

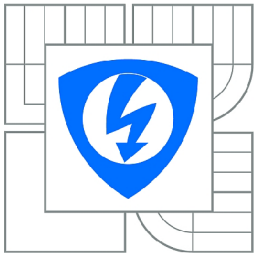
Packet analyzer for Wireless Sensor Networks based on the standard IEEE 802.15.4

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. MARTIN BEDNAŘÍK
AUTHOR

VEDOUCÍ PRÁCE Ing. ĽUBOMÍR MRÁZ
SUPERVISOR

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Martin Bednařík	***ID:***	73091
Ročník:	2	***Akademický rok:***	2010/2011

NÁZEV TÉMATU:

Návrh paketového analyzátoru pro bezdrátové senzorové sítě založené na standardu IEEE 802.15.4

POKYNY PRO VYPRACOVÁNÍ:

Student v práci prostuduje standard pro nízkopříkonové bezdrátové senzorové sítě IEEE 802.15.4. Dále navrhne a realizuje paketový analyzátor na bázi IEEE 802.15.4 pomocí rádiových modulů AT86xx a obslužního mikrokontroléru AVR od firmy Atmel. Je vhodné zvážit vhodnou volbu rozhraní pro komunikaci s počítačem a optimální návrh firmwaru pro mikrokontrolér. Dále student zaintegruje navrhnutý analyzátor do analyzačního softwaru Wireshark případne jiného dostupného produktu. Na závěr student analyzuje a vyhodnotí parametry navrženého paketového analyzátoru.

DOPORUČENÁ LITERATURA:

[1] Jose A. Gutierrez. IEEE 802.15.4 Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensor Networks: Institute of Electrical & Electronics Enginee, 2003. 155p. ISBN 0738135577
[2] Mauri Kuorilehto, Mikko Kohvakka, Jukka Suhonen, Panu Hämäläinen, Marko Hännikäinen, Timo D.Hamalainen. Ultra-Low Energy Wireless Sensor Networks in Practice: Theory, Realization and Deployment. Wiley, 2008.396 p.ISBN 0470057866.
[3] Šandera, Josef. Návrh plošných spojů pro povrchovou montáž. BEN, 2006. 272p. ISBN: 8073001810

Termín zadání:	7.2.2011	***Termín odevzdání:***	26.5.2011

Vedoucí práce: Ing. Ľubomír Mráz

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.

Anotace

Cílem tohoto projektu je seznámit se s technologiemi bezdrátových senzorových sítí, se standardem IEEE 802.15.4. a principem komunikace v bezdrátových senzorových sítích podle tohoto standardu.

Hlavní částí projektu je navrhnout paketový analyzátor, který bude schopen zachytávat data na vybraném kanále a tato data analyzovat. Součástí je též tvorba potřebného programového vybavení pro mikrokontrolér.

Dalším cílem projektu je udělat průzkum na trhu dostupných paketových analyzátorů a porovnat tyto proti analyzátoru, vytvořeného v této práci.

Abstract

The objective of this thesis is to get familiar with wireless sensor networks technologies and with standard IEEE 802.15.4. and communication principle in wireless sensor networks built on this standard.

Main goal of this project is to design a packet analyzer, which is capable to catch data on chosen channel and this data is able to analyze. Part of this project is production of necessary microcontroller software equipment.

Another output of this project is do a research of available packet analyzers on market and compare them with analyzer build by this thesis.

Klíčová slova

Bezdrátová senzorová síť, WSN, IEEE 802.15.4, Zigbee, WPAN, CSMA/CA, PAN koordinátor, analyzátor, Wireshark, Atmel, AVR, USB, µracoli

Keywords

Wireless sensor network, WNS, IEEE 802.15.4, Zigbee, WPAN, CSMA/CA, PAN coordinator, analyzer, Wireshark, Atmel, AVR, USB, µracoli

Bibliografická citace mé práce

BEDNAŘÍK, M. *Návrh paketového analyzátoru pro bezdrátové senzorové sítě založené na standardu IEEE 802.15.4.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 52 stran, 3 přílohy. Vedoucí diplomové práce Ing. Ľubomír Mráz.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Návrh paketového analyzátoru pro bezdrátové senzorové sítě založené na standardu IEEE 802.15.4“ 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 tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Ľubomírovi Mrázovi za poskytnutí cenných rad a připomínek při zpracování mé diplomové práce.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Paketový analyzátor bezdrátové senzorové sítě je schopen zachytit data, konkrétně rámce dle standardu 802.15.4 a obsah těchto rámců zobrazit, analyzovat a dekódovat ve vhodném softwaru.

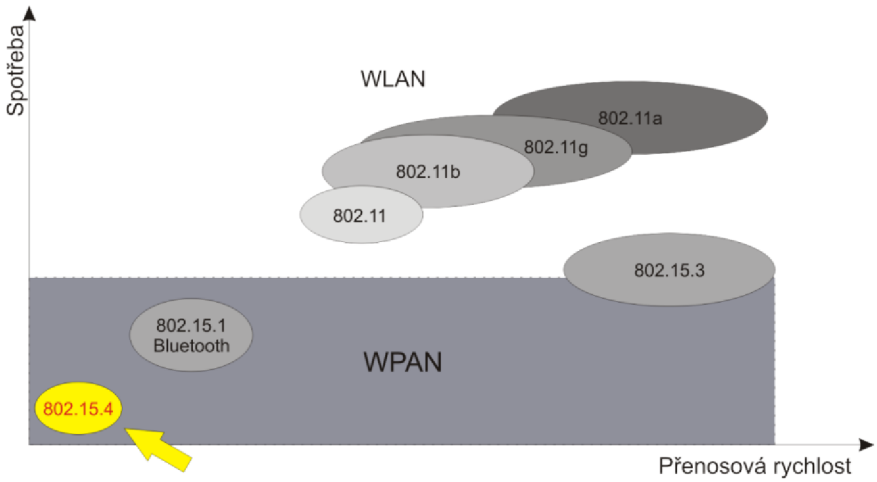
Rozbor diplomové práce je rozdělen do osmi kapitol. První kapitola pojednává obecně o bezdrátových sítích a dále jsou zde rozebrány právě sítě senzorové. Kapitola druhá rozebírá podrobně standard IEEE 802.15.4, podle kterého probíhá komunikace v bezdrátových senzorových sítích. Třetí kapitola se zaměřuje na samotný možný návrh realizace paketového analyzátoru. Existuje celá řada možností, jak tento analyzátor realizovat, přičemž byly vybrány 4 varianty. Kapitola čtvrtá se zabývá popisem vybrané varianty řešení. V kapitole páté je popsán software nutný pro použití analyzátoru a návrh tohoto softwaru. Z možných variant v kap. 3, byla vybrána ta, která zahrnovala návrh hardwaru pro analyzátor a tento je popsán v kapitole šesté. Jak je třeba tento analyzátor ovládat, pokud uživatel s ním přijde poprvé do styku, je popsáno v kapitole sedmé. Kapitola osmá se zabývá analýzou navrženého paketového analyzátoru, popsáním nedostatků zjištěných při této analýze a možnými řešeními, jak tyto nedostatky odstranit.

1 Bezdrátová senzorová síť

Bezdrátové sítě vznikly jako alternativa klasických drátových sítí. Skupina IEEE definuje dvě základní skupiny bezdrátových sítí – WLAN (Wireless Local Area Network) a WPAN (Wireless Personal Area Network). [1]

V současnosti jsou podle IEEE personální bezdrátové sítě (WPAN) rozdělené do 3 skupin podle přenosové rychlosti, spotřeby el. energie a QoS:

1. High-data rate WPAN (IEEE 802.15.3) je vhodná pro multimediální aplikace, vyžadující QOS,
2. Medium-data rate WPAN (IEEE 802.15.1 / Bluetooth) je vhodná jako náhrada kabelového spoje pro spotřební elektroniku, a to především mobilní telefony či PDA,
3. Low-data rate WPAN (LR-WPAN) (IEEE 802.15.4) je zamřena na aplikace, které nepotřebují vysokou přenosovou rychlost, jsou levné a spotřebují minimum el. energie. Do této kategorie spadají právě bezdrátové senzorové sítě.



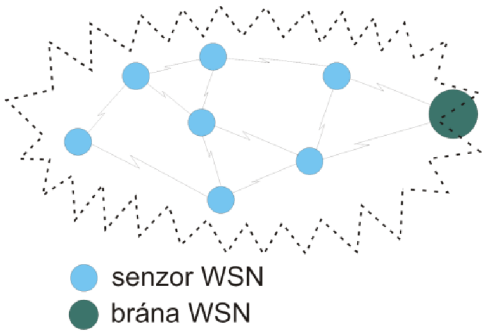
Obr. 1 Srovnání bezdrátových standardů

Obr. 1 zobrazuje operační prostor jednotlivých sítí. Lze vidět, že Std 802.15.4 je navržen tak, aby nezasahoval a nekonkuroval jiným vyšším a především rychlejším síťovým standardům.

Bezdrátová senzorová síť (angl. WSN – Wireless Sensor Network) je soustava prostorově rozmístěných autonomních senzorů, sloužících k řízení a monitorování fyzikálních či přírodních podmínek, jako jsou například teplota, zvuk, tlak, pohyb, emise apod. Monitorování ve smyslu sběru dat ze senzorů a odeslání těchto dat nadřazenému systému ke zpracování. Řízení ve smyslu ovládání určitého akčního členu (spínání osvětlení, ovládání žaluzií, apod.). Vývoj této sítě byl motivován armádními aplikacemi, ale nyní jsou používány především v mnoha industriálních a civilních aplikacích, jako je monitorování výroby, ve zdravotnictví ke sledování stavu pacientů či například k automatizaci domácnosti.

1.1 Design

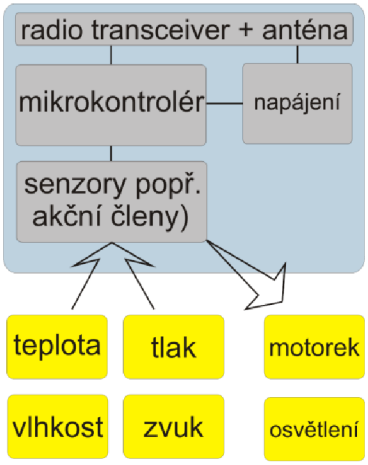
Hlavní prvky bezdrátové senzorové sítě jsou tedy dílčí uzly (senzory), snímající požadované veličiny, popřípadě ovládající další zařízení a hlavní uzel (brána), sloužící k přijímání/vysílání dat.



Obr. 2 Design WSN

Uzel WSN se ve většině případů sestává z následujících prvků:

- RF modul (přijímač+vysílač) popřípadě jiný bezdrátový prvek,
- senzor popř. akční člen,
- mikrokontrolér s příslušným firmwarem,
- zdroj napájení, obvykle baterie.



Obr. 3 Blokové schéma uzlu WSN

Přičemž takovýto uzel může pracovat ve 4 režimech:

- sběr dat – uzle snímá ze senzoru údaje, převádí do digitální podoby, zpracuje a případně uloží,
- přenos dat – data jsou kódována, zapouzdřena do rámců a bezdrátově vyslána,
- příjem dat – dekódování dat, kontrola chyb,
- úsporný režim.

Mezi hlavní požadavky na každý uzel patří především velmi nízká spotřeba. Jak lze vidět na Obr. 3, tak každý uzel je napájen baterií (blok *napájení*). Dlouhá výdrž zařízení je ještě podtržena faktem, že přibližně 99% času se uzel nachází ve stavu spánku a odebírá tak minimální množství energie. Dále je potřeba, aby toto zařízení mělo co nejmenší rozměry, potažmo aby byly téměř nepostřehnutelné. WSN musí být schopna komunikovat se stovkami uzlů, přičemž musí být také schopna rozpoznat

nové uzly v síti či uzly odstraněné. Jelikož přenos se provádí bezdrátově, pak je záhodno, aby tato data byla zabezpečena.

Standardů bezdrátové senzorové sítě existuje celá řada, ovšem mezi nejvíce používané patří:

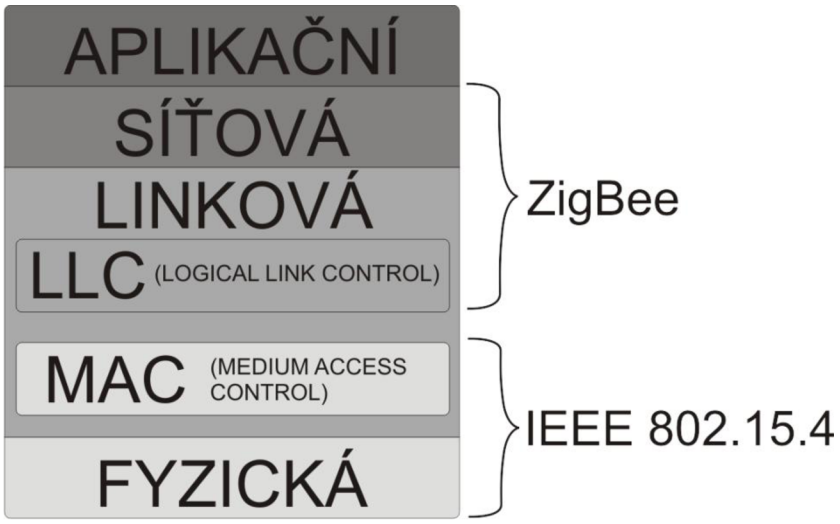
- **Standard ZigBee** – ZigBee je standardem specifikujícím protokoly pracující na vyšších vrstvách. Je založen na standardu IEEE 802.15.4. Tento standard produktem ZigBee aliance, což je seskupení průmyslových společností Honeywell, Freescale, Texas Instruments a další.

- **Standard IEEE 802.15.4** – Standard IEEE 802.15.4 specifikuje fyzickou vrstvu a podvrstvu MAC (medium access control) linkové vrstvy pro LR-WPAN sítě. Tento standard je vytvořen organizací IEEE a udržován pracovní skupinou 4 spadající pod 802.15. Tento standard je základem pro ostatní standardy pracující na vyšších vrstvách jako jsou například ZigBee, WirelessHART a MiWi a těmto standardům poskytuje služby.

2 Standard IEEE 802.15.4

2.1 Úvod

Std IEEE 802.15.4 specifikuje nejnižší vrstvy v bezdrátové senzorové síti (WPAN), které jsou zaměřeny na nízkonákladovou a nízkorychlostní komunikaci. Základní myšlenkou je vytvořit komunikační oblast o průměru desítek metrů s přenosovou rychlostí do 250kbit/s. [1]



Obr. 4 Vrstvový model ISO/OSI vs. WSN

Definice vrstvového modelu pro Std IEEE 802.15.4 vychází z OSI modelu a definuje pouze fyzickou a Medium Access Control (MAC) podvrstvu linkové vrstvy, kterou ale v tomto standardu neoznačujeme jako podvrstvu, ale jako celou vrstvu. Komunikací na vyšších vrstvách se standard nezabývá, viz Obr. 4

2.2 Komponenty

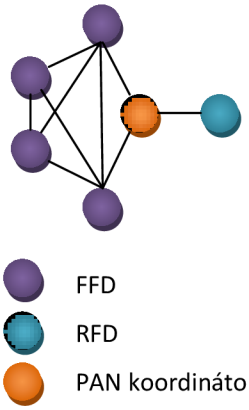
- FFD (Full Function Device) – Plně funkční zařízení: obsahuje kompletní sadu MAC služeb a umí pracovat v jakémkoliv z tří možných statusů:
  - PAN koordinátor – vytváří a řídí síť; v celé síti pouze jeden,
  - Koordinátor – poskytuje služby dalším zařízením; v síti jich může být více,
  - Network Device – komunikuje pouze s nadřazeným koordinátorem.

- RFD (Reduced Function Device) – Zařízení se sníženou funkcionalitou: obsahuje redukovanou sadu MAC služeb a umí pracovat pouze jako tzv. network device.

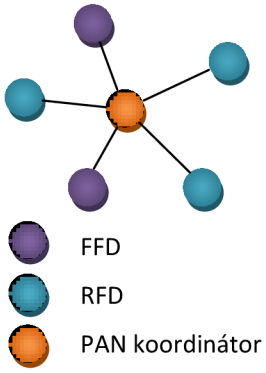
2.3 Síťové topologie

- **Hvězda** – PAN koordinátor je centrálním uzlem, jenž komunikuje s ostatními FFD či RFD zařízeními. Veškerá komunikace zde probíhá přes PAN koordinátora.

- **Peer-to-perr** – jednotlivá zařízení jsou si rovna, neexistuje zde žádný centrální uzel. PAN koordinátor se zde však musí nacházet. V případě výpadku jednoho uzlu se nepřeruší komunikace, jako by se to mohlo stát u hvězdy, v případě, že by vypadl PAN koordinátor. Jelikož Std IEEE 802.15.4 nedefinuje síťovou vrstvu, směrování zde není tedy přímo podporováno, ovšem tvoří tak základ pro další standardy pracující na vyšších vrstvách jako je například ZigBee.



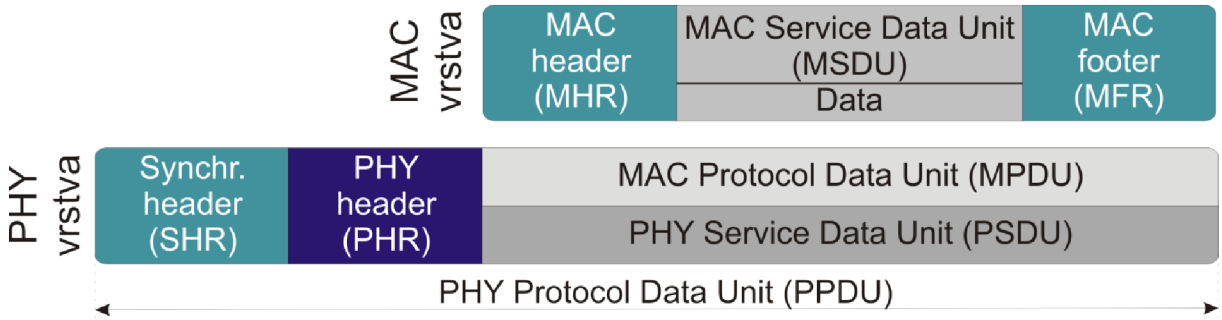
Obr. 6 Topologie peer-to-peer



Obr. 5 Topologie hvězda

2.4 Datové jednotky

Standard definuje 4 druhy rámců, každý navrhnutý jako PHY Service Data Unit (PSDU). Tento je zabalen do PHY Protocol Data Unit (PPDU), který je složen ze synchronizační hlavičky (Synchronization header – SHR), PHY hlavičky (PHY header – PHR) a PHY Service Data Unit (PSDU). PSDU naopak v sobě obsahuje MAC Protocol Data Unit (MPDU), což je datová jednotka MAC vrstvy. Ta se skládá z MAC hlavičky, MAC zápatí a MAC Service Data Unit (MSDU), což jsou v podstatě samotná užitečná data. [1]



Obr. 7 Struktura datových jednotek podle vrstev

Podrobnější popisy datových jednotek jsou uvedeny v kapitolách 2.6.1 a 2.7.1.

2.5 Metoda přístupu ke sdílenému médiu

Nezáležíce na typu sítě, každé sítové zařízení využívá metodu CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). Tato metoda je založena na sdíleném přenosovém kanálu. Pokud stanice chce komunikovat, tak před začátkem vyslání dat zjistí, zda je médium volné. Pokud zjistí, že ano, začne okamžitě vysílat data. Pokud je médium obsazeno, čeká se na jeho uvolnění a zahájí se exponenciální čekání. Po uplynutí této doby se opět zjistí, zda je kanál volný a proces se opakuje. [1]

2.6 Fyzická vrstva

Fyzická vrstva (PHY) poskytuje rozhraní mezi vrstvou linkovou MAC a fyzickým médiem, kde skutečně probíhá komunikace, což v případě WSN je vzduch. PHY vrstva je nejnižší vrstvou v ISO/OSI modelu a má na starosti kontrolu (aktivaci/deaktivaci) RF modulu, kvalitu linky, výběr vhodného komunikačního kanálu a příjem a vysílání zpráv přes fyzické médium. Přenos dat tedy probíhá pomocí rádiových vln, jejichž frekvence je uvedena v Tab. 1

Tab. 1 Frekvenční pásma pro standard IEEE 802.15.4

Stránka	Kanál	Pásmo [MHz]	Šířka pásma [MHz]	Přenos. rychlost [kb/s]	Modulace	Čipová rychlosti
0	0	868	868-868,6	20	BPSK	300 kchips/s
1				100	Q-QPSK	400 kchips/s
2				250	PSSS	400 kchips/s
0	1-10	915	902-928	40	BPSK	600 kchips/s
1				250	Q-QPSK	1 Mchip/s
2				250	PSSS	1,6 Mchip/s
0	11-26	2400	2400-2483,5	250	Q-QPSK	2 Mchip/s

Z Tab. 1 tedy vyplývá, že Std IEEE 802.15.4 definuje 3 frekvenční pásma:

- 868 – 868,6 MHz: bezlicenční pásmo pro Evropské země, 1 komunikační kanál
- 902 – 928 MHz: bezlicenční pásmo pro severní Ameriku, 10 komunikačních kanálů
- 2400 – 2483,5 MHz: bezlicenční pásmo pro celosvětové použití, 16 komunikačních kanálů

Pásmo 2,4GHz se může zdát jako nejvhodnější volba pro většinu IEEE 802.15.4 aplikací, především pro produkty, pohybující se skrz jednotlivé „frekvenční regiony“. Nicméně toto pásmo je sdíleno mezi mnoha dalšími aplikacemi, jako jsou sítě WLAN či jiné WPAN. Což může vyústit v problémy při komunikaci. Tedy proto pásma 868/915 MHz jsou definována jako alternativa tam, kde 2,4GHz není možné, resp. by způsobovalo problémy. [1]

Kvůli fyzikálním charakteristikám každého pásma a regulačním protokolům, kde jsou použity, Std IEEE 802.15.4 specifikuje různé přenosové rychlosti a modulace – viz Tab. 1. Základní přenosovou rychlostí v pásmu 868 MHz je 20 kb/s, ale samozřejmě uživatelé si můžou vybrat i vyšší rychlosti využívající různé modulace.

Std IEEE 802.15.4 využívá kombinaci čísel kanálů a tzv. stránek kanálu ke specifikaci frekvence. Existuje celkově 32 stránek, přičemž pouze 3 z nich jsou definovány standardem, ostatních 29 stránek je rezervováno pro případné budoucí rozšíření. Každá stránka je rozdělena do 27 kanálů, očíslovaných 0 až 26. Stránky jsou číslovány 0 až 31.

2.6.1 Datová jednotka fyzické vrstvy

Jak bylo předznamenáno v kapitole 2.4, datová jednotka fyzické vrstvy se nazývá PHY Protocol Data Unit (PPDU). Tedy formální pojmenování jednotky zní paket. Z pohledu OSI modelu, se to může zdát, jako nesprávné pojmenování, jelikož datovou jednotkou této vrtsvy bývá rámec. Ovšem pojem paket na fyzické vrtsvě WSN je běžně používán. Obsahuje v sobě veškerá data z vyšších vrstev. [2]

PPDU se skládá z:

- synchronizační hlavička (SHR),
- hlavička fyzické vrstvy (PHR),
- data poskytnutá vyšší vrstvou.

Obr. 8 Struktura datové jednotky fyzické vrstvy

Synchronizační hlavička (SHR): skládá se ze dvou polí – záhlaví a oddělovač začátku paketu.
Záhlaví obsahuje 32 bitů, které jsou samé log. 0. Záhlaví poskytuje přijímači dostačující počet bitů, aby dosáhl čipové a bitové synchronizace.
Oddělovač začátku paketu obsahuje 8bitové slovo „0xe6“ (11100101), což přijímači umožňuje stanovit začátek paketu.

Hlavička fyzické vrstvy (PHR): je to jedno pole obsahující 8 bitů. MSB bit je rezervován a zbývajících 7 bitů určuje délku paketu PPDU v bajtech. Pakety délky 0 až 4 a 6 až 8 bajtů jsou rezervovány. Pakety délky 5 bajtů jsou MPDU potvrzovací pakety a pakety s 9 nebo více bajty jsou MPDU, tedy samotná data poskytnutá MAC vrstvou.

Data: jedno pole, které se nazývá PSDU – má proměnnou délku v závislosti na množství dat, které poskytne MAC vrstva.

Maximální velikost PPDU paketu je 136 bajtů.

2.6.2 Komunikace na fyzické vrstvě

Fyzická vrstva standardu 802.15.4 je zodpovědná za navázání bezdrátové radiofrekvenční (RF) linky mezi dvěma zařízeními. Zároveň poskytuje modulaci, demodulaci, synchronizaci mezi přijímačem a vysílačem, jak na úrovni frekvence, tak na úrovni paketu.
Jak již vyplývá z Tab. 1 tak standard specifikuje čtyři různé přenosové rychlosti, které mohou být použité ve třech pásmech. [1]

2.6.2.1 Komunikace v pásmu 868 a 915 MHz

Zařízení pracující v pásmech 868/915 MHz mohou komunikovat třemi různými přenosovými rychlostmi ve třech různých modulacích. Přičemž BPSK modulace tvoří jakýsi základ, tedy je vyžadována od všech zařízení a modulace Q-QPSK a PSSS jsou volitelné. [1]

BPSK modulace

Tato modulace tedy tvoří jakýsi základ v tomto komunikačním pásmu. BPSK modulace využívá techniku DSSS (přímé rozprostření spektra) poskytující přenosovou rychlost 20 kb/s v 868 MHz a 40 kb/s v 915 MHz pásmu.
Datové bity, tedy logické 0 a 1 jsou kódovány následujícím způsobem:

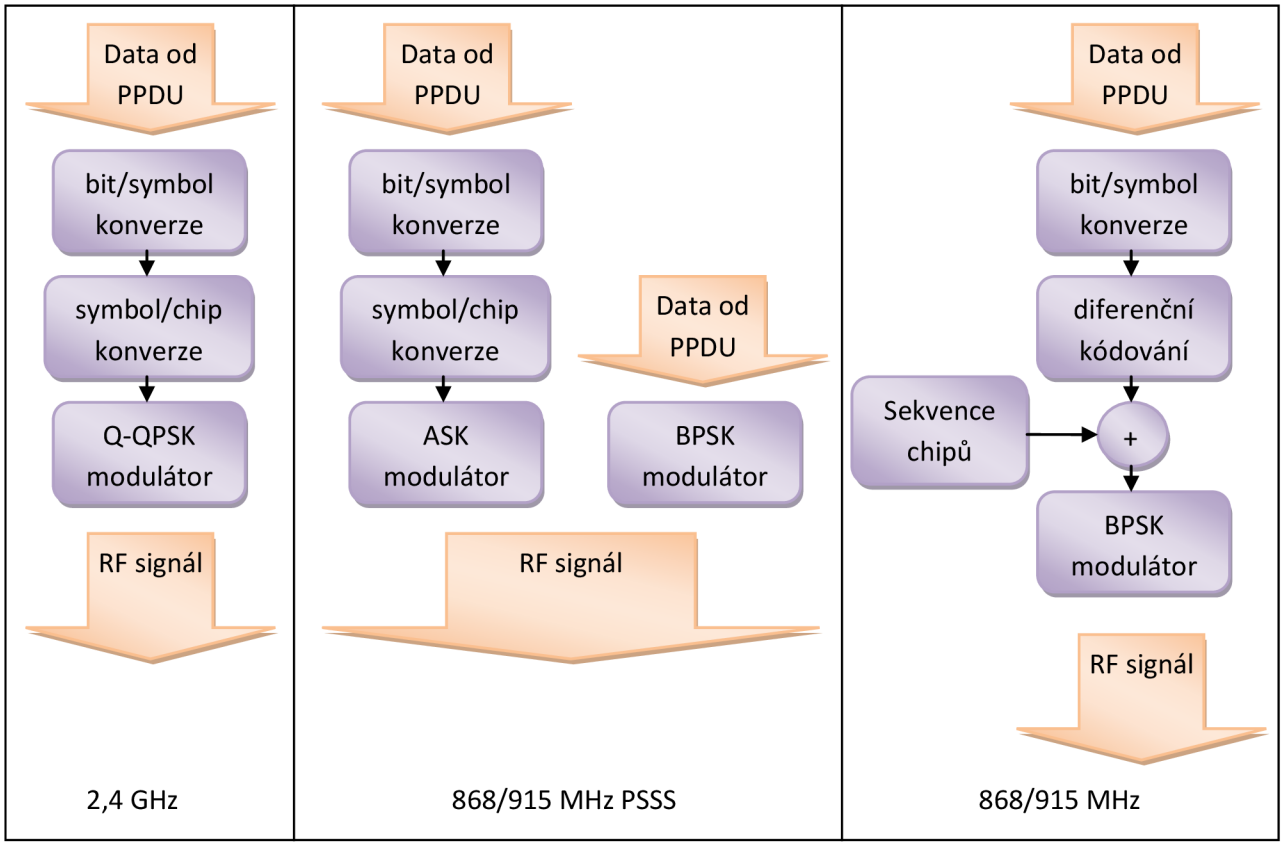
- pokud je datový bit 0, BPSK bit je modulován se stejnou fází jako předcházející BPSK bit,
- pokud je datový bit 1, BPSK bit je modulován s opačnou fází jako předcházející BPSK bit.

Q-QPSK modulace

V tomto pásmu je to volitelná modulace, která je odvozena z modulace používané v pásmu 2,4 GHz a poskytuje zde přenosovou rychlost 100 kb/s v 868 MHz a 250 kb/s v 915 MHz pásmu.
Využívá 16bitovou kvazi-ortogonální modulaci s 16čipovou pseudonáhodnou sekvencí k modulaci 4 datových bitů do každého symbolu.

PSSS modulace

Parallel Sequence Spread Spectrum používá tzv. multikódové modulační schéma, založené na amplitudovém klíčování (ASK). Mezi výhody této modulace patří, že má větší výkonnost oproti ostatním dvěma modulacím. Popis této modulace by vystačil na celou kapitolu, ovšem tímto se tato práce nezabývá, a proto je uvedeno na Obr. 9 pouze blokové schéma této modulace.



Obr. 9 Bloková schémata modulací

2.6.2.2 Komunikace v pásmu 2,4 GHz

V tomto pásmu se používá pouze jeden druh modulace (Q-QPSK) a jedna přenosová rychlost (250 kb/s).

2.6.3 Služby fyzické vrstvy

Fyzická vrstva tvoří rozhraní mezi přenosovým médiem a MAC vrstvou, přičemž k tomuto používá dva druhy služby:

- Datové služby fyzické vrstvy (PHY data services),
- Řídící služby fyzické vrstvy (PHY management services).

Tyto služby jsou dostupné prostřednictvím dvou přístupových bodů:

- PHY layer data service access point (PD-SAP),
- PHY layer management entity access point (PLME-SAP).

Standard definuje čtyři typy služeb primitiv:

- žádost (request): tento primitiv je směrován od volajícího (také uživatel) jako žádost iniciování služby
- indikace (indication): tento primitiv je směrován od služby uživateli jako indikace interní události

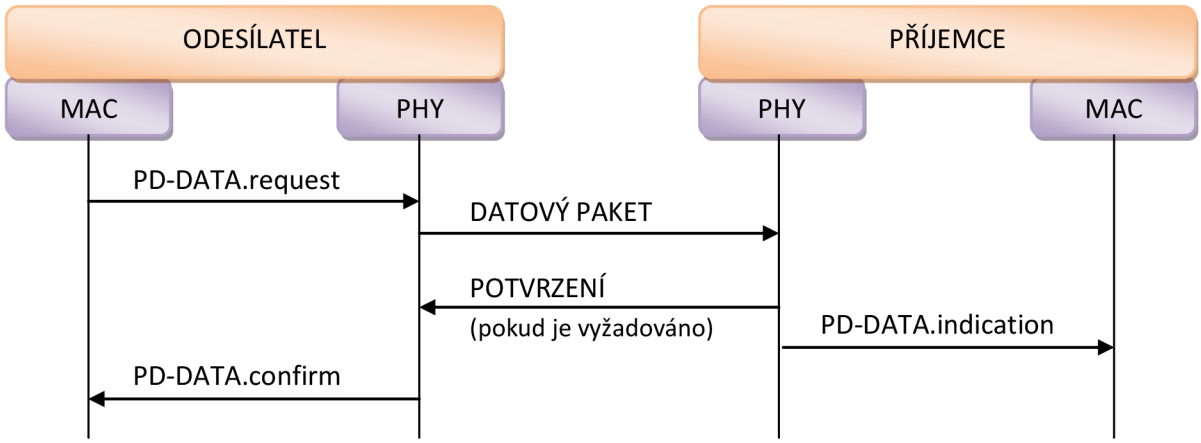
- odpověď (response): tento primitiv je směrován od uživatele ke službě ke kompletaci procedury vyvolané primitivem indikace
- potvrzení (confirm): tento primitiv je směrován od služby k uživateli k oznámení výsledků jednoho či více přidružených předchozích žádostí.

2.6.3.1 Datové služby fyzické vrstvy

Datové služby poskytují tři primitiva:

- PD-DATA.request
- PD-DATA.confirm
- PD-DATA.indication

Na obrázku níže je zobrazena komunikace mezi dvěma uzly v síti.



Obr. 10 Diagram komunikace uzlů s primitivy fyzické vrstvy

2.6.3.2 Řídící služby fyzické vrstvy

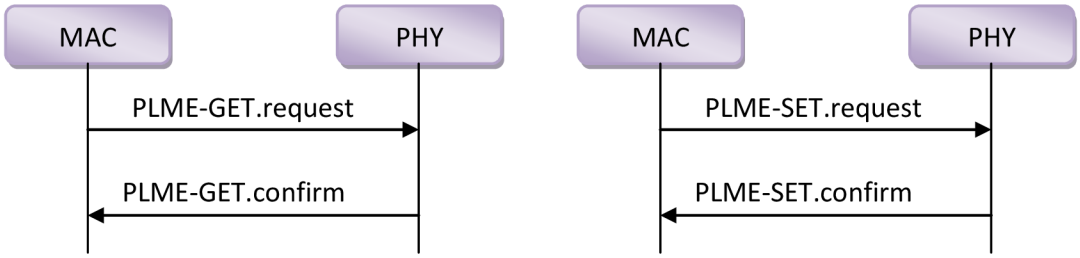
Řídící služby fyzické vrstvy poskytují kontrolu pro příkazy řízení komunikačních nastavení a funkčnost rádia.

Standard definuje pět PLME primitiv, jejichž popis je uveden v Tab. 2, přičemž každý primitiv má pouze dva typy služeb a to *request* a *confirm*.

Tab. 2 Primitiva řídících služeb fyzické vrstvy

Primitiv	Kategorie	Popis
PLME-GET	Nastavení komunikace	PIB řízení
PLME-SET		
PLME-SET-TRX-STATE	Ovládání rádia	zapnutí/vypnutí rádio
PLME-CCA	RF detekce	Detekce radiového signálu
PLME-ED		

PIB (PHY PAN Information Base) obsahuje konfigurační atributy pro řízení fyzické vrstvy. Tyto atributy mohou být zapsány, reps. přečteny primitivy PLME-SET, resp. PLME-GET, jak zobrazuje Obr. 11.



Obr. 11 Mechanismus výměny primitiv s PIB atributy

Primitiv PLME-SET-TRX-STATE slouží k vypnutí či zapnutí rádia. Účel tohoto primitiva je řídit rádiový modul a zajistit tak minimální odběr zařízení.

Primitiv PLME-CCA (Clear Channel Assessment), jak název napovídá, slouží ke zjištění volného kanálu pro přenos. Tedy před přenosem dat, MAC vrstva vyšle žádost (PLME-CCA.request) fyzické vrstvě, která zapne rádio, provede CCA měření a poté rádio vypne. Jakmile je toto měření dokončeno, tak MAC vrstvě je vysláno potvrzení (PLME-CCA.confirm) obsahující informace, zda je kanál volný či nikoliv.

Primitiv PLME-ED (Energy Detection) umožňuje zařízení provést radiového signálu v kanálu, ve kterém momentálně pracuje. MAC vrstva vyšle žádost (PLME-ED.request) a vrstva fyzická na ní odpoví (PLME-ED.confirm) obsahujíc v sobě informaci o úrovni radiového signálu, která může nabývat 256 úrovní (0 až 255).

2.7 MAC vrstva

MAC vrstva je druhou vrstvou ve standardu IEEE 802.15.4. Mezi klíčové úlohy této vrstvy patří přístup k médiu (standard využívá CSMA/CA), zapouzdření dat do rámců, adresace, zabezpečení, řízení sítě a PAN asociace/disasociace. [1]

Každé zařízení v síti má svou unikátní 64bitovou IEEE adresu. Tato adresa může být nahrazena kratší 16bitovou adresou přidělenou PAN koordinátorem, který zajistí, že tato adresa bude jedinečná v celé síti. Tento proces je řízen asociační procedurou, která je blíže popsána v kap. 2.7.3.

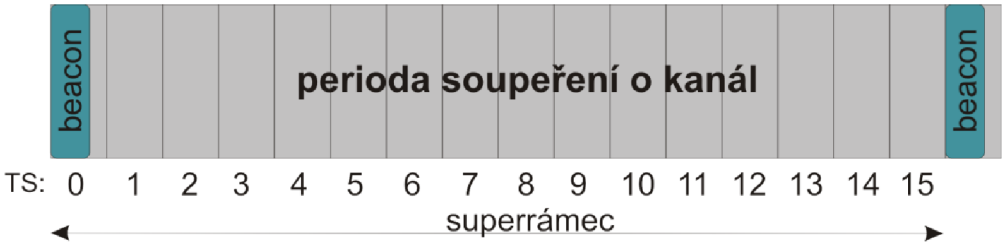
Jak již bylo popsáno v kap. 2.3, existují tedy dvě síťové topologie. V případě topologie hvězda je komunikace řízena jediným PAN koordinátorem, který funguje jako síťový master a který vysílá beacon rámce.

Jakékoliv FFD může vytvořit svou vlastní sít po tom, co se stane PAN koordinátorem. Každá síť v topologii hvězda funguje nezávisle na sousedních sítích. Během procesu vytváření nové „sítě hvězda“ si musí PAN koordinátor vybrat svůj identifikátor nazývaný PAN ID, který ale nesmí být používaný žádnou jinou okolní sítí. Toto je provedeno skenováním všech dostupných či vybraných kanálů existujících sítí a poté je vybráno takové PAN ID, které nebylo nalezeno během skenování. Po této proceduře může PAN koordinátor začít vysílat beacon rámce (blíže vysvětleno v kap. 2.6.2) v pravidelných intervalech a povolit zařízením, žádajícím o připojení, ke spojení.
Byly zde zmíněny tzv. beacon rámce. Tedy z pohledu beacon rámců standard definuje dva typy sítí:

- beacon-enable: PAN koordinátor vysílá v pravidelných intervalech beacon rámce, které jsou zachytávány FFD či RFD zařízením. Vyšší vrstvy v těchto zařízeních rozhodnou, zda se připojí do sítě a to tak, že vyšlou požadavek na asociaci. Naopak PAN koordinátor rozhodne, zda toto zařízení připojí či nikoliv.

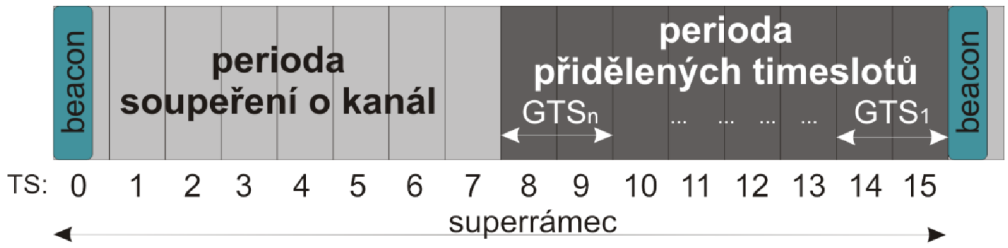
- non-beacon-enable: PAN koordinátor používá beacon rámce pouze k účelům asociace. Synchronizace zařízení je dosažena tak, že PAN koordinátor vysílá žádosti na data v pravidelných intervalech.

Jak bylo řečeno, tak beacon rámce jsou vysílány v pravidelných intervalech. Tento interval, tedy čas mezi dvěma beacon rámci, se nazývá superrámec (superframe). Je rozdělen do 16 timeslotů (TS), přičemž první timeslot TS0 začíná beacon rámcem (první timeslot se nerovná beacon rámec – pouze je v něm obsažen) a končí 15. Timeslotem – TS15. Zařízení, které chce komunikovat s PAN koordinátorem musí toto vykonat v čase mezi dvěma beacon rámci, což vyplývá z metody přístupu ke sdílenému médiu, viz kap. 2.5. Tato perioda se nazývá CAP (Contention Access Period) – perioda soupeření o kanál – viz Obr. 12.



Obr. 12 Struktura superrámce

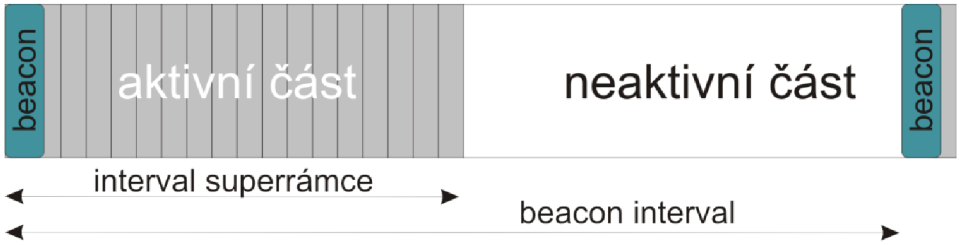
PAN koordinátor může na žádost přidělit určitou část superrámce nějakému zařízení, ve kterém bude pouze ono komunikovat a nemusí tak soutěžit o volný kanál. Takovýto časový segment se nazývá GTS (Guaranteed Time Slot). Jednotlivé GTS jsou složeny z jednoho či více timeslotů a jsou soustředěny ke konci superrámce, tedy před příchodem dalšího beacon rámce, viz Obr. 13.



Obr. 13 Struktura superrámce s GTS

Perioda přidělených timeslotů se nazývá CFP (Content Free Period). Jelikož CFP může zabrat významnou část superrámce, standard vyžaduje, aby délka CAP byla minimálně 440 symbolů a mohla tak komunikovat i zařízení, jenž nemají přiřazena žádný GTS.

Přiřazení GTS slotů je tedy plně v kompetenci PAN koordinátora a používá se to především u aplikací vyžadující nízkou odezvu či určitou šířku pásma. Ovšem u aplikací, které nemají tyto kritéria, superrámec může být jednoduše rozdělen na dvě části – aktivní a neaktivní. Aktivní část je složena z 16 timeslotů, které dohromady tvoří superrámec. Neaktivní část neslouží v podstatě k ničemu a během této doby nelze komunikovat. Délka těchto částí může, ale nemusí být stejná, přičemž délka



Obr. 14 Struktura superrámce s aktivní a neaktivní částí

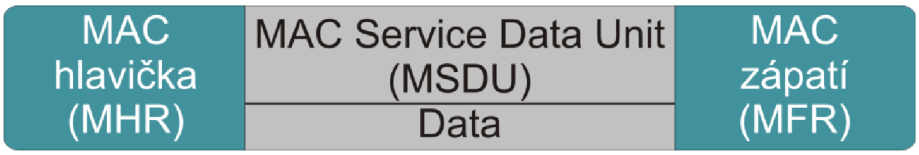
aktivní částí tvoří interval superrámce a interval mezi beacon rámci je beacon interval, viz Obr. 14. Takovýto mechanismus umožňuje zařízení redukovat komunikační dobu a zvýšit tak výdrž zařízení, ovšem za cenu vetší odezvy a menší přenosové rychlosti.

2.7.1 Datová jednotka MAC vrstvy

Datová jednotka této vrstvy se nazývá MPDU (MAC Protocol Data Unit), který je navržen tak, aby odrážel jednoduchost a flexibilitu protokolu. [2]

MPDU jednotka se skládá ze tří částí:

- hlavička (MHR),
- datová část (MSDU),
- zápatí (MFR).



Obr. 15 Struktura rámce MAC vrstvy

Hlavička (MHR): skládá se ze čtyř polí – kontrolní, sekvenční číslo, adresy a zabezpečení (volitelné).
Kontrolní pole specifikuje typ rámce, formát a obsah pole adresy. Dále se v něm také indikuje, zda je vyžadováno potvrzení od příjemce.
Pole *sekvenční číslo* obsahuje identifikátor pořadí rámce. Je inkrementováno v každém následujícím rámci.
Pole *adresy* obsahuje zdrojovou a cílovou adresu uzlu popřípadě PAN ID (viz Obr. 17), což je právě blíže určeno v kontrolním poli.
Pokud je vyžadována bezpečnost dat, tak hlavička obsahuje pole *zabezpečení*, jehož velikost je variabilní, v závislosti na úrovni zabezpečení.

Datová část (MSDU): obsahuje užitečná data.

Zápatí (MFR): obsahuje 16bitové FCS číslo (Frame Check Sequence), založené na CRC (cyklická redundantní kontrola), které slouží ke kontrole správného příjmu dat.

Standard definuje čtyři typy MAC rámců, přičemž maximální délka každého rámce je 127 bajtů a maximální velikost MSDU je 118 bajtů.

Beacon rámec

Přenos beacon rámce (beacon frame) probíhá pouze směrem k FFD zařízení, přičemž nezáleží na topologii sítě. Beacon rámec slouží k přenosu informací o síti a synchronizaci.

Hlavička (MHR)				Data (MSDU)				Zápatí MFR
2B	1B	4/10B	0-14B	2B	proměnná	proměnná	proměnná	2B
kontrolní pole	sekv. číslo	adresy	zabez-pečení	specifikace superrámce	GTS	očekávaná adresa	data	FCS

Obr. 16 Struktura beacon rámce

Datový rámec

Přenos datového rámce (data frame) probíhá mezi všemi zařízeními v síti, opět nezáležíc na topologii. Jak název napovídá, tak slouží k přenosu užitečných dat.

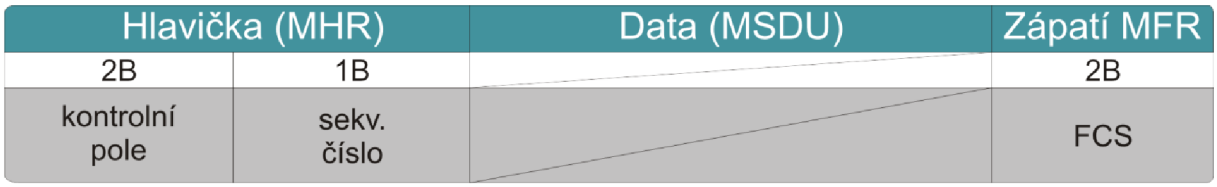
Hlavička (MHR)				Data (MSDU)	Zápatí MFR
2B	1B	4-20B	0-14B	proměnná	2B
kontrolní pole	sekv. číslo	adresy	zabez-pečení	data	FCS

0/2B	0/2/8B	0/2B	0/2/8B
PAN ID příjemce	adresa příjemce	PAN ID odesílatele	adresa odesílatele

Obr. 17 Struktura datového rámce

Potvrzovací rámec

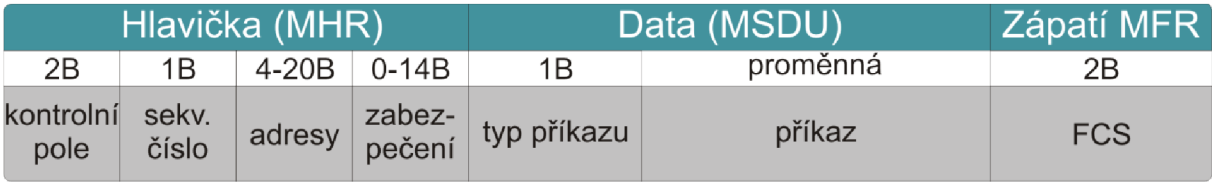
Přenos potvrzovacího rámce (acknowledgment frame) opět probíhá mezi všemi zařízeními v síti. Používá se k potvrzení přijatých dat. Rámec neobsahuje MSDU.



Obr. 18 Struktura potvrzovacího rámce

Příkazový rámec

Přenos příkazového rámce (MAC command frame) probíhá mezi všemi zařízeními v síti. Slouží k řízení sítě, přičemž k tomu využívá předem definovaných 9 příkazů: *Association request, Association response, Disassociation notification, Data request, PAN ID conflict notification, Orphan notification, Beacon request, Coordinator realignment, GTS request*.



Obr. 19 Struktura příkazového rámce

2.7.2 Modely přenosu dat

V sítích pracujících na Std IEEE 802.15.4 existují 3 typy přenosu dat, přičemž ještě záleží na topologii sítě. [2]

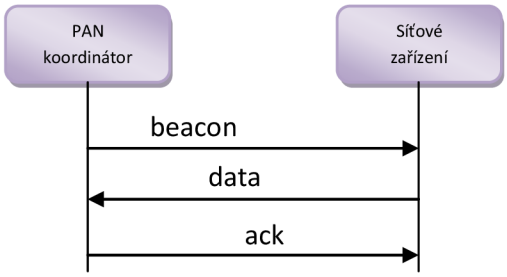
Přenos dat ke koordinátorovi

K tomuto dochází u topologie hvězda. V beacon-enabled síti zařízení, které data vysílá koordinátorovi, musí být synchronizováno pomocí beacon rámců. Pokud zařízení má přiřazen GTS slot, počká si tedy na tento svůj slot a vyšle data bez použití CSMA. Jinak jsou data přenesena v CAP čase, tak jak definuje metoda CSMA/CA. Po přijetí dat koordinátorem, tento vyšle potvrzovací rámec (ACK) zařízení jako signál, že přenos dat je kompletní.

V non-beacon-enabled síti zařízení jednoduše zkontroluje volnost kanálu a pokud je kanál volný, tak vyšle data a poté ACK rámec.

Mechanismus přenosu dat pro beacon-enabled síť ilustruje Obr. 20. Pro non-beacon-enabled síť je totožný s tím rozdílem, že v této není vysílán beacon rámec.

Standard nazývá tento proces přímým přenosem dat.



Obr. 20 Přenos dat ke koordinátorovi

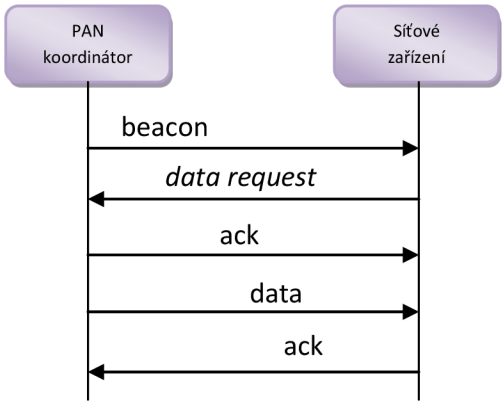
Přenos dat od koordinátora

K tomuto dochází u topologie hvězda. V beacon-enabled síti koordinátor, který chce vyslat data zařízení, musí nejprve vyslat adresu příjemce v poli beacon rámce – očekávaná adresa. Toto indikuje příjemci, že se má připravit na příjem dat od koordinátora. Jakmile zařízení přijme beacon rámec a v poli si přečte svou adresu, vyšle koordinátorovi příkazový rámec s příkazem *Data request*, načež koordinátor odpoví ACK rámcem následovaným datovým rámcem. Přenos je ukončen, jakmile koordinátor přijme ACK rámec od zařízení.

V non-beacon-enabled síti je tato procedura odlišná, jelikož se zde nenacházejí beacon rámce, ve kterých právě se adresuje onen příjemce. V těchto sítích se tedy zařízení musí periodicky dotazovat koordinátora, zda pro něj nemá data.

Mechanismus přenosu dat pro beacon-enabled síť ilustruje Obr. 21. Pro non-beacon-enabled síť je totožný s tím rozdílem, že v této není vysílán beacon rámec.

Standard nazývá tento proces nepřímým přenosem dat.



Obr. 21 Přenos dat od koordinátora

Přenos dat mezi zařízeními

Je možný pouze u topologie peer-to-peer. Zařízení mohou kdykoliv vysílat data pomocí neslotové metody CSMA/CA. Toto samozřejmě přináší s sebou problémy, protože přenos není synchronizován a může docházet ke kolizím. Synchronizace může být zajištěna vyššími vrstvami.

2.7.3 Služby MAC vrstvy

MAC vrstva tvoří rozhraní mezi vrstvou fyzickou a vrstvami vyššími, které standard už ovšem nedefinuje. Ke komunikaci mezi vrstvami využívá dva druhy služeb [2] :

- Datové služby MAC vrstvy (MAC data services),
- Řídící služby MAC vrstvy (MAC management service).

Tyto služby jsou dostupné prostřednictvím dvou přístupových bodů:

- MAC common part sublayer service access point (MCPS-SAP),
- MAC management service access point (MLME-SAP).

2.7.3.1 Datové služby MAC vrstvy

Datové služby poskytují tři primitiva:

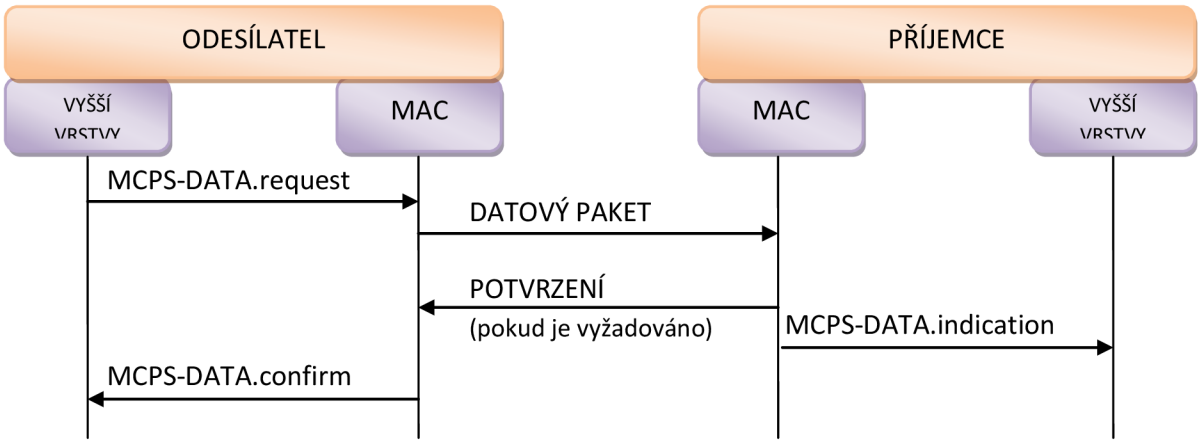
- MCPS-DATA.request
- MCPS-DATA.confirm
- MCPS-DATA.indication

Na Obr. 22 je zobrazena přenos data mezi dvěma zařízeními z pohledu MAC vrstvy.

2.7.3.2 Řídící služby MAC vrstvy

Řídící služby MAC vrstvy poskytují příkazy pro kontrolu komunikačních nastavení, rádia a funkčnost sítě.

Standard definuje patnáct MLME primitiv, jež jsou popsány v Tab. 3.



Obr. 22 Diagram komunikace uzlů s primitivy MAC vrstvy

Tab. 3 Primitiva řídících služeb MAC vrstvy

Primitiv	Kategorie	Popis	request	confirm	response	indication
MLME-GET	nastavení komunikace	nastavení MAC PAN	•	•		
MLME-SET			•	•		
MLME-RESET			•	•		
MLME-RX-ENABLE	ovládání rádia	vyp./zap. rádio	•	•		
MLME-SCAN		skenování kanálů	•	•		
MLME-ASSOCIATE	správa sítě	asociační proces s koordinátorem	•	•	•	•
MLME-DISASSOCIATE			•	•		•
MLME-GTS		správa GTS	•	•		•
MLME-ORPHAN		správa sirotka			•	•
MLME-SYNC		synchronizace s koordinátorem	•			
MLME-SYNC-LOSS						•
MLME-START		správa beacon	•	•		
MLME-BEACON-NOTIFY						•
MLME-POLL		synchronizace v non-beacon-enabled síti	•	•		
MLME-COMM-STATUS		stav komunikace				•

Nastavení MAC PAN – konfigurační atributy pro řízení MAC vrstvy

Ovládání rádia – slouží k vypnutí či zapnutí rádia

Skenování kanálů – 4 druhy skenování:

- Energy detection scan: používán PAN koordinátorem ke zjištění vhodného kanálu při vytváření sítě.
- Active channel scan (aktivní skenování): toto skenování hledá PAN koordinátory nebo koordinátory v síti. Zařízení prvně vyšle příkazový rámec (viz kap. 2.7.1) s příkazem *beacon request,* který řekne koordinátorům, aby vyslali beacon rámec. Jestliže se jedná o síť becon-enabled, tak beacon rámec je vyslán v dalším slotu. Pokud je síť non-beacon-enabled, becon rámec je vyslán neslotovanou CSMA metodou.
- Passive channel scan (pasivní skenování): stejný jako aktiv, ovšem zde není vyslán příkazový rámec.
- Orphan channel scan: umožňuje sirotkovi (zařízení, které ztratilo spojení se svým koordinátorem) nalézt koordinátora.

Asociace a disasociace – před připojením nového uzlu do sítě, musí tento projít procesem asociace. Tento proces je odlišný v beacon-enabled a non-beacon-enabled síti, a proto jsou zde popsány postupy pro obě tyto sítě:

- asociace v beacon-enabled síti: zařízení provádí pasivní skenování, jehož výstupem jsou přijaté beacon rámce, které jsou předány vyšší vrstvě, a ta rozhodne, ke komu se připojí. Pomocí přijatých beacon rámců se toto zařízení synchronizuje s vybraným koordinátorem pomocí primitivu MLME-SYNC.request a poté požádá o asociaci pomocí MLME-ASSOCIATE.request. Koordinátor po přijmutí této žádosti odešle potvrzovací rámec ACK o přijmutí žádosti jako takové, ne jako odpověď na asociaci. ACK není povinný ovšem pro tuto proceduru ano. Koordinátor může schválit či odmítnout asociaci, což už záleží na požadavcích dané aplikace případně kapacitě sítě, pomocí zprávy MLME-ASSOCIATE.response s odpovídajícími parametry.

- asociace v non-beacon-enabled síti: zařízení provádí aktivní skenování, jehož výstupem jsou přijaté beacon rámce, které jsou předány vyšší vrstvě, a ta rozhodne, ke komu se připojí. Jelikož se v této síti nepoužívají beacon rámce pro synchronizaci, je vyslána žádost o asociaci MLME-ASSOCIATE-request s odpovídajícími parametry. Takto musí koordinátor odpovědět během doby *macResponseWaitTime.*

Pokud koordinátor asociuje zařízení, je tomuto přidělena 16bitová adresa ve zprávě MLME-ASSOCIATE.response. Jestliže zařízení nepožaduje tuto zkrácenou adresu, musí používat rozšířenou vlastní 64bitovou adresu.

Disasociaci může iniciovat jak koordinátor, tak síťové zařízení pomocí primitivu MLME-DISSASSOCIATE. Vyšší vrstva tedy požádá MAC vrstvu o disasociaci pomocí MLME-DISASSOCIATE.request, která vyšle příkazový rámec s příkazem *Disassociation notification.* Přijde-li ACK rámec od příjemce zprávy, tak MAC vrstva indikuje o disasociaci vrstvu vyšší pomocí primitivu MLME-DISASSOCIATE.confirm.

Správa GTS – alokaci, realokaci a dealokaci GTS poskytuje primitiv MLME-GTS. O správu se stará pouze PAN koordinátor, který kontroluje, kolik z 16 timeslotů je zahrnuto do CFP periody (zbytek je v CAP). Může alokovat až 7 GTS slotů. O alokaci GTS slotů žádá pouze síťové zařízení, kdežto o dealokaci může požádat jak síťové zařízení, tak PAN koordinátor.

Správa sirotka – pokud síťové zařízení ztratí spojení s PAN koordinátorem, provede orphan channel scan, během něhož vysílá příkazové rámce s příkazem *Orphan notification* na každém z dostupných kanálů. Po přijmutí této zprávy koordinátorem MAC vrstvou, předá pomocí primitivu MLME-ORPHAN.indication vrstvě vyšší, která zjistí, zda toto zařízení nebylo již asociováno. Pokud bylo, předá se MAC vrstvě primitivit MLME-ORPHAN.response, která vyšle zařízení příkazový rámec s příkazem *Coordinator realignment.* Pokud nebylo dříve asociováno, neprovede se nic.

Synchronizace s koordinátorem – pro beacon-enabled síť. Primitiv MLME-SYNC umožňuje zařízení vyhledat beacon rámce. Tento proces je iniciován primitivem MLME-SYNC.request, jehož výsledkem je aktivace rádia, načež se čeká na přijmutí beacon rámce od koordinátora. Pokud zařízení ztratí

synchronizaci s koordinátorem, MAC vrstva předá zprávu vrstvě vyšší pomocí MLME-SYNC-LOSS.indication.

Správa beacon – generování beacon rámců je iniciováno primitivem MLME-START, jehož parametry umožňují nastavení zařízení jako koordinátora, výběr kanálu, nastavení beacon intervalu a superrámce. Vyšší vrstva předá MAC vrstvě zprávu MLME-START.request, načež odpoví pomocí MLME-START.confirm. Pokud zařízení přijme beacon rámec obsahující PAN ID sítě, ke které je asociován, tento je předán dále ke zpracování. Jestli beacon rámec obsahuje jeden nebo více bajtů v poli *data,* MAC vrstva předá vrstvě vyšší zprávu MLME-BEACON-NOTIFY.indication.

Synchronizace v non-beacon-enabled síti – jak je napsáno v kap. 2.7.2, tak zařízení se musí periodicky dotazovat, zda pro něj nemá koordinátor data. Toto je iniciováno primitivem MLME-POLL.request, načež je vyslán příkazový rámec s příkazem *Data request*. Koordinátor odpoví potvrzovacím rámcem, v jehož kontrolním poli je nastavena tvz. *flag pending* na *0* (*FP=0*), což znamená, že žádná data pro něj nemá. MAC vrstva předá toto v primitivu MLME-POLL.confirm. Pokud by koordinátor měl data, tak pošle ACK rámec s *FP=1* a hned poté vyšle datový rámec.

Stav komunikace – primitiv MLME-COMM-STATUS.indication je předán vyšší vrstvě po přenose iniciovaným služebním primitivem response nebo pokud se přijala data, která neprošla zabezpečovacím procesem.

2.7.4 Zabezpečení

Jak vyplývá ze struktury MAC rámce, tak lze vidět, že hlavička obsahuje pole zabezpečení. [1] Standard poskytuje tři hlavní pilíře zabezpečení:

Diskrétnost dat je provedena šifrováním dat symetrickou šifrou, tedy tentýž klíč je použit k šifrování i dešifrování dat v tzv. plaintextu. Zařízení, které tento klíč nezná, nemůže provést dešifrování. Standard definuje šifrování pole *data* v beacon rámci, příkazovém rámci a datovém rámci. Další data, jako jsou data v potvrzovacím rámci, hlavička a zápatí rámců, šifrována nejsou.

Autenticita dat, také nazývána integrita dat, je služba, která umožňuje příjemci detekovat modifikaci zprávy, připojením tzv. message integrity code (MIC) ke zprávě. Standard definuje kontrolu integrity pro MAC hlavičku, pole zabezpečení a nezabezpečené pole dat u beacon, datového a příkazového rámce.

Aby mohla být vysvětlena následující položka, tak je třeba si říci, že v případě, kdy jsou data šifrována, je pole dat v rámci rozděleno ještě na tři pole a to čítač rámce, čítač klíče (což jsou v podstatě sekvenční čísla) a samotná zašifrovaná data. Oba čítače se inkrementují s každým odeslaným rámce.

Ochrana proti přehrání je v podstatě zajištěna čítačem rámce a klíče. Např. pokud by útočník odposlechl data a za nějaký čas je vyslal znovu, tak příjemce toto odmítne, jelikož se nebude shodovat sekvenční číslo, které by mělo být vyšší, než předcházející, což nebude.

Ani ne tak diskrétnost, jako spíše právě autentičnost dat je druh zabezpečení, nejvíce používaný v aplikacích pracujících ve standardu 802.15.4. Mnohem důležitější je zajistit, aby zpráva byla

autentická, tedy aby zdroj dat byl známý a důvěryhodný a jakékoliv ovlivnění nebo modifikace zprávy mohlo být detekováno.

Vrstva MAC poskytuje 8 úrovní zabezpečení, které jsou rozděleny na dva bezpečnostní módy – zabezpečený a nezabezpečený.

Nezabezpečený mód

Jak název napovídá, tak data nejsou nijak šifrována ani není zajištěna autenticita dat. Tento mód je vhodný pro aplikace, u kterých je důležitá cena a na bezpečnost se neklade důraz.

Je to úroveň zabezpečení 0, kdy data nejsou nijak šifrována, není zajištěna jejich integrita ani ochrana proti přehrání.

Rámec neobsahuje pole *zabezpečení* v hlavičce.

Zabezpečený mód

Patří sem úrovně zabezpečení 1 až 7. Pro šifrování dat je použit standard AES s délkou klíče 128 bitů. Délka kódu integrity dat může být 32 (MIC-32), 64 (MIC-64) nebo 128 (MIC-128) bitů.

Rámec obsahuje pole *zabezpečení* v hlavičce. Toto pole se dále skládá ze tří částí – Security control (určuje úroveň zabezpečení), Frame counter („sekvenční číslo“ – viz Ochrana proti přehrání), Key identifier (identifikuje druh klíče).

Možné úrovně zabezpečení jsou popsány v Tab. 4.

Tab. 4 Úrovně zabezpečení v IEEE 802.15.4

Úroveň zabezp.	Atributy zabezpečení	Diskrétnost dat	Integrita dat	Ochrana proti přehrání
0	Žádný	✗	✗	✗
1	MIC-32	✗	MIC-32	✓
2	MIC-64	✗	MIC-64	✓
3	MIC-128	✗	MIC-128	✓
4	ENC	✓	✗	✗
5	ENC-MIC-32	✓	MIC-32	✓
6	ENC-MIC-64	✓	MIC-64	✓
7	ENC-MIC-128	✓	MIC-128	✓

3 Možné realizace paketového analyzátoru

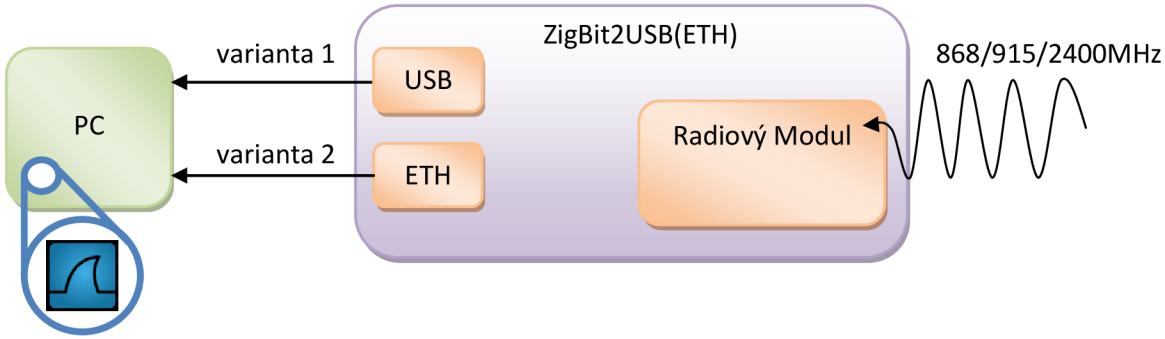
Návrh paketového analyzátoru probíhal prozkoumáním různých variant řešení, ale z hlediska ceny, dostupnosti a složitosti byly vybrány následující varianty:

1) Rádiový modul Zigbit + deska Zigbit2USB
2) Rádiový modul Zigbit + deska Zigbit2Ethernet
3) Atmel AVR RZ USBstick
4) Atmel RZ600 + Atmel STK600 nebo Ethernut Reference Board (Ethernut 1)

Všechny výše uvedené varianty mají společného činitele, a sice program Wireshark, na němž bude probíhat sběr dat. Jednak obsahuje podporu pro protokol IEEE 802.15.4 a především je hojně rozšířený, volně dostupný a uživatelsky modifikovatelný různými doplňky.

3.1 Varianta 1 + 2

Tato varianta je kombinací rádiového modulu Zigbit [3] a desky ZigBit2USB(ETH), ke které tento modul bude připojen – viz Obr. 22.



Obr. 23 Nákres řešení varianty 1 a 2

3.1.1 Radiový modul ZigBit

Rádiové moduly určené pro standard IEEE 802.15.4. vyrábí celá řada firem, např. Atmel, Texas Instruments, Freescale Semicondutor, Radiocrafts a další. Pro realizaci byl vybrán modul od firmy Atmel. Jejich komerční název zní Zigbit moduly (Zigbit Wireless modules). Atmel vyrábí celkem 5 typů modulů rozdělených do tří kategorií podle frekvence, na které pracují.

Obr. 24 Zigbit moduly s čipovou anténou (vlevo) a bez antény (vpravo)

Název „modul“ mají tyto zařízení z toho důvodu, že v sobě integrují mikrokontrolér a RF přijímač/vysílač (dále jen RF modul; nezaměňovat pojmy RF modul a rádiový modul!) s příslušnými pasivními součástkami umístěný ve stínícím kovovém krytu.

Samozřejmě tento rádiový modul neumí sám o sobě nic, a proto je potřeba do něj nahrát příslušný software, v tomto případě se mluví spíše o firmwaru. Existují různé projekty či pracovní skupiny, zabývající se tvorbou firmwaru pro tyto účely, a cílem bude vybrat ten nejvhodnější.
Rádiový modul obsahuje dva integrované obvody – mikrokontrolér a RF modul.

Mikrokontrolér ATmega1281 – nízkopříkonový CMOS 8bitový mikrokontrolér založený na AVR architektuře. Obsahuje 7 vstupně-výstupních 8bitových portů a dvě USART rozhraní. Z hlediska modulu jsou ovšem důležité především funkce těchto portů. Na portu B se například nachází rozhraní SPI, přes které komunikuje s RF přijímačem/vysílačem a přes nějž se může programovat MCU. Dále jsou zde dvě USART rozhraní, jedno I2C, JTAG IRQ a analogové vstupy do AD převodníku. [4]

RF modul – rádiový přijímač/vysílač. Existují dva typy pro různá frekvenční pásma. Jsou to AT86RF230 pro pásmo 2,4 GHz a AT86RF212 pro pásmo 868/915 MHz. Liší se od sebe nejen komunikační frekvencí, ale také spotřebou a citlivostí. Spotřeba je v této aplikaci nevýznamná, jelikož analyzátor nebude napájen z baterií, ale přímo z PC, ke kterému bude připojen. Právě citlivost je hlavním faktorem, proč bylo k těmto RF modulům, potažmo rádiovým modulům, přihlédnuto. Ze všech dostupných variant RF modulů na trhu patří právě tyto mezi nejcitlivější, což je pro aplikaci analyzátoru velmi vhodné. [5] [6]

3.1.2 Deska ZigBit2USB(ETH)

Tato deska slouží v podstatě k připojení rádiového modulu k počítači, na kterém bude probíhat sběr dat.

Je uvažováno nad dvěma typy připojení, a sice přes USB nebo přes ethernet.

3.1.2.1 Varianta s USB

Deska obsahuje jen velmi málo součástek. Jelikož data ze Zigbit modulu jsou posílána sériovou linkou UART, lze v nejjednodušším případě použít převodník napětí úrovní mezi UART mikrokontroléru a rozhraním RS232 počítače. Ovšem toto rozhraní se dnes už běžně nevyskytuje v počítačích, a proto bylo zvoleno USB.

Nejdůležitějším prvkem desky je převodník UART/USB tvořený obvodem FT232R. Tento obvod má také vyvedeny všechny piny CBUS, což jsou konfigurovatelné I/O piny, na konektor.
Obvod FT232R jednak přizpůsobuje napěťové úrovně, jelikož signály na USB mají amplitudu 5V, zatímco na UART modulu 3,3V, a také komunikační protokol, který je na těchto sběrnicích odlišný.

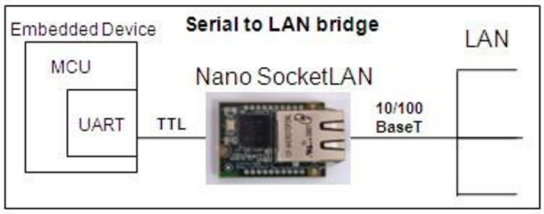
Při použití USB a Wiresharku je potřeba spolupráce programu, který bude přeposílat data z USB do Wiresharku, jelikož tento umí standardně zachytávat pouze na síťových rozhraních. Proto bylo uvažováno na možností využití ethernetu.

3.1.2.2 Varianta s ethernetem

Nevýhody použití USB pro zasílání dat do PC byly vysvětleny v předcházející kapitole. Proto byla navrhnuta možnosti zasílat zachycená data prostřednictvím technologie Ethernet, resp. přes síťové rozhraní. Jelikož Wireshark ve výchozím stavu a bez nějakého komplikovaného nastavování umožňuje zachytávat data právě na síťových rozhraních počítače, je záhodno tohoto využít.

K realizaci tohoto rozhraní by byl využit modul Nano SocketLAN od firmy Connect One Ltd.

Tento modul obsahuje mikrokontrolér, který je již z výroby naprogramován, aby plnil určité funkce. Tyto funkce se poté nastavují jednoduše pomocí uživatelského rozhraní. Modul může pracovat ve třech módech, přičemž v tomto případě je důležitý mód SerialNET – Serial to LAN bridge. Tento mód umožňuje transparentní přenos sériové linky, v tomto případě UART, přes LAN síť – viz Obr. 25. [7]



Obr. 25 Blokové schéma SerialNET módu (převzato z [7])

Jak je vidět na obrázku výše, tak vše důležité obstará právě tento modul. Stačí tedy pouze připojit piny UART ze Zigbit modulu na příslušné piny Nano SocketLAN modulu.

Jistým problémem by mohl být fakt, že data by samozřejmě byla zabalena do Ethernet rámců. Wireshark sice hierarchicky zobrazuje data ze všech vrstev (dle modelu OSI), tedy Ethernet rámce dekóduje bez problému. Jak již bylo zmíněno, Wireshark obsahuje podporu pro standard IEEE 802.15.4, ale zda by tyto data dekódoval korektně, není prozatím jasné.

3.2 Varianta 3

Tato varianta uvažuje využití modulu RZUSBstick [3] od firmy Atmel. Tento modul obsahuje mikrokontrolér AT90USB1287 a RF modul AT86RF230. Označení RF modulu tedy říká, že tento modul

Obr. 26 RZUSBstick

pracuje na frekvenci pouze 2,4 GHz, což by samozřejmě znemožňovalo využití v sítích 868/915 MHz. Komunikační rozhraní zde tvoří USB, jehož výhody/nevýhody použití s programem Wireshark byly již popsány v kap. 3.1.2.1.

3.3 Varianta 4

Tato varianta uvažuje využití kitu RZ600 od firmy Atmel [3] a to buď s vývojovým modulem STK600 firmy Atmel nebo modulem Ethernut 1 [8], což je zařízení vyvíjené skupinou v Německu.

Sada RZ600 obsahuje 6 rádiových modulů a 2 USB adaptéry. Každý jeden pár modulů se liší v RF modulu, který obsahují. Přičemž jsou zde 3 RF moduly – AT86RF212 (868/915 MHz), AT86RF230 (2,4 GHz) a AT86RF231 (2,4 GHz). Všechny tyto rádiové moduly mají konektor pro připojení antény a konektor, na který je vyvedeno SPI rozhraní. Tyto moduly jsou tedy v podstatě pouze jakési převodníky SPI – IEEE 802.15.4. Právě díky tomu, že jako komunikační rozhraní je použito SPI, lze je připojit k jakémukoliv zařízení obsahující toto rozhraní.

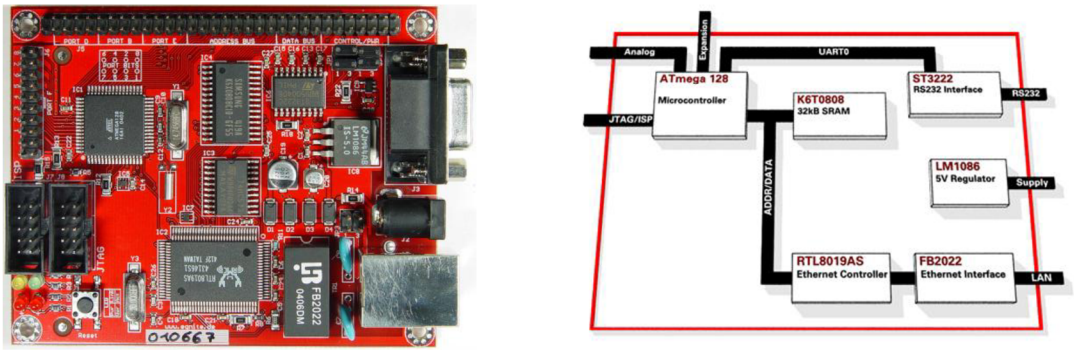
Sada mimoto obsahuje ještě 2 USB adaptéry, obsahující 32bitový mikrokontrolér AT32UC3A3256 a konektor USB. Kombinace tohoto USB adaptéru a RF modulu v podstatě tvoří RZUSBstick rozebraný v kap. 3.2., ale v tomto případě, lze komunikovat ve všech frekvenčních pásmech pro WSN.

Obr. 27 Atmel RZ600

Opět se zde dostáváme k USB, a proto byla zvolena ještě možnost připojit RF modul k jedné ze dvou zmíněných vývojových desek. Existuje tato možnost, například pokud některou z uvedených desek již uživatel vlastní, ale při použití pouze jako paketový analyzátor by to v konečném důsledku prodražilo tuto aplikaci a to celkem zbytečně. Proto je zde pouze uvedeno, že tato možnost existuje.

Tyto desky byly zvoleny z toho důvodu, že obsahují rozhraní SPI vyvedené na konektor kompatibilní s RF moduly. Samozřejmě takovýchto vývojových desek existují kvanta, zde je to zmíněno pouze jako jedna z možností.

Deska Ethernut 1 obsahuje rozhraní Ethernet, RS232, JTAG/ISP a analogové. V tomto případě jsou důležité především ISP (což je v postatě připojeno na SPI rozhraní) a RS232, potažmo Ethernet. Modul obsahuje mikrokontrolér ATmega 128 a další příslušné obvody – viz Obr. 28.



Obr. 28 Fotografie Ethernut 1 (vlevo) a blokové schéma (vpravo)

Vývojová sada STK600 je určena pro mikrokontroléry z rodiny AVR a AVR32. Lze k němu připojit jakýkoliv mikrokontrolér, přičemž každý pin takového MCU je vyveden na konektor, který lze soustředit do jednoho místa a může sloužit jako SPI rozhraní.

Obr. 29 Fotografie vývojové kitu STK600

4 Návrh paketového analyzátoru

4.1 Výběr platformy

V kapitole 3 byly rozebrány možné způsoby realizace analyzátoru, ze kterých byla vybrána varianta *Radiový modul Zigbit + deska ZigBit2USB*. K této variantě bylo přihlédnuto z následujících důvodu:

- modifikovatelnost: lze vyměnit moduly ZigBit a možnost analyzovat tak více frekvenčních pásem (868/915MHz i 2,4GHz) aniž by se muselo měnit celé zařízení,
- funkce jako vývojová platforma: podrobněji popsáno v kap. 4.3,
- jednoduchost: k PC se připojuje přes sériový port, který má nejjednodušší protokol a je podporován každým systémem,
- cena: jedná se o nejlevnější variantu ze všech představených.

Analyzátor je tedy tvořen modulem ZigBit a základní deskou, k níž se modul připojuje a která slouží k připojení k PC. Tato základní deska je dále nazývána jako deska ZigBit2USB.

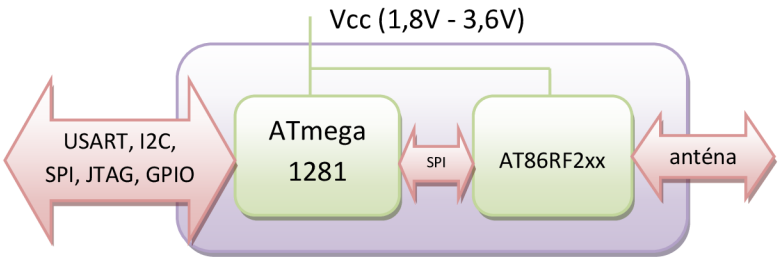
4.2 Rádiový modul ZigBit

ZigBit modul je kompaktní, nízkopříkonový modul s vysokou citlivostí pracující v pásmu 700/800/900MHz či 2,4GHz podle standardu IEEE 802.15.4/ZigBee. Takovéto moduly vyrábí řada firem, přičemž při vývoji byl využit modul od firmy MeshNetics [9]. Tento modul v sobě ukrývá součástky firmy Atmel, a sice mikrokontrolér ATmega1281 a RF modul AT86RF2xx (tento je závislý na pásmu, ve kterém probíhá komunikace). Při vývoji byly využity moduly od firmy MeshNetics, což jsou ale v podstatě přejmenované moduly firmy Atmel. V následující tabulce jsou stručně rozebrány dostupné moduly firmy MeshNetics, které mohou být použity pro analyzátor. Tyto moduly jsou identické s moduly Atmel, pouze se liší v sériovém číslu a všechny mají mikrokontrolér ATmega1281.

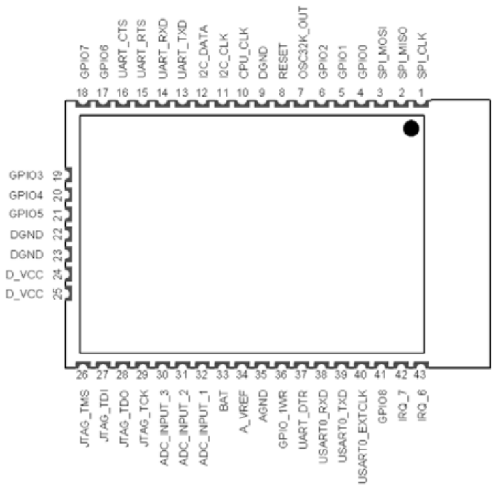
Tab. 5 Přehled dostupných ZigBit modulů

Modul	Sériové číslo	Frekv. pásmo	RF modul	Poznámka
ZigBit 900 Module	MNZB-900-B0	868/915MHz	AT86RF212	Balancovaný výstup, bez antény
ZigBit Amp Module	MNZB-A24-UFL	2,4GHz	AT86RF230	Obsahuje zesilovač a konektor na anténu
ZigBit Module with Dual Chip Antenna	MNZB-24-A2	2,4GHz	AT86RF230	Modul obsahuje čipovou anténu
ZigBit Module with Balanced RF Output	MNZB-24-B0	2,4GHz	AT86RF230	Balancovaný výstup, bez antény

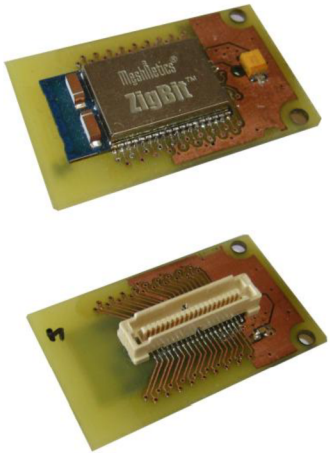
Tyto moduly samy o sobě nelze připojit k desce ZigBit2USB. Je potřeba je ještě připájet na destičku, kde z jedné strany je tento modul a z druhé strany je konektor, viz Obr. 31. Tato destička s konektorem a připájeným ZigBit modulem není předmětem návrhu této práce.



Obr. 30 Blokové schéma ZigBit modulu



Obr. 32 Profil ZigBit modulu s pojmenovanými piny



Obr. 31 ZigBit modul na destičce s konektorem

Srdcem modulu je tedy mikrokontrolér (MCU) ATmega1281. Tento MCU sám o sobě má 10 osmibitových vstupně/výstupních portů, jeden šestibitový a 2 USART porty. Ovšem v modulu by tolik bran bylo zbytečných, a proto je v ZigBit modulu pouze zjednodušená verze a vyvedeny externě jsou pouze určité piny, jak lze vidět na Obr. 32., více v [9].

4.3 ZigBit2USB deska

Aby bylo možno ZigBit modul připojit k PC, na kterém probíhá sběr dat, bylo potřeba navrhnout základní desku. Tato deska je modulární, jelikož obsahuje konektor pro připojení různých modulů rozebraných v kapitole 4.2. A to především z toho důvodu, aby bylo možno analyzovat provoz jak v sítích 868/915MHz, tak v 2,4GHz. ZigBit moduly lze tedy jednoduše vyměnit, protože jsou navzájem pinově kompatibilní.

Podrobnější popis desky je uveden v kap. 6.

5 Návrh softwaru pro analyzátor

Na internetu je dostupná řada knihoven, určená pro osmibitové mikrokontroléry. Tyto knihovny jsou v podstatě zdrojové kódy, které umožňují komunikaci podle standardu 802.15.4. Jedná se o předpřipravené kódy pro mikrokontrolér. Většina z nich je šířena pod licencí, která dovoluje jakýkoliv zásah do kódu a tudíž jeho libovolnou modifikaci.

Při návrhu analyzátoru byly podrobně prozkoumány dvě knihovny. Knihovna µracoli [10] a Chibi [11]. Na knihovnu µracoli se vztahuje tzv. modifikovaná BSD licence. Tato licence v podstatě říká, že se jedná o volně šiřitelný, open-source, Debian Free Software Guidelines, GPL kompatibilní software. Přesné znění licence dostupné na [10]. Knihovna Chibi je též volně šiřitelná.

V obou těchto knihovnách je již připravený analyzátor. U knihovny µracoli se data posílají na sériový port, na PC jsou zachytávaná programem a poté přes tzv. rouru (více v kap. 5.2.1) posílána do Wiresharku. Stejný princip je uplatněn i u knihovny Chibi. Rozdíl mezi těmito dvěma knihovnami je pouze v programu, který přeposílá data ze sériového portu do Wiresharku. U µracoli je to skript napsaný v jazyku Python, zatímco u Chibi je to program napsaný v jazyku C# .NET. Pokud by tedy byl využit proprietární SW µracoli, tak by bylo potřeba do PC doinstalovat ještě překladač Python a rozšíření Pywin32 a Pyserial. Toto by bylo až příliš zbytečně komplikované a pokud by bylo potřeba použít analyzátor na jiném PC, tak by se na něj samozřejmě muselo toto vše nainstalovat, jelikož se jedná o nestandardní aplikace a také ne vždy to funguje bez problémů, jak bylo zjištěno při vývoji. Z těchto důvodu bylo přikloněno spíše k softwaru ke knihovně Chibi. Tento SW se nenachází přímo v knihovně, ale spolupracuje s ním a lze ho nalézt na [11]. Je na něj uplatněna stejná licence a lze ho tedy modifikovat dle potřeby – jeho funkce je blíže popsána v kap. 5.2.
Princip zasílání dat z analyzátoru do Wiresharku popisuje následující obrázek.

Obr. 33 Princip komunikační cesty od uzlu do Wiresharku

5.1 Řídící firmware pro mikrokontrolér

K vývoji firmwaru pro analyzátor byla využita knihovna µracoli. Jedná se o balíček zdrojových kódů, umožňujících komunikaci pouze mezi uzly a jejich periferiemi. Tato knihovna obsahuje podporu pro MCU ATmega 1281 a oba RF moduly AT86RF212 i AT86RF230.

Vývoj kódu probíhal ve vývojovém prostředí AVR Studio 4.18. Zdrojový kód firmwaru lze tedy otevřít přes soubor projektu AVR Studia, přičemž tak dojde k otevření všech níže zmíněných souborů.

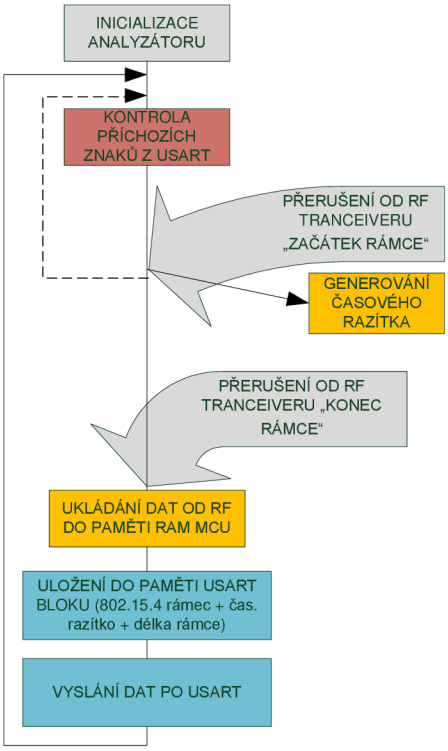
V případě analyzátoru je zapotřebí, aby uzel pracoval v tzv. promiskuitním[1] módu a data posílal na sériovou linku. Vývojový tým µracoli dodatečně vyvinul ještě tři zdrojové soubory, které spolupracují s knihovnou µracoli a které dané zařízení přepnou právě do promiskuitního módu a data, zachycená na zvoleném kanále, posílá na sériovou linku.
Jedná se o tyto soubory:

1 *sniffer.c* – hlavní zdrojový soubor analyzátoru
2 *sniffer_ctrl.c* – slouží k ovládání analyzátoru, zpracovává příkazy
3 *sniffer.h* – hlavičkový soubor pro předešlé zdrojové soubory

[1] Pokud zařízení pracuje v promiskuitním módu, tak to znamená, že na přijatá data nijak nereaguje, nemusí mít přiřazenou ani žádnou adresu, ale pouze naslouchá na zvoleném kanále a případně data přeposílá dále.

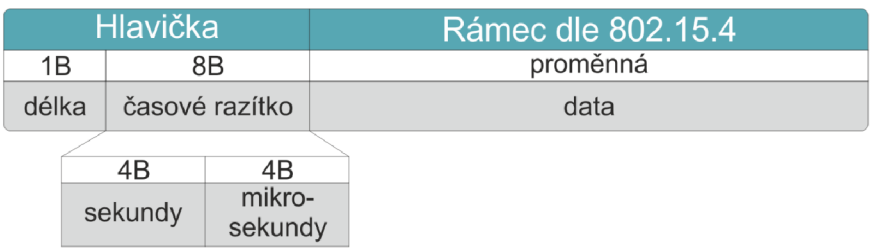
Princip funkce ukazuje blokové schéma hlavní smyčky programu na Obr. 34.

Červený blok je vykonáván stále v nekonečné smyčce.
Oranžové bloky jsou vykonány tehdy, když přijde do MCU přerušení od RF modulu.
Modré bloky jsou vykonány, pokud jsou uložena data v RAM paměti MCU.



Obr. 34 Bloková diagram hlavní smyčky programu analyzátoru

Rámec přijatý mikrokontrolérem, je jednak označkován časovým razítkem a také je zjištěna jeho délka. Časové razítko je v podstatě počet impulsů přečtených z registru časovače mikrokontroléru a přepočítaných na sekundy a mikrosekundy. Toto časové razítko má délku 8 bajtů, přičemž 4 bajty tvoří sekundy a 4 bajty mikrosekundy a začne se počítat, jakmile je analyzátor připojen k napájení. 8 bajtů je dostatečná délka, aby analyzátor mohl v jednom kuse zachytávat data po dobu 2^{32} (4 mld.) sekund. Délka rámce je reprezentována jedním bajtem a je složena z délky samotného 802.15.4 rámce + délka časového razítka. Celková struktura dat posílaných z analyzátoru do PC je na Obr. 35.



Obr. 35 Struktura data vysílaných analyzátorem

5.2 Softwarový most

5.2.1 Roura

Než bude přistoupeno k objasnění samotné roury, tak je třeba nejdříve objasnit, co je to meziprocesová komunikace. Každý proces či program spuštěný v počítači ukládá data do paměti interpretovanou nějakými proměnnými. Tyto proměnné jsou lokální v rámci daného procesu. Tedy například v procesu A je definována proměnná *x*, ke které chceme přistoupit z procesu B. Toto nelze jednoduše provést, a proto, aby bylo možno komunikovat mezi procesy, je třeba vytvořit nějakou techniku k předávání dat mezi nimi. Tato technika se nazývá meziprocesová komunikace (angl. Inter-Process Communication, IPC). Existují různé metody, jak k předávání dat může docházet. Těchto metod existuje celá řada, přičemž jednou z nich je právě roura (angl. pipe), o níž bude v následujícím textu řeč. Roura je v podstatě jakási mezipaměť pro onu meziprocesovou komunikaci, kdy data z jednoho procesu jsou ukládána do této mezipaměti (k čemuž je používaná fronta typu FIFO) a jiný proces tyto data čte. Jedná se pouze o jednosměrnou komunikaci, tedy pokud by bylo potřeba komunikovat i opačným směrem, je třeba vytvořit další rouru. Tato metoda byla převzata z unixových systémů.

Jelikož Wireshark neumí ve výchozím nastavení zachytávat na jiných rozhraních než síťových na bázi ethernet, tak z tohoto důvodu je zapotřebí spolupráce programu, který vytvoří právě tuto rouru.

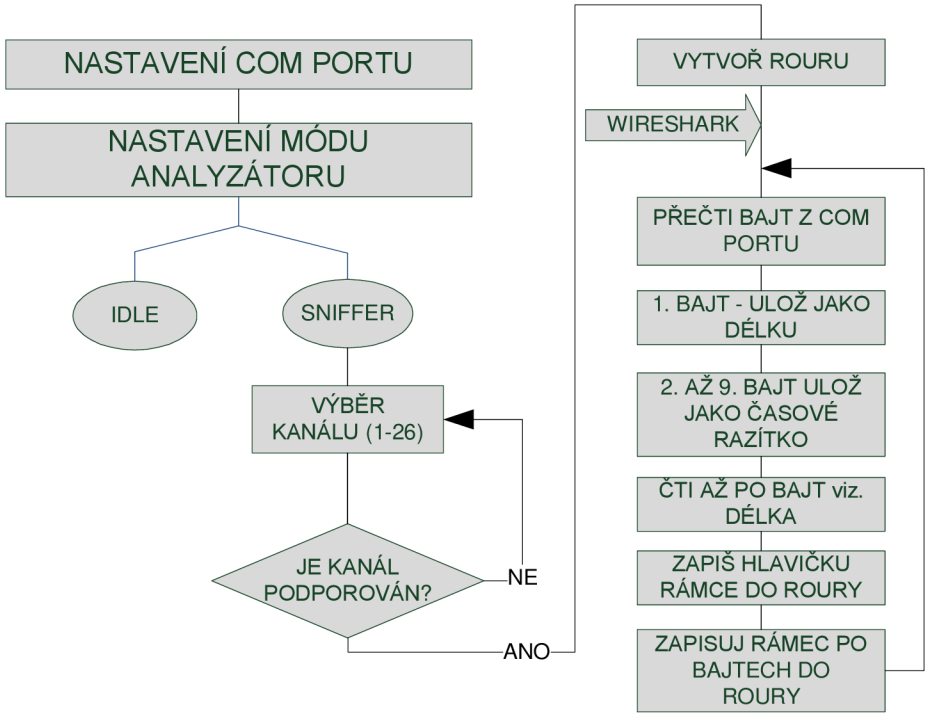
5.2.2 Návrh programu softwarového mostu

Jak bylo zmíněno v úvodu kapitoly, tak pro vývoj bylo využito programu z projektu Freaklabs. Tento software je naprogramován v jazyce C# .NET, ale neměl žádné grafické rozhraní. Fungoval pouze v příkazové řádce, kde probíhalo nastavení a výpis zachycených dat. Jako rozšíření v rámci diplomové práce bylo vyvinuto grafické rozhraní, jež umožňuje řadu nastavení, které nebylo dříve možné bez zásahu do kódu programu.

Program byl napsán ve vývojovém prostředí MS Visual Studio 2008 v jazyce C# .NET. Z toho důvodu je zapotřebí mít v počítači nainstalován balík knihoven *.NET Framework* alespoň ve verzi 3.5. Tento program byl nazván **Wireshark bridge**, jelikož tvoří jakýsi komunikační most mezi sériovým portem a wiresharkem.

Jelikož tento program není naprogramován jako vícevláknový proces a pracuje pouze v jednom vlákně, tak má jistou nevýhodu v tom, že s programem není možno pracovat v intervalu od vytvoření roury po připojení Wiresharku do této roury.

Princip funkce programu *Wireshark bridge* ukazuje blokové schéma na Obr. 36.



Obr. 36 Blokový diagram programu Wireshark bridge

Dále bylo potřeba upravit původní program tím směrem, aby byl schopen zpracovat data, ke kterým je přidáno časové razítko. Časové razítko je přenášeno za bajtem vyjadřujícím délku rámce a má velikost 4 + 4 bajty. Ovšem časové razítko, které vyšle analyzátor je ve tvaru tzv. big-endian, a proto je nutné změnit endianitu obou čtveřic bajtů do tvaru tzv. little-endian.

Jelikož Wireshark spolupracuje s knihovnou *WinPcap*, tak *Wireshark bridge* musí data, která přijme z analyzátoru poslat do roury, tedy i samotnému Wiresharku, v určitém formátu a pořadí, viz Obr. 37.

Obr. 37 Struktura dat posílaných Wiresharku

Globální hlavička – vyslána pouze jednou při připojení wiresharku do roury a popisuje endianitu dat, verze formátu dat, časová zóna pro časová razítka, maximální délka rámce a typ hlavičky dat na linkové vrstvě.

Hlavička rámce – vysílána před každým rámcem a obsahuje časové razítko a délku rámce

Rámec – rámec dle 802.15.4

5.3 Wireshark

Aplikace Wireshark je síťový protokolový analyzátor, jenž umožňuje analyzovat provoz tím, že surová data v binární formě dekóduje a zobrazuje pro uživatele čitelnou formou. [12]

Wireshark byl pro tento projekt vybrán z toho důvodu, že již od verze 1.0.0 obsahuje podporu pro standard 802.15.4. Jinými slovy, dokáže dekódovat data, která přijme, podle tohoto standardu.

Jak bylo řečeno v kap. 5.2.1, tak wireshark umí zachytávat pouze na síťových rozhraních počítače. Ovšem tím je míněno pouze to, že při otevření programu jsou načtena všechna síťová rozhraní a uživatel si pak z těchto vybere, na kterém chce zachytávat. Ale jelikož kolonka *Interface:* je editovatelná, tak lze do ní napsat i název uživatelsky definovaného rozhraní, což právě v tomto případě bude ona roura. Bohužel název tohoto rozhraní se po vypnutí programu vymaže a při opětovném spuštění je potřeba jej opět vypsat. Toto lze obejít tak, že v nastavení programu se toto rozhraní zapíše jako výchozí a při startu se bude zachytávat vždy z tohoto, pokud se nedefinuje jinak. Toto se provede pomocí: *Edit – Preferences – Capture: Default interface*, kde se vyplní název roury.

Ve Wiresharku je obsažena podpora i pro protokoly na vyšších vrstvách (síťová, aplikační). Tudíž pokud uživatel chce analyzovat rámce, které obsahují data z vyšších vrstev, například podle Std Zigbee či 6LoWPAN, tak zapnutí podpory těchto protokolů se provede v nastavení *Analyze – Enabled Protocols,* kde již vybere požadované. Ve výchozím nastavení jsou povoleny všechny protokoly.

Wireshark obsahuje též možnost kontroly CRC každého rámce a nastavení dešifrovacího klíče, pokud analýza probíhá v zabezpečené síti.

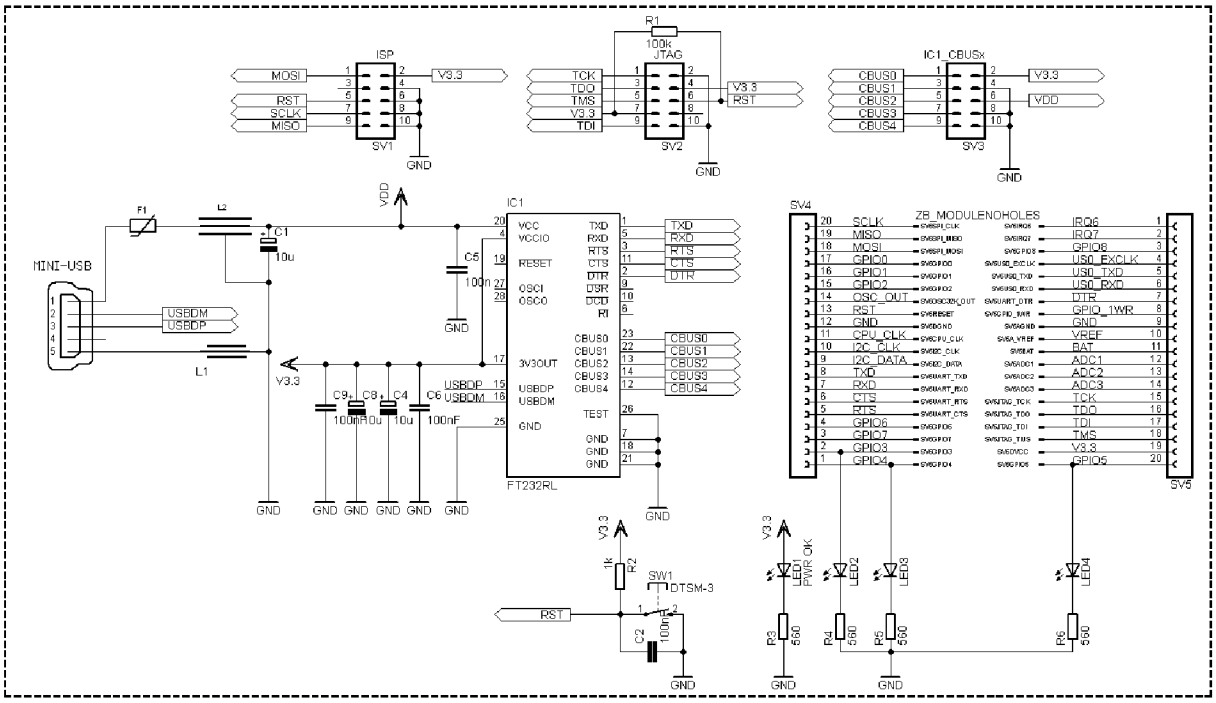
6 Návrh hardwaru analyzátoru

Návrh schématu i DPS probíhal v programu Eagle 5.4.0. Analyzátor je umístěn na oboustranné desce o rozměrech 80 x 60 mm. Elektronické součástky jsou v pouzdře pro povrchovou montáž, aby velikost analyzátoru byla co nejkompaktnější, a jsou osazeny pouze z jedné strany DPS. Po bocích jsou vyvedeny dvě dutinkové lišty 20pin, které jsou připojeny na všechny piny ZigBit modulu. Dále se na desce nachází 3 konektory pro plochý kabel. Na jednom jsou vyvedeny piny sběrnice CBUS obvodu FT232R + napájení, jeden konektor pro JTAG a jeden pro ISP programátor.

Tato deska není v podstatě nic jiného, než převodník UART/USB. Srdcem desky je integrovaný obvod FT232R, který se stará o přizpůsobení komunikačního protokolu a napěťových úrovní. Obvod FT232R se stará v podstatě o transparentní přenos sériové linky přes USB. V PC se po připojení a nainstalování potřebných ovladačů vytvoří virtuální COM port. Na tento COM port se poté jednoduše lze připojit například přes terminálový program.

Na desce je konektor mini-USB, přes který je vedena komunikace do PC i napájení.

Na desce se nachází 4 LED diody, z nichž jedna je indikační nazvaná PWR OK a indikuje, že deska je připojena k napájení, které se bere právě z USB. Další 3 LED diody jsou připojeny na piny GPIO (General Purpose Input Output) Zigbit modulu, což jsou piny, jak název napovídá, sloužící k nějakému obecnému použití. Tedy například aby indikovali nějakou událost, která bude definována ve firmwaru mikrokontroléru.



Obr. 38 Schéma desky Zigbit2USB

Schéma analyzátoru je uvedeno v Příloze A.
Deska plošných spojů je uvedena v Příloze B.

7 Ovládání analyzátoru

K provozu analyzátoru je tedy potřeba celkově tří věcí:

1) HW analyzátor tvořený ZigBit modulem a deskou ZigBit2USB
2) Wireshark bridge
3) Wireshark

Následující popis bude předpokládat provozování analyzátoru na počítači s čerstvou instalací systému MS Windows bez jakéhokoliv dalšího softwaru a se ZigBit modulem bez nahraného firmwaru.

Nejdříve je potřeba si nainstalovat vývojové prostředí AVR Studio, aby bylo možno otevřít projekt. Do ZigBit modulu nahrát příslušný firmware. Je třeba jít do adresáře *src* a zde přes příkazovou řádku otevřít program make s parametrem zigbit900 či zigbit2400. Tedy *„make zigbit900“* pokud bude použit modul pro frekvenční pásmo 898/915MHz či *„make zigbit2400“* pro pásmo 2,4GHz. Tento příkaz zkompiluje zdrojové soubory v podadresářích adresáře *src* do adresáře *lib*, v němž vytvoří knihovny **.a*. Ve složce *main* se nachází hlavní zdrojové soubory analyzátoru. Otevře se soubor projektu *sniffer.aps* a v tomto projektu se nachází zdrojové soubory analyzátoru. V dialogovém okně *Project Options*, které se otevře přes *Project – Configuration Options* lze v kolonce *Active Configuration* vybrat modul, pro který má probíhat kompilace. Lze si zde vybrat ze dvou možností, a

sice *zigbit900* pro modul pracující v 898/915MHz pásmu a *zigbit2400* pro 2,4GHz pásmo. Samozřejmě toto předpokládá mít vytvořené příslušné knihovny pro dané pásmo v adresáři *\lib* dle výše popsaného postupu, jinak kompilátor vypíše chybu, že nemůže najít tyto knihovny. Po kompilaci se tedy nahraje tento firmware do modulu.

Dále je třeba si připravit software v PC. Jelikož v čerstvé instalaci Windows se nenachází knihovny, které jsou potřeba ke spuštění programu Wireshark bridge, je potřeba tyto doinstalovat přes balík .NET Framework 3.5 dustupný na [19]. Dále je potřeba nainstalovat protokolový analyzátor Wireshark s knihovnami WinPcap dostupný z [20]. Po připojení analyzátoru do USB portu, dojde k automatickému nainstalování ovladačů, starající se o komunikaci s obvodem FT232R a které v počítači vytvoří virtuální COM port. Virtuální z toho důvodu, že je provozován přes USB, ale s plnou funkcionalitou jako na fyzickém sériovém portu. Pokud nedojde k automatickému nainstalování ovladačů, tak je lze najít na stránkách výrobce FTDI Chip [13]. Číslo tohoto COM portu je potřeba si poznačit, jelikož tento je poté vybrán v aplikaci Wireshark bridge. Číslo portu lze zjistit například ve Správci zařízení.

Otevřeme tedy Wireshak bridge, ve kterém se nastaví číslo COM portu a bitová rychlost (tato musí být stejná jako ve firmwaru modulu). Připojíme se na tento port. Poté vybereme mód, v němž požadujeme, aby analyzátor pracoval. Analyzátor může pracovat ve dvou operačních módech:

1 **IDLE** – výchozí stav; v tomto módu se nachází po připojení k počítači a jedná se o mód, ve kterém nezachycuje žádná data ze sítě WSN.

2 **SNIFFER** – na vybraném kanále zachytává data ze sítě WSN

Pro zachytávání dat vybereme *sniffer* a nastavíme kanál, na kterém má zachytávat data. Pokud nastavíme kanál, který není podporovan modulem, tak jsme na toto upozorněni chybovou hláškou. Lze si ještě nastavit název roury ve tvaru: *\\.\pipe\xxx*, kde *xxx* je uživatelem modifikovatelný název. Klikneme na *Vytvoř rouru*, čímž se vytvoří roura. Otevřeme Wireshark a do kolonky Interface (viz kap. 5.3) napíšeme *\\.\pipe\xxx* (místo *xxx* samozřejmě námi definovaný název) a klikneme na *Start*. Wireshark se poté připojí k rouře a pokud jsou zachycena analyzátorem nějaká data, tak jsou zde zobrazována po jednotlivých rámcích.

V rámcích zobrazených ve Wiresharku jsou dvě nepřesnosti:

- hodnota *Arrival Time*, což je hodnota počítaná z časového razítka. Přičemž aby tato hodnota byla správná a odpovídala aktuálnímu datu a času při analýze, pak by v časovém razítku musel být počet sekund uběhnuvší od 1. 1. 1970. Bohužel tomu tak není, jelikož analyzátor generuje tato razítka a MCU v něm neví, kolik sekund uběhlo.

- délka rámce *Length*. Respektive jsou zde dvě délky: *bytes on wire* a *bytes captured*. V hodnotě *bytes on wire* je započtena délka rámce + čas. razítko + FCS, a proto je tato délka vždy o 10B větší než *bytes captured*. Při analýze je důležitá hodnota *bytes captured*, což je právě délka pouze 802.15.4 rámce.

Pro ukončení zachytávání je důležité neukončovat přes Wireshark, ale klinutím na *Zruš rouru* v programu Wireshark bridge.

8 Analýza paketového analyzátoru

Prvně bylo potřeba zjistit, zda je vůbec schopen MCU komunikovat rychlostí alespoň 250kb/s na UART lince bez hardwarové kontroly toku dat a bez paritních bitů. Díky 8MHz krystalu je schopen dokonce komunikovat rychlostí až 500kb/s. Tato rychlost byla nastavena a analyzována korektnost znaků poslaných touto rychlostí do počítače. Tímto testem MCU prošel. Je třeba si ale uvědomit, že tato přenosová rychlost je rychlost na fyzické vrstvě a jsou do ní tedy započítány start bity i stop bity každého poslaného bajtu a mezera mezi každým bajtem. Z posledního řádku v Tab. 6 vyplývá, že přenosová rychlost je tedy ve skutečnosti v rozmezí 400kb/s až 450kb/s.

Analyzátor byl podroben testu propustnosti. Tento test znamená komunikaci mezi dvěma uzly rychlostí 250kb/s, s velikostí rámce 127B. Přičemž vysílač vytvoří kontinuální tok dat. Samozřejmě nejedná se tak úplně o kontinuální tok, jelikož vysílač vždy musí počkat, než mu od přijímače přijde potvrzení o přijetí dat, což trvá jistou dobu. Přenos 127B rámců touto rychlostí trvá přibližně 4ms. Tento rámec je přijat RF modulem, kde je ukládán do paměti a jakmile je přijat celý rámec, tak tento je poslán do MCU. MCU si přečte z rámce cílovou adresu a pokud souhlasí, tak vygeneruje ACK rámec. Velikost ACK rámce je 11B (na PHY) a jeho přenos tedy trvá 352µs. Vysílač tedy v tomto testu vysílá tok dat s mezerou přibližně 352µs.

Byly změřeny časy jednotlivých částí zpracování dat – od příjmu RF modulem po vyslání na UART linku. Tyto časy byly změřeny pro 4 velikosti rámců a jsou uvedeny v Tab. 6.

Tab. 6 Zpoždění jednotlivých části při přenosu dat z analyzátoru do PC

Velikost rámce [B]	**5**	**30**	**70**	**127**
$t_{čekání\_na\_rámec}$ **[µs]**	204	1000	2300	4120
$t_{RF\_buffer \rightarrow MCU\_buffer}$ **[µs]**	74	268	576	1010
$t_{MCU\_buffer \rightarrow UART\_buffer}$ **[µs]**	51	76	118	324
$t_{UART\_vyslani}$ **[µs]**	248	744	1540	2700
$t_{celkovy}$ **[µs]**	374	1088	2234	4034
Přenos. rychlost UART [kb/s]	451	419	410	403

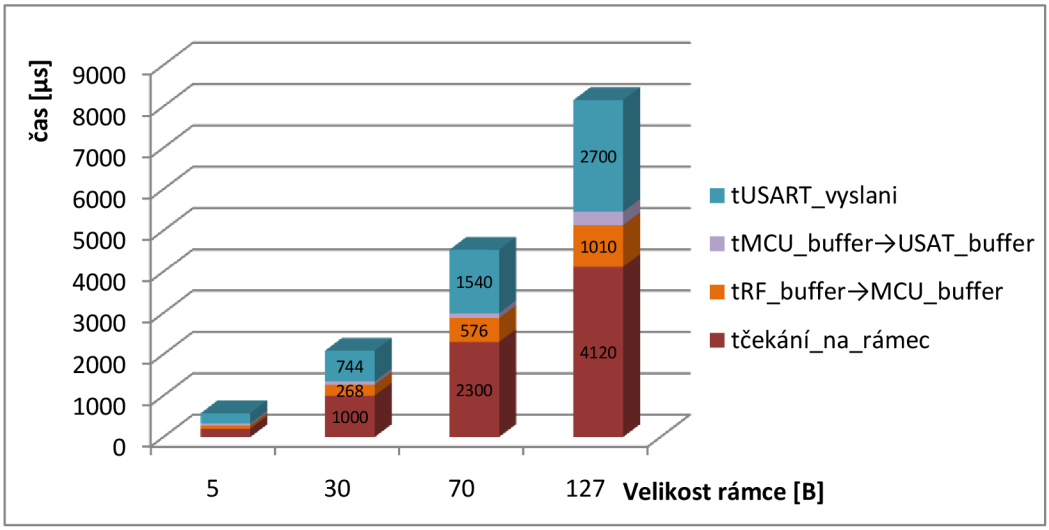
RF modul generuje dvě přerušení. První přerušení *začátek-příjmu* je generováno jako indikace začátku příjmu rámce a druhé přerušení *konec-příjmu* jako indikace, že kompletní rámec je uložen v paměti RF modulu. Interval mezi těmito dvěma přerušeními je $t_{čekání\_na\_rámec}$. Po přerušení *konec-příjmu* jsou data z paměti RF modulu poslána do MCU přes sběrnici SPI, kde jsou ukládána do RAM paměti. Tento samotný přenos je interval – $t_{RF\_buffer \rightarrow MCU\_buffer}$. Z toho jenom přenos přes samotnou SPI sběrnici taktovanou na 4MHz trvá pro 127B rámec asi 254µs.

Dále jsou data z RAM přesunuta do paměti UART bloku v MCU – $t_{MCU\_buffer \rightarrow UART\_buffer}$. Samotné vyprázdnění této paměti a vyslání dat trvá také určitý čas – $t_{UART\_vyslani}$. Je důležité si uvědomit, že po UART je vyslán nejen 802.15.4 rámec, ale navíc i délka rámce a časové razítko, což prodlouží data o 9 bajtů a prodlužuje tak i interval vysílání dat (interval $t_{UART\_vyslani}$ tedy vytvoří 802.15.4 rámec + 1B + 8B). Součet těchto tří časů tvoří $t_{celkovy}$ a je důležité, aby

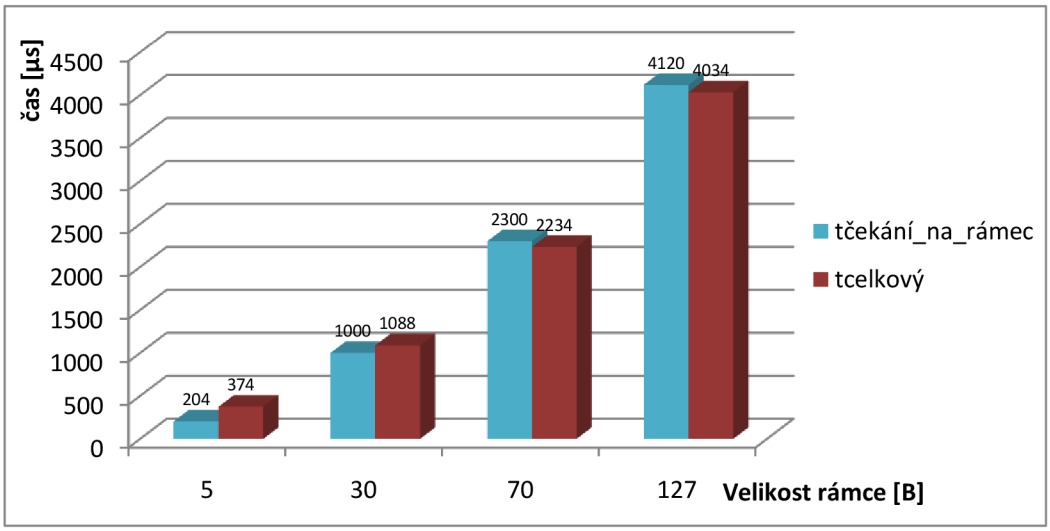
$$t_{čekání\_na\_rámec} > t_{celkový} \tag{1}$$

Během doby $t_{čekání\_na\_rámec}$ může MCU zpracovat data od RF modulu a vyslat po UART lince. Právě tato doba je kritická a ovlivňuje potenciální následující příchozí data. Pokud tedy MCU nestihne zpracovat *n*-tý rámec v čase menším než je $t_{čekání\_na\_rámec}$, pak *n+1* rámec může přepsat *n*-tý rámec svými daty a dojde tak ke znehodnocení dat. Jak vyplývá z grafu na Obr. 40, tak pro velikosti rámců 5B a 30B není nerovnice (1) dodržena a dochází k přepsání dat. Jelikož 5B rámec přijde do analyzátoru rychlostí 250kb/s, ale MCU k těmto datům přidá 9B, tudíž by bylo potřeba, aby byla data vyslána rychlostí ještě vyšší než 500kb/s. K tomuto přepisu dochází i v sítích, kde jsou posílány potvrzovací rámce. Například pokud jeden uzel vysílá 127B datový rámec a protější uzel mu potvrdí příjem ACK rámcem, tak opět může dojít k přepsání prvních 5B zachyceného datového rámce.

Data z Tab. 6 jsou vynesena do grafů na Obr. 39 a Obr. 40. Z Obr. 39 vyplývá, že pokud analyzátor zachytí rámec o velikosti 127B, tedy maximální velikosti, jaká se může vyskytnout v síti podle Std 802.15.4, tak interval mezi jeho příjmem a vysláním je dlouhý přibližně 8ms. Zatímco rámec o velikosti 5B, tedy naopak nejmenší možná velikost, je zpracováván 0,6ms. Toto jsou tedy časy, které rámec fyzicky stráví v analyzátoru.



Obr. 39 Graf zpoždění jednotlivých části při přenosu dat z analyzátoru do PC



Obr. 40 Porovnání časů *čekání na rámec* a *celkový*

Toto přepisování rámce je samozřejmě nežádoucí, protože dojde ke znehodnocení dat. Dochází k tomu z důvodu principu, jakým je naprogramován MCU. Jelikož v MCU jsou neustále vyvolávána přerušení od UART, která potřebují určitou režii (uchovat si stav Program Counter, Status Registru a dalších registrů), a po vykonání činností v přerušení si musí MCU předešlý stav opět načíst, přičemž toto dělá po každém odeslaném bajtu. Toto by tedy mohlo být příčinou, proč není dodržena nerovnice (1). Řešením tohoto problému by tedy bylo změnit obsluhu UART tak, aby nevytvářel přerušení, ale aby byl řízen pomocí dotazování.

Dalším limitujícím faktorem může být časové razítko, jelikož toto prodlužuje interval vysílání dat po UART lince o několik µs, které právě už mohou být ty nadbytečné a kvůli kterým není dodržena nerovnice (1). Samozřejmě časové razítko je důležité a nelze ho tedy odstranit jen z důvodu, aby byla dodržena podmínka nerovnice (1). Tvorba tohoto časového razítka by mohla být sice přenechána PC, ale tento princip je nevýhodný z toho důvodu, že čas, kdy by se vygenerovalo toto časové razítko je jednak ovlivněn dobou zpracování rámce analyzátorem, která je pro každou délku rámce jiná, a jednak operačním systémem počítače. Pro vyřešení problému nízké propustnosti by bylo možno využít volnou paměť RAM v MCU, jelikož tato je velká 8kB a lze tedy do ní uložit 50 rámců o velikosti 127B (6,8B) s rezervou pro samotný program.

Při rychlosti 250kb/s je analyzátor schopen zpracovat přibližně 242 rámců o velikosti 127B za sekundu, což vyplývá z času $t_{čekání\_na\_rámec}$ pro tuto velikost rámce.

Jak bylo zmíněno v kap. 5.3, tak v možnostech disektoru[2] lze nastavit, aby byly dekódovány resp. zobrazeny pouze bezchybné rámce. V tomto případě by tedy disektor spočítal hodnotu CRC pro každý rámec a porovnal s hodnotou v poli FCS a pouze pokud by souhlasila, tak by tento rámec zobrazil. Pokud by byl rámec chybný, tak by byl zahozen. Toto je ale nevýhodné v případech, kdy by byl analyzátor použit při vývoji sítě, ve které bychom chtěli samozřejmě analyzovat všechny, tedy i chybné rámce, aby uživatel vývojář měl zpětnou vazbu a mohl případně toto opravit v uzlu sítě. Bylo by tedy potřeba zobrazit někde tyto chybné rámce a k tomuto se přímo nabízí program Wireshark bridge. Wireshark bridge by tedy musel každý rámec kontrolovat a pokud by byl bez chyb, tak by ho poslal do roury, zatímco chybný rámec by zobrazil v okně v hexa tvaru. Knihovna µracoli umožňuje vypnutí kontroly CRC každého rámec, ale toto již nebylo implementováno. V současné verzi, pokud analyzátor zachytí chybný rámec, tak tento je zahozen a uživatel o něm není informován. Pokud nenastane chyba na přenosovém kanále, tak by se v síti ani neměly žádné chybné rámce vyskytnout, protože, jak bylo zmíněno, již RF modul v rádiovém modulu kontroluje CRC a pokud zjistí chybu, tak ho ani nevyšle do sítě. Jedná se tedy o nedostatek současné verze analyzátoru.

Disektor dále obsahuje možnost nastavit dešifrovací klíč, ale tato funkce nebyla blíže prozkoumána.

V programu *Wireshark bridge* se nachází jisté nedostatky a aby se těmto předešlo, tak je třeba dodržet pravidlo, že je dobré analyzátor resetovat tlačítkem na Zigbit2USB desce po každém ukončení analýzy případně při změně kanálu.

[2] Disektor – modul Wiresharku, který dekóduje data z binární formy

Různí výrobci na trhu poskytují své vlastní řešení paketového analyzátoru WSN, které je shrnuto do Tab. 7. Přičemž do neprofesionální sféry a pro žádné kriticky důležité projekty je vhodný právě touto prací navržený analyzátor. Jednak z hlediska ceny, která je desetinová oproti nejlevnějšímu komerčnímu analyzátoru, a jednak kvůli možnosti úpravy analyzátoru a to jak programové, tak hardwarové. Jistou nevýhodou by mohl ale být fakt, že současná verze softwaru analyzátoru nedokáže zobrazit chybné rámce se špatným kontrolním součtem (CRC).

Tab. 7 Srovnání dostupných analyzátorů

Název	2,4GHz	898/ 918MHz	Indikace chybných rámců	6LoWPAN	Vlastnosti	Cena
q51 PANalyzer	✓	✗	✗	✓	+ PoE, SW zdarma - napájení	$500 (8500Kč)
WiSens® Classic Packet Sniffer/Analyzer	✓	✗	✓	✗	+ USB stick	$995 – HW (17000Kč) $500 – SW podpora
WhizNets Zigbee network analyzer	✓	✗	✓	✗		$999 (17000Kč)
ZigBit2USB + ZigBit	✓	✓	✗	✓	+ SW zdarma	$60 – (1000Kč)

9 Závěr

Touto prací navržený paketový analyzátor pro bezdrátové senzorové sítě nemá přílišné ambice, aby mohl v nějaké větší míře konkurovat na trhu dostupným komerčním analyzátorům. Oproti nim má ovšem jednu velkou výhodu, a tou je cena. Tato je nesrovnatelně nižší než nejlevnější dostupný komerční analyzátor. Toto je dáno hlavně tím, že analyzátor využívá počítačový software s otevřeným zdrojovým kódem Wireshark, který je šířen bezplatně. Proto je tento analyzátor výhodný z hlediska ceny a také z hlediska uživatelské modifikovatelnosti. Nejdůležitější roli zde hraje především disektor pro standard IEEE 802.15.4. Jeho kód je volně dostupný na internetu a každý uživatel si ho tedy může upravit dle svých požadavků. Jistou nevýhodou by mohl být fakt, že se u takového softwaru uživatel nedočká žádné profesionální podpory, tak jak je tomu právě u komerčních analyzátorů, ale také díky které je v některých případech cena těchto řešení tak vysoká.

Cena tohoto paketového analyzátoru je vytvořena pouze hardwarem a činí přibližně 1000Kč, z toho 500Kč jenom za samotný ZigBit rádiový modul. Veškerý software je poskytován zdarma.

Paketový analyzátor byl sestaven a ověřena jeho funkčnost. Po sestavení byl podroben analýze, při níž se vyskytovaly jisté nedostatky, jako je nízká propustnost v sítích s velkou hustotou provozu a pro krátké rámce a nedokonalost programu Wireshark bridge. Tyto nedostatky jsou sepsány v kap. 8 a možný princip řešení je zde vysvětlen, ale nebyl již implementován.

Výsledkem této práce je funkční zařízení, schopné zachytávat data bezdrátové senzorové sítě na všech kanálech definovaných standardem IEEE 802.15.4, podle použitého rádiového modulu, a tato data analyzovat a zobrazovat pro uživatele čitelnou formou.

Seznam použité literatury

[1] **Gutiérrez, José A., Edgar, H. Callaway and Raymond, L. Barrett, jr.** *Low-Rate Wireless Personal Area Netorks, Enabling Wireless Sensors with IEEE 802.15.4 Second Edition.* New York : IEEE Press, 2007. p. 169. ISBN 0-7381-4977-2.

[2] Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). [Online] 8. 9 2006. <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>. ISBN 0-7381-4997-7.

[3] Atmel Products - MCU Wireless. [Online] 2010. [Citace: 07. 12 2010.] <http://www.atmel.com/products/zigbee/default.asp?family_id=676>.

[4] Atmel ATmega1281 datasheet. [Online] 2010. <http://www.atmel.com/dyn/resources/prod_documents/doc2549.pdf>.

[5] AT86RF230 datasheet. [Online] 2009. <http://www.atmel.com/dyn/resources/prod_documents/doc5131.pdf>.

[6] AT86RF212 datasheet. [Online] <http://www.atmel.com/dyn/resources/prod_documents/doc8168.pdf>.

[7] Connect One Ltd. - Embedded TCP/IP Internet Controllers & Embedded & Serial Device Servers. [Online] Connect One Ltd, 1996. [Citace: 07. 12 2010.] <http://connectone.com/>.

[8] Embedded Ethernet. [Online] 2010. [Citace: 07. 12 2010.] <http://www.ethernut.de>.

[9] MeshNetics / ZigBee Modules / ZigBit Module with Chip Antenna. *MeshNetics.* [Online] 2009. <http://www.meshnetics.com/netcat_files/Image/M-251~01-(ZigBit%20OEM%20Module%20Product%20Datasheet).pdf>.

[10] µracoli - The µController Radio Communication Library. [Online] [Citace: 10. 04 2011.] <http://www.nongnu.org/uracoli/index.html>.

[11] Freaklabs – Open Source Wirelless. [Online] 2011. [Citace: 10. 04 2011.] <http://freaklabs.org/>.

[12] The Wireshark Wiki. [Online] 2010. [Citace: 07. 12 2010.] <http://wiki.wireshark.org/>.

[13] FT232R datasheet. [Online] <http://www.ftdichip.com/Products/ICs/FT232R.htm>.

[14] **Mauri Kourilehto, Mikko Kohvakka, Jukka Suhonen.** *Ultra-Low Energy Wireless Sensor Networks in Practice.* místo neznámé : John Wiley & Sons, Ltd. ISBN 978-0-470-05786-5.

[15] Wireless sensor networks, research group. [Online] [Citace: 10. 04 2011.] <http://www.sensor-networks.org>.

[16] **Naveen Sastry, David Wagner.** Security Consideration for IEEE 802.15.4 Networks. [Online] [Citace: 10. 04 2011.] <http://citeseerx.ist.psu.edu>.

[17] **Ergen, Sinem C.** ZigBee/IEEE 802.15.4 Summary. [Online] <www.sinemergen.com/zigbee.pdf>.

[18] Transmission of IPv6 Packets over IEEE 802.15.4 Networks. *RFC 4944.* [Online] Září 2007. <http://tools.ietf.org/html/rfc4944>.

[19] Microsoft Corporation. [Online] 2011. <http://www.microsoft.com>

[20] Wireshark · Go deep. [Online] 2011. <http://www.wireshark.org/>

Seznam zkratek

IEEE	Institute of Electrical and Electronics Engineers
Std	Standard
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
QoS	Quality of Service
WSN	Wireless Sensor Network
ISO/OSI	International Organization for Standardization/Open Systems Interconnection
FFD	Full Function Device
RFD	Reduced Function Device
PHY	Physical layer
MAC	Medium Access Control
LLC	Logical Link Control
PSDU	PHY Service Data Unit
PPDU	PHY Protocol Data Unit
SHR	Synchronization header
PHR	PHY header
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
MHR	MAC Header
MFR	MAC Footer
CSMA/CA	Carrier sense multiple access with collision avoidance
RF	Radio-Frequency
BPSK	Binary-Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
PSSS	Parallel Sequence Spread Spectrum
TS	Timeslot
GTS	Guaranteed Time Slot
CFP	Content Free Period
CAP	Contention Access Period
FCS	Frame Chech Sequence
CRC	Cyclic Redundant Check
ACK	Acknowledgement
AES	Advanced Encryption Standard
AVR	Advanced Virtual RISC (Alf Vegard RISC)
USB	Universal Serial Bus
CMOS	Complementary Metal–Oxide–Semiconductor
USART	Universal Synchronous / Asynchronous Receiver and Transmitter
JTAG	Joint Test Action Group
I2C	Inter-Integrated Circuit
MCU	Microcontroller Unit
RS232	Recommended Standard 232
ISP	In-System Programming
SPI	Serial Peripheral Interface

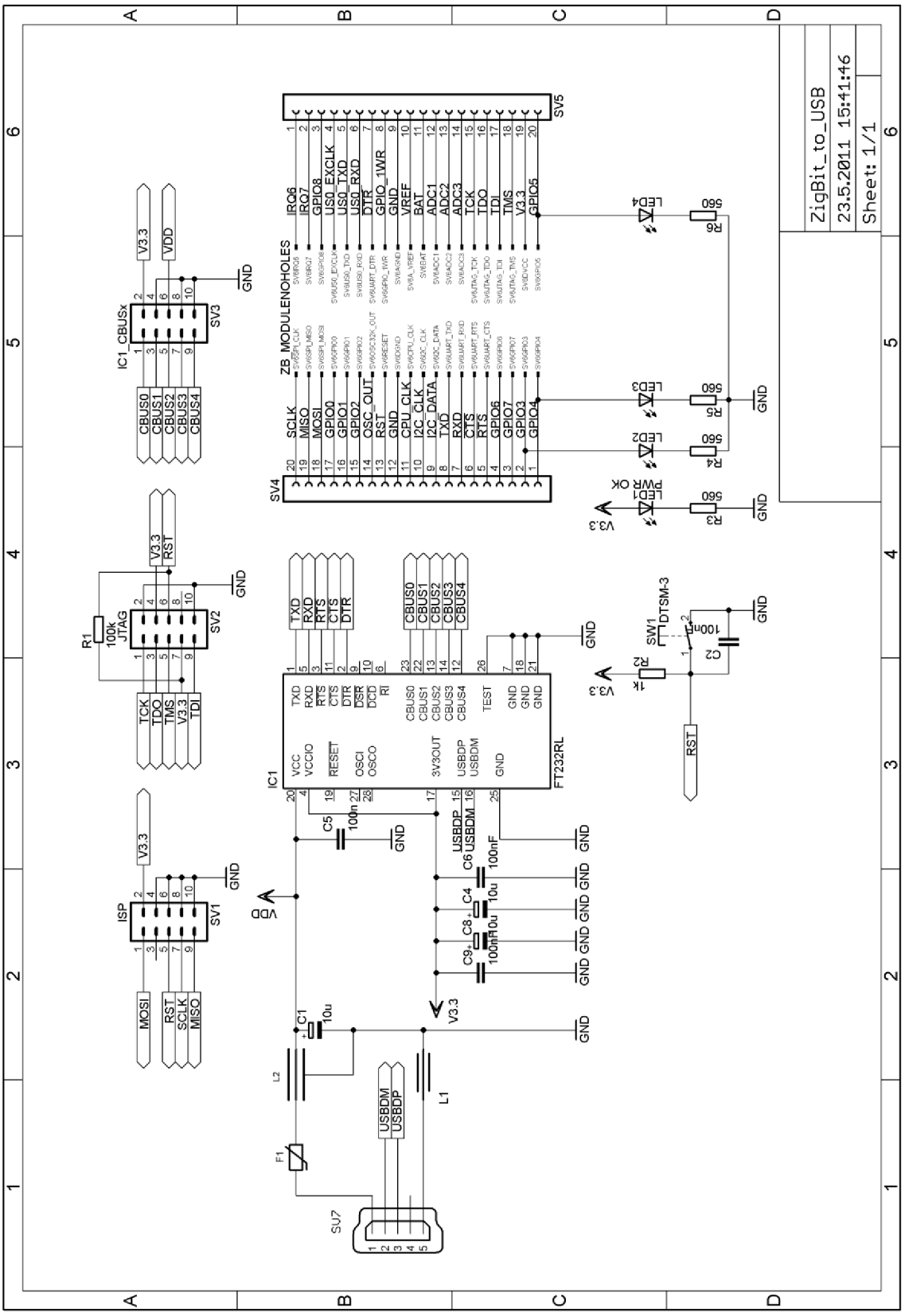
Seznam příloh

A Schéma paketového analyzátoru
B Deska plošných spojů
C Seznam součástek

Obsah přiloženého DVD

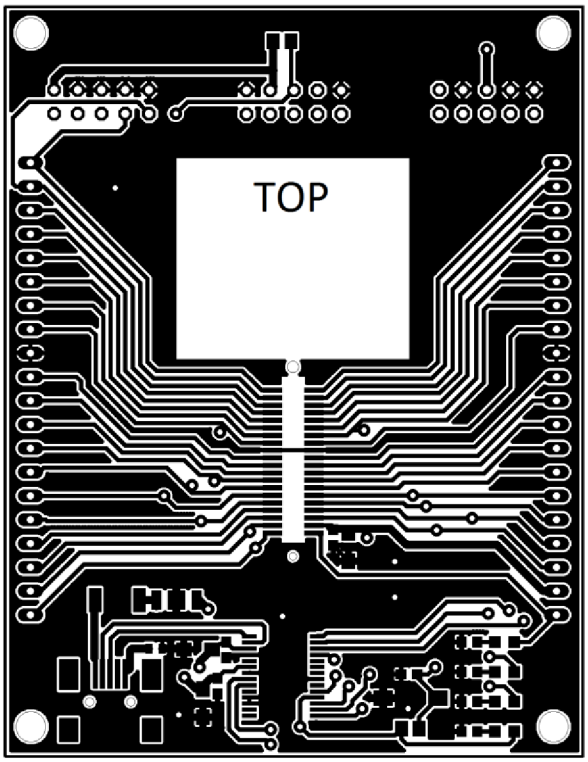
\Datasheety – datasheety použitých součástek
\Software – potřebný software + firmware pro ZigBit modul
\Paketovy analyzator – elektronická verze diplomové práce
\DPS – schéma + DPS

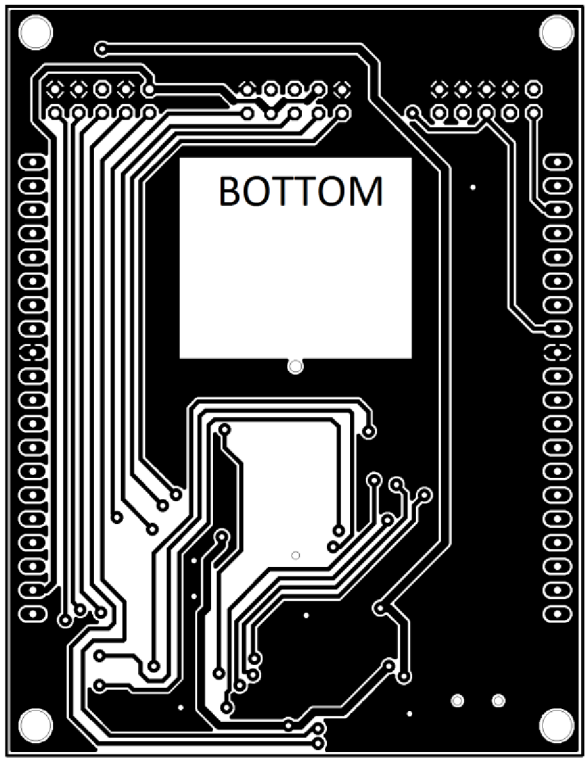
A Schéma analyzátoru

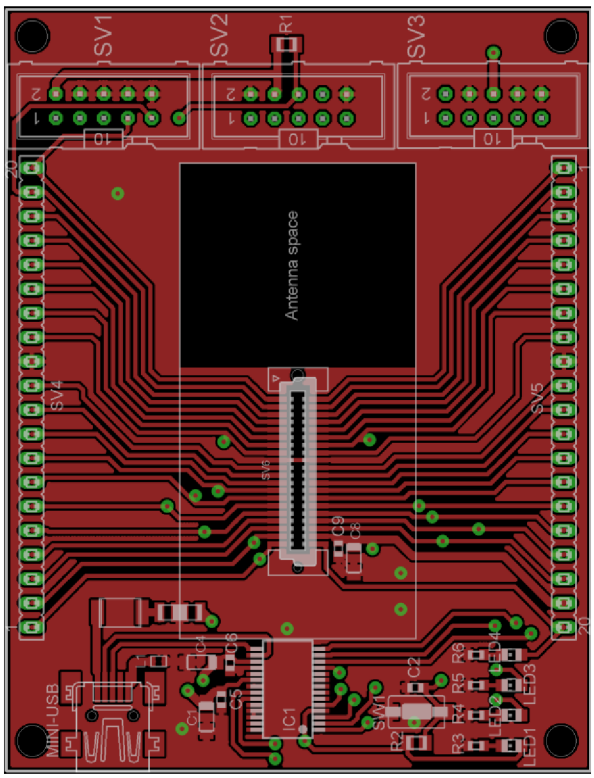


B Deska plošných spojů

Strana TOP (strana součástek), stran BOTTOM (spodní) a osazovací plán.







C Seznam součástek

R1	100kΩ (0805)
R2	1kΩ (0805)
R3, R4, R5, R6	560Ω (0603)
C1, C4, C8	10µF/10V
C2, C5, C6, C9	100nF (0603)
L1	CW0603L1206
L2	NFM41PC204F1H3L
F1	PPTC1812SMD014/15V
IC1	FT232R
SV1, SV2, SV3	ML10
SV4, SV5	BL20G
SV6	5177983-1
SV7	MINI-USB (SMD)
LED1	zelená LED (0805)
LED2, LED3, LED4	červená LED (0805)
SW1	P-DT2112C SMD