

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Bezdrátové sítě ZigBee

Ladislav Beran

Bakalářská práce 2011

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ladislav BERAN**
Osobní číslo: **I08292**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**
Název tématu: **Bezdrátové sítě ZigBee**
Zadávající katedra: **Katedra elektrotechniky**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je seznámit se s možnostmi tvorby bezdrátových sítí na bázi technologie ZigBee, a sice s ohledem na její budoucí využití při komunikaci několika inteligentních senzorů rozmístěných po lidském těle. Důraz musí být kladen na co nejnižší spotřebu zařízení (napájení z baterií). Předpokládá se, že jednu síť bude tvořit cca 6 modulů. Součástí práce budou i testy funkčnosti a spolehlivosti sítě v ne zcela ideálních podmínkách šíření signálu (např. skrz překážky z různých materiálů). Pro první testy využijte komerčně dostupné moduly.

Teoretická část: Popis technologie ZigBee

Praktická část: Vývoj testovací aplikace, praktické testy sítě

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

ZigBee. Wikipedie, otevřená encyklopedie. [Online]
<http://cs.wikipedia.org/wiki/ZigBee>.

ZigBee Specification (Document 053474r17). ZigBee Standards Organization, 2007.

ZigBee Alliance Home Page. [Online] <http://www.zigbee.org/>.

Bradáč, Zdeněk., Bezdrátový komunikační standard ZigBee. Automatizace. 2005, Sv. 4.

Atmel, Inc., BitCloud - ZigBee PRO stack. [Online]
http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4495&source=redirect.

Z-Stack - ZigBee Protocol Stack. [Online]
<http://focus.ti.com/docs/toolsw/folders/print/z-stack.html>.

ZigBee 2006 Protocol Stack. [Online] <http://www.microchip.com/>.

ZigBee Alliance., ZigBee Home Automation Public Application Profile (rev. 26). ZigBee Alliance, Inc, 2010.

ZigBee Alliance Membership FAQ. [Online]
<http://www.zigbee.org/Join/MembershipFAQ.aspx>.

ZigBee Alliance Membership Forms. [Online]
<http://www.zigbee.org/Join/MembershipForms.aspx>.

ZigBee Certification FAQ. [Online]
<http://www.zigbee.org/Products/CertificationFAQ.aspx>.

Vedoucí bakalářské práce:

Ing. Martin Hájek
Katedra elektrotechniky

Datum zadání bakalářské práce:

15. prosince 2010

Termín odevzdání bakalářské práce:

13. května 2011



prof. Ing. Simeon Karamazov, Dr.
děkan



Ing. Zdeněk Němec, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2011

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne

Ladislav Beran

Poděkování

Rád bych poděkoval Ing. Martinu Hájkovi za jeho velmi velkou pomoc při zpracování této práce a profesionální rady a pomoc při návrhu desky plošných spojů. Dále bych chtěl poděkovat všem učitelům, kteří mne učili za vynikající odvedenou práci a vstřícnost. V neposlední řadě bych chtěl poděkovat mé rodině a přítelkyni za podporu při studiu.

Anotace

Práce pojednává o vlastnostech a využití přenosové technologie ZigBee. V práci práce jsou rozebrány základní komunikační protokoly a implementace SDK prostředí pro vytvoření základní funkční aplikace. Dále se práce zabývá návrh desky plošných spojů pro vývojový kit, na kterém bude možné testovat vývoj ZigBee zařízení.

Klíčová slova

ZigBee, IEEE 802.15.4, Bezdrátové sítě, Personální bezdrátové sítě, nízkovýkonové sítě, bezdrátové senzory.

Title

ZigBee Wireless Network

Annotation

This work deals with properties new wireless technology called ZigBee. Elementary communication protocols and implementation of SDK workspace for creating of elementary applications are described in the text of thesis. Simple development kit was designed in the context of the work as well.

Keywords

ZigBee, IEEE 802.15.4, Personal area network, Low-Rate Wireless network, Low-Energy Network, Wireless Sensors.

Obsah

Seznam zkratk	8
Seznam tabulek	10
1 Úvod	11
2 WPAN typu ZigBee	12
2.1 Úvod.....	12
2.2 Historie.....	12
2.3 IEEE 802.15.4.....	13
2.3.1 Typy zařízení.....	13
2.3.2 Úvod.....	13
2.3.3 Verze.....	13
2.4 Referenční ISO/OSI ZigBee model.....	14
2.4.1 Fyzická vrstva.....	15
2.4.2 MAC Vrstva.....	18
2.4.3 Zabezpečení na MAC vrstvě.....	25
2.4.4 Síťová vrstva (NWK).....	31
2.4.5 Aplikační vrstva ZigBee.....	34
3 Vývojový kit	37
3.1 Úvod.....	37
3.2 Výběr ZigBee modulu.....	37
3.3 ZigBee modul Atmel typ ATZB-24-A2.....	37
3.4 Blokové schéma zapojení.....	40
3.5 Elektrické schéma zapojení.....	40
3.5.1 Napájení.....	41
3.6 Diody, Sedmisegmentový displej.....	43
3.7 Sériová linka.....	43
3.8 Tlačítka.....	43
3.9 Vstupy.....	44
4 Návrh desky plošného spoje	44
5 Firmware	45
5.1 Úvod.....	45
5.2 Testovací firmware.....	45

5.2.1	Ukázka kódu pro oživení sedmissegmentového LCD displeje.....	45
5.3	FW pro ZigBee síť	46
5.3.1	Úvod	46
5.3.2	ZigBee Bitcloud Stack.....	47
5.3.3	Základní kostra aplikace.....	49
6	Závěr.....	57
	Seznam použitých zdrojů.....	58
7	Přílohy	61
7.1	Schéma zapojení vývojového kitu.....	61
7.2	Vrstva TOP.....	62
7.3	Vrstva BOTTOM.....	62
7.4	Osazovací motiv	63
7.5	Srovnání vysílacích čipů.....	64

Seznam zkratek

ASCII – American Standard Code for Information Interchange

BLUETOOTH – bezdrátový standart pro přenos dat

WIFI – Bezdrátový standart typu IEEE802.11

ZIGBEE – Bezdrátový standart typu IEEE802.15.4

AVR – Advanced Virtual RISC

RISC – Reduced Instruction Set Computer

EEPROM – Electrically Erasable Programmable Read-Only Memory

JTAG – Joint Test Action Group

OCD – On Chip Debugging

LSB – Least Significant bit

MSB – Most Significant bit

SMT – Surface Mount Technology

SMD – Surface Mount Device

THT – Through-hole technology

THD – Through-hole Device

SPI – Serial Peripheral Interface

TWI – Two-wire Serial Interface = v principu to samé jako I2C

USART – Universal Synchronous Asynchronous Receiver Transmitter

UART – Universal Asynchronous Receiver Transmitter

PAN – Personal Area Network

LAN – Local Area Network

MAN – Metropolitan Area Network

WAN – World Area Network

CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance

DSSS – Direct Sequence Spread spectrum

UWB – Ultra Wide Band

MIC – Message Integrity Code

AES – Advanced Encryption Standard

ZDO – ZigBee Device Object

CSS – Chirp Spread Spectrum

RSSI – Received Signal Strength Indication

LQI – Link Quality Indication

IEEE – The Institute of Electrical and Electronics Engineer

LDO – Low Drop Out – stabilizátor s malým úbytkem napětí

LR-WPAN – Low Rate Wireless Personal Area Network

LQI – Link Quality Indicator

Seznam obrázků

Obrázek 1 – Schéma referenčního modelu ZigBee [20]	15
Obrázek 2 – Frekvenční spektrum IEEE802.15.4 vs IEEE802.11 [22]	16
Obrázek 3 – Schéma vysílacího řetězce pro pásmo 2,4 GHz [30]	17
Obrázek 4 – Schéma převodu symbolu pomocí systému DSSS [30].....	18
Obrázek 5 – Schéma O-QPSK modulátoru [27]	18
Obrázek 6 – Schéma základního MAC rámce [14].....	19
Obrázek 7 – Schéma základního MAC rámce – Frame Control [14]	19
Obrázek 8 – typy rámců [14].....	19
Obrázek 9 – Schéma Beacon frame [14].....	21
Obrázek 10 – MAC vrstva – schéma superrámce [18].....	22
Obrázek 11 – Schéma Data Frame [14].....	23
Obrázek 12 – Schéma Acknowledgment Frame [14].....	23
Obrázek 13 – Schéma MAC Command Frame [14]	23
Obrázek 14 – Seznam MAC Command rámců [14].....	24
Obrázek 15 – Schéma MAC rámce [19]	26
Obrázek 16 – Schéma části rámce pro zabezpečení MAC hlavičky [19]	26
Obrázek 17 – Schéma části rámce pro různé zabezpečení dat [19].....	27
Obrázek 18 – Komunikace koncové zařízení → koordinátor bez Beacon Frame	29
Obrázek 19 – Komunikace koordinátor → koncové zařízení bez Beacon Frame	29
Obrázek 20 – Komunikace koncové zařízení → koordinátor s Beacon Frame	30
Obrázek 21 – Komunikace koordinátor → koncové zařízení s Beacon Frame	30
Obrázek 22 – Topologie ZigBee sítě [31]	31
Obrázek 23 – Schéma CCM zabezpečeného rámce na NWK vrstvě[1]	33
Obrázek 24 – ZigBee modul firmy Atmel [26]	37
Obrázek 25 – Schéma vnitřního zapojení ATZB-24-A2 modulu [23].....	38
Obrázek 26 – Blokové schéma RF modulu AT86RF230 [24]	39
Obrázek 27 – Blokové schéma vývojového kitu	40
Obrázek 28 – Popis konektorů desky vývojového kitu	44
Obrázek 29 – Blokové schéma BitCloud stacku [29].....	48
Obrázek 30 – Schéma komunikace v ZigBee.....	56

Seznam tabulek

Tabulka 1 – Frekvenční pásma dle normy IEEE 802.15.4 9[16],[17][14].....	15
Tabulka 2 – Tabulka definice kanálů pro ZigBee [22].....	16
Tabulka 3 - ZigBee profily[21]	36

1 Úvod

Dnešní doba je dobou bezdrátových technologií. Dnes již skoro každé nové zařízení vydané na trh je opatřeno bezdrátovou technologií určenou pro přenos uživatelských dat. Každé zařízení má svoje vlastní požadavky a proto byla nutnost díky kompatibilitě zařízení vytvořit nové přenosové standardy. Jedním z těchto nových standardů je i ZigBee.

Cílem této práce je prostudovat tento standard a ze získaných znalostí navrhnout vývojový kit, s pomocí kterého bude možné vytvořit vlastní bezdrátovou síť. ZigBee je relativně novou technologií, která nekonkuruje stávajícím jako je Bluetooth či Wifi, ale snaží se vyplnit mezeru na trhu. ZigBee je technologie speciálně navržená pro provoz zařízení s krátkým dosahem v rámci sítí typu PAN. Jejím hlavním úkolem je bezdrátová automatizace v různých oborech lidské činnosti, kde se kladou požadavky na dlouhodobý bateriový provoz a zároveň nízké pořizovací náklady. Při maximálním úsporném provozu se dá zařízení provozovat až dva roky na jednu alkalickou baterii. Typická zařízení vhodná pro integraci do ZigBee sítí mohou být např. teplotní čidla, senzory pohybu či automatizace v rámci budov (osvětlení, ovládání garážových vrat).

Vlastní text práce je rozdělen do několika základních kapitol. První kapitola se zabývá vlastním popisem ZigBee s důrazem na OSI/ISO model a standart IEEE802.15.4, na němž je ZigBee postaveno. Vzhledem k rozsahu práce není možné podat vyčerpávající popis celého standardu. Ten by zabral několik set stran. Zájemci mohou podrobnější informace získat např. z [14][15],[16],[17].

V další kapitole je popsán vývojový kit, který byl v rámci práce vyvinut, postaven a oživen. Jeho hlavním úkolem je poskytnout platformu pro ověřování praktického vývoje ZigBee aplikací/sítí a testování provozu ZigBee sítí v praxi.

V poslední kapitole je rozebrán firmware pro výše zmíněný kit. Firmware umožňuje vytvořit jednoduchou síť.

2 WPAN typu ZigBee

2.1 Úvod

ZigBee je standard určený pro tvorbu sítí krátkého dosahu (PAN). ZigBee je relativně nová technologie, vystavěná na standartu IEEE 802.15.4 [14]. Primární využití ZigBee sítě pro spojení nízkovýkonových zařízení v síti PAN na vzdálenost maximálně několik desítek metrů, pro použití v průmyslových a senzorových sítích. ZigBee technologie je založena na standardu IEEE 802.15.4, který definuje spodní dvě vrstvy – Fyzickou a Linkovou. Nad tímto standardem již leží vrstvy definované ZigBee Aliancí. Celý koncept ZigBee je navržen s ohledem na minimální spotřebu celého komunikačního řetězce.[1]

2.2 Historie

Vývoj ZigBee začal již v roce 1997, avšak na trh byl standart uveden až v roce 2003, ne jako konkurence již stávajících technologií jako je Bluetooth a Wifi, ale jako doplněk bezdrátových sítí na aplikace, na které není vhodné Wifi ani Bluetooth. Standart IEEE 802.15.4 definuje dvě spodní vrstvy ISO/OSI modelu - spojovou a fyzickou vrstvu. V lednu 2005 byla zveřejněna konečná podoba specifikace. V dnešní době se na vývoji standardu ZigBee podílí více než 300 firem sdružené do korporace ZigBee Aliance [3]. V této alianci jsou zástupci světoznámých automatizačních firem jako (Omron, Honeywell, Siemens, dále Texas Instruments, Analog Device, Cisto Systems atd.). Díky tomu, že v ZigBee Alianci jsou firmy zabývající se automatizací, tak standart ZigBee vyhovuje svými vlastnostmi potřebám automatizace nejvíce – sběr dat z bezdrátových snímačů, které jsou bateriově napájeny po velmi dlouhou dobu.[1],[32]

2.3 IEEE 802.15.4

2.3.1 Typy zařízení

Koordinátor

Centrální prvek ZigBee sítě, zajišťující založení, nastavení a start ZigBee sítě. Při založení sítě vygeneruje jedinečný identifikátor konkrétní sítě nazývaný PANID.

Router

Prvek ZigBee sítě, zajišťující rozšíření sítě na vyšší počet prvků.

End-Device

Koncový prvek sítě. Typický koncový prvek je čidlo teploty, senzor pohybu apod.

2.3.2 Úvod

Standart IEEE802.15.4 byl navržen pro zařízení s nízkou přenosovou rychlostí, velkou dobou provozu na bateriové napájení a nízkou složitostí. Jeho použití spadá pod nelicencovaná mezinárodní přenosová pásma. Potenciální zařízení jsou bezdrátové senzory pro domácí automatizaci, náhrada za IČ ovladače a spousta dalších potencionálních aplikací. Norma IEEE 802.15.4 je pro ZigBee technologii stěžejní, protože definuje spodní dvě vrstvy referenčního modelu ZigBee (Fyzická a linková vrstva) a způsob komunikace [14].

2.3.3 Verze

Standard IEEE 802.15.4 se neustále inovuje, a přizpůsobuje požadavkům vývojářů. V následující kapitole jsou rozebrány inovace základního dokumentu nazývaného IEEE 802.15.4-2003. V době psaní této práce je aktuální verze IEEE 802.15.4d.

IEEE 802.15.4 standart 2006

Původní standart z roku 2003 specifikuje dvě fyzické vrstvy, založené na technice Direct Sequence Spread Spectrum (DSSS) pro 868/915 MHz pásma s přenosovou rychlostí 20 kb/s a 40 kb/s, a jedno pásmo v 2450 MHz s rychlostí 250 kb/s. Standart z roku 2006 tyto vlastnosti rozšiřuje v Evropském 868 MHz pásmu o rychlosti 100/250 kb/s, v USA pásmu 915 MHz z deseti přenosových kanálů na celkem třicet. Kromě rozšíření pásem a rychlostí také definuje již celkem 4 typy fyzických vrstev, které jsou závislé na použité modulaci signálu. Tři z nich si zachovávají systém rozprostřeného spektra (DSSS) s použitím buď BPSK nebo O-QPSK modulace. Pro pásmo 868/915 MHz je volitelná alternativa s modulacemi BSK a ASK (založené nikoliv na DSSS, ale na Parallel Sequence Spread Spectrum. [14]

IEEE802.15.4.a – standart 2007

Tento standart upravuje dříve používaný standart 2006 a rozšiřuje jej o další dvě možnosti fyzických vrstev pro ZigBee zařízení – vrstvu pro použití UWB (Ultra Wide Band) a CSS (Chirp Spread Spectrum). UWB fyzická vrstva je rozdělena na tři pásma – pásmo pod 1 GHz, pásmo 3-5 GHz a pásmo 6-10.6 GHz. CSS fyzická vrstva využívá pásmo 2450 MHz [15]

Změny oproti předchozímu standardu:

- Rozšíření o přenosovou rychlost 851 kb/s
- Vytvoření UWB přístupu k fyzickému médiu
- Vytvoření 14 překrývajících se pásem s CSS (Chirp Spread Spektrum) v pásmu 2,4 GHz
- Vytvoření 16 UWB kanálů ve třech pásmech (250-750 MHz, 3.1-10.6 GHz)

IEEE802.15.4.c

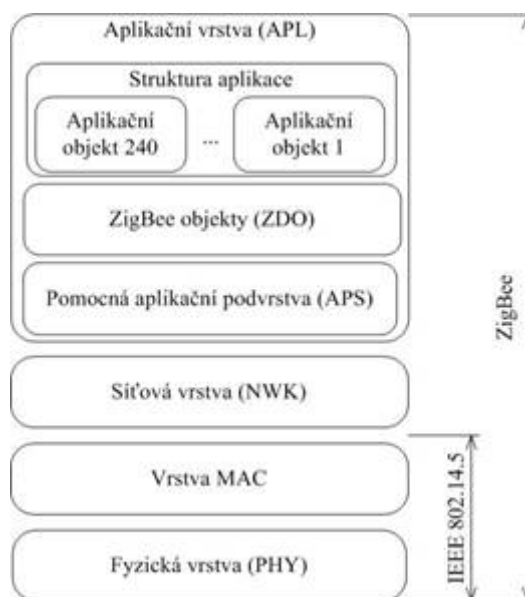
Vytvoření nového přenosového pásma pro Čínskou Republiku. Frekvenční rozsah 779- 787 MHz [16].

IEEE802.15.4.d standart 2009

Vytvoření nového přenosového pásma pro Japonsko. Frekvenční rozsah 950-956MHz [17].

2.4 Referenční ISO/OSI ZigBee model

Referenční model vychází ze standardního sedmivrstvého ISO/OSI modelu, který je definován mezinárodní normou ISO 7498 [33]. Norma IEEE 802.15.4 definuje spodní dvě vrstvy referenčního OSI/ISO modelu – fyzickou a spojovou. Vyšší vrstvy jako síťovou a aplikační již definuje vlastní specifikace ZigBee. V práci je popisován základní standart IEEE 802.15.4-2003 [14] s doplňováním informací z novějších standardů – [15],[16],[17].



Obrázek 1 – Schéma referenčního modelu ZigBee [20]

2.4.1 Fyzická vrstva

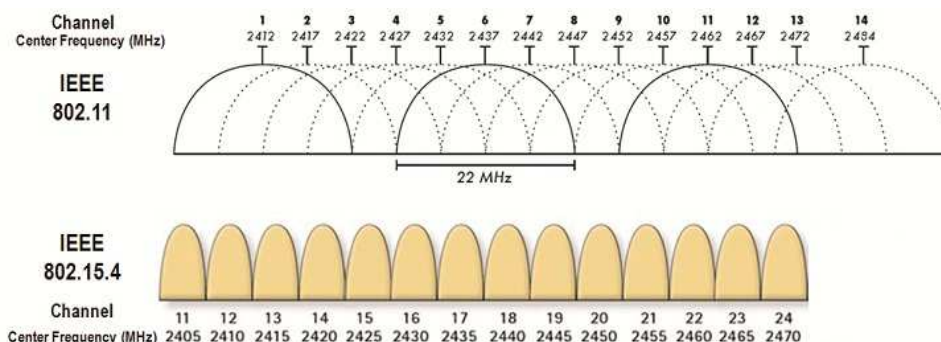
Fyzická vrstva zajišťuje komunikaci mezi protilehlou fyzickou vrstvou skrz přenosové médium. Fyzická vrstva je přímo závislá na použité technologii rádiového spoje – frekvenci přenosového pásma a metodou přenosu skrz rádiový kanál. Dále fyzická vrstva slouží pro řízení RF vysílače/přijímače pro nastavení kanálu pro vysílání dat, vysílaném výkonu, či řízení toku dat a zabezpečení jejich odeslání.[1][10][13]

V Tabulka 1 jsou popsána frekvenční pásma, která jsou definována normou IEEE 802.15.4.

Pro ilustraci jsou na Obrázek 2 zobrazeny kanály pro komunikaci pomocí ZigBee a jako referenci jsou přidány komunikační kanály Wifi.

Region	Pásmo	Počet kanálů/ max. rychlost	
Globální	2400,0 – 2483,5 MHz	16 kanálů / 250 kb/s	DSSS
Globální	2450 MHz	14 překrývajících se kanálů	CSS
USA	902-928 MHz	10 kanálů / 40 kb/s pro standart 2003 30 kanálů / 100/250 kb/s pro standart 2007	DSSS
Evropa	868.0-868.6 MHz	Pouze jeden kanál 20 kb/s pro 2003 100/250 kb/s pro 2007	DSSS
Čína	779-787 MHz	8 kanálů	DSSS
Japonsko	950-959 MHz	22 kanálů	DSSS

Tabulka 1 – Frekvenční pásma dle normy IEEE 802.15.4 9[16],[17][14]



Obrázek 2 – Frekvenční spektrum IEEE802.15.4 vs IEEE802.11 [22]

Hlavní úkoly fyzické vrstvy:

- Aktivace a deaktivace RF modulu
- Přenos fyzických dat
- Detekce nosné (Carrier sensing)
- Oprava chyb vzniklých během přenosu
- Modulace a kódování dat
- přepínání kanálů
- Vyhodnocení úrovně energie v kanálu (RSSI)
- Generování LQI – LQI je indikátor kvality spojení mezi dvěma zařízeními. Generuje se pomocí počtu chybně přijatých paketů. Hodnota LQI se pohybuje v rozmezí <0-255> (více je lepší spojení)

Channel page (decimal)	Frequency Band	Channel numbers (decimal)	Modulation scheme	Data rate, kbps
0	868 MHz	0	BPSK	20
	915 MHz	1 - 10	BPSK	40
	2.4 GHz	11 - 26	O-QPSK	250
2	868 MHz	0	O-QPSK	100
	915 MHz	1 - 10	O-QPSK	250
5	780 MHz	0 - 3	O-QPSK	250

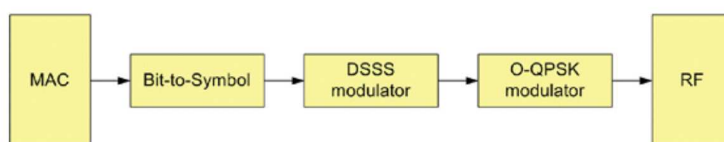
Tabulka 2 – Tabulka definice kanálů pro ZigBee [22]

Komunikace

Pro zvýšení spolehlivosti přenášených dat je vysílání provedeno systémem DSSS. Jednotlivé bity jsou pak nahrazeny početnější sekvencí bitů, které se pak vysílají do fyzického média. Signál se rozprostře do spektra a je více odolný proti rušení. Dále je i díky systému rozprostřeného spektra zabezpečen proti odposlechu, protože potenciálním špehům signálu, kteří neznají mechanismus vytváření pseudonáhodných sekvencí, se signál jeví jako šum.

Používané typy modulací

Pro rádiové přenosy se hlavně používá DSSS v pásmech pod 1 GHz se používá BPSK modulace a v pásmu 2,4 GHz se používá QPSK modulace, v novějších zařízeních se již používá vylepšená modulace typu O-QPSK.

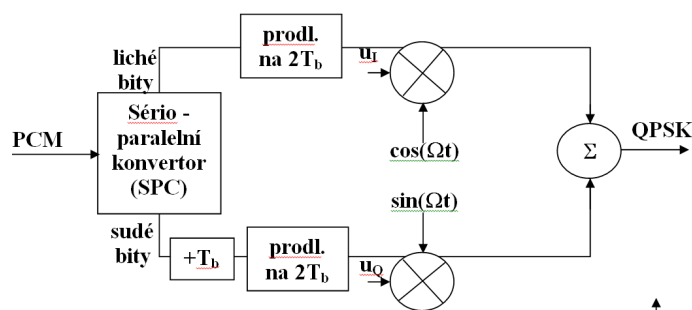


Obrázek 3 – Schéma vysílacího řetězce pro pásmo 2,4 GHz [30]

Na Obrázek 3 je patrný vysílací řetězec pro pásmo 2,4 GHz. Data z MAC vrstvy se přivedou na jednotlivé symboly v hodnotě $\langle 0-F \rangle$. V dalším členu se symboly převedou na bitovou posloupnost pomocí systému rozprostřeného spektra. Tabulka převodu je na Obrázek 4. Převedená posloupnost bitů se následně rozdělí na sudé a liché bity a vytvoří se pomocí modulátoru O-QPSK modulace (viz Obrázek 5).

symbol (decimal)	symbol (binary)	Chip values (c0, c1 ... c31)
0	0000	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1000	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0100	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	1100	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0010	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
5	1010	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 0
6	0110	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 0 1
7	1110	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 1 0 1 1 0 1
8	0001	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1001	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0101	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
11	1101	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0011	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 1 0
13	1011	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
14	0111	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
15	1111	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 0 0 0

Obrázek 4 – Schéma převodu symbolu pomocí systému DSSS [30]



Obrázek 5 – Schéma O-QPSK modulátoru [27]

2.4.2 MAC Vrstva

MAC vrstva definuje komunikaci mezi dvěma zařízeními, zajišťuje přístup k přenosovému médiu (radiovému kanálu), ověřuje, zda přijatý rámec je platný, řídí spojení a potvrzuje přijetí rámce [14]. Na Obrázek 6 je zobrazena základní struktura MAC rámce. MAC rámec se skládá ze základních částí, které jsou popsány v následujících kapitolách. Dále je v rámci označena část MHR, což je část rámce, která je stejná pro všechny vysílané rámce v ZigBee síti. Jedná se o hlavičku rámce. Tyto rámce jsou popsány v kapitole „Základní typy rámců na MAC vrstvě“.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Obrázek 6 – Schéma základního MAC rámce [14]

Frame Control

Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame type	Security enabled	Frame pending	Ack. request	Intra-PAN	Reserved	Dest. addressing mode	Reserved	Source addressing mode

Obrázek 7 – Schéma základního MAC rámce – Frame Control [14]

Na Obrázek 7 je zobrazena nejdůležitější struktura části základního MAC rámce. Tato část rámce musí být u všech vysílaných rámců. Položka Frame Control určuje:

- Frame Type

Frame type value $b_2 b_1 b_0$	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100–111	Reserved

Obrázek 8 – typy rámců [14]

Význam rámců v Obrázek 8 je popsán níže v kapitole základní typy rámců

- Security Enabled : 1 bit určující zda je rámec cryptograficky šifrován již na MAC vrstvě.
- ACK request : 1 bit určující zda odesílatel požaduje potvrzení příjmu rámce
- Intra-PAN : 1 bit určující zda je rámec určen do vnitřní sítě (1 = IntraPAN, 0 = InterPAN)
- Dest. Addr.mode: Určuje typ adresy (16 bit , 64 bit) cílového uzlu
- Source. Addr.mode: Určuje typ adresy (16 bit , 64 bit) zdrojového uzlu

Sequence Number

Sequence Number obsahuje unikátní sekvenční číslo identifikující vysílaný rámec. Délka je 8bitů.

Destination PAN identifier

Destination PAN identifier je 16 bitový unikátní PAN identifikátor cílové adresy rámce. V případě hodnoty 0xFFFF je rámec adresován na Broadcast síť. Této hodnoty využívá koordinátor při zakládání sítě a hledání ostatních koordinátorů.

Destination address

Destination address (cílová adresa) je buď 16 bitová nebo 64 bitová unikátní adresa cíle. V případě 16 bitového adresování je 0xFFFF adresa Broadcastu sítě.

Frame Payload

Frame Payload je položka s proměnnou délkou a obsahem závislým na typu přenášeného rámce. V případě povoleného šifrování (bit Security Enabled v části Frame control) je celá tato položka šifrována.

FSC

FSC položka obsahuje kontrolní součet. Vytváří se z položky MHR a MAC Payload.

Základní typy rámců na MAC vrstvě

Níže uvedené rámce vycházejí ze základního MAC rámce. Popis každého je níže. ZigBee využívá čtyři následující rámce[1],[9],[11]:

- **Beacon Frame**
- **Data Frame**
- **Acknowledgment Frame**
- **MAC Command Frame**

Beacon Frame

Tento rámeček slouží pro synchronizaci sítě a může jej vyslat pouze koordinátor. Beacon frame se vysílá v periodicky se opakujících intervalech. Používá se pro probouzení a synchronizaci zařízení. Interval synchronizačních sekvencí se pohybuje v intervalu od přibližně 15 ms po dobu až 15 minut. Jeho využití může být dvojitý. První typ, koordinátor probudí všechny zařízení, a pokud nějaké z nich má data k odeslání tak soupeří o to, aby data mohla odeslat. Tento typ komunikace je nevýhodný z hlediska soupeření zařízení o přístup k fyzickému médiu a odvysílání dat. Druhý typ je, že se pomocí Beacon Frame vzbudí jen určité zařízení, kterému koordinátor pošle dotaz, zda má data. Po případné komunikaci je zařízení zase usnáváno. Tento typ není moc výhodný z více hledisek, v případě, že zařízení dlouhou dobu nemá nic k odeslání, tak musí přijmout data a odeslat odpověď, že žádná data nemá. Tento způsob snižuje výdrž baterie, avšak i v tomto případě je výdrž baterie hodně vysoká. Druhý problém může být periodické procházení připojených zařízení. Může se stát, že nějaká data potřebujeme takřka okamžitě po jejich příjmu. Řešením může být tabulka kritických zařízení, které se mají obsluhovat nejčastěji, aby nedošlo ke ztrátě dat. Pro úplnost je na Obrázek 9 zobrazeno schéma rámce.

Octets: 2	1	4/10	2	variable	variable	variable	2
Frame control	Sequence number	Addressing fields	Superframe specification	GTS fields (Figure 38)	Pending address fields (Figure 39)	Beacon payload	FCS
MHR			MAC payload				MFR

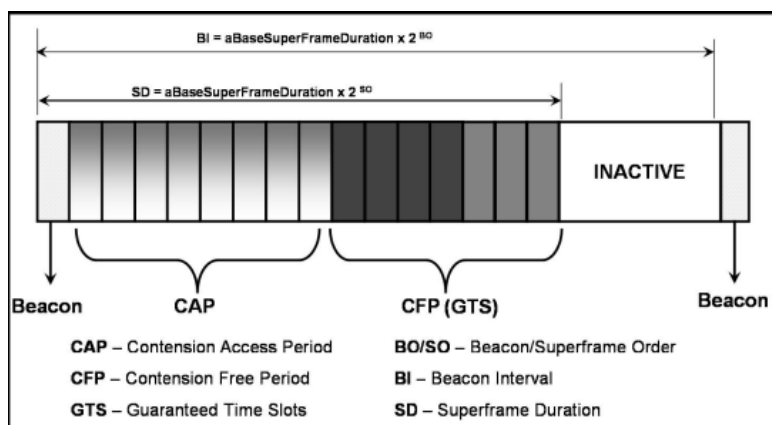
Obrázek 9 – Schéma Beacon frame [14]

Pro koordinátory je možnost vytvořit pomocí Beacon Frame tzv. Superrámeček.

Super rámec

Standart IEEE802.15.4 vytváří možnost pro koordinátory v použití tzv. Super rámec [18]. Superrámec může vyslat pouze koordinátor se strukturou danou konkrétní instancí koordinátora. Rámec je ohraničen synchronizačními Beacon intervaly, kde se všechna zařízení připraví na příjem dat. Superrámec je rozdělen na 16 stejných slotů. Každý superrámec definuje svou aktivní a neaktivní část. Během aktivní části rámce koordinátor komunikuje se všemi zařízeními v rámci PAN sítě - na obrázku sekvence CAP (Content Access Period). Zařízení odesílají data pomocí CSMA/CA přístupu k médiu. Další sekvencí aktivní části je volný čas pro odeslání dat – sekvence CFP (Content Free Period). Sekvence CFP startuje ihned po dokončení sekvence CAP a dělí se na tzv. garanční časové intervaly GTS (Guaranteed Time Slot), které slouží pro prioritní a pomalá zařízení. Maximální počet GTS je omezen na 7 slotů. Po skončení CFP sekvence následuje neaktivní sekvence, kdy všechna zařízení přejdou do úsporného režimu, a celá síť se uspí.

Doba trvání superrámce se vypočítá z doby trvání CAP sekvence, doby trvání CFP sekvence a startovací synchronizační sekvence Beacon. Do doby trvání se nepočítá neaktivní část. Schéma vysílaného super rámce je na Obrázek 10.



Obrázek 10 – MAC vrstva – schéma superrámce [18]

Data Frame

Na Obrázek 11 je zobrazeno schéma datového rámce. Data Frame je nejjednodušší vysílaný rámec. Slouží pro přenos užitečných dat, kterých může být v jednom rámci až 104 B. Část MHR je stejná se všemi vysílanými rámci [14].

Octets: 2	1	(see 7.2.2.2.1)	variable	2
Frame control	Sequence number	Addressing fields	Data payload	FCS
MHR			MAC payload	MFR

Obrázek 11 – Schéma Data Frame [14]

Acknowledgment Frame

Na Obrázek 12 je zobrazeno schéma Acknowledgment Frame - potvrzovacího rámce. ACK Frame je odeslán vysílacímu zařízení jako potvrzení příjmu ihned po přijetí dat v tzv. mrtvém čase. Jedná se o rámec, který se odesílá vždy po rámcich Data Request, Mac Command Request a Data Frame. Jedná se o jediný vysílaný rámec, který neobsahuje informace o zdrojové a cílové adrese v části nazvané MHR.[14]

Octets: 2	1	2
Frame control	Sequence number	FCS
MHR		MFR

Obrázek 12 – Schéma Acknowledgment Frame [14]

MAC Command Frame

Na Obrázek 13 je zobrazeno schéma řídicího MAC Command rámce. MAC Command Frame slouží k centralizovanému nastavení a řízení klientských zařízení v síti. Jedná se o vysílaný rámec koordinátorem, pomocí kterého koordinátor posílá zařízením síťové klíče a různé další konfigurační nastavení pro síť. Pro popis jsme byly vybrány dva řídicí rámce ze seznamu, který je na Obrázek 14, konkrétně Beacon request a Data request. Tyto rámce byly vybrány z důvodu nejčastějšího vysílání v ZigBee síti a jedná se o velmi důležité řídicí rámce[14]

Octets: 2	1	(see 7.2.2.4.1)	1	variable	2
Frame control	Sequence number	Addressing fields	Command frame identifier	Command payload	FCS
MHR			MAC payload		MFR

Obrázek 13 – Schéma MAC Command Frame [14]

Command frame identifier	Command name	RFD		Subclause
		Tx	Rx	
0 x 01	Association request	X		7.3.1.1
0 x 02	Association response		X	7.3.1.2
0 x 03	Disassociation notification	X	X	7.3.1.3
0 x 04	Data request	X		7.3.2.1
0 x 05	PAN ID conflict notification	X		7.3.2.2
0 x 06	Orphan notification	X		7.3.2.3
0 x 07	Beacon request			7.3.2.4
0 x 08	Coordinator realignment		X	7.3.2.5
0 x 09	GTS request			7.3.3.1
0 x 0a–0 x ff	Reserved			—

Obrázek 14 – Seznam MAC Command rámců [14]

Beacon Request

Rámcem se vysílá ve dvou možných alternativách, které jsou závislé na typu zařízení. Prvním typem je rámcem vysílaný koordinátorem. Koordinátor tento typ rámcem vysílá při vytváření ZigBee sítě. Jedná se o aktivní prohledání okolního prostředí na ostatní koordinátory. Koordinátor odvysílá pouze jeden tento frame, adresovaný na broadcast PANID. Pokud do stanoveného času nedostane odpověď, je daný kanál volný a může na něm založit novou ZigBee síť. V případě, že obdrží odpověď od jiného koordinátora, přepne na další kanál, na kterém může vysílat a zopakuje odeslání Beacon Request. Druhou alternativou je vysílání Beacon Request u koncového zařízení. Koncové zařízení vysílá Beacon Request neustále a postupně periodicky na všech kanálech, na kterých může komunikovat. Vysílání Beacon Request je považováno za aktivní hledání sítě PAN, ke které by se zařízení mohlo připojit. V případě, že zařízení nalezne vhodnou PAN síť, ke které se může připojit, obdrží od koordinátora Beacon Frame.

Data request

Data request je řídicí rámcem vysílaný od koncového zařízení ke koordinátorovi a od koordinátora ke koncovému zařízení. Jedná se o požadavek na data.

2.4.3 Zabezpečení na MAC vrstvě

Standart IEEE 802.15.4 definuje různé typy zabezpečení. Standart definuje, jak má být zabezpečení provedeno, ale již nedefinuje, jak mají být klíče spravovány v zařízeních či jaké zabezpečovací techniky mají být použity. Tyto problémy jsou (a musí) být řešeny ve vyšších vrstvách definované ZigBee aliancí [19].

Během přenosu přes fyzické médium může docházet k chybnému přenosu dat vlivem náhodných či periodických rušivých jevů. Pro odhalení chyb se používá CRC kódování, kdy se každý vyslaný rámec doplní vygenerovaným cyklickým redundantním kódem. V ZigBee je použit polynom ve tvaru:

$$C_{16}(x) = x^{16} + x^{12} + x^5 + 1$$

Pro zabezpečení dat se používá primárně šifrovací algoritmus AES 128bit. Pomocí AES algoritmu se nešifrují jen přenášená data, ale i ověřovací údaje. Tento koncept se nazývá Data Integrity s použitím metody nazývané MIC (Message Integrity Code), občas nazývané jako MAC (Message Authentication Code). Tento algoritmus zabezpečuje integritu hlavičky a přiložená data. Využití spočívá v tom, že se přijatá data od uzlu zkontrolují, zda nebyla narušena a to tím, že se použije síťový klíč (viz. zabezpečení síťové vrstvy) a v případě, že se data neshodují, tak se poté použije skrytý klíč pro případ, že by se do sítě chtěl připojit nový uzel. Když se data nebudou shodovat, tak se zpráva zahodí. MAC (Message Authentication Code) může mít velikost 32,64 nebo 128 bitů, ale vždy je vytvořen 128-bitovým AES kódováním.[1][9][10][11][13][19]

Šifrování dat

Přenášená data lze i šifrovat proti zcizení. IEEE standart definuje tři bezpečnostní režimy.

1. režim: nezabezpečený přístup
2. režim: přístup na základě práv
3. režim: zabezpečený přístup

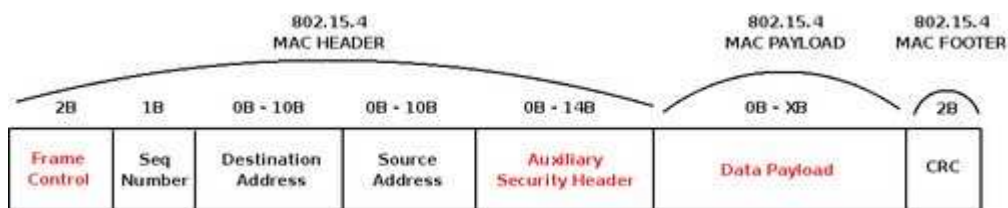
Zabezpečený režim podporuje více služeb zabezpečení, mezi které patří :

- odmítnutí opakujících se rámců
- přístup na základě práv
- šifrování dat pomocí AES 128bit (Advanced Encryption Standard)
- použití MIC (Message Integrity Code)

Šifrovací standart AES nahrazuje dříve používaný a v dnešní době nespolehlivý standart DES (Data Encryption Standart), který byl pouze 64bit, kde posledních 8 bitů bylo kontrolních a 56 efektivních. Tento algoritmus je možné prolomit do 24h. Algoritmus AES využívá tzv. symetrického klíče – stejný klíč je použit pro šifrování i dešifrování dat. Délka

klíče může být 128bit, 192bit nebo až 256bit. Metoda šifruje data postupně v blocích s pevnou délkou 128bitů. Šifrování je velmi rychlé.

Dále je možné použít pro šifrování algoritmus MIC. MIC je kryptografický kontrolní součet, který je zahrnut do vysílacího rámce. Po přijetí se MIC zkontroluje na shodu, pokud se data během přenosu změnila, či pokud se útočník snaží podstrčit vlastní rámec tak mikrokontrolér rámec zahodí. Na Obrázek 15 je zobrazeno schéma základního MAC rámce s vyznačením možností zabezpečení rámce. Na Obrázek 16 je zobrazeno schéma zabezpečení MAC hlavičky pomocí rozšíření bezpečnostní kontroly.



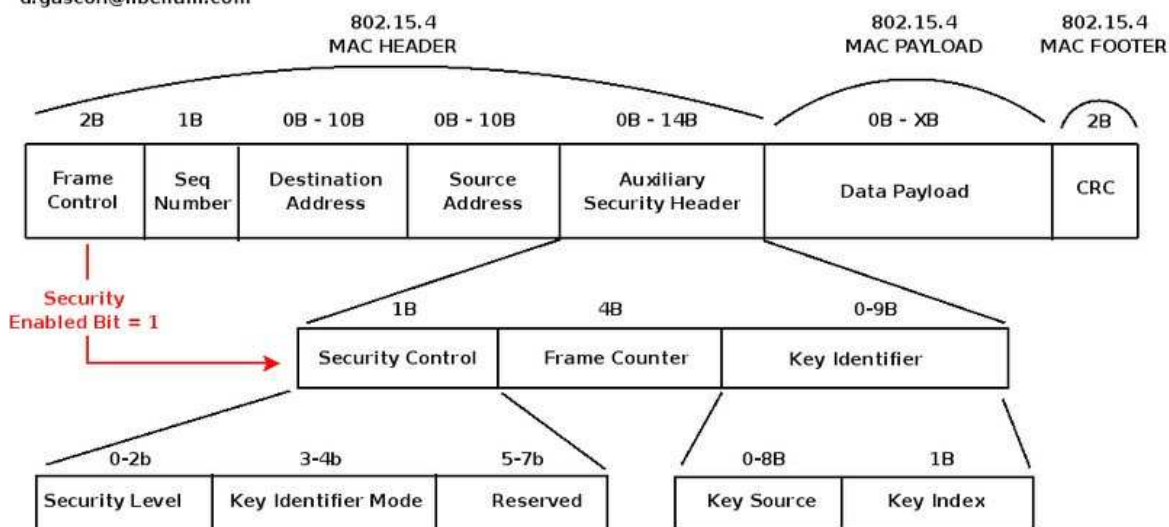
Obrázek 15 – Schéma MAC rámce [19]

IEEE802.15.4 definuje tři možné typy zabezpečení na MAC vrstvě:

- Frame Control – umístěno v hlavičce MAC
- Rozšířená bezpečnostní kontrola – v MAC hlavičce
- Zabezpečení samotných dat

Security in the IEEE 802.15.4 MAC FRAME
<http://www.sensor-networks.org>

Author: David Gascón
 d.gascon@libelium.com



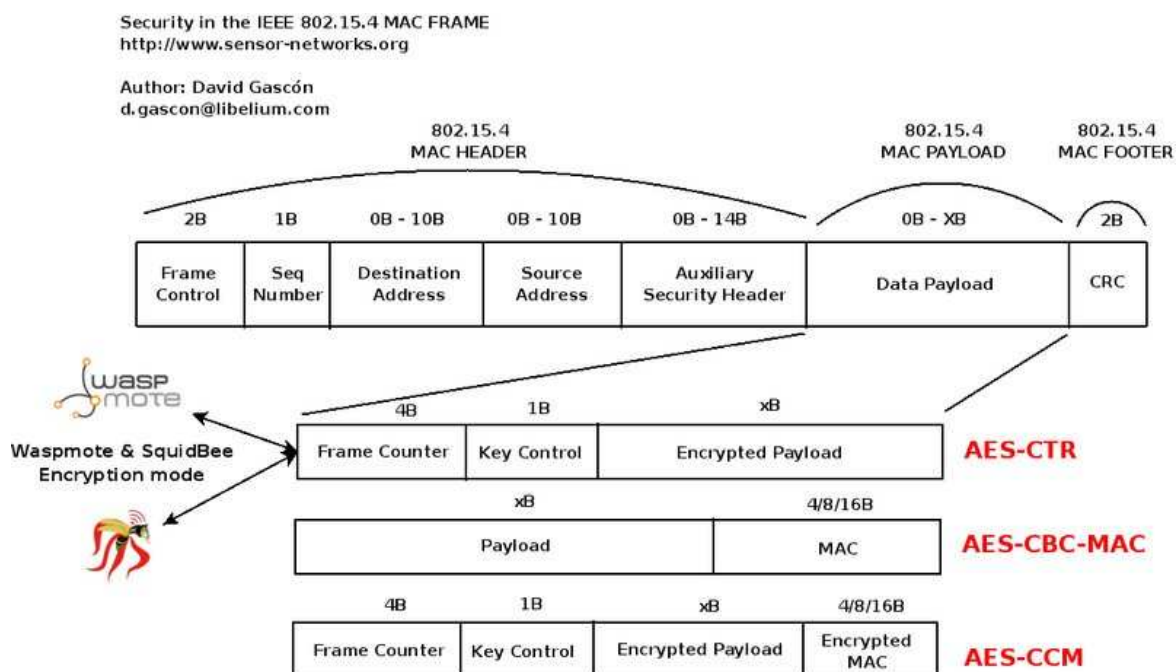
Obrázek 16 – Schéma části rámce pro zabezpečení MAC hlavičky [19]

Rozšířené zabezpečení dat je povoleno pouze když je ve Frame Control povoleno. Má tři části:

- 1) Security control (1B) : Část ve které se nastavuje aktuální použité kódování
- 2) Frame Counter (4B) : Obsahuje unikátní číslo rámce. Zabezpečuje ochranu zprávy před opakovaným přeposláním
- 3) Key Identifier (0-9B) – Specifikuje klíč který je potřeba pro komunikaci s uzlem

Typy zabezpečení – Security Control:

Na Obrázek 17 jsou zobrazeny struktury rámců podle typu použitého zabezpečení přenášených dat.



Obrázek 17 – Schéma části rámce pro různé zabezpečení dat [19]

Identifikace klíče – Key Identifier Mode

- 0x0 – Klíč je znám vysílačem i přijímačem, ve zprávě není uveden
- 0x1 – Klíč má 1B index klíče a je určen z pole identifikátoru klíče a macDefaultKeySource
- 0x2 – Klíč má 1B index klíče a je určen 4B z identifikátoru klíče
- 0x3 - Klíč má 1B index klíče a je určen 8B z identifikátoru klíče

Data Payload:

Zabezpečení dat může mít tři různé konfigurace v závislosti na definované oblasti

- a) **AES-CTR** – Všechna data jsou šifrována pomocí 128bit AES algoritmu. Pro každou zprávu se nastaví unikátní ID zprávy a Key Control.
- b) **AES-CBC-MAC** - Informace o pravosti rámce. Informace jsou vygenerovány z MAC hlavičky a poskytuje příjemci informace o tom, že rámec byl vyslán z důvěryhodného uzlu.
- c) **AES-CCM** – Jedná se o směsici předchozích dvou.

Acces control list:

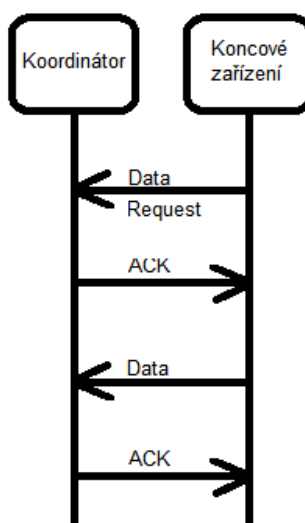
Každý uzel sítě má vlastní seznam důvěryhodných zařízení ve kterém jsou uloženy:

- a) **Address** – adresa uzlů s kterými může daný uzel komunikovat
- b) **Security Suite** – typ používaného zabezpečení pro přenos dat
- c) **Key** – Klíč pro AES algoritmus
- d) **Last Initial Vector** – číslo předchozího přijatého rámce. Díky této položce se zamezí opakovaným požadavkům na data (při možných pokusech o hack sítě)

Schéma komunikace koordinátor ↔ koncové zařízení

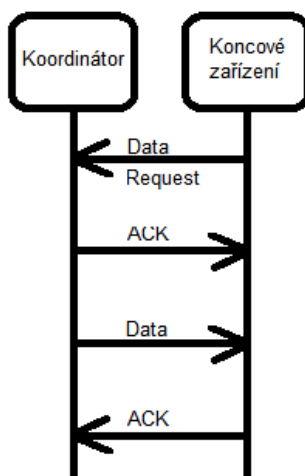
a) Komunikace bez Beacon Frame

Přenos dat ke koordinátoru (viz Obrázek 18) probíhá tak, že se koncové zařízení periodicky dotazuje na koordinátora. Při dotazu je nejdříve poslán požadavek na odeslání dat. Po odeslání požadavku odeslání dat koncové zařízení čeká na potvrzovací rámec. Po přijetí potvrzovacího rámce jsou data poslána a z koordinátora je odeslán potvrzovací rámec ACK [1],[19],[20].



Obrázek 18 – Komunikace koncové zařízení → koordinátor bez Beacon Frame

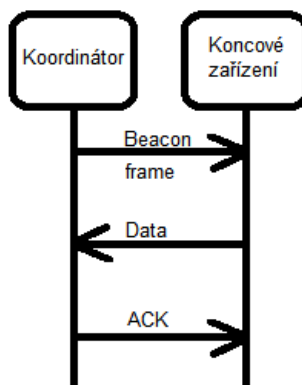
Přenos dat z koordinátora do koncového zařízení (viz Obrázek 19) probíhá tak, že se koncové zařízení odešle dotaz, zda má koordinátor pro něj nějaká data. V případě, že data jsou připravena, odešle potvrzovací rámec a začne odesílat data. Po odeslání očekává od koncového zařízení potvrzovací rámec [1],[19],[20].



Obrázek 19 – Komunikace koordinátor → koncové zařízení bez Beacon Frame

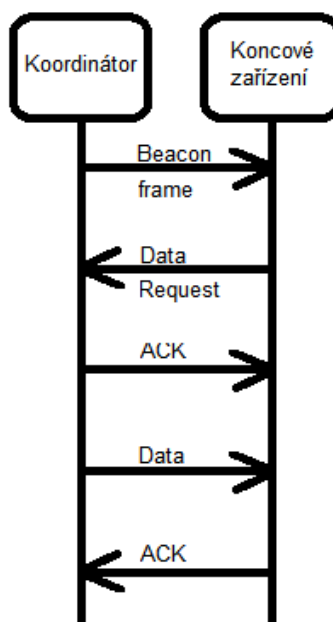
b) Komunikace s použitím Beacon frame

Přenos dat ke koordinátoru (viz Obrázek 20) probíhá tak, že nejdříve koordinátor vyšle Beacon Frame, který zachytí všechny zařízení a synchronizují se. Poté zařízení vygenerují náhodný čas a poslouchají fyzické médium, zda je volno. Zařízení, které začne vysílat jako první, odešle své data do koordinátora, který je přijme a odešle potvrzovací rámeček [1],[19],[20].



Obrázek 20 – Komunikace koncové zařízení → koordinátor s Beacon Frame

Přenos dat z koordinátora (viz Obrázek 21) do koncového zařízení probíhá tak, že koordinátor odešle Beacon Frame a přijímá zprávu od koncového zařízení. Po přijetí zprávy odešle potvrzovací rámeček a začne odesílat data. Koncové zařízení data přijme a odešle potvrzovací rámeček [1],[19],[20].



Obrázek 21 – Komunikace koordinátor → koncové zařízení s Beacon Frame

2.4.4 Síťová vrstva (NWK)

Mezi hlavní úkoly síťové vrstvy patří směrování a zabezpečení rámců k jejich cílům, Síťová vrstva ZigBee koordinátoru přiděluje síťové adresy nově připojeným zařízením. Síťová vrstva prohledává okolí a shromažďuje si informace o okolních zařízeních [1],[9],[11],[13][15],[20].

Seznam možností síťové vrstvy dle typu zařízení:

Koordinátor, Router, koncové zařízení

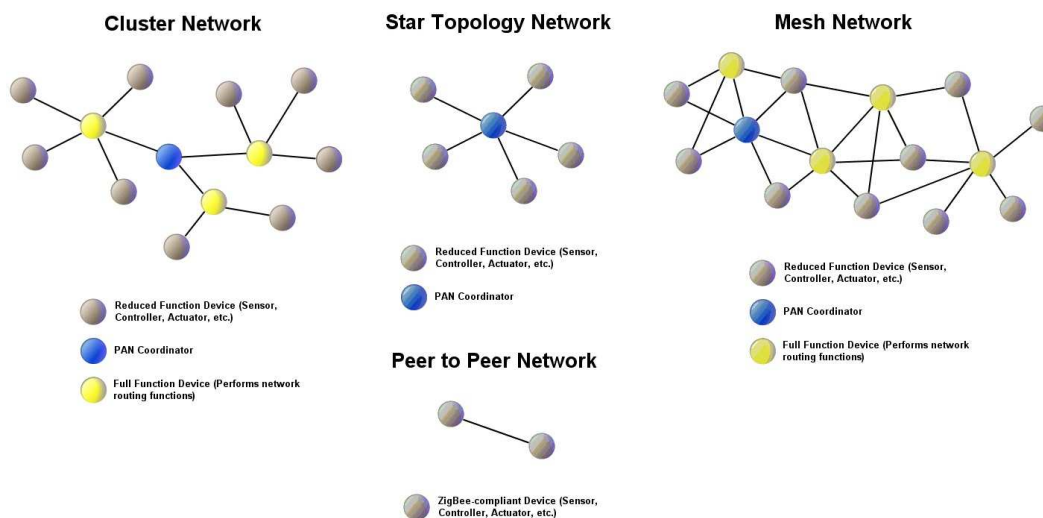
- Vysílá požadavek pro připojení do existující sítě a odpojení z ní
- Prohledávání okolních zařízení
- Hledání nejlepší cesty pro přenos dat – Routování
- Řízení příjmu

Koordinátor navíc

- Zakládá a spouští síť
- Vyřizuje požadavky od zařízení pro požadavek připojení či odpojení
- Přiřazuje adresy zařízením

Topologie sítě

Topologie sítě je možnost, jak mohou býtí zařízení spojena. ZigBee standart definuje tři možné topologie: Hvězda (v obrázku Star), Cluster a Mesh. Na Obrázek 22 jsou graficky zobrazeny základní struktury topologií.



Obrázek 22 – Topologie ZigBee sítě [31]

Topologie Peer to Peer (rovný s rovným):

Nejjednodušší typ sítě. V síti jsou pouze dva prvky, které mezi sebou komunikují. Ve specifikaci tento typ sítě není zaveden, avšak tento typ sítě lze vytvořit v případě, že jedno ze zařízení bude koordinátor a druhé bude koncové zařízení. Mezi prvky lze usadit tzv. opakovače signálu, které pouze přijaté data poslechnou a odešlou dále, čímž se dá zvýšit dosah sítě. Nahrazuje pouze kabel.

Topologie typu Mesh:

Nejsložitější typ síťového propojení. Každé zařízení může komunikovat se všemi zařízeními. Výhoda této sítě je ta, že v případě výpadku jednoho či více klíčových zařízení (routeru) je síť stále v provozu díky možné komunikaci přes okolní prvky sítě. I přes velkou odolnost proti výpadkům sítě je potřeba klíčová zařízení jako koordinátor a router napájet externím zdrojem zálohovaným bateriemi. Díky tomuto řešení dojde k velké spolehlivosti sítě nezávisle na napájení a vysoké odolnosti a schopnosti komunikovat i přes výpadky určitých členů, kde se routováním před odesláním dat, zjišťuje nejvhodnější cesta. Zároveň tento typ sítě umožňuje komunikaci zařízení peer to peer mezi okolními členy, takže se zbytečně nezatěžují kritické prvky[11].

Topologie typu Hvězda:

Nejjednodušší typ víceprvkové sítě. Tato síť je velmi náchylná na výpadky klíčových zařízení. Zařízení mohou komunikovat pouze přes koordinátora. V případě výpadku koordinátora dojde k totálnímu výpadku sítě a její nefunkčnosti. V tomto případě, je nutnost napájení koordinátora z tvrdého zdroje tj. z elektrické napájecí sítě se zálohou v bateriích. Nevýhoda této sítě oproti struktuře Mesh je ta, že veškerá komunikace funguje přes koordinátora, což výrazně ovlivňuje propustnost sítě, proto se tato struktura využije v aplikacích, kde dochází k periodickému čtení dat ve velkém rozmezí. Příkladem může být snímání teploty z místností, kde nevádí, zda zařízení odešle data se zpožděním 30s a případně sepne servomotor na uzavření topení za minutu.[11]

Topologie Cluster (Strom) :

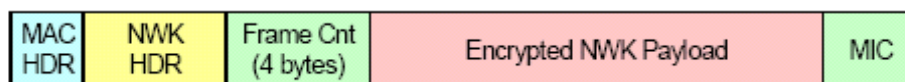
Tato topologie se též nazývá stromová. Jedná se o rozšíření topologie hvězda, kde v krajních bodech dojde k nahrazení koncových zařízení za router. V případě výpadku jednoho či více klíčových prvků sítě dojde k tomu, že určité sekce sítě jsou alespoň částečně funkční přes router, který funguje.[11]

Adresace v ZigBee síti

Adresa slouží k jednoznačné identifikaci zařízení v rámci sítě. ZigBee zařízení podporují celkem dva možné typy adres – plnou adresu a zkrácenou. Plná adresa je 64 bitová a lze jí použít například při komunikaci přes internet jako bezdrátové čidlo. Při 64bitové adrese je maximální počet zařízení $2^{64} - 2$ kde první adresu (0x00000000) má koordinátor a poslední (0xFFFFFFFF) Broadcast. Broadcast je adresa, na které naslouchají všechny zařízení. Zařízení totiž při příjmu rámce nejdříve maskují adresové pole vlastní adresou a v případě, že vymaskovaná adresa se shoduje s původní, dojde ke zpracování dat. V případě, že se odešlou data na Broadcast, kde adresa je 64 logických „1“, tak při maskování jakýmkoliv číslem (adresou) vyjde původní číslo (adresa zařízení). Dále ZigBee podporuje zkrácené adresování, které je pouze 16 bitové a je jednodušší. Využívá se v rámci jedné sítě, kde potom počet zařízení nesmí nepřekročit 65536 zařízení. Dále se používá 16bitový identifikátor sítě (PAN ID). Díky 16bitovému identifikátoru lze paralelně provozovat až 65536 sítí. [11]

Zabezpečení síťové vrstvy

Možností, jak zabezpečit komunikaci na síťové vrstvě proti narušení je seznam důvěryhodných zařízení. V principu to je to samé, jak funguje seznam povolených MAC adres u Wifi. Síťová vrstva používá mírně modifikovaný CCM mód používaný na MAC vrstvě. Využívá všechny vlastnosti CCM a přidává navíc možnost využít pouze část CCM. Tento princip zjednodušuje zabezpečení, protože se klíč neváže pouze na jeden zabezpečovací algoritmus, ale lze použít na každý rámec jiný šifrovací algoritmus [1][19]. Schéma zabezpečeného síťového rámce je na Obrázek 23.



Obrázek 23 – Schéma CCM zabezpečeného rámce na NWK vrstvě[1]

Zabezpečení obstarává služba SSP, která se po zapnutí sítě spojí s protistranou a získá její zabezpečovací klíč, pomocí šifrovacího algoritmu zahesluje data a odešle. SSP služba je řízena z aplikační vrstvy, kde jí je předán klíč a také šifrovací algoritmus, který se na zabezpečení bude používat. Celý tento proces je řízen pomocí ZDO.

Bezpečnostní klíče

Bezpečnostní klíče slouží k šifrování komunikace mezi uzly. Používají se klíče:

- Hlavní klíč - Hlavní klíč je před započítím komunikace uložen ve všech zařízeních a slouží jako zabezpečený přenos pro spojovací klíče.
- Spojovací klíč – Unikátní klíč pro každý pár zařízení
- Síťový klíč – klíč o délce 128b. Sdílený klíč pro všechny zařízení v síti. Zařízení, které má požadavek na připojení do existující sítě připojit tento klíč musí znát. Klíč je generován v různých časových intervalech důvěryhodným zařízením – většinou koordinátorem.

Použití klíčů:

- Domácí provoz – Koordinátor sdílí pouze síťový klíč. Umožňuje snížit paměťové nároky na minimum.
- Komerční provoz – Koordinátor sdílí hlavní a spojovací klíč. Tento způsob je velmi náročný na paměť.

2.4.5 Aplikační vrstva ZigBee

Aplikační vrstva ZigBee se skládá z uživatelského programu, aplikační pomocné podvrstvy, ZDO (ZigBee Device Object) a Uživatelských aplikačních objektů.[3],[20][32]

Aplikační pomocná podvrstva

Aplikační podvrstva (APS) je vrstva, která se stará o párovací (Binding) tabulky, které obsahují záznam, jak propojit dvě zařízení v závislosti na jejich parametrech, jejich potřebách a podporovaných službách. Dále díky párovacím tabulkám přeposílá zprávy dalším zařízením. Díky tomuto způsobu ZigBee podporuje tzv. multi-hop komunikaci.

Aplikační objekt

Na jednom ZigBee zařízení může pracovat více aplikací (senzory, spínače, apod.), Každá tato jednotlivá aplikace se nazývá tzv. aplikační objekt.

ZDO

ZDO definují přesnou roli zařízení v síti. Definují, zda je zařízení Koordinátor, Router či Koncové zařízení. Dále ZDO vytváří a odpovídá na žádosti o spojení mezi zařízeními, zřizuje zabezpečené spojení mezi dvěma prvky sítě, či také zajišťuje hledání v síti a spravuje používané zabezpečovací klíče.

Koordinátor

Prvek ZigBee sítě, zajišťující založení, nastavení a start ZigBee sítě. Koordinátor vybírá komunikační kanál, na kterém mohou zařízení komunikovat, nastavuje PANID sítě, přijímá požadavky na připojení a v případě šifrování, sdílí zabezpečovací klíče. V ZigBee síti může existovat pouze jeden koordinátor. Jelikož koordinátor je klíčovým prvkem na kterém záleží celá síť, neměl by (a nesmí) být napájen bateriově.

Router

Prvek ZigBee sítě, zajišťující rozšíření sítě na vyšší počet prvků. Důvod vytvoření routeru je takový, že každý prvek má omezený počet potomků, se kterými může komunikovat. V ZigBee síti může být více zařízení typu router. Router by neměl být jako koordinátor napájen z baterií, jelikož na něm stojí celá větev ZigBee sítě.

End-Device

Koncový prvek sítě. Typický koncový prvek je čidlo teploty, senzor pohybu apod. End-Device je jediným možným prvkem v ZigBee síti, který může být napájen pouze bateriově.










Uživatelské aplikační objekty

Uživatelské aplikační objekty implementují do sítě konkrétní požadavky pro aplikaci. Tyto požadavky se nazývají ZigBee Aplikační profily. Každý profil definuje typy použitých zařízení, typy zpráv a jejich formáty. Každý profil má vlastní 16bitové identifikační číslo, které přiděluje ZigBee Alliance. Aplikační profil je dohoda o formátech zpráv a akcích, které umožní zařízením od různých výrobců komunikovat mezi sebou a tím zajistit kompatibilitu. Každé zařízení, než je vydáno na trh, musí vyhovovat předem danému profilu. [21]

ZigBee aliance definuje dva druhy profilů:

- Průmyslové specifické
- Veřejné

Veřejné profily pokrývají základní využití technologie v různých oborech. Specifikace si definuje výrobce sám, avšak musí profil projít schválením. V současné době jsou k dispozici následující ZigBee profily uváděné originálními názvy [21]:

Název Profilu	Popis	Logo
ZigBee Building Automation	Profil používaný pro komerční použití v ovládní budov	
ZigBee Remote Control	Profil určen pro náhradu infračervených dálkových ovladačů	
ZigBee Smart Energy	Profil určený pro vysoce úsporné projekty	
ZigBee Health Care	Profil zaměřen pro nekritické služby ve zdravotnictví a v oblasti fitness či péči o chronicky nemocné. Profil vyhovuje standardu ISO/IEEE 11073. Typická zařízení jsou měřiče cukru či informativní EKG.	
ZigBee Home Automation	Profil určen pro domácí automatizaci dostupné pro inteligentní dům jako je automatické rolování žaluzií, osvětlení, otevírání garážových vrat atd.	
ZigBee Input Device	Profil určen pro využití ZigBee technologie v ovládacích prvcích pro osobní počítače.	
ZigBee Telecom Services	Profil určen pro služby typu placení či lokalizace pomocí mobilních telefonů v budovách. Předpokládá se, že telefon bude mít speciální ZigBee SIM kartu	
ZigBee 3D Sync	Profil určený pro bezdrátové brýle pro sledování 3D videa	
ZigBee Retail Services	Profil pro chytré monitorování, kontroly a automatizace objednání a doručení zboží.	

Tabulka 3 - ZigBee profily[21]

3 Vývojový kit

3.1 Úvod

Vývojový kit je deska plošných spojů, jejímž hlavním účelem je vytvořit hardware pro pozdější vývoj zařízení s ZigBee funkcionalitou. Pro možnost vytvořit zařízení s jistou funkčností jsou na DPS přidány tlačítka, sedmissegmentový displej, led diody a sériová linka.

3.2 Výběr ZigBee modulu

Klíčovým prvkem vývojového kitu je tzv. ZigBee modul. ZigBee modul obvykle obsahuje na plošném spoji mikrokontrolér, RF Transceiver a výstupní RF části (např. plošná anténa). ZigBee modul je připraven pro zapájení na uživatelskou desku plošných spojů. Jako ilustrace může sloužit Obrázek 24, kde je zobrazen modul od firmy Atmel, typ ATZB-24-A2. Rozměry modulu jsou 18.8 x 13.5 x 2.0 mm.



Obrázek 24 – ZigBee modul firmy Atmel [26]

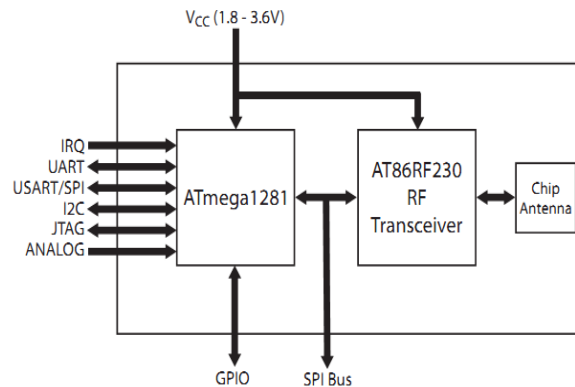
V době vývoje kitu bylo možné na trhu sehnat několik modulů od různých výrobců. Nakonec byl pro aplikaci vybrán modul od firmy Atmel. Hlavním důvodem pro výběr daného modulu byl fakt, že na pozici mikrokontroléru je použit čip ATmega1281 z řady AVR, se kterým umí autor práce pracovat. To by mělo usnadnit vývojové práce, kdy bude možné soustředit pouze na problematiku ZigBee.

Nejdůležitějším prvkem modulu je RF transceiver, což je specializovaný integrovaný obvod navržený pro realizaci všeho potřebného pro bezdrátovou komunikaci. Ve zvoleném modulu je transceiver AT86RF230. Pro úplnost je v příloze uvedeno stručné srovnání transceiverů, ze kterého vyplývá, že AT86RF230 má z dostupných RF čipů nejlepší parametry – Spotřebu ve spícím režimu, citlivost, či vysílaný výkon.

3.3 ZigBee modul Atmel typ ATZB-24-A2

Na Obrázek 25 je zobrazeno vnitřní zapojení ZigBee modulu. Pro potřeby řízení je v ZigBee modulu použit mikroprocesor ATMEGA1281V, který má k dispozici 128kB paměti programu 8kB RAM a 4kB EEPROM. Označení „V“ u mikroprocesorů značí technologii mikroprocesoru, která dovoluje nižší napájecí napětí, u tohoto typu až do 1,8V. Mikroprocesor má jako samotný čip k dispozici velké množství periférií a vstupně/výstupních bran, pro uživatelské rozhraní a pro připojení komponent jich je však již mnohem méně. Pro vývojáře je k dispozici pouze 9 GPIO linek, jedna sběrnice 1wire, I2c, UART, USART/SPI a JTAG. Uvnitř modulu je zapojen přes SPI sběrnici RF

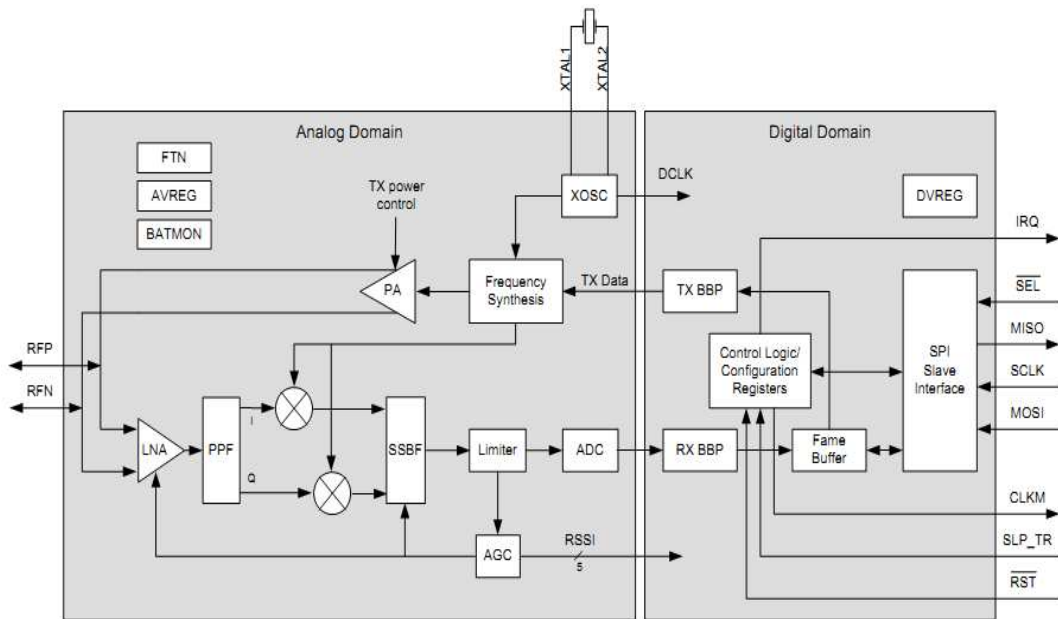
transceiver AT86RF230, který na rozdíl od modelu 231 nepodporuje šifrování komunikace. Z tohoto důvodu je patrné, že tyto moduly jsou hlavně pro prototypové a studijní účely a pro použití v aplikacích, kde nevádí, že jsou přes fyzické médium přenášena nezabezpečená data. Dále na modulu je na rozdíl od modelu ATZB-24-B0 integrována plošná anténa. Vysílač s plošnou anténou má citlivost -101 dBm [23].



Obrázek 25 – Schéma vnitřního zapojení ATZB-24-A2 modulu [23]

Popis RF transceiveru AT86RF230

Blokové schéma je na Obrázek 26. Ve schématu je patrné rozdělení na dvě dominantní části – digitální a analogovou. Digitální část obsahuje konfigurační registry, přijímové a vysílací vyrovnávací buffery a je spojena s mikrokontrolérem přes sběrnici SPI. Analogová část obsahuje obvody pro frekvenční syntézu a modulaci vysílaného signálu a obvody pro detekci a demodulaci přijímaného signálu. Dále obsahuje společnou VF část, ze které je patrné, že modul může fungovat pouze v poloduplexním režimu (poloduplexní režim, je režim, kdy zařízení může v jeden okamžik buď pouze přijímat nebo pouze vysílat data). Podrobný popis blokového schéma lze najít v dokumentaci k RF transceiveru [24].

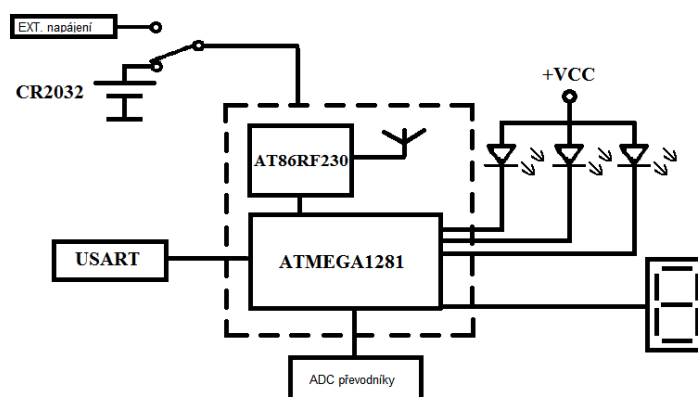


Obrázek 26 – Blokové schéma RF modulu AT86RF230 [24]

Srovnání RF čipů

Tabulka srovnání RF čipů a jejich parametrů od výrobců Atmel, Freescale, Texas Instruments, Oki a Ubec je v příloze. Při srovnání parametrů dostupných RF čipů je jasné, že Atmel produkuje jedny z nejlepších čipů. Pro naše požadavky by byl vhodný čip AT86RF231, avšak nesehnali jsme jej na ZigBee modulu, proto je použit AT86RF230, který se od RF231 liší jen v podpoře AES zabezpečení, podporou nižšího napájecího napětí, menší spotřebou v RX/TX režimu a vyšší maximální přenosovou rychlostí, která dosahuje až 2 Mbit/s. Alternativní volbou by mohlo být využití RF transceiveru CC2520 od firmy Texas Instruments, který má srovnatelné parametry s produktem od firmy Atmel. Při vysílání sice vysílá s úrovní +5dbm, avšak má nižší citlivost příjmu a vyšší odebíraný proud při RX/TX režimu [5],[6],[24].

3.4 Blokové schéma zapojení



Obrázek 27 – Blokové schéma vývojového kitu

Na Obrázek 27 je zobrazeno blokové zapojení vývojového kitu. Hlavní komponentou DPS je ZigBee modul, který je na obrázku označen čárkovanou čarou. ZigBee modul může být napájen ze dvou nezávislých napájecích zdrojů, jejichž volba se volí pomocí zkratovací propojky. Z modulu jsou na DPS vyvedeny vstupy pro ADC převodníky a sériová linka. K desce jsou rovněž připojeny 3 barevné diody a sedmissegmentový displej.

3.5 Elektrické schéma zapojení

Kompletní elektrické zapojení vývojového kitu je v příloze.

Srdcem vývojového kitu je ZigBee modul od firmy Atmel s kódovým označením ATZB-24-A2. Uvnitř modulu se skrývá mikrokontrolér řady AVR ATmega1281V a vysokofrekvenční RF čip AT86RF230. Vývojový kit má dvě možnosti napájení. První možností je napájení pomocí externího zdroje se stejnosměrným napětím 7-15V. Napětí je z konektoru K375A přivedeno na ochrannou diodu D1, která slouží jako ochrana proti prepólování. Kondenzátory C1 a C9 jsou filtrační kondenzátory. Pro stabilizaci napětí byl použit LDO stabilizátor od firmy Texas Instruments s kódovým označením TVL1117-33CDCYR a označením ve schématu U2. Stabilizátor má referenční hodnotu 3,3V. Kondenzátor C10 slouží jako vyhlazovací kondenzátor. Vyhlazené a stabilizované napětí je přivedeno na jumper J3, kde se pomocí zkratovací propojky volí mezi bateriovým a externím napájením. Druhou možností je napájení z baterie CR2032. Z důvodu klesání napětí při vybití baterie je nutné použít stabilizátor. Stabilizátor je nutné použít se zvyšujícím měničem. Pro mikroprocesor by ani nebyl potřeba, protože pracuje až do 1,8V, avšak je nutný pro ostatní součástky, jako je řadič sedmissegmentového lcd displeje, či pro převodník na sériovou linku. Jako stabilizující měnič je použit integrovaný obvod od firmy Texas Instruments s označením TSP60125 s výstupním napětím 3,3V. Pro jeho zapojení je použito doporučené zapojení z dokumentace firmy. V pouzdře integrovaného obvodu se nacházejí celkem dva měniče, které jsou zapojeny paralelně. Označení ve schématu je U4. Ve schématu je přidán také jumper J2, pro možnost vypnutí funkce měniče pomocí zkratovací propojky. Měnič je vypnut, když je na vstupu ENABLE přivedena logická zem.

Výstup měniče je přiveden na jumper J3. Na signalizaci správné funkce napájecího zdroje je připojena zelená dioda D2 a je k ní připojen předřadný rezistor R2 o hodnotě 470 Ω. Pro komunikaci pomocí sériové linky je nutno použít převodník. Jako převodník byl použit obvod od firmy Texas Instruments s označením MAX3232MPWREP. K zapojení převodníku je použito doporučené zapojení z dokumentace firmy. Kondenzátory C11-C14 jsou keramické kondenzátory pro nábojovou pumpu. C15 a C16 jsou kondenzátory filtrační a vyhlazovací pro stabilizaci napětí pro integrovaný obvod. Označení integrovaného obvodu ve schématu je U5. K mikrokontroléru jsou připojeny tři barevné diody a sedmsegmentový displej. Světelné LED diody jsou zapojené tak, že se v mikrokontroléru spínají logickou nulou. Rezistory R3-R5 jsou předřadné rezistory a volí se podle barvy diody. Druhou možností optického výstupu je sedmsegmentový displej. Displej je ovládán pomocí převodníku CD4094CB z firmy Texas Instruments. Integrovaný obvod 4094 je sério-paralelní převodník. Ovládání posuvného registru je pomocí 4 vodičů. Signál Strobe, který převede data z vnitřních D-registrů do výstupního bufferu. Signál DATA jako vstup dat. Posledním signálem je OE, kde se signál fyzicky zapíše z výstupního bufferu na výstupní porty. Označení integrovaného obvodu ve schématu je U3. Jako zobrazovací prvek je použito sedmsegmentového displeje s předřadnými rezistory. Hodnota rezistorů R9-R16 se volí podle typu displeje. Displej HDSP5501 je displej se společnou anodou. Označení displeje ve schématu je D7. Jako vstupy jsou na vývojovém kitu zapojeny celkem tři tlačítka. Tlačítko RESET sloužící k resetu mikrokontroléru. Tlačítko RESET při stisku uzemní vstup mikrokontroléru. Zapojení tlačítka je přes Pull-UP rezistor R1 s hodnotou 10 kΩ. Označení ve schématu je S1. Další vstupy jsou dvě uživatelské tlačítka. Tlačítka jsou připojeny přes Pull-up rezistory R7 a R8 s hodnotou 10 kΩ. Tlačítka jsou ve schématu označena S2 a S3. Jako dalšími vstupy jsou tři vstupy pro AD převodníky označení AD1-AD3. Pro správnou funkci převodu je nutné na vstup AVCC přivést referenční napětí. Vstup referenčního napětí je připojen přes sériově zapojenou tlumivku L1 s hodnotou 10 μH, která slouží s kondenzátory C4 a C7 jako pasivní filtr. Posledním vstupem je 10pinový programovací JTAG konektor s typovým označením MLW10G.

3.5.1 Napájení

Pro napájení vývojového kitu byla zvolena kombinace bateriového a externího napájení, které se volí pomocí zkratovací propojky.

Bateriové napájení

Pro bateriové napájení byla zvolena knoflíková baterie CR2032, která má při plném nabití napětí cca 3,3V, avšak s vybíjením napětí hodně klesne na hodnotu cca 2,8 V což by bylo dostačující pro napájení mikroprocesoru, který pracuje až do 1,8 V, avšak ne pro napájení převodníku sériové linky a posuvného registru 4094B pro sedmisegmentový displej. Proto bylo nutné použít stabilizaci napětí na hodnotu 3,3 V. Člověk by mohl namítnout, že by se na stabilizaci mohl použít klasický stabilizátor, avšak klasický LDO (low drop out – stabilizátor s malým úbytkem napětí) není schopen pracovat s napětím menším než je jeho referenční napětí, proto pro stabilizaci byl použit měnič od firmy Texas Instruments s technickým označením TPS60125. Jedná se o spínaný zvyšující měnič napětí, který má avšak tu výhodu, že dokáže pracovat v rozmezí napětí 1,8 V až 3,6 V. Zajímavé je to, že tento zvyšující měnič dokáže pracovat i s napětím vyšším než je jeho referenční hodnota. V krajních hodnotách měnič má ale slabou účinnost (kolem 60%). Nejvyšší účinnosti dosáhneme při použití napájecího napětí v rozmezí 2,3 V, až 2,8 V, kde měnič podle odebíraného proudu dosahuje účinnosti až 95%. Samotný měnič obsahuje v pouzdru celkem dva měniče, které jsou nezávislé. Pro naše účely byly měniče spojeny paralelně, aby v případě potřeby byl měnič schopen dodat požadovaný proud, který může být až 200mA na modul tj. maximální proud 400mA. V případě, že by došlo k náhodnému zkratu či bylo potřeba z měniče odebrat vyšší proud, je nutné zapojit chlazení měniče, které je zesponu čipu a poznáme ho jako kovové okénko. Jelikož je chlazení zesponu čipu, je nutné vytvořit chladicí plochu na plošném spoji a „thermal-pad“ propojit s chladicí plochou pomocí teplovodivé pasty. S postupem času je celkem jisté, že odběr zařízení je v plném provozu (mikroprocesor pracoval bez úsporného režimu, všechny diody sepnuté a všechny segmenty na sedmisegmentovém displeji svítí) cca 30 mA. Do spotřeby nebyl započítán proud, který by odebíral mikrokontrolér při vysílání a příjmu dat. Podle tabulky se na vysílání a příjem spotřebuje cca 15 mA, takže můžeme hovořit o maximální spotřebě kolem 45 mA, z čehož vyplývá, že by pravděpodobně nebylo chlazení potřeba. Pro chlazení byla zvolena celkem velká chladicí plocha s mnoha prokovenými otvory na rozlitou zem na straně Bottom. Díky takto velké chladicí ploše nemusíme mít strach z možného teplotního přetížení měniče.

Napájení z adaptéru

Jako alternativní napájení bylo použito napájení z externího zdroje napětí s rozmezím napětí (4,7 V až 15 V). Pro stabilizaci napětí byl zvolen opět produkt firmy Texas Instruments s typovým označením TVL1117-33, kde koncové dvojčíslí označuje výstupní napětí 3,3 V. Tento měnič se vyznačuje vysokým možným dodávaným proudem (až 0,8 A), teplotní ochranou a malým zvlněním výstupního napětí, které je při zátěži typově 1 mV s maximální hodnotou 10 mV.

3.6 Diody, Sedmisegmentový displej

Na DPS je osazena trojice barevných LED diod ve velikosti SMD 0805. Diody se spínají inverzní logikou (Log. 0 : svítí, Log 1: nesvítí. Pro celkové snížení spotřeby byly pro signalizaci použity diody s jmenovitým proudem 5 mA. Diody jsou připojeny tak, že zelená se ovládá pinem GPIO0 (na desce LED3), oranžová GPIO1(LED 2) a červená GPIO2(LED1).

Pro otestování sedmisegmentového displeje byla vytvořena funkce, která dokáže na displeji zobrazit textový řetězec, avšak znaky jsou velmi nedokonalé a zobrazit celou abecedu pomocí 7 segmentů je nemožné. Ovládání posuvného registru CD4094B je pomocí pinů GPIO5 jako signál Output-Enable, GPIO6 jako signál STROBE, GPIO7 jako signál DATA a GPIO8 jako hodinový vstup.

3.7 Sériová linka

Jako převodník je použit integrovaný obvod od firmy Texas Instruments MAX3232-EP, který v jedné pouzdře má převodníky, takže tento čip je možné použít na dva mikroprocesory a dvě nezávislé sériové linky. Pro funkci převodníku jsou potřeba pouze čtyři kondenzátory. V zapojení jsou původně nakreslené elektrolytické, avšak lze použít i jiné typy, pokud splní kapacitu. Díky minimalizaci byly použity kondenzátory keramické ve velikosti 0805 s hodnotami 100 nF a jeden keramický pro filtraci napájení s hodnotou 1uF. Bohužel došlo k chybě návrhu a byly přehozeny signály TXD a RXD, díky skutečnosti, že na řídicím mikroprocesoru je sériová linka označena jako pro druhý mikrokontrolér, místo vlastních pinů sériové linky, takže prohození signálů TXD a RXD je provedeno již na desce ZigBee modulu. V příloze je již chyba opravena.

3.8 Tlačítka

Na vývojovém kitu jsou použity celkem tři tlačítka. Jedno je využito na případný reset mikroprocesoru, zbylá dvě tlačítka jsou použita jako uživatelské vstupy. Tlačítka jsou připojena na vstupní brány mikroprocesoru na GPIO3 (SW1) a GPIO4 (SW2).

3.9 Vstupy

Externí vstupy pro ADC převodníky

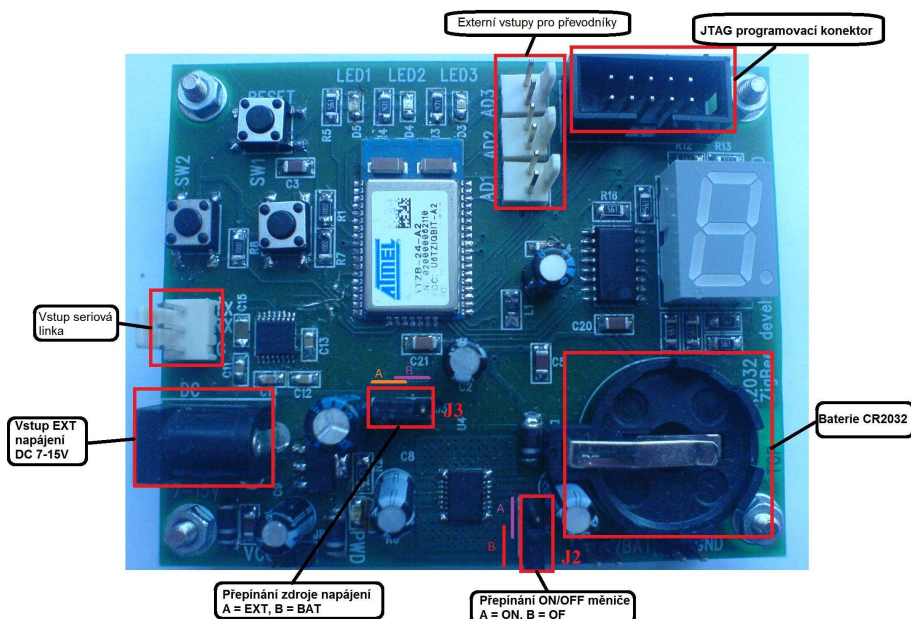
Z desky vývojového kitu jsou vyvedeny tři PSH konektory na možné připojení externích vstupů pro interní AD převodníky v ZigBee modulu. Konektory jsou odděleny na vlastní kanály a každý kanál má vlastní zem, takže připojení externího měřeného napětí je spolehlivější a jednodušší.

Vstup pro měření napětí baterie

ZigBee modul má na desce vyhrazen jeden ADC převodník pro měření napětí baterie. Tento převodník má v mikroprocesoru vnitřní označení ADC0 a jeho vstup je vyveden na pin 33 pod označením BAT.

4 Návrh desky plošného spoje

Pro kit byla navržena dvoustranná DPS o rozměrech 82x67 mm. Motivy a osazovací výkresy jsou v přílohách. Pro vývoj desky plošného spoje byl použit balík programů PADS od firmy Mentor Graphics. Fotografie osazené desky spolu označením polohy důležitých prvků je na Obrázek 28.



Obrázek 28 – Popis konektorů desky vývojového kitu

Pro řízení zdroje napájení desky jsou přepínače J2 a J3. Přepínač J2 se stará o povolení měniče. V poloze „A“ je měnič zapnut a stabilizované napětí je přivedeno na přepínač J3. V poloze „B“ přepínače J2 je měnič vypnut. Přepínač J3 je použit jako rozhodování jaké stabilizované napětí použít. V poloze „A“ je přepínač přepnut na Externí zdroj napájení. V poloze „B“ je jako zdroj napětí použita baterie.

5 Firmware

5.1 Úvod

Pro snadnější oživení vývojového kitu byl navržen pomocný, testovací FW, jehož jediným úkolem je otestovat správnou funkci veškerého HW na desce kitu. Tento FW je nazýván „testovací firmware“. Firmware vůbec nevyužívá části kitu, zodpovědného za implementaci ZigBee sítě.

Druhý firmware již implementuje funkčnost ZigBee sítě. Pro vývoj firmware byl zvolen jednodušší způsob v použití ZigBee stacku. Použitím ZigBee stacku byla práce ulehčena, protože je již většina kódů předpřipravena od výrobce HW. Pro ZigBee modul firmy Atmel je k dispozici ZigBee Bitcloud Stack.

Bitcloud stack lze použít pouze na mikroprocesory od firmy Atmel. Avšak je nutné dávat pozor na verze Bitcloud Stacku. Atmel vyvíjí více Stacků, každý pro jinou architekturu RF čipu a případně i mikrokontroléru. Příkladem může být Bitcloud Stack pro AVRRAVEN. AVRRAVEN je sada vývojových kitů pro ukázkou ZigBee technologie. Jedná se již o kompletně oživené a funkční zařízení. Pro vytvoření funkční aplikace postavené na modulu ATZB-24-A2 je nutné použít ZigBee Bitcloud Stack pro ZigBit.

Bitcloud stack se chová jako elementární operační systém vybudovaný speciálně pro danou architekturu modulu [29].

5.2 Testovací firmware

Testovací firmware byl napsán nezávisle na vývojové prostředí pro vývoj ZigBee aplikací. Firmware má za úkol pouze otestovat funkční celky na vývojové desce.

5.2.1 Ukázka kódu pro oživení sedmissegmentového LCD displeje

Ukázka oživovacího kódu pro oživení sedmissegmentového LCD displeje. Jako první je potřeba provést inicializaci sedmissegmentového displeje. Ve skutečnosti dojde pouze k nastavení ovládacích pinů na výstupní režim. Jako následující stačí zavolat funkci `show_char('a')`. Funkce nejdříve otestuje, zda zadaný znak je zobrazitelný na displeji (zda je to abecední znak). V případě, že znak je nezobrazitelný, vrátí funkce hodnotu 0. Když je znak zadán v pořádku, dojde k postupnému odeslání znaku bit po bitu na vývod mikrokontroléru a systém vypíše zadaný znak. Data se odesílají ve formátu MSB jako první. Vypsání znaku nerozlišuje velká a malá písmena, vše vypíše jako velké. V ovládacích funkcích je připsána rutina výpisu řetězce textu. Pro vypsání textu je nutný časovač a plný přístup k němu. Celý projekt je obsažen na přiloženém CD.

```

void init_sedmiseg(void){
DDR0 |= (1<<PG2);
DDR1 |= (1<<PE3);
DDR2 |= ((1<<PD6)|(1<<PD7));
}

int show_char(char znak){
//funkce prijima znaky ,nasledne poslany znak zobrazi na segmentovce
//vraci 1 pokud znak zobrazen, 0 pokud znak nelze zobrazit - je mimo rozsah zobrazitelnych
znaků

    if((znak>='0')&&(znak<='9')) znak -= '0';
    else if((znak >= 'A')&&(znak <= 'Z')) znak =(znak - 'A')+10;
    else if((znak >= 'a')&&(znak <= 'z')) znak = (znak-'a')+10;
    else return 0;
    char tempdata = char_table[(int)znak];

    CLR_OE_SEGMENTOVKA ; //clr OE
    CLR_CLK_SEGMENTOVKA ; //clr clk
    _delay_us(5);
    for(char i=0;i<8;i++){
        CLR_CLK_SEGMENTOVKA; //clr clk
        if(tempdata>=128) SETB_DATA_SEGMENTOVKA; //je MSB nastaven?
        else CLR_DATA_SEGMENTOVKA;
        tempdata=tempdata<<1;

        SETB_CLK_SEGMENTOVKA; //setb clk
        _delay_us(5);
    }

    SETB_STROBE_SEGMENTOVKA; //setb strobe -zapiš do výstupního bufferu
    _delay_us(2);
    CLR_STROBE_SEGMENTOVKA;
    _delay_us(2);
    SETB_OE_SEGMENTOVKA; //setb OE - zapiš na porty.
    return 1;
}

```

5.3 FW pro ZigBee síť

5.3.1 Úvod

Jak již bylo řečeno, v praxi se obvykle FW staví na bázi předpřipraveného kódu, nazývaného „stack“. Stack je soubor funkcí, které umožňují vývojáři rychlejší vývoj požadované aplikace. Stack lze rozdělit na dva typy. Prvním typem je stack, který obsluhuje pouze spodní dvě vrstvy. Vyšší vrstvy si vývojář musí vytvořit sám. Příkladem takovýchto stacků může být stack SMAC. Tento způsob vývoje je velice náročný, proto pro tuto práci byl zvolen druhý typ stacku. Druhý typ je kompletní stack. Tento typ stacku implementuje celý referenční ZigBee model a vývojář napíše již pouze koncovou uživatelskou aplikaci. Příkladem kompletního stacku je v práci rozebíraný Bitcloud stack.

5.3.2 ZigBee Bitcloud Stack

Práce se stackem

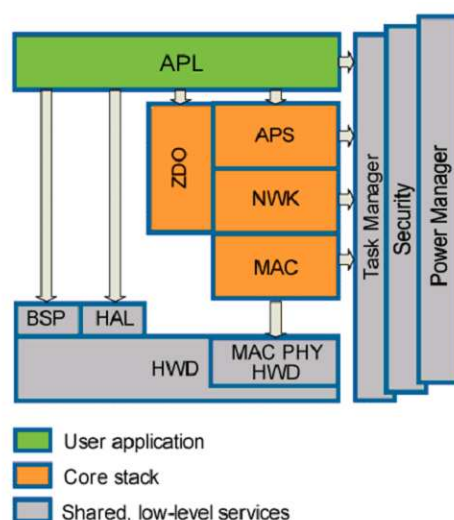
Bitcloud funguje jako skutečný operační systém s kooperativním multitaskingem, kde se programátor musí zabývat vrácením řízení jádru operačního systému a též musí žádat o přidělení strojového času. Velice důležité je časování délky uživatelských funkcí. Bitcloud definuje maximální čas, který může trvat uživatelská funkce. Časy jsou následující. Pro uživatelskou funkci je limit 10ms a vrácení řízení „task handleru“ musí být do 50ms. V případě, že dojde k překročení času či nedojde k vrácení řízení, tak dojde k přerušení aktuálních požadavků na aplikační vrstvě a systém si řízení vezme zpátky. Při tomto úkonu dojde k ukončení nedokončených uživatelských aplikačních funkcí a v důsledku dojde ke ztrátě dat. Pro práci s Bitcloud Stackem je nutno se držet určitých pravidel[29].

Pravidla pro práci

Pro korektní práci programu postaveného na Bitcloud stacku je nutné dodržovat následujících 12 pravidel.

1. Všechny uživatelské funkce jsou volány jádrem. Nutno je tedy zaregistrovat pointer na uživatelskou funkci.
2. Při požadavku systémových událostí je nutné napsat na získání informací o této události uživatelskou funkci.
3. Všechny uživatelské funkce musí být zpracovány do 10 ms.
4. Volané funkce pracují s prioritou vrstvy, která danou funkci zavolala.
5. Aplikační vlákno bude spuštěno, až když jsou vykonány všechny procesy s vyšší prioritou.
6. Aplikační vlákno musí trvat méně než 50 ms. Jedná se o čas, než bude řízení programu vráceno zpět jádru.
7. Kritické sekce nesmí trvat déle než 50 μ s .
8. Celé struktury by neměly být předávány na zpracování jádru. Vždy se předává pouze pointer na strukturu.
9. Pokud je to možné používat globální proměnné místo lokálních.
10. Uživatelská aplikace nesmí obsahovat rekurzivní volání. Doporučené maximální množství vnořených funkcí je omezeno na 10 a každá funkce by neměla obsahovat více než dva vstupní parametry.
11. Dynamická alokace paměti je přísně zakázána. Uživatel nesmí používat standartní funkce jazyka C jako malloc, calloc, realloc apod.
12. Hardware používaný stackem nesmí být přímo obsluhován uživatelskou aplikací. Veškeré hardwarové komponenty musí být obsluhovány přes stack.

Struktura BitCloud stacku



Obrázek 29 – Blokové schéma BitCloud stacku [29]

Z Obrázek 29 je patrné rozdělení celé architektury aplikace na 3 části, na obrázku barevně odlišené. První část, zelená, je uživatelská aplikační vrstva, kde jsou vytvořené kompletní uživatelské obslužné kódy jako „Rozsviť zelenou diodu“, „Zahaj měření teploty“ či „časovač přetekl“.

Oranžovou barvou je označena vrstva definovaná standardem ZigBee. V této vrstvě jsou celkem 4 sekce, kde každá sekce přidá vlastní informace k datům, která se budou odesílat.

Vrstva HAL

Hardware Abstraction Layer – vrstva obsahující kompletní ovládací rutiny základních hardwarových zařízení jako je EEPROM paměť, Watchdog, IRQ, USART apod.[29]

Vrstva BSP

Board Support Package – Vrstva obsahující ovladače periférii připojených na mikrokontrolér jako jsou různé senzory teploty, osvětlení, tlačítka či diody. [29]

Task Manager

Task manager(správce úloh) – Vrstva zprostředkávající interakci mezi aplikační vrstvou a mikroprocesorovým časem. Správce úloh v sobě implementuje tabulku prioritních funkcí a podle této tabulky přidělují strojový čas mikroprocesoru. [29]

Power Management

Vrstva starající se o správné uvedení systému do režimu spánku jako je uložení všech dat do paměti a zároveň opětovné nahrání dat a vzbuzení systému při požadavku dat. [29]

5.3.3 Základní kostra aplikace

V níže uvedené ukázce kódu je uvedena základní kostra programu, na které je nutné postavit každou novou aplikaci s využitím Bitcloud stacku [29]. V Bitcloud Stacku funguje na většinu událostí jen asynchronní volání funkcí, kde se po dokončení požadované funkce skočí na zaregistrovaný pointer na funkci, který je nutné zaregistrovat v příslušné struktuře. Událostí se myslí cokoliv, co vznikne jako požadavek na přidělení strojového času, jako třeba přetečení časovače, připojení či odpojení koncového zařízení, nebo příjem dat.

V programu je jako první implementace základních vrstev ZigBee stacku. Proměnná `appState` je řídicí proměnná pro stavový automat. V této proměnné se nachází v jaké fázi se program nachází. Proměnná `blinkTimer` je struktura časovače, ve které jsou všechny potřebné informace pro nastavení časovače. `APL_TaskHandler` je hlavní program. V hlavním programu se nachází stavový automat. Stavový automat má stavy:

- `APP_INITING_STATE`
Stav, ve kterém je program po zapnutí. V Ukázce se zde nastaví a spustí časovač.
- `APP_STARTING_NETWORK_STATE`
Stav, ve kterém se program nachází po nastavení základních parametrů sítě. Ve stavu dochází k odeslání požadavku na zapnutí sítě. V tomto stavu se nachází Koordinátor před vytvořením sítě, koncové zařízení před tím, než se připojí k existující síti.
- `APP_IN_NETWORK_STATE`
Stav kdy jsou už zařízení spojena.
- `APP_LEAVING_NETWORK_STATE`
V ukázce stav uveden není, jedná se o stav, ve kterém se nachází zařízení, než dojde ke korektnímu odpojení. Ve stavu se čeká, než koordinátor odešle odpověď na požadavek odpojení.

ZDO_StartNetworkConf

Jedná se o funkci, kam program skočí, když došlo ke zpracování požadavku spuštění sítě. Jako vstupní parametr funkce je pointer na strukturu, kde jsou uložena data o stavu sítě, kanálu, PANID, adrese konkrétního zařízení a adrese nadřazeného prvku (pouze v případě Routeru a Koncového zařízení, u Koordinátora je položka nevyplněna). Hlavička této funkce je zaregistrována ve struktuře, která se posílá při startu sítě nižším vrstvám.

SYS_PostTask(APL_TASK_ID)

Příkaz je patrný ve funkci `ZDO_StartNetworkConf()`. Jedná se o registraci požadavku na přiřazení strojového času hlavnímu programu (stavovému automatu). Tento příkaz se dává pouze v případě stavů, kdy nedojde k připojení sítě a dalších podobných událostí, kdy je potřeba program vykonávat v hlavní smyčce – znovu nastavení sítě atd.

ZDP LeaveResp

Funkce, volaná tehdy, když došlo ke zpracování požadavku na opuštění sítě. Dává možnost uživateli signalizovat, že zařízení se korektně odpojilo.

ZDO WakeUp Ind

Funkce, volaná tehdy když došlo k probuzení spícího zařízení. Tato funkce má smysl pouze u koncového zařízení. Ve funkci se musí provést znovu připojení ke koordinátoru a odeslat/přijmout případná data.

ZDO MgmtNwkUpdateNotf

Funkce, volaná asynchronně a to v případě, když dojde k detekci kolize adresy či adresy sítě (PANID).

blinkTimerFired

Jedná se o funkci, která se volá tehdy, když dojde k přetečení uživatelského časovače. Ve funkci může být periodické měření dat ze senzorů.

```

#include <taskManager.h>
#include <zdo.h>
#include <configServer.h>
#include <aps.h>
AppState_t appState = APP_INITING_STATE; //přepnutí stav.automatu do inicializace
static HAL_AppTimer_t blinkTimer; //struktura časovače

void APL_TaskHandler()
{
    switch (appState)
    {
        case APP_IN_NETWORK_STATE: // když je zařízení připojeno.
            ...
            break;
        case APP_INITING_STATE: //inicializační část
            blinkTimer.interval = 250; // Timer interval in milisecond
            blinkTimer.mode = TIMER_REPEAT_MODE; //Repeating mode (TIMER_REPEAT_MODE
            //TIMER_ONE_SHOT_MODE)
            blinkTimer.callback = blinkTimerFired; // Callback function for timer fire
            eventHAL_StartAppTimer(&blinkTimer);
            ...
            break;
        case APP_STARTING_NETWORK_STATE: //část kódu,kde se zapíná síť, případně zde
            //čeká ED na na připojení
            ...
            break;
    }
}
/***** Confirm callbacks*****/
static void ZDO_StartNetworkConf(ZDO_StartNetworkConf_t *confirmInfo) //asynchronně volaná
//funkce, kontrola zda se připojení k síti zdařilo,nutno zaregistrovat pointer
{
    ...
    if (ZDO_SUCCESS_STATUS == confirmInfo->status)
    {
        appState = APP_IN_NETWORK_STATE; //připojení proběhlo v pořádku
        ...
        SYS_PostTask(APL_TASK_ID); //předání řízení jádru,
    }
}
void ZDP_LeaveResp(ZDP_ResponseData_t *zdpRsp) //funkce opouštění sítě, asynchronně volaná
{...}

/***** Indication callbacks*****/
void ZDO_WakeUp_Ind(void) //funkce pro vzbuzení zařízení(pro ED)
{
    ...
    if (APP_IN_NETWORK_STATE == appState) //v případě že zařízení bylo před usmáním
        //připojeno, provede se znovu připojení
    {
        appState = APP_STARTING_NETWORK_STATE; //znovu připojení
        ...
        SYS_PostTask(APL_TASK_ID); //registrace požadavku předání řízení
// hlavnímu programu
    }
}
void ZDO_MgmtNwkUpdateNotf(ZDO_MgmtNwkUpdateNotf_t *nwkParams)
{...}

static void blinkTimerFired //obsluha přetečení časovače,
{
    PORTB ^= (1 << LED_RED); //invertování stavu vývodu
}

```

Funkce firmware

FW byl vytvořen na základě dodaných dem, ze kterých bylo vybráno pouze to nejnútnejší pro danou potřebu. V rámci práce byl vytvořen jednak firmware pro koordinátora a jednak firmware pro koncové zařízení.

Firmware pro koordinátora

Základní firmware koordinátora pracuje tak, že se nejdříve nastaví všechny potřebné nastavení jako popis uzlu (deskriptor), typ zařízení, nastavení adresy (dynamická) a nastavení uspávání. Všechny následující části kódu jsou vybrány z projektu obsaženého na příloženém CD. [29]

1. Nastavení popisu typu uzlu

Pro testy ZigBee síť byly použity modifikované demo kódy, proto musí být použity i stejné deskriptory.

```
simpleDescriptor.endpoint = WSNDemo_ENDPOINT;  
simpleDescriptor.AppProfileId = WSNDemo_PROFILE_ID;  
simpleDescriptor.AppDeviceId = WSNDemo_DEVICE_ID;  
simpleDescriptor.AppDeviceVersion = WSNDemo_DEVICE_VERSION;  
endpointParams.simpleDescriptor = &simpleDescriptor;
```

2. Nastavení typu uzlu – Koordinátor

Jako nejdůležitější část lze považovat nastavení typu zařízení. Pro zapisování typu zařízení je ve Stacku implementován CS server, kterému se dá jaká proměnná se má zapsat, předloží se pointer kde, je obsah proměnné a zavolá se funkce. O správné zapsání do paměti se postará jádro Stacku. Mimo zapsání do jádra programu se nastavení také zapisuje do aplikační struktury.

```
appMessage.data.nodeType = DEVICE_TYPE_COORDINATOR;  
endpointParams.APS_DataInd = APS_DataIndCoord;  
deviceType = appMessage.data.nodeType;  
CS_WriteParameter(CS_DEVICE_TYPE_ID, &deviceType);
```

3. Nastavení typu adresy

Pro nastavení adresy jsou dvě možné řešení. První, problémové, je nastavení statické adresy, problémové z důvodu možné kolize adres. Druhé řešení je nastavení adres dynamicky, jak je uvedeno v následujícím kódu. Teoreticky by nastavení adresy být nemuselo, koordinátor má adresu vždy nula, avšak pro případy neošetřených stavů je vhodné adresu nastavit, klidně i jinou než 0x0000, protože dynamické přidělení adresy se o správné nastavení adresy koordinátora postará.

```
#define CS_NWK_UNIQUE_ADDR false  
#define CS_NWK_ADDR 0x0000
```

4. Nastavení „režimu spánku“

Nastavení „režimu spánku“ je provedeno zapsáním hodnoty pomocí konfiguračního serveru. Pro koordinátora a Router se používá nastavení bez režimu spánku, z důvodu kritického významu těchto prvků.

```
rxOnWhenIdleFlag = true;
CS_WriteParameter(CS_RX_ON_WHEN_IDLE_ID, &rxOnWhenIdleFlag);
```

5. Start sítě

Následující instrukcí je již registrace požadavku na start sítě. Jako první je nutnost registrace pointeru na funkci, která se zavolá, až bude požadavek vykonán. Následně předáme ZDO vrstvě požadavek na start sítě.

```
networkParams.ZDO_StartNetworkConf = ZDO_StartNetworkConf;
ZDO_StartNetworkReq(&networkParams);
```

6. Kontrola stavu sítě

Jako poslední je kontrola stavu, zda došlo k vytvoření sítě. V případě že došlo k chybě, dojde k registraci požadavku přidělení času uživatelské vrstvě. V případě že je síť úspěšně vytvořena, bude ve vráceném pointeru na strukturu v položce status ZDO_SUCCESS_STATUS. Tento status oznamuje úspěšné vytvoření sítě. Po tomto testu je již síť funkční. Koordinátor získal dynamickou adresu(ShortAddr) 0x0000 a vygeneroval náhodné PANID sítě. Vrácená struktura obsahuje informace o komunikačním kanále, na kterém se koordinátoru podařilo vytvořit síť a adresu rodiče, která je ovšem nevyplněna, protože koordinátor nikam připojen není.

```
ZDO_StartNetworkConf_t *startInfo = param; //přetypování pointeru
if (ZDO_SUCCESS_STATUS == startInfo->status){ //kontrola úspěchu startu
appState = APP_IN_NETWORK_STATE; //start úspěšný, přepnutí FSM
failedTransmission = 0; //vynulování počtu špatných pokusů
appMessage.data.panID = startInfo->PANId; //uložení PANID
appMessage.data.shortAddr = startInfo->shortAddr; //uložení 16b adresy
appMessage.data.parentShortAddr = startInfo->parentAddr; //adresa nadřazeného prvku
appMessage.data.workingChannel = startInfo->activeChannel; //komunikační kanál

initSubStateMachine();
} else appPostGlobalTask(); //požadavek o přidělení času pro už. vrstvu
```

Firmware pro koncové zařízení

Základná firmware koncové zařízení pracuje tak, že se nejdříve nastaví všechny potřebné nastavení jako popis uzlu (deskriptor), typ zařízení, nastavení adresy (dynamická) a nastavení uspávání. Všechny následující části kódu jsou vybrány z projektu obsaženého na příloženém CD .

1. Nastavení popisu typu uzlu

Pro testy ZigBee sítě byly použity modifikované demo kódy, proto musí být použity i stejné deskriptory.

```
simpleDescriptor.endpoint = WSNDEMO_ENDPOINT;  
simpleDescriptor.AppProfileId = WSNDEMO_PROFILE_ID;  
simpleDescriptor.AppDeviceId = WSNDEMO_DEVICE_ID;  
simpleDescriptor.AppDeviceVersion = WSNDEMO_DEVICE_VERSION;  
endpointParams.simpleDescriptor = &simpleDescriptor;
```

2. Nastavení typu uzlu – End-Device

Jako nejdůležitější část lze považovat nastavení typu zařízení. Pro zapisování typu zařízení je ve Stacku implementován CS server, kterému se dá jaká proměnná se má zapsat, předloží se pointer, kde je obsah proměnné a zavolá se funkce. O správné zapsání do paměti se postará jádro Stacku. Mimo zapsání do jádra programu se nastavení také zapisuje do aplikační struktury.

```
appMessage.data.nodeType = DEVICE_TYPE_END_DEVICE;  
endpointParams.APS_DataInd = APS_DataIndDevice;deviceType =  
appMessage.data.nodeType;  
CS_WriteParameter(CS_DEVICE_TYPE_ID, &deviceType);
```

3. Nastavení typu adresy

Pro nastavení adresy jsou dvě možné řešení. První, problémové, je nastavení statické adresy, problémové z důvodu možné kolize adres. Druhé řešení je nastavení adres dynamicky, jak je uvedeno v následujícím kódu. Teoreticky by nastavení adresy být nemuselo, dynamické přidělování adres přidělí adresu zařízení, která je unikátní v síti. Výhodou dynamického přidělování je vybrání unikátní adresy v případě, že je v síti velké množství zařízení a mohlo by dojít ke kolizi adres.

```
#define CS_NWK_UNIQUE_ADDR false  
#define CS_NWK_ADDR 0xAAAA
```

4. Nastavení „režimu spánku“

Nastavení „režimu spánku“ je provedeno zapsáním hodnoty pomocí konfiguračního serveru. Pro koncové zařízení je možnost použití uspávání zařízení, díky čemuž se dosáhne velké úspory baterií.

```
rxOnWhenIdleFlag = false;
CS_WriteParameter(CS_RX_ON_WHEN_IDLE_ID, &rxOnWhenIdleFlag);
```

5. Start sítě

Následující instrukcí je již registrace požadavku na start sítě. Jako první je nutnost registrace pointeru na funkci, která se zavolá, až bude požadavek vykonán. Následně předáme ZDO vrstvě požadavek na start sítě.

```
networkParams.ZDO_StartNetworkConf = ZDO_StartNetworkConf;
ZDO_StartNetworkReq(&networkParams);
```

6. Kontrola stavu sítě

Jako poslední je kontrola stavu, zda došlo k vytvoření sítě. V případě, že došlo k chybě, dojde k registraci požadavku přidělení času uživatelské vrstvě. V případě, že je síť úspěšně vytvořena, bude ve vráceném pointeru na strukturu v položce status ZDO_SUCCESS_STATUS. Tento status oznamuje úspěšné připojení do sítě. Při připojení je již vyplněna struktura s údaji jako jsou PANID, přidělená adresa, komunikační kanál a adresa nadřazeného prvku. Adresou nadřazeného prvku se myslí adresa, kam je zařízení připojeno. V tomto případě bude zařízení přidělena adresa nadřazeného prvku 0x0000, což je koordinátor. V případě rozsáhlejší sítě s routery, by v položce parentAddr byla adresa routeru, ke kterému je koncové zařízení připojeno.

```
ZDO_StartNetworkConf_t *startInfo = param; //přetypování pointeru
if (ZDO_SUCCESS_STATUS == startInfo->status){ //kontrola úspěchu startu
appState = APP_IN_NETWORK_STATE; //start úspěšný, přepnutí FSM
failedTransmission = 0; //vynulování počtu špatných pokusů
appMessage.data.panID = startInfo->PANID; //uložení PANID
appMessage.data.shortAddr = startInfo->shortAddr; //uložení 16b adresy
appMessage.data.parentShortAddr = startInfo->parentAddr; //adresa nadřazeného prvku
appMessage.data.workingChannel = startInfo->activeChannel; //komunikační kanál

initSubStateMachine();
} else appPostGlobalTask(); //požadavek o přidělení času pro už. vrstvu
```


Vytvoření sítě

Vytvořit ZigBee síť může pouze koordinátor, avšak pouze za předpokladu, že již není členem jiné sítě. Před vytvořením nové sítě koordinátor proměří energii v pásmu, na kterém se zakládá síť. V případě, že energie v kanálu přesahuje určitou mez, koordinátor přeladí na jiný povolený kanál, který opět otestuje. V případě, že je kanál z hlediska vysílané energie dostupný, odešle Beacon Request rámec s dotazem na jiné koordinátory ve vysílaném kanálu (Na Obrázek 30 je vysílaný Beacon Request na řádce 1). Pokud v kanálu je již jiná ZigBee síť, koordinátor vybere kanál na kterém je nejmenší počet ZigBee sítí pro předcházení kolizí. Koordinátor síti přiřadí unikátní PANID číslo sítě a síť je vytvořena. Nyní se již mohou připojit zařízení. Koncové zařízení před svým připojením do již existující sítě vysílá na každém kanálu, na kterém může komunikovat Beacon Request, který slouží jako aktivní prohledání kanálů na dostupné koordinátory. V případě, že je na daném kanálu založena síť, na kterou se zařízení může připojit, obdrží koncové zařízení Beacon Frame. Nyní je již síť zformována a lze začít komunikovat. V testované síti se používá komunikace bez Beacon Frame. Ukázka části komunikace je na Obrázek 30. Obrázek byl získán pomocí vývojových modulů AVRRAVEN, vyráběných firmou Atmel z prostředí nazývaného AVR wireless studio. Pro získání obrázku byl z vývojové sady použit USB modul, který fungoval jako ZigBee sniffer, což je zařízení schopné monitorovat provoz v okolním prostředí a zachytávat pouze ZigBee rámce.

ID	Date	Time	Delta	Se...	Channel	Source PAN	Source Address	Destination PAN	Destination Address	IEEE Type	Payload	
1	2011-05-08	14:59:51.59890	00:00:00.00000	0	15			0xFFFF	0xFFFF	Beacon request	07 Start koordinatora	
2	2011-05-08	14:59:02.28998	00:00:10.32608	0	15			0xFFFF	0xFFFF	Beacon request	07 End-Device scan na koordinator	
3	2011-05-08	14:59:02.29215	00:00:10.32825	0	15	0x7ED1	0x0000			Beacon	FF CF 00 00 00 22 84 AA AA AA AA AA AA AA FF...	Koordinator odpovi Beacon Frame
4	2011-05-08	14:59:02.60909	00:00:11.02019	1	15	0x7ED1	0x4BF2	0x7ED1	0x0000	Data	09 10 00 00 F2 4B 01 69 67 45 23 91 78 56 34 12 06...	
5	2011-05-08	14:59:02.60991	00:00:11.02101	1	15					Acknowledgement	Koordinator odešle potvrzení prijmu dat	
6	2011-05-08	14:59:03.29345	00:00:11.32955	2	15	0x7ED1	0x4BF2	0x7ED1	0x0000	Data request	04 End-Device oozaduje data od koordinatora	Koordinator odpovi na zadosť o data
7	2011-05-08	14:59:03.29393	00:00:11.33003	2	15					Acknowledgement	ACK Frame a data odešle	
8	2011-05-08	14:59:03.29574	00:00:11.33184	1	15	0x7ED1	0x0000	0x7ED1	0x4BF2	Data	09 18 F2 4B 00 00 01 17 67 45 23 91 78 56 34 12 67...	
9	2011-05-08	14:59:03.29676	00:00:11.33286	1	15					Acknowledgement		
10	2011-05-08	14:59:03.29964	00:00:11.33574	3	15	0x7ED1	0x4BF2	0x7ED1	0x0000	Data	08 00 FD FF F2 4B 0A 6A 08 00 13 00 00 00 00 36 0...	ED odpovi ACK frame a odešle vlastni data na C
11	2011-05-08	14:59:03.30066	00:00:11.33676	3	15					Acknowledgement	Koordinator potvrdi prijem dat	
12	2011-05-08	14:59:03.30325	00:00:11.33935	4	15	0x7ED1	0x4BF2	0x7ED1	0x0000	Data request	04 ED ma pro C data	
13	2011-05-08	14:59:03.30373	00:00:11.33983	4	15					Acknowledgement	C odešle potvrzení zadosťi	
14	2011-05-08	14:59:03.32531	00:00:11.36141	5	15	0x7ED1	0x4BF2	0x7ED1	0x0000	Data	48 00 00 00 F2 4B 0A 68 40 01 01 00 01 00 01 37 01...	C prijme data a odešle potvrzení
15	2011-05-08	14:59:03.32631	00:00:11.36301	5	15					Acknowledgement		
16	2011-05-08	14:59:03.35171	00:00:11.38781	2	15	0x7ED1	0x0000	0x7ED1	0xFFFF	Data	08 00 FD FF F2 4B 09 6A 08 00 13 00 00 00 00 36 00...	
17	2011-05-08	14:59:03.40262	00:00:11.43872	3	15	0x7ED1	0x0000	0x7ED1	0xFFFF	Data	08 00 FD FF F2 4B 09 6A 08 00 13 00 00 00 00 36 00...	
18	2011-05-08	14:59:03.45271	00:00:11.48881	4	15	0x7ED1	0x0000	0x7ED1	0xFFFF	Data	08 00 FD FF F2 4B 09 6A 08 00 13 00 00 00 00 36 00...	C odešlã data na Broadcast - všem zařizenim
19	2011-05-08	14:59:04.32406	00:00:12.36016	6	15	0x7ED1	0x4BF2	0x7ED1	0x0000	Data request	04	
20	2011-05-08	14:59:04.32454	00:00:12.36064	6	15					Acknowledgement		

Obrázek 30 – Schéma komunikace v ZigBee

6 Závěr

V práci je teoreticky rozebrána ZigBee technologie s možnostmi různých typů zabezpečení. V rámci práce byl vytvořen a oživen vývojový kit v množství dvou kusů. Pro otestování základních funkčních bloků byl vytvořen elementární firmware, který nespolupracuje se ZigBee, ale slouží pouze jako kontrola funkčnosti modulu před samotným vývojem ZigBee sítě. Pro zprovoznění ZigBee sítě byla použita demo aplikace od výrobce, která byla upravena na míru použitého hardwarového řešení. Demo aplikace je rozdělena na dva samostatné projekty, kde první projekt je pro zařízení typu koordinátor a druhý projekt je pro koncové zařízení. V práci je popsáno základní nastavení a vytvoření ZigBee sítě s následnou ukázkou reálné ZigBee komunikace vytvořeného kódu.

Zadání práce se podařilo splnit pouze částečně. Původně zamýšlený směr testování vhodnosti ZigBee jakožto technologie pro provozování personální sítě senzorů na lidském těle, byl po dohodě s vedoucím práce redukován na prosté prostudování a základní zprovoznění sítě na vyvinutých vývojových kitech. Důvodem byla jednak objevená vysoká náročnost problematiky překračující svým rozsahem bakalářskou práci a jednak také nedostatek času, neboť spoustu věcí bylo nutné zjistit vlastním experimentováním a nikoli např. čtením dokumentace od výrobce. Nicméně výstupem práce je funkční hardware vývojového kitu a pro něj předpřipravený základní firmware, na kterém lze dále stavět, čili vytvářet plnohodnotné ZigBee aplikace s požadovanou funkčností. Vše tak může sloužit jako základ pro pokračující experimenty v této oblasti. Lze tedy dokončit i všechny body, požadované v původním zadání této bakalářské práce.

Pro další práci se ZigBee technologií je nutné prostudovat způsob přenosu uživatelských dat mezi moduly. Pravděpodobně bude nutností důkladně prostudovat další demo aplikace od výrobce. Pro další vývoj je nutné zkonstruovat další vývojové kity. V příloze na CD jsou již chyby objevené v prototypové desce opraveny.

Seznam použitých zdrojů

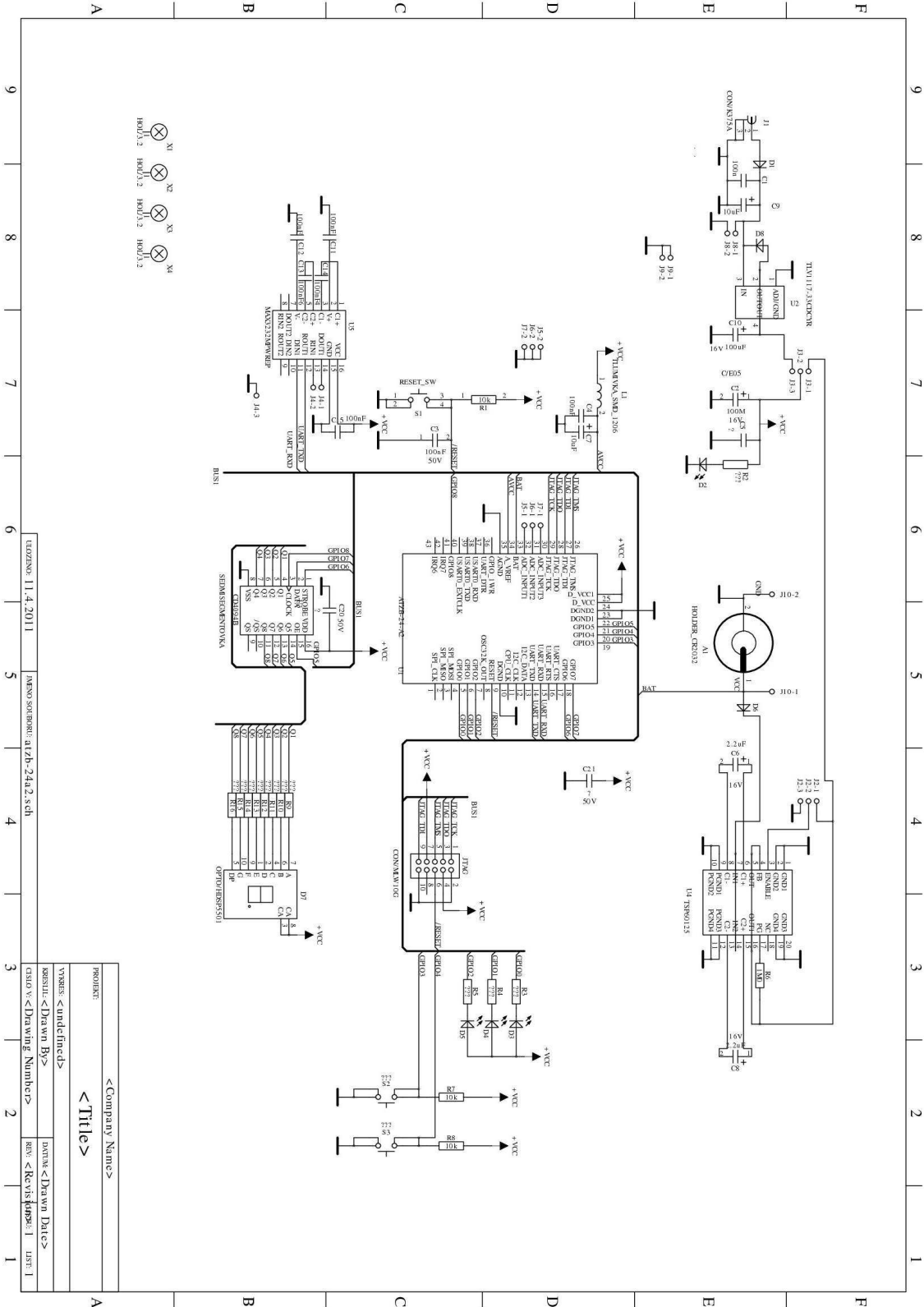
- [1] ŠEDIVÝ, Jiří. *Bezdrátový přenos Zigbee*. Praha, 2010. 39 s. Bakalářská práce. České Vysoké Učení v Praze. Dostupné z WWW: <http://support.dce.felk.cvut.cz/mediawiki/images/1/15/Bp_2010_sedivy_jiri.pdf>.
- [2] Main Page. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-04-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Main_Page>.
- [3] *ZigBee Alliance* [online]. 2011 [cit. 2011-04-11]. ZigBee Alliance. Dostupné z WWW: <<http://zigbee.org/>>.
- [4] Freaklabs - Open Source Wireless [online]. 20 June 2008 [cit. 2011-04-11]. *Atmel AT86RF230 and AT86RF231 Review*. Dostupné z WWW: <http://freaklabs.org/index.php/Articles/Zigbee/atmel-at86rf230_231-review.html>.
- [5] *Freaklabs - Open Source Wireless* [online]. 18 March 2008 [cit. 2011-04-11]. Zigbee/802.15.4 Chip Comparison Guide . Dostupné z WWW: <<http://freaklabs.org/index.php/Articles/Zigbee/Zigbee-Chip-Comparison.html>>.
- [6] Comparison of 802.15.4 radio modules. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-04-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Comparison_of_802.15.4_radio_modules>.
- [7] *ZigBee Learning* [online]. 2009 [cit. 2011-04-11]. MeshNetics. Dostupné z WWW: <<http://www.meshnetics.com/zigbee-learning/>>.
- [8] *ZigBee & RF4CE* [online]. 1995-2011 [cit. 2011-04-11]. RF/IF and ZigBee® Solutions. Dostupné z WWW: <http://focus.ti.com/analog/docs/gencontent.tsp?familyId=367&genContentId=24190&DCMP=hpa_rf_general&HQS=NotApplicable+OT+zigbee>.
- [9] *HW.cz* [online]. 8. Červen 2005 [cit. 2011-04-11]. ZigBee - novinka na poli bezdrátové komunikace. Dostupné z WWW: <<http://hw.cz/Rozhrani/ART1299-ZigBee---novinka-na-poli-bezdratove-komunikace.html>>
- [10] *Noel.feld.cvut.cz* [online]. 2007 [cit. 2011-04-11]. 802.15.4 & ZigBee. Dostupné z WWW: <noel.feld.cvut.cz/vyu/scs/prezentace2007/MC1321x_ZigBee/802.15.4.htm>

- [11] BRADÁČ, Zdeněk. Bezdrátový komunikační standard ZigBee. *Automatizace.cz* : *Odborný časopis pro automatizaci, měření a inženýrskou informatiku* [online]. Duben 2005, 2005, 48, [cit. 2011-04-11]. Dostupný z WWW: <<http://www.automatizace.cz/article.php?a=638>>.
- [13] BRADÁČ, Zdeněk; FIEDLER, Petr; KAČMÁŘ, Milan. Bezdrátové komunikace v automatizační praxi VII: další standardy . /*www.odbornecasopisy.cz* [online]. 2004, 07, [cit. 2011-04-11]. Dostupný z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=32437>.
- [14] IEEE802.15.4. *IEEE802.15.4*. 1 October 2003: IEEE 3 Park Avenue New York, NY 10016-5997, USA, 2003. 679 s. Dostupné z WWW: <<http://standards.ieee.org/about/get/802/802.15.html>>.
- [15] IEEE802.15.4. *IEEE802.15.4a-2007*. 31.8.2007 : IEEE 3 Park Avenue New York, NY 10016-5997, USA, 2007. 203 s. Dostupné z WWW: <<http://standards.ieee.org/about/get/802/802.15.html>>.
- [16] IEEE802.15.4. *IEEE802.15.4c-2009*. 17 April 2009 : IEEE 3 Park Avenue New York, NY 10016-5997, USA, 2009. 33 s. Dostupné z WWW: <<http://standards.ieee.org/about/get/802/802.15.html>>.
- [17] IEEE802.15.4. *IEEE802.15.4d-2009*. 17 April 2009 : IEEE 3 Park Avenue New York, NY 10016-5997, USA, 2009. 39 s. Dostupné z WWW: <<http://standards.ieee.org/about/get/802/802.15.html>>.
- [18] *Simulation of IEEE802.15.4/ZigBee* [online]. 2005 [cit. 2011-04-11]. ZigBee Tutorial. Dostupné z WWW: <<http://www.ifn.et.tu-dresden.de/~marandin/ZigBee/ZigBeeTutorial.html>>.
- [19] GASCÓN, David. *Wireless Sensor Networks Research Group* [online]. 5.2.2009 [cit. 2011-04-17]. Security in 802.15.4 and ZigBee networks. Dostupné z WWW: <<http://sensor-networks.org/index.php?page=0903503549>>.
- [20] KOTON, J.; ČÍKA, P.; KŘIVÁNEK, V. *Feld CVUT* [online]. 18. 04. 2006 [cit. 2011-04-17]. Standard nízkorychlostní bezdrátové komunikace ZigBee. Dostupné z WWW: <<http://access.feld.cvut.cz/view.php?cisloclanku=2006032001>>.
- [21] ZigBee alliance. *ZigBee Alliance* [online]. 2011 [cit. 2011-04-17]. Overview. Dostupné z WWW: <<http://www.zigbee.org/Standards/Overview.aspx>>.
- [22] *National Instruments* [online]. Oct 28, 2010 [cit. 2011-04-19]. Five Factors to Consider When Implementing a Wireless Sensor Network. Dostupné z WWW: <<http://zone.ni.com/devzone/cda/tut/p/id/10789>>.
- [23] Atmel. *ZigBit 2,4 GHz Wireless modules datasheet* [online]. [s.l.] : [s.n.], 2009, 06/09 [cit. 2011-04-19]. ZigBit modul Overview, s. . Dostupné z WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc8226.pdf>.

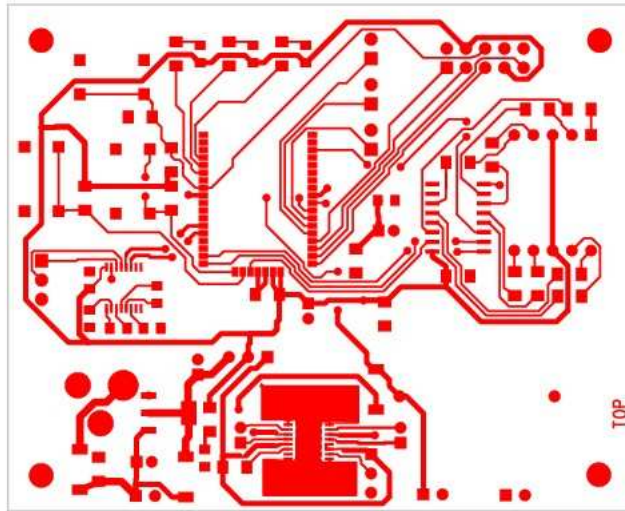
- [24] Atmel. *Low power 2,4GHz tranceiver for ZigBee* [online]. [s.l.] : [s.n.], 2009, 02/09 [cit. 2011-04-19]. AT86RF230 - General Circuit Description, s. . Dostupné z WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc5131.pdf>.
- [25] Atmel. *Low power 2,4GHz tranceiver for ZigBee* [online]. [s.l.] : [s.n.], 2009, 09/09 [cit. 2011-04-19]. AT86RF231 - General Circuit Description, s. . Dostupné z WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc8111.pdf>.
- [26] *Octopart Base e-shop* [online]. 2009 [cit. 2011-05-04]. ATZB-24-A2. Dostupné z WWW: <<http://octopart.com/atzb-24-a2r-atmel-12068335>>.
- [27] NĚMEC, Zdeněk. *Sdělovací technika : Přednáškové materiály* [online]. [s.l.] : [s.n.], 2010 [cit. 2011-05-08]. Dostupné z WWW: <portal.upce.cz>.
- [28] BRADÁČ, Zdeněk, et al. Kam kráčí standard ZigBee . /www.odbornecasopisy.cz [online]. 2007, 06, [cit. 2011-04-11]. Dostupný z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=34255>.
- [29] Atmel. *Bitcloud User Guide* [online]. [s.l.] : [s.n.], 2009, 11/09 [cit. 2011-04-19]. Network formation and Join, s. . Dostupné z WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc8199.pdf>.
- [30] *ZigBee* [online]. 15.11.2006 [cit. 2011-05-10]. ZigBee. Dostupné z WWW: <<http://blog.naver.com/PostView.nhn?blogId=choseonje&logNo=20030502788&parentCategoryNo=11&viewDate=&Page=1&listtype=0>>.
- [31] *ZigBee wireless networking topologies* [online]. 2009 [cit. 2011-05-10]. Types of ZigBee Networks. Dostupné z WWW: <http://www.stg.com/wireless/ZigBee_netw.html>.
- [32] ZigBee. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 9.7.2005, last modified on 13.4.2010 [cit. 2011-05-18]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/ZigBee>>.
- [33] *International Standard : ISO/EIC 7494-1*. [s.l.] : [s.n.], 15.11.1994. 68 s. Dostupné z WWW: <http://mobiledevices.kom.aau.dk/uploads/media/ISOSI_02.pdf>.

7 Přílohy

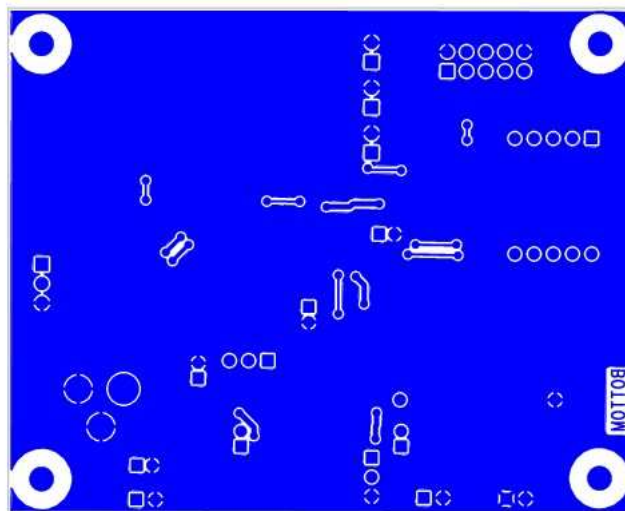
7.1 Schéma zapojení vývojového kitu



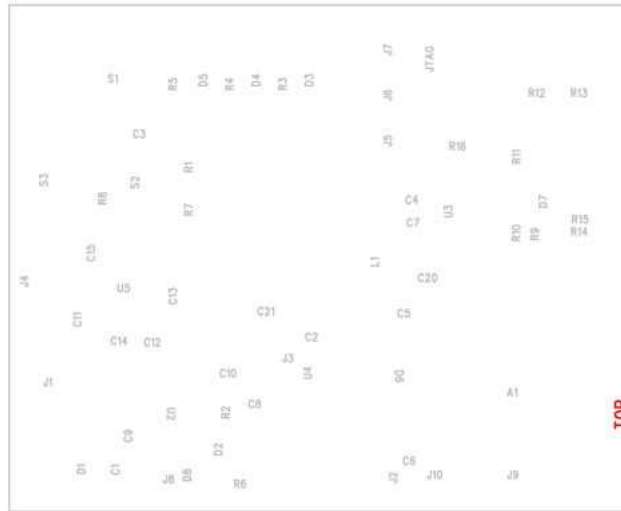
7.2 Vrstva TOP



7.3 Vrstva BOTTOM



7.4 Osazovací motiv



MANUFACTURER	PART NUMBER	SUPPLY VOLTAGE (V)	STEEL CURRENT (mA)	TX CURRENT (MAX)	RX CURRENT (MAX)	TX POWER (DBM)	RX SENSITIVITY (DBM)	SECURITY RITY	PACKAGE	DIMENSIONS (MM)	TEMPERATURE RANGE (DEG C)	COMMENTS
AT&M	AT86RF230	1.8-3.6	.02 (TYP)	16.5 @ 3DBM	15.5	3	-101	NONE	32 QFN	5X5	-40 TO 85	NICE SPEC. NO SECURITY.
AT&M	AT86RF231	1.3-3.6	0.02	14.5 @ 3DBM	13.2	3	-101	AES	32-QFN	5X5	-40 TO 85	SECURITY MODULE ADDED. ANTENNA DIVERSITY UP TO 2 MBPS.
FRRESALE	MCI3201	2.0-3.4	1 (TYP) 22 (MAX)	30 @ 0DBM	37	4	-91	NONE	32 QFN	5X5	-40 TO 85	MCI3201 - NON-ZIGBEE/802.15.4 APPS. PA AND LNA OUTPUT PINS FOR EXT AMPS. TX/RX SWITCH
	MCI3202/3	2.0-3.4	1 (TYP) 6 (MAX)	30 @ 0DBM	37	4	-91	NONE	32 QFN	5X5	-40 TO 85	MCI3202 - 802.15.4 ONLY
	MCI3191/2/3	2.0-3.4	1 (TYP) 6 (MAX)	30 @ 0DBM	37	4	-92	NONE	32 QFN	5X5	-40 TO 85	MCI3191 - NON 802.15.4/ZIGBEE. MCI3192 - 802.15.4 ONLY. MCI3193 - 802.15.4 + ZIGBEE
TEKSA INSTRUMENTS	CC2420	2.1-3.6 (VREG) 1.6-3.6 (IO)	20	17.4 @ 0DBM	18.8	0	-95	CTR CCM AES	48 QLP	7X7	-40 TO 85	ONE OF THE FIRST 802.15.4 CHIPS OUT
	CC2520	1.8-3.8	.09(LPM1) 175(LPM2)	23.8 @ 0DBM	22.3 (NORMAL) 18.8 (LOW CURRENT)	5(TYP)	-98	AES	28 QFN	5X5	-40 TO 125	SEE MY REVIEW ABOUT THIS CHIP'S SPECS
OKI	ML7065	2.5	NO INFO	NO INFO	NO INFO	0	-90	AES	48 QFN	7X7	-25 TO 70	POSSIBLE BUG ON WEBSITE WHERE DATASHEET LINK GIVES PRODUCT INTRO ONLY.
UBEC	UZ2400	2.4-3.6	2	23 @ 0DBM	19	0	-95	AES CTR CCM	40 QFN	6X6	-40 TO 85	CHECK ERRATA ON COMPANY WEBSITE FOR ISSUES WITH SECURITY ENGINE AND ZIGBEE 2006 SPEC.

7.5 Srovnání vysílacích čipů