

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky

Zdeněk Šíma

ZigBee & Bluetooth

Bakalářská práce

Knihovna JU - PF



3115172683

Vedoucí bakalářské práce: Ing. Václav Novák
České Budějovice 2006

JIHOČESKÁ UNIVERZITA
V ČESKÝCH BUDĚJOVICÍCH
PEDAGOGICKÁ FAKULTA
KATEDRA INFORMATIKY
- 33 -

26. 4. 2006 *muš*!

PODĚKOVÁNÍ

Děkuji panu Ing.Václavu Novákovi za rady a pomoc při vypracování této bakalářské práce.

PROHLÁŠENÍ

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a použil jsem pouze materiály uvedené v seznamu literatury.

V Českých Budějovicích 25. dubna 2006



.....
Podpis

ANOTACE

NÁZEV PRÁCE:	ZigBee & Bluetooth
PRACOVISTĚ:	Jihočeská univerzita v Českých Budějovicích Fakulta pedagogická Katedra informatiky
AUTOR:	Zdeněk Šíma
STUDIJNÍ OBOR:	Výpočetní technika a informatika
VEDOUCÍ BAKALÁŘSKÉ PRÁCE:	Ing. Václav Novák
ROK OBHAJOBY:	2006

V této bakalářské práci se věnuji dvěma standardům pro vytváření osobních bezdrátových sítí (WPAN), jimiž jsou ZigBee a Bluetooth. Mým úkolem je popsat komunikační protokol ZigBee, seznámit s jeho architekturou, s topologií jeho sítí a s dalšími klíčovými vlastnostmi daného protokolu. Dále porovnáím a vymezím možnosti technologií ZigBee a Bluetooth v reálných aplikacích na základě jejich teoretických a praktických parametrů. Také popíši implementaci protokolu ZigBee do reálné aplikace.

In my thesis ZigBee & Bluetooth I treat two standards for the creation of wireless personal area networks (WPAN), which are ZigBee and Bluetooth. My task is to describe ZigBee stack, introduce its architecture, its network topology and other key attributes of this stack. Further on, I will compare and determine the possibilities of the use of ZigBee and Bluetooth in real applications on the basis of their theoretic and practical characteristics. I will also pay attention to the ZigBee stack implementation into real applications.

Obsah

1. Úvod.....	1
2. ZigBee.....	2
Použité zkratky:.....	2
2.1 Historie.....	3
2.2 Základní vlastnosti ZigBee a oblasti jeho využití.....	3
2.3 IEEE 802.15.4-2003.....	4
2.3.1 Obecný popis.....	4
2.3.2 Fyzická vrstva (PHY).....	7
2.3.3 Linková vrstva (MAC).....	9
2.4 Protokol ZigBee.....	16
2.4.1 Architektura protokolu.....	16
2.4.2 Topologie sítě.....	17
2.4.3 Aplikační vrstva (APL).....	18
2.4.4 Síťová vrstva (NWK layer).....	26
2.4.5 Způsob zabezpečení ZigBee zařízení.....	27
3. Bluetooth.....	28
3.1 Historie.....	28
3.2 Technické parametry.....	28
3.3 Popis.....	29
4. Porovnání a vymezení využití.....	30
5. Aplikace ZigBee.....	32
5.1 Než začneme.....	32
5.2 Sestavení hardware.....	33
5.3 Programování.....	35
6. Závěr.....	40

Použité materiály

ZigBee:

IEEE 802.15.4-2003 – The Institute of Electrical and Electronics Engineers

ZigBee Specification verze 1.0 – ZigBee Alliance

www.zigbee.org – oficiální stránky ZigBee Alliance

www.chipcon.com – oficiální stránky společnosti Chipcon

www.ember.com – oficiální stránky společnosti Ember

www.atmel.com – oficiální stránky společnosti Atmel

www.mouser.com – oficiální stránky internetového obchodu Mouser

www.freescale.com – oficiální stránky společnosti Freescale (Motorola)

Bluetooth:

Specifikace protokolu Bluetooth ve verzích 1.2 a 2.0+EDR – Bluetooth.org

www.bluetooth.org – oficiální stránky organizace Bluetooth

Seznam tabulek

Tabulka 1: Obecný formát MAC rámce.....	10
Tabulka 2: Formát pole Frame control.....	10
Tabulka 3: Beacon frame.....	12
Tabulka 4: Struktura superframu.....	12
Tabulka 5: Data frame.....	13
Tabulka 6: MAC command frame.....	14
Tabulka 7: MAC commands.....	15
Tabulka 8: Obecný formát APS rámce.....	21
Tabulka 9: Obecný formát AF rámce.....	23
Tabulka 10: Formát KVP rámce.....	24
Tabulka 11: Typy příkazů KVP.....	24
Tabulka 12: Obecný formát NWK rámce.....	27

Seznam ilustrací

Ilustrace 1: Schéma protokolu ZigBee.....	17
Ilustrace 2: Podvrstva APS.....	20

1. Úvod

Svět okolo nás je plný zařízení, která mezi sebou komunikují bez použití kabelů a drátů. Ať už jsou to obyčejné televize, rádia, satelity, vysílačky či mobilní telefony. V poslední době se tento druh komunikace velice rozšířil. Takový způsob je velice výhodný pro aplikace, ve kterých není možné z různých technických důvodů kabely použít, i pro zařízení, která jsou od sebe velice vzdálená a propojení kabelem by bylo nákladné, v mnohých případech by dokonce ani nešlo použít. Někde je bezdrátové propojení vzhledem k mobilitě zařízení nebo obrovské vzdálenosti nezbytností, jako například u mobilních telefonů a satelitů.

Dvě různá zařízení, ačkoli spolu nemají na první pohled nic společného, mohou tvořit rozlehlé sítě a navzájem si předávat data. Mezi obrovské výhody bezdrátového spojení patří hlavně univerzálnost. Zatímco v případě kabelů platí co výrobce, to jiný konektor, v bezdrátovém světě je většina technologií standardizována. Při koupi telefonu nebo PDA s Bluetooth víte, že jej připojíte ke kterémukoli počítači podporující Bluetooth, ať je od kteréhokoli výrobce.

Ovšem bezdrátová komunikace má i řadu nevýhod. Mezi největší problémy patří bezpečnost. Pakety volně vysílané vzduchem lze snadno odposlouchávat, aniž by o tom měl adresát nebo odesílatel nějaké tušení. Další z nevýhod je zhoršení kvality signálu vlivem špatného počasí. I přes tyto komplikace je používání bezdrátových technologií stále častější.

Právě rozmach „bezdrátů“ je jedním z důvodů, proč jsem si zvolil toto téma jako zadání mé bakalářské práce. Mým cílem je popsat standard a ZigBee a porovnat jej s Bluetooth. Bluetooth je starší norma pro komunikaci v bezdrátových osobních sítích (WPAN) a je velice rozšířená při propojování osobních počítačů s mobilními telefony nebo PDA. Oproti tomu norma ZigBee je zcela nová. Obě technologie nemají za úkol soupeřit ve stejných typech aplikací, ale navzájem se doplňují.

2. ZigBee

Použité zkratky:

AES – Advanced Encryption Standard	GTS – Guaranteed Timeslots
AF – Application Framework	ISM – Industrial Science Medical
AIB – APS Information Base	KVP – Key Value Pair
APL – Application Layer	LQI – Link Quality Indication
APS – Application Support Sublayer	MFR – MAC Footer
APSD – APS Data Entity	MHR – MAC Header
APSM – APS Management Entity	MSG – Message Service
BPSK – Binary Phase-shift Keying	NLDE – Network Layer Data Entity
CAP – Contention Access Period	NLME – Network Layer Management Entity
CCA – Clear Channel Assessment	NWK – Network Layer
CFP – Contention-free Period	O-QPSK - Offset Quadrature
CID – Cluster Identification Number	Phase-shift Keying
CLH – Cluster Head	OSI – Open Systems Interconnection
CRC – Cyclic Redundancy Check	PAN – Personal Area Network
CSMA-CA – Carrier Sense Medium	PDA – Personal Digital Assistant
Access – Collision Avoidance	RFD – Reduced-function Device
ED – Energy Detection	WPAN – Wireless PAN
FCS – Frame Check Sequence	ZDO – ZigBee Device Object
FFD – Full-function Device	

2.1 Historie

ZigBee Alliance byla založena 21.10.2002 společnostmi InvenSense, Ember, Freescale, Honeywell, Mitsubishi, Motorola, Philips and Samsung. Technologie je postavena na standardu IEEE 802.15.4-2003, který obsahuje fyzickou a linkovou vrstvu. Nápad na založení vznikl už v roce 1998, protože technologie Bluetooth a WiFi byly pro některé aplikace nevhodné nebo nepoužitelné. 1.května roku 2003 byla dokončena specifikace standardu IEEE 802.15.4-2003.

Schválení specifikace ZigBee bylo dosaženo 14.12.2004. V červnu roku 2005 se stal tento dokument volně dostupným a v současné době má organizace zhruba 200 členů.

2.2 Základní vlastnosti ZigBee a oblasti jeho využití

Cílovými aplikacemi ZigBee by podle návrhu měly být různé senzory tepla, tlaku, vlhkosti, nebezpečných látek atd. v průmyslových budovách, dále například řízení vytápění v domácnostech, zavlažování zahrady, řešení oprávnění vstupů do objektů, dálkové ovladače pro domácí elektroniku (tzn. nahrazení IrDA), připojení PC periférií apod.

Z toho vyplývají také vlastnosti tohoto standardu. Těmi základními jsou spolehlivost, nízká cena, velmi malá spotřeba, vysoká bezpečnost a odolnost proti rušení, rychlý přechod mezi vysílacími režimy a otevřený standard.

Zařízení ZigBee mohou dosahovat maximálních přenosových rychlostí 20-250kb/s podle použité frekvence. Komunikace probíhá ve volných ISM pásmech. Zabezpečení dat je proveditelné několika způsoby, které jsou založeny na algoritmu AES.

Modul ZigBee je schopen přejít do vysílacího resp. přijímacího režimu během cca 0,5ms, což zajišťuje velice malou prodlevu. Jeho spotřeba se pohybuje kolem 20mA při vysílání resp. přijímání a zhruba 0,5mA v idle modu.

2.3 IEEE 802.15.4-2003

Přímá obsluha ZigBee modulu je založena na standardu IEEE 802.15.4-2003. Tento standard definuje protokol a propojení zařízení, která používají radiovou komunikaci v sítích typu PAN. Přístup k médiu probíhá pomocí mechanismu CSMA-CA. ZigBee používá tohoto standardu k definování dvou nejspodnějších vrstev.

2.3.1 Obecný popis

2.3.1.1 Typy zařízení

Zařízení komunikující v síti mohou být typu FFD, nebo RFD. FFD je zařízení s plnou funkcí, používá se pro složitější aplikace nebo pokud slouží jako koordinátor dané sítě. Naproti tomu RFD je zařízení s částečnou funkcí, které se využívá pro jednoduché aplikace a nemůže sloužit jako koordinátor sítě. FFD může komunikovat s RFD i s FFD, kdežto RFD pouze s FFD.

2.3.1.2 Topologie sítě

Síť se skládá minimálně ze dvou zařízení komunikujících na jednom kanálu, z nichž alespoň jedno je typu FFD, fungující jako koordinátor sítě. Podle potřeby může mít síť topologii „star“ nebo „peer-to-peer“. Standard ZigBee pak definuje další topologie.

V síti typu star je založeno spojení mezi ostatními zařízeními a jedním centrálním, které je nazýváno koordinátorem. Ten slouží především k zahájení a ukončení sítě a k řízení komunikace v síti.

V síťovém uspořádání typu peer-to-peer je také koordinátor, ale komunikace probíhá odlišným způsobem. V této síti může jakékoli zařízení komunikovat s kterýmkoli

dalším.

Každé zařízení musí mít unikátní 64-bitovou rozšířenou adresu. Tato adresa je určena pro přímou komunikaci v síti. Může být ovšem nahrazena 16-bitovou (přiděluje koordinátor sítě), pokud je zařízení asociováno v nějaké konkrétní síti.

Unikátní identifikátor má také každá síť. Ten umožňuje komunikaci zařízením v jedné síti, která používají zkrácenou 16-bitovou adresu, a také dovoluje komunikaci mezi zařízeními napříč nezávislými sítěmi.

Přestože utvoření sítě má na starosti síťová vrstva (network layer), popíší, jakým způsobem k tomu dochází.

Síť typu star může být vytvořena poté, co si FFD zařízení zvolí kanál, na němž ustanoví vlastní síť a stane se PAN koordinátorem. Všechny sítě tohoto typu fungují nezávisle na dalších takovýchto sítích, což je zajištěno právě unikátním PAN identifikátorem. Pak se mohou do sítě připojit další FFD a RFD zařízení.

V peer-to-peer síti (peer = kolega), kde může komunikovat každý s každým, je jedno zařízení ustanoveno koordinátorem (například to, které začalo komunikaci). Možným využitím je síť cluster-tree (cluster = shluk), což je speciální případ peer-to-peer sítě, kde je většina zařízení FFD. Taková síť se skládá z částí nazývaných cluster. Každá část je řízena koordinátorem clusteru (CLH) a každý cluster má podobně jako síť své unikátní číslo (CID). Není to však 16-bitová hodnota, ale pouze jakýsi index, kde hlavní cluster (ten, který obsahuje koordinátor celé sítě) má index 0 a každý další o jedničku větší. Poté, co se další zařízení připojí k síti, může ustanovit nový cluster s CID a je zároveň jeho koordinátorem. Výsledkem pak je jedna velká síť, která se skládá z clusterů (cluster je vlastně také síť). Výhoda takovýchto sítí je, že mají velký plošný rozsah, ovšem naproti tomu nevýhodou je větší časová prodleva při posílání dat.

2.3.1.3 Přenos dat

Při komunikaci záleží na tom, zda je síť beacon-enabled. Beacon (beacon = signál) je synchronizační rámec vysílaný koordinátorem, kde je specifikována adresa sítě a koordinátora, interval mezi synchronizačními rámci a další důležitá data. Interval mezi

beacon rámcí (včetně tohoto framu) je rozdělen na 16 stejně dlouhých časových úseků (slotů), z nichž první je vyhrazen právě pro vysílání synchronizačního rámce. Těchto 16 slotů je nazýváno superframe. Prostor mezi beacon rámcí (CAP) využívají zařízení k přenosu dat. Když chce zařízení vysílat data, musí pro sebe získat kanál použitím metody slotted-CSMA-CA. Všechny transakce musí být ukončeny do dalšího synchronizačního rámce. Pokud nechce koordinátor využívat superframe strukturu, vypne vysílání beacon rámců.

Superframe však může být rozdělený na prostor pro aktivitu zařízení v síti (CAP + CFP) a prostor, kdy komunikace neprobíhá. Během CAP mohou zařízení volně komunikovat, ale část CFP se skládá ze slotů vyhrazených jednotlivým zařízením. Během neaktivní části se mohou zařízení přepnout do modu s nízkým odběrem elektrického proudu. Typickým příkladem může být aplikace, kdy čidla jednou za určitý interval (definovaný v beacon rámcí, který vysílá koordinátor) odešlou centrálnímu zařízení (koordinátorovi) naměřená data a poté se na zbytek intervalu přepnou do úsporného režimu, aby šetřila baterie.

Existují tři typy přenosu dat. První je přenos dat směrem ke koordinátorovi, kdy zařízení vysílá data. Druhý způsob je směrem k zařízení, kdy koordinátor data odesílá. Třetí způsob je komunikace mezi sobě rovnými zařízeními (peer-to-peer).

Když chce zařízení poslat data koordinátorovi v beacon-enabled síti, nejprve čeká na beacon rámeček, s kterým se synchronizuje. Pak pošle data využitím metody CSMA-CA, jejichž příjem může koordinátor potvrdit (potvrzování doručení je volitelný prvek). Pokud chce poslat data v nonbeacon-enabled síti, jednoduše data pošle (opět pomocí CSMA-CA). Doručení je též koordinátorem potvrzeno. Tím je transakce ukončena.

Pokud chce poslat data koordinátor v beacon-enabled síti, označí tuto skutečnost v beacon rámcí, jemuž zařízení naslouchají. Při přijetí tohoto synchronizačního rámce zařízení zjistí, že mu chce koordinátor data poslat, vyšle příkaz, že požaduje data od koordinátora a ten data odešle. V případě nonbeacon-enabled sítě uchová koordinátor data, dokud o ně zařízení nepožádá vysláním příkazu. V případě, že zařízení požádá koordinátora o data, ale ten žádná data pro dané zařízení nemá, vyšle datový rámeček s prázdným tělem.

Při komunikaci v peer-to-peer síti pošle jednoduše odesílatel cílovému zařízení data v datovém rámcí.

2.3.2 Fyzická vrstva (PHY)

Dle standardu je fyzická vrstva zodpovědná za aktivaci a deaktivaci rádiového vysílače, detekci energie (ED) na vysílacím kanálu, indikaci kvality přenosu přijatých rámců (LQI), určení, zda je kanál volný pro komunikaci (CCA pro CSMA-CA), volbu kanálu a vysílání a příjem dat.

2.3.2.1 Frekvenční rozsah vrstvy

Fyzická vrstva ve skutečnosti obsahuje dvě vrstvy. První z nich operuje na frekvenci 868/915MHz, její maximální rychlost je 20kb/s (868MHz) respektive 40kb/s (915MHz). Modulace dat probíhá metodou BPSK. Tyto pásma jsou určena pro použití v Evropě (868MHz) respektive v Severní Americe (915MHz). Druhá vrstva je definována na frekvenci 2,4GHz, přenosová frekvence dosahuje 250kb/s a modulace je zajištěna metodou O-QPSK.

2.3.2.2 Frekvence a číslování kanálů

Vrstva určuje celkem 27 komunikačních kanálů, které jsou číslovány od 0 do 26. Pásmo 868MHz obsahuje jeden kanál (kanál 0 – rozsah 868-868,6MHz), v pásmu 915MHz se nachází 10 kanálů (kanály 1-10 – rozsah 902-928MHz) a v pásmu 2.4GHz je definováno 16 kanálů (kanály 11-26 – rozsah 2400-2483,5MHz).

Zařízení nemusí umět vysílat na takových kanálech, na kterých by to bylo v rozporu s předpisy daného regionu (státu).

2.3.2.3 Modulace

Modulaci na obou vrstvách předchází postupná konverze bitů na chip hodnoty.

Vrstva 868/915MHz – Každý bit zvlášť je mapován do tzv. symbolů. Symbol je dále převeden na 15 možných chip hodnot, které jsou modulovány. Bit s hodnotou 1 odpovídá chipům hodnot 111101011001000 ($c_0 - c_{15}$), bit o hodnotě 0 chipům 0000101000110111. Převáděn je nejprve bit s nižším příznakem, stejně tak i probíhá vysílání chipů v rámci symbolu. Chipy jsou nakonec modulovány algoritmem BPSK.

Vrstva 2,4GHz – V této vrstvě jsou mapovány 4 bity do jednoho symbolu. Existuje tedy 16 různých symbolů, každý symbol je převeden na 32 chip hodnot. Zde probíhá modulace pomocí O-QPSK.

Standard IEEE 802.15.4-2003 uvádí veškeré časové údaje v počtech symbolů. Přenosová symbolová rychlost na 868/915MHz vrstvě je stejná jako datová přenosová rychlost (tzn. 20kb/s resp. 40kb/s), kdežto na 2,4GHz vrstvě, kde se mapují zároveň 4 bity do jednoho symbolu, je přenosová symbolová rychlost čtyřikrát nižší než datová.

2.3.2.4 Metody pro přístup na kanál a informace o něm

Energy Detection (ED) – Zjištění energie na kanálu je určeno pro použití síťovou vrstvou. Když vybírá zařízení kanál, na kterém založí síť, je pro něj nejvýhodnější vybrat co možná nejvolnější kanál. Měřením je zjištěn odhad síly přijímaného signálu na daném kanálu. Účelem měření není jakkoli dekodovat přijatý signál. Délka měření ED se rovná intervalu osmi symbolů. Jeden symbol je (pro PHY 2,4GHz) z hlediska doby ekvivalentní 4 bitům, tedy délka měření ED je rovna 4 bajtům, což převedeno na čas znamená 0,125ms.

Link Quality Indication (LQI) – Toto měření charakterizuje sílu a kvalitu přijatého paketu. LQI může být implementováno kombinací ED a odhadu poměru signálu/šum a je použito pro každý přijatý paket zvlášť. Použití výsledků LQI měření je ponecháno vyšším vrstvám (mimo tento standard).

Clear Channel Assessment (CCA) – Zařízení kompatibilní s IEEE 802.15.4-2003 musí být schopno poskytovat alespoň jednu ze tří CCA metod.

CCA Mode 1 : Zařízení indikuje kanál jako obsazený, pokud je hranice signálu (zjištěná pomocí ED) nad určitou hranicí (threshold).

CCA Mode 2 : Kanál je označen jako obsazený, pokud je rozpoznán rámec, který má stejné charakteristiky jako rámec IEEE 802.15.4-2003 a to i tehdy, pokud je úroveň signálu pod hranicí ED.

CCA Mode 3 : Tato metoda je kombinací obou předchozích. Kanál je obsazený, pokud je rozpoznán rámec IEEE 802.15.4-2003 s úrovní signálu nad hranicí ED.

Úroveň ED může být nastavena maximálně 10dB nad hranici citlivosti přijímače, přičemž délka zjištění, zda je kanál volný (metoda CCA), je rovna intervalu 8 symbolů.

2.3.3 Linková vrstva (MAC)

MAC vrstva řídí přímý přístup k radio transceiveru (přijímač/vysílač) a jejím hlavním úkolem je vytvářet komunikační rámce. Musí poskytovat spolehlivé připojení mezi dvěma zařízeními, být schopna použít CSMA-CA metodu k přístupu na kanál a synchronizovat se s beacon rámci. Dále musí podporovat připojení zařízení k síti a jeho odpojení a také zajišťovat zabezpečení přenášených dat, tedy jejich kódování a dekodování.

2.3.3.1 Obecný formát rámce MAC

Každý MAC rámec sestává z třech základních částí. První je hlavička (MHR), která obsahuje kontrolní prvky rámce včetně adresních polí. Následují užitečná data (MAC payload) a celý rámec je uzavřen patou rámce (MFR). Přehled rámce je znázorněn v tabulce 1.

Byty: 2	1	0 / 2	0 / 2 / 8	0 / 2	0 / 2 / 8	různý	2
Frame control	Sequence number	Dest.PAN ident.	Dest. address	Sour.PAN ident.	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Tabulka 1: Obecný formát MAC rámce

Hlavička komunikačního rámce se skládá z kontrolních údajů a adresních polí.

Pole Frame control (řízení rámce) je 16-bitový údaj pro řízení daného rámce, jeho struktura je zachycena v tabulce 2.

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

Tabulka 2: Formát pole Frame control

Pole Frame type je dlouhé 3 bity a udává typ rámce. Bity 000 (v pořadí bit2 bit1 bit0) znamenají, že rámec je Beacon frame (synchronizační rámec), bity 001 značí Data frame (datový rámec), hodnota 010 označuje Acknowledgement frame (potvrzení, že byl předchozí rámec doručen) a konečně bity 011 přísluší MAC command frame (MAC příkaz). Příznak Security enabled je nastaven na 1, pokud je rámec kryptograficky zabezpečen. V opačném případě je bit nastaven na hodnotu 0. Další příznak Frame pending oznamuje příjemci, zda má pro něj odesílatel připravena ještě nějaká data. V tom případě si příjemce data vyzvedne opětovným požadavkem. V pátém bitu se nachází příznak Acknowledgement request, jenž nastaven na 1 oznamuje příjemci, že je po něm požadováno potvrzení o přijetí dat. Pokud by tak totiž neučinil, odesílatel by se domníval, že data nedorazila a poslal by je znovu. Hodnota nastavená v příznaku Intra-PAN určuje, jestli je rámec určen pro zařízení ve stejné síti. Pokud je bit nastaven na 1 a zároveň rámec obsahuje i adresu příjemce a odesílatele, rámec nemá obsahovat identifikátor cílové sítě. Bity 10-11 (respektive 14-15) nesou informaci o typu adresy příjemce (respektive odesílatele). Bity 10 (v pořadí bit1 bit0) značí 16-bitovou zkrácenou adresu, bity 01 pak

64-bitovou. Pokud je toto pole (resp. pole typ adresy odesílatele) nastaveno na 00 a rámec není Acknowledgement frame nebo Beacon frame, hodnota typ adresy odesílatele (resp. typ adresy příjemce) nesmí mít hodnotu 00. To by znamenalo, že je rámec směřován ke (resp. směřován od) koordinátorovi sítě.

8-bitová hodnota Sequence number (pořadové číslo) je index rámce. Beacon rámce jsou číslovány zvlášť, takže si koordinátor uchovává hodnotu indexu pro Beacon frame a po každém odeslání takového rámce ji zvýší o jedničku. Před prvním vysláním tohoto rámce inicializuje index náhodným číslem. Pro ostatní typy rámců je index společný, nezáleží na tom, s kolika zařízeními budeme komunikovat. Tato hodnota se používá proto, aby bylo možné zjistit, ke kterému rámcí patří případné potvrzení. Pokud zařízení vysílající rámec nedostane do určité doby potvrzení o doručení (a požaduje jej), vyše opětovně stejný rámec se stejným indexem.

Adresa sítě, ve které se nachází příjemce (Destination PAN identifier), má velikost 16 bitů. Pokud má adresa hexadecimální hodnotu 0xffff, je vysílaný rámec celoplošný (broadcast), tedy určený všem zařízením na kanálu.

Pole pro adresu příjemce je dlouhé 16 nebo 64 bitů podle typu adresy. Pokud je adresa 0xffff, jde též o broadcast, tentokrát ale pro zařízení nacházející se pouze v konkrétní síti.

Adresa sítě, kterou rámec odesílá, je též 16-bitová hodnota, stejně tak jako adresa zařízení je 16- nebo 64-bitová, to podle typu adresy, který je obsažen již dříve v rámcí.

Frame payload je pole přenášející užitečná data rámce, jejichž obsah záleží na daném typu rámce.

Poslední částí MAC rámce je pole FCS. V něm je uložen 16-bitový kontrolní součet CRC. Ten je počítán z částí MHR a MAC payload podle standardu ITU-T použitím polynomu stupně 16: $G_{16}(x) = x^{16} + x^{12} + x^5 + 1$.

2.3.3.2 Formát speciálního rámce MAC – beacon frame

Tento typ rámce slouží k synchronizaci zařízení a předávání údajů o síti. Jeho struktura je znázorněna v tabulce 3.

<i>Byty: 2</i>	<i>1</i>	<i>4/10</i>	<i>2</i>	<i>různý</i>	<i>různý</i>	<i>různý</i>	<i>2</i>
Frame control	Sequence number	Addres. fields	Superfr. spec.	GTS fields	Pending addresses	Beacon payload	FCS
MHR			MAC payload				MFR

Tabulka 3: Beacon frame

Údaje Frame control a Sequence number se nijak neliší od formátu obecného rámce. Třetí pole v pořadí, Addressing fields, obsahuje adresu sítě, kterou koordinátor řídí, a adresu samotného koordinátora. Superframe specification nese velice důležité informace o struktuře superframu, tedy o tom, jak je dlouhý a jakým způsobem je možno s koordinátorem komunikovat. Superframe lze sestavit podle tabulky 4.

<i>Bity: 0-3</i>	<i>4-7</i>	<i>8-11</i>	<i>12</i>	<i>13</i>	<i>14</i>	<i>15</i>
Beacon order	Superframe order	Final CAP slot	Battery life extension	Reserved	PAN coordinator	Association permit

Tabulka 4: Struktura superframu

Čtyřbitová hodnota Beacon order určuje délku superframu. Pokud má hodnotu 15, síť je typu non-beacon enabled, tedy koordinátor tyto synchronizační rámce nevysílá. V ostatních případech je délka superframu (v počtu symbolů) vypočítána podle následujícího vzorce: $60 \cdot 16 \cdot 2^{\text{Beacon order}}$. Šedesát je počet symbolů pro jeden slot. Číslo 16 udává počet slotů v superframu (z nichž první je vyhrazen pro beacon frame), obě jsou konstantní. Hodnota $2^{\text{Beacon order}}$ je tedy násobek. Pro případ fyzické vrstvy s rychlostí 250kb/s (pásmo 2,4GHz) a tedy 62,5ksymbol/s je minimální délka superframu (Beacon order = 0) 0,015 vteřiny, zatímco maximální (Beacon order = 14) je zhruba 251,5 sekundy. Údaj Superframe order udává aktivní délku superframu, tedy dobu, ve které je možné komunikovat. Poté bude již koordinátor všechny rámce ignorovat. Délka aktivní části superframu se spočítá analogicky k délce celého superframu, podmínkou ovšem je, že hodnota Superframe order je menší než Beacon order. Aktivní část by neměla být kratší než 440 symbolů, aby bylo umožněno všem zařízením komunikovat s koordinátorem. V případě potřeby jsou ovšem povoleny výjimky.

Čtyřbitový údaj Final CAP slot označuje číslo posledního CAP slotu. Po něm aktivní část superframu končí. Bit PAN coordinator udává, zda beacon frame vyslal koordinátor (1 = ano, 0 = ne). Další bitový příznak, Association permit, dává informaci, zda je povoleno se asociovat se sítí.

Pole GTS fields s proměnnou délkou definuje sloty, které jsou pro komunikaci vyhrazeny výhradně určitému zařízení. Takových slotů může být garantováno pro jedno zařízení i více za sebou. GTS fields se skládá z GTS specification (1 byte), GTS directions (0/1 byte, značí směr přenosu dat při daném GTS) a GTS list (proměnná délka). Pole GTS specification nese informaci o tom, zda koordinátor přijímá požadavky na garantování slotů a také údaje o již garantovaných slotech. GTS list obsahuje 3 bytové popisovače jednotlivých GTS, které obsahují zkrácenou 16-bitovou adresu zařízení, číslo počátečního slotu a počet slotů, které jsou danému zařízení garantována.

Pole Pending addresses obsahuje adresy zařízení, pro která má koordinátor data. Poté si zařízení o tato data požádá a budou mu koordinátorem zaslána. Takovýchto adres může být v jednom beacon framu nejvýše sedm. V případě, že má koordinátor zpráv více, informuje o tom postupně, v každém synchronizačním rámci vždy po sedmi adresách.

2.3.3.3 Formát speciálního rámce MAC – data frame

Schéma datového rámce je znázorněno v tabulce 5.

<i>Byty: 2</i>	<i>1</i>	<i>různý</i>	<i>různý</i>	<i>2</i>
Frame control	Sequence number	Addressing fields	Data payload	FCS
MHR			MAC payload	MFR

Tabulka 5: Data frame

Hlavička i patka datového rámce jsou stejné jako u beacon framu. Účel tohoto typu rámce je pouze přenést data, proto kromě nezbytných kontrolních a adresovacích položek nese pouze data (v poli Data payload). Pole Addressing fields obsahuje cílovou adresu a podle nastavení ve Frame control i adresu zařízení, které data odesílá.

2.3.3.4 Formát speciálního rámce MAC – acknowledgement frame

Tento typ rámce obsahuje pouze tři pole. Prvním z nich, zaujímající dva byty, je jako u všech ostatních Frame control. Údaje a příznaky, které nemá cenu uvádět, jako například nastavení bezpečnosti nebo požadování potvrzení, jsou nastaveny na hodnotu 0, případně jsou při přijetí ignorovány. Další pole v rámci je Sequence number, které obsahuje číslo rámce, jehož doručení tímto potvrzujeme. Celý rámec je jako všechny ostatní ukončen kontrolním součtem, tedy polem FCS.

2.3.3.5 Formát speciálního rámce MAC – command frame

Příkazy, požadavky a odpovědi na ně probíhají pomocí MAC command frame.

<i>Byty: 2</i>	<i>1</i>	<i>různý</i>	<i>1</i>	<i>různý</i>	<i>2</i>
Frame control	Sequence number	Addressing fields	Cmd. frame identifier	Command payload	FCS
MHR			MAC payload		MFR

Tabulka 6: MAC command frame

Frame control, Sequence number a Addressing fields jsou skládána stejně jako u beacon a data framu. Jaký příkaz je rámcem přenášen, je skryto v poli Command frame identifier. Zařízení typu FFD musí být schopno vysílat i přijímat všechny typy příkazů, schopnosti RFD zařízení a typy příkazů včetně jejich hodnot jsou znázorněny v tabulce 7.

<i>Command frame identifier</i>	<i>MAC command frame</i>	<i>RFD</i>	
		<i>Tx</i>	<i>Rx</i>
0x01	Association request	x	
0x02	Association response		x
0x03	Disassociation notification	x	x
0x04	Data request	x	
0x05	PAN ID conflict notification	x	
0x06	Orphan notification	x	
0x07	Beacon request		
0x08	Coordinator realignment		x
0x09	GTS request		
0x0a – 0xff	Reserved		

Tabulka 7: MAC commands

Sloupec RFD je rozdělen podle toho, jestli se jedná o vysílání (Tx) nebo přijímání (Rx) příkazu, kde označení 'x' znamená, že takový požadavek musí umět RFD zařízení obsloužit.

První z příkazů, Association request, slouží k přiřazení zařízení do sítě. Je tedy vysílán koordinátorovi zařízením, které o přiřazení žádá. Pokud je již však se sítí asociováno, ale spojení ztratí, nepoužívá opětovně tento příkaz, ale Orphan notification. V žádosti zařízení uvádí, jestli je FFD nebo RFD, jakým způsobem je napájeno (trvalý zdroj nebo omezený, např. baterie), zda je schopno komunikovat zabezpečeně, vypíná přijímač v neaktivních částech a jestli chce přiřadit zkrácenou 16-bitovou adresu, nebo bude komunikovat pouze pomocí 64-bitové. V odpovědi na požadavek uvádí koordinátor, zda byla asociace se sítí úspěšná a v opačném případě důvod neúspěchu (plná síť atp.).

Data request, tedy požadavek na zaslání dat, vysílá zařízení v beacon-enabled síti tehdy, když je v beacon framu uvedena jeho adresa, což právě znamená, že má u koordinátora nevyzvednuté zprávy. V nonbeacon-enabled síti může tento příkaz periodicky vysílat v rámci pollingu.

Příkazy PAN ID conflict notification a Coordinator realignment spolu velice úzce souvisí. Pokud totiž prvním z nich dává koordinátor najevo, že zaznamenal síť se stejným PAN ID jako má ta jeho, vyšle právě tento příkaz koordinátorovi oné sítě. Aby se zamezilo konfliktům, musí jeden z koordinátorů těchto konkurenčních sítí změnit adresu své sítě,

případně i komunikační kanál. K tomu slouží druhý z příkazů. Koordinátor může samozřejmě změnit síť i z jiných důvodů.

Orphan notification vysílá zařízení, které již bylo asociováno se sítí, ale kontakt ztratilo a snaží se ho obnovit. Tento příkaz vysílá tak dlouho, dokud koordinátor neodpoví, nebo zařízení ztratí zájem kontaktu se sítí.

Při zakládání nové sítě vyšle budoucí koordinátor Beacon request, čímž přinutí všechny koordinátory na stejném kanálu, aby vyslali beacon frame se svými údaji a také údaji o síti. Tím může zvolit parametry vhodně pro svou síť. V případech, kdy toto neudělá, může dojít ke konfliktům.

Posledním z příkazů je GTS request. Tímto žádá zařízení koordinátora o přidělení time slotů výhradně pro něj. Tento MAC příkaz může zařízení vyslat pouze v beacon-enabled síti.

2.4 Protokol ZigBee

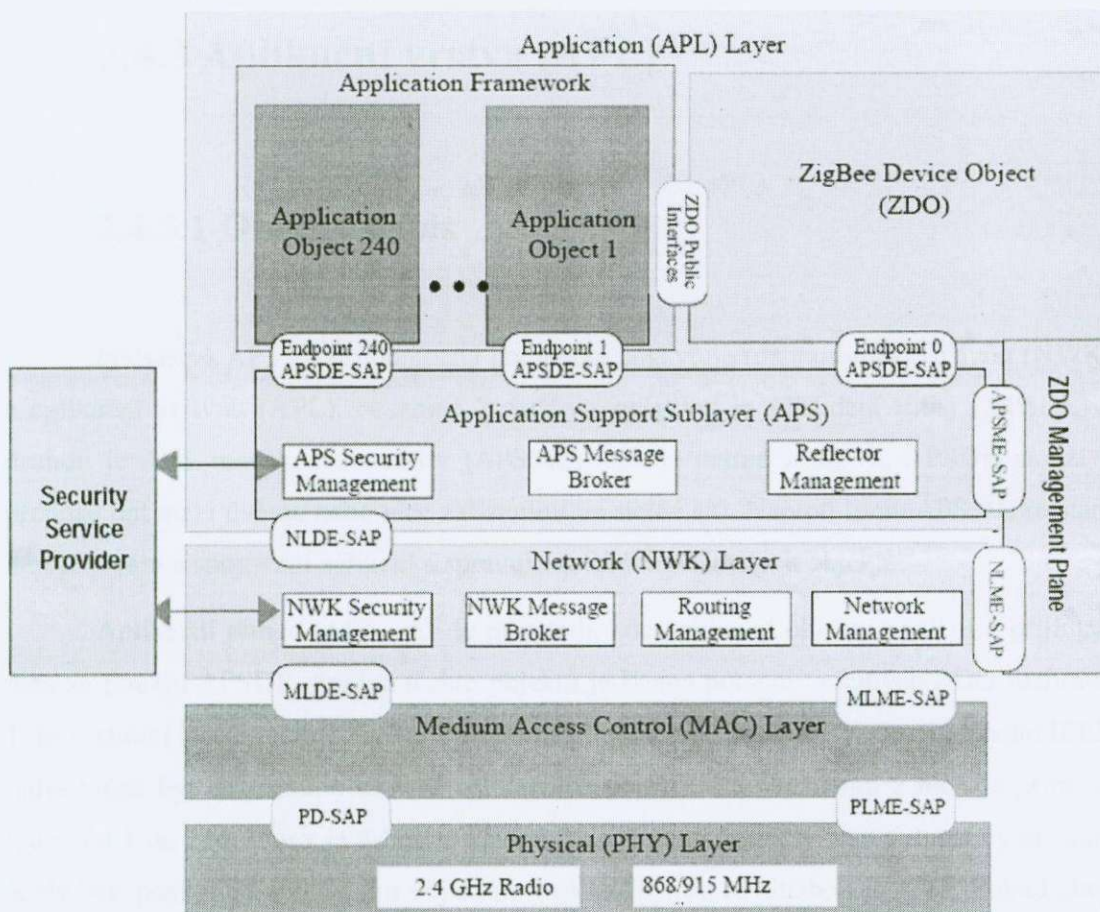
2.4.1 Architektura protokolu

Architektura protokolu ZigBee je založena na standardu OSI, který obsahuje 7 vrstev. Ovšem pro svou potřebu využívá jen dvě. Jak již bylo uvedeno, standard IEEE 802.15.4-2003 definuje fyzickou (PHY) a linkovou (MAC) vrstvu. Na tom staví ZigBee další vrstvy.

V první řadě je to síťová vrstva (NWK). Mezi její úkoly patří zakládání nových sítí, mechanismus připojení a opuštění sítě, je také zodpovědná za směrování paketů do cílového místa a hledání a udržování cest mezi jednotlivými zařízeními.

Druhou vrstvou je aplikační vrstva (APL), která se skládá z podvrstvy pro podporu aplikace (APS), aplikačního rámce (AF), objektu ZigBee zařízení (ZDO) a dalších objektů, které definuje výrobce (aplikační objekty). ZDO definuje roli zařízení v síti (jestli je koordinátor atd.), zahajuje nebo odpovídá na požadavky ke spojení a zřizuje bezpečné vazby mezi zařízeními v síti. Pro další pochopení je důležité mít představu, jak jsou

jednotlivé prvky tohoto protokolu uspořádané. K tomu poslouží ilustrace 1.



Ilustrace 1: Schéma protokolu ZigBee

2.4.2 Topologie sítě

Sítě ZigBee v podstatě překrývají sítě definované v IEEE 802.15.4-2003 a dál je rozvíjejí. ZigBee NWK podporuje 3 topologie. Jsou to star, tree a mesh. Síť typu star obsahuje zařízení, které síť řídí (koordinátora), a ostatní zařízení komunikují přímo s ním. V sítích mesh a tree zakládá koordinátor síť a určuje klíčové parametry. Ovšem síť může být rozšířena routery. Zatímco v síti s topologií tree může používat beacon frames, v síti typu mesh musí být zavedena komunikace peer-to-peer. Routery, které síť dále rozšiřují, nevysílají standardní beacon rámce. Jinak se tyto sítě chovají tak, jak je specifikováno v

2.4.3 Aplikační vrstva (APL)

2.4.3.1 Obecný popis

Podvrstva APS - Tato pomocná podvrstva poskytuje rozhraní mezi sít'ovou (NWK) a aplikační vrstvou (APL), obsahuje 2 služby. Tou první je APS data entity (APSD) a druhou je APS management entity (APSM). Jak je patrné z názvů, APSD slouží k přenosu dat mezi dvěma nebo více zařízeními ve stejné síti. Naproti tomu APSM se stará o objevování a spojování zařízení a spravuje databázi ovládaných objektů.

Aplikační rámec (AF) – AF je prostředí, kde aplikační objekty vysílají a přijímají data za použití APSD. Správa těchto objektů je řízena pomocí veřejných ZDO rozhraní. Tyto rozhraní jsou součástí ZigBee standardu a budou dále popsány. V rámci jednoho IEEE rádia může být definováno až 240 aplikačních objektů, kde každému z nich je přiřazen index od 1 do 240. Index je 8-bitové kladné číslo, ostatní hodnoty jsou vyhrazeny pro jiné účely. AF poskytuje aplikačním objektům dvě služby. První službou je KVP. Pokud chce zařízení zjistit hodnotu nějakého atributu z jiného zařízení, použije právě KVP. Tato služba má určitou strukturu, vhodnou k použití právě k tomuto účelu, navíc používá komprimované XML. Při návrhu normy ZigBee se počítalo také s tím, že touto technologií budou nahrazovány různé proprietární standardy (např. počítačové periferie – klávesnice a myši). Pro případ, že by pro ně použití KVP bylo nevhodné, definuje standard ještě druhou službu, jíž je MSG. MSG je transportováno stejným způsobem jako KVP, ovšem nechává aplikaci naprostou volnost ve struktuře služby podle toho, jak to vyhovuje jejím požadavkům.

V reálné aplikaci může nastat komplikace s adresováním, kdy například 2 spínače ovládají 4 žárovky. 2 spínače obsahují jedno IEEE 802.15.4-2003 rádio, stejně tak jako žárovky umístěné na stropě. Protože adresovat lze právě rádio, a ne nějaká další zařízení umístěné za rádiem, pokud bychom chtěli prvním spínačem ovládat tři žárovky, nastává problém. ZigBee umožňuje jakýsi způsob sub-adresování, kdy je každé koncové zařízení

(žárovka nebo spínač) aplikační objekt se svým indexem. Každý takový aplikační objekt má pak atributy, které ho určují (spínač má atribut definující, jestli je sepnutý, topení může mít atributy ukazující teplotu okruhu a místnosti atd.). Tím je pak možné adresovat každé jednotlivé zařízení, i když se dělí o rádio.

Adresování zpráv zařízením v síti je možno několika způsoby. Tím základním je přímé adresování. Zařízení uvádí adresu rádia zařízení (adresa sítě plus adresa zařízení) a k tomu ještě adresu koncového zařízení (endpoint address). Malým jednoduchým zařízením ovšem nemusí vyhovovat ukládání adres pro všechna zařízení, se kterými komunikují, a označí ve zprávě, že je nepřímě adresovaná. Neuvádí tedy cílovou adresu, ale pouze svou. Ta je přeložena pomocí spojovací tabulky (binding table) a zpráva je poslána všem zařízením, která jsou obsažena ve spojovací tabulce. Posledním způsobem adresování je broadcast, kdy je zpráva doručena všem aplikačním objektům daného zařízení (tzv. aplikační broadcast).

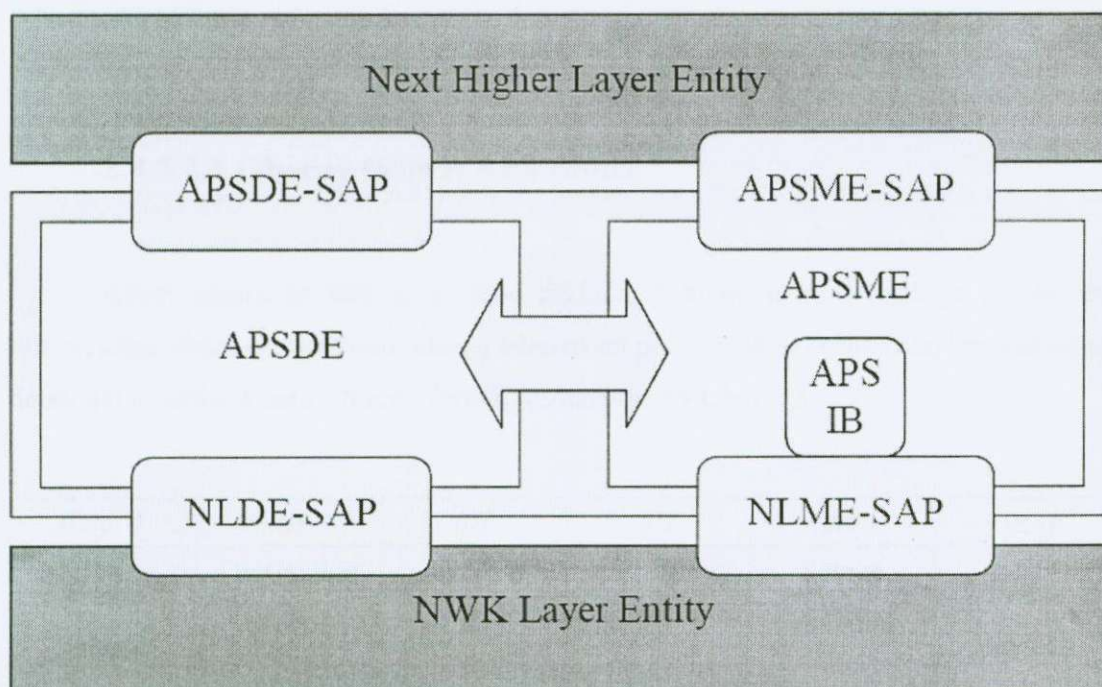
Aby byla možná komunikace mezi zařízeními různých výrobců, je potřeba standardizovat i způsob, jak budou jednotlivé druhy produktů komunikovat, tedy sjednotit strukturu zpráv a požadavků. To se nazývá profil. Například kotel a tepelné senzory spolu tvoří vytápěcí systém. Tedy je v tomto profilu definováno, jakým způsobem budou mezi sebou komunikovat kotel a tepelné senzory a jakou strukturu zprávy budou používat. Profily jsou definované jednak organizací ZigBee, ale hlavně se počítá s tím, že různé profily budou vytvářet právě výrobci dalších zařízení. Každý profil je samozřejmě potřeba schválit a zveřejnit společností ZigBee.org.

ZigBee objekty (ZDO) jsou jakousi základní třídou, která poskytuje rozhraní pro aplikační objekty. Toto rozhraní určuje, jakým způsobem inicializovat nižší vrstvy, jak posílat zprávy atp. To pak aplikační objekty implementují podle vlastní potřeby. Rozhraní obsahuje především management objevování dalších zařízení, jejich spojování a také management zabezpečení.

2.4.3.2 Podvrstva APS

Jak bylo již uvedeno, APS je podvrstva, která zajišťuje rozhraní mezi sítovou (NWK) a aplikační (APL) vrstvou. Rozhraní poskytuje sadu služeb, které jsou přístupné

skrže dvě entity – data entity (APSDE) a management entity (APSME). První služba APSDE vytváří z dat APS protokol přidáním příslušných údajů. Druhá služba spojí dvě zařízení na základě jejichž požadavků a služeb, které nabízejí, a je schopna doručit zprávu od jednoho zařízení k druhému. APSME má na starosti vytvořit a starat se o spojovací tabulku. K tomu nabízí také dvě služby. AIB Management zjišťuje a upravuje atributy daného zařízení, zatímco druhá služba navazuje vztahy s dalšími zařízeními pomocí bezpečnostních klíčů. Na ilustraci 2 je znázorněn model podvrstvy APS.



Ilustrace 2: Podvrstva APS

Pokud chce vyšší vrstva odeslat data jinému zařízení, použije k tomu právě APSDE. Přitom musí specifikovat následující údaje. Za prvé typ adresace cílového zařízení. Adresování může probíhat pomocí zkrácené 16-bitové adresy (pak je nutná adresa PAN sítě), nebo rozšířené 64-bitové adresy, a nebo může zařízení zvolit nepřímé adresování. Poté uvede konkrétní adresu zařízení a také adresu cílového objektu (endpoint). Další atribut je identifikační číslo profilu, kterým mezi sebou zařízení komunikují. Buďto využije jeden z již definovaných standardem ZigBee, nebo deklaruje, že tento profil je specifický pouze pro tuto aplikaci.

Pokud je použito nepřímé adresování, musí být uvedena také adresa clusteru, na němž je

uložena spojovací tabulka. Naopak v případě přímého adresování je tento parametr vynechán. Aby příjemce dokázal identifikovat, od koho data dostává, musí zařízení uvést i vlastní endpoint – tedy číslo objektu. Následuje hodnota, která udává délku dat, samotná data a na konec vysílací volby, zda má být paket šifrovaný a potvrzený příjemcem.

APSME slouží k nastavování různých atributů a ke spojení nebo odpojení dvou zařízení. Jak je vidět z obrázku modelu podvrstvy APS, jak APSDE, tak i APSME je využívána vyšší vrstvou. Při požadavku na spojení nebo odpojení vyšší uvede vrstva adresační údaje obou zařízení a pokud chce zjistit nebo změnit nějaký atribut, zadá jméno (i hodnotu) atributu.

2.4.3.2.1 Obecný formát APS rámce

Každý rámec se skládá ze dvou základních komponent. Jednak je to hlavička rámce, která obsahuje kontrolní údaje a adresovací pole, a pak užitečná data, která obsahují délku dat a samotná data. Obecný formát je znázorněn v tabulce 8.

<i>Byty: 1</i>	<i>0/1</i>	<i>0/1</i>	<i>0/2</i>	<i>0/1</i>	<i>různý</i>
Frame control	Destination endpoint	Cluster ID	Profile ID	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Tabulka 8: Obecný formát APS rámce

Důležité řídicí údaje jsou obsaženy v poli Frame control. Typ rámce (data, command, acknowledgement) je obsažen v prvních dvou bitech. Další dva bity v tomto poli určují typ adresování – přímé, nepřímé a broadcast. Následně je v rámci označeno, zda jsou data zabezpečena a jestli je požadováno potvrzení doručení (acknowledgement). Dále je v rámci obsažena adresa koncového a zdrojového objektu, identifikační číslo clusteru a profilu. Celý rámec je ukončen daty konkrétního rámce. Toto platí pro rámce obecně, podle konkrétního typu mohou být různá pole rámce vypuštěna.

2.4.3.2.2 Spojování (binding)

Smysl spojování a spojovací tabulky je v ušetření paměťového místa pro zařízení, která ho potřebují využít jinak, než pro uchování adres zařízení, se kterými komunikují. U koordinátorů se předpokládá, že jako zařízení, která řídí síť, mají dostatek zdrojů. Samotné spojování a využití takovéto tabulky probíhá následujícím způsobem. Zařízení, které chce adresovat nepřímo, požádá o záznam to spojovací tabulky. Záznam může být veden buďto koordinátorem, nebo zařízením, které je v požadavku uvedeno jako cílové.

Každý řádek (tedy záznam) spojovací tabulky obsahuje adresu zdrojového zařízení (toho, které vysílá paket adresovaný nepřímo), číslo jeho aplikačního objektu, index clusteru, kam má být paket doručen a nakonec dvojice údajů adresa cílového zařízení a číslo cílového aplikačního objektu. Těchto dvojic může být několik, v důsledku to tedy znamená, že paket adresovaný nepřímo může být doručen několika zařízením nebo více aplikačním objektům v rámci jednoho zařízení. V případě, že prostor pro záznamy tabulky je již obsazen, je požadavek odmítnut.

2.4.3.3 Aplikační rámec (AF)

Klíčovým bodem pro komunikaci v ZigBee síti je, aby se zařízení domluvila na profilu, jímž budou komunikovat. Každý profil musí být registrován společností ZigBee.org. To je potřeba proto, aby byla identifikační čísla jednotlivých profilů jedinečná. Po přiřazení identifikačního čísla je možné definovat vlastnosti profilu.

Před tím, než proběhne předávání dat mezi zařízeními, je potřeba zjistit, jaké služby dané zařízení poskytuje. V případě, že více aplikací sdílí jedno rádio, každé aplikaci je přidělen jeden endpoint. Každý endpoint má jednoduchý popisovač, který jej specifikuje, a je předáván při zjišťování služeb.

2.4.3.3.1 Popisovače ZigBee

Standard definuje pět popisovačů. Node nese informace o typu a možnostech uzlu. Node power udává vlastnosti týkající se napájení zařízení. Tyto popisují uzel jako celek. Další popisovače jsou vedeny o každém endpointu zvlášť. Simple obsahuje základní informace o endpointu, zatímco Complex je rozšíření k Simple. User je uživatelský popisovač, vyhrazen pro další informace. Complex a User nejsou povinné, jejich případné použití je označeno v popisovači Simple.

2.4.3.3.2 Obecný formát AF rámce

Přehled obecného rámce je znázorněn v tabulce 9.

<i>Bity: 4</i>	<i>4</i>	<i>různý</i>	<i>různý</i>	<i>různý</i>
Transaction count	Frame type	Transaction 1	...	Transaction n

Tabulka 9: Obecný formát AF rámce

Čtyřbitové pole Transaction count je číselná hodnota, počet tzv. transakcí. Frame type informuje o použitém typu zprávy, tedy zda se jedná o KVP nebo MSG. Poté následují za sebou všechny transakce. Každá obsahuje číslo transakce, které je 8-bitové a s každou transakcí se inkrementuje o 1, a samotná data. Formát těchto dat závisí na tom, jestli je použito KVP nebo MSG.

2.4.3.3.3 Formát KVP rámce

KVP je přímo určen pro manipulaci s atributy, čemuž odpovídá i jeho struktura (tabulka 10).

Bity: 4	4	16	0 / 8	různý
Command type identifier	Attribute data type	Attribute identifier	Error code	Attribute data

Tabulka 10: Formát KVP rámce

Na začátku příkazu je určen typ příkazu. Za ním následují údaje týkající se konkrétního atributu. Nejprve je to jeho typ (číselný, 8-bit/16-bit s/bez znaménka, nebo řetězec znaků atd.), dále identifikátor, který je specifikovaný v popisu jednotlivého zařízení. Pole Error code je obsaženo pouze v odpovědích (get/set/event response) a oznamuje výsledek příkazu. Ten je ukončen hodnotou daného atributu. Typ příkazů, které je možno pomocí KVP provádět, je znázorněn v tabulce 11.

Command type identifier value $b_3b_2b_1b_0$	Description
0000	Reserved
0001	Set
0010	Event
0011	Reserved
0100	Get with acknowledgement
0101	Set with acknowledgement
0110	Event with acknowledgement
0111	Reserved
1000	Get response
1001	Set response
1010	Event response
1011 – 1111	Reserved

Tabulka 11: Typy příkazů KVP

2.4.3.3.4 Formát MSG rámce

Tento způsob zjišťování a nastavování atributů je použit zařízeními, která nejsou připraveny na KVP, nebo jim tento způsob nevyhovuje. Proto je velice jednoduchý a dává

volnost profilu. Rámec se skládá pouze ze 2 polí. První je 8-bitová hodnota, udávající délku příkazu. Po ní již následují (druhé pole) blíže nespecifikovaná data.

2.4.3.4 Objekty ZigBee zařízení (ZDO)

Objekt ZigBee zařízení je řešení, které implementuje ZigBee zařízení, tedy koncové zařízení, routery a koordinátory. ZBO má za úkol inicializovat všechny vrstvy protokolu kromě aplikačních objektů. Dále je zodpovědná za implementaci řady funkcí, kterými je zařízení vlastně řízeno.

2.4.3.4.1 Objevování zařízení a služeb

Tuto funkci musí být zařízení schopné provádět v rámci sítě. Pokud se koncové zařízení během provozu přepíná do sleep modu, aby šetřilo energii, neodpovídá po tu dobu na požadavek objevení zařízení a služeb, které poskytuje. Pro tento případ může všechny důležité údaje (adresy a popisovače) uchovávat koordinátor nebo router, v případě požadavku na objevení zařízení a služeb odpovídá i za dané koncové zařízení. Požadavek na objevení zařízení může být adresován buďto jednomu zařízení (unicast), nebo všem zařízením v síti (broadcast). Odpověď je ovšem ve všech případech unicast.

Při žádosti o objevení služeb mohou být dotazovány různé záležitosti. Cílem dotazu může být síťová adresa plus výčet všech endpointů, dále dotaz, které endpointy komunikují prostřednictvím určitého profilu a také může být dotaz směřován k získání popisovačů.

2.4.3.4.2 Správa zabezpečení

ZDO rozhodne, jestli bude použito zabezpečení při komunikaci. Pokud ano, správa zahrnuje následující úkony. Nejprve je potřeba získat Master key od koordinátora. Není tomu tak v případě, že je klíč již přednastaven. Poté ustanoví ostatní klíče pro komunikaci s ostatními zařízeními. O tom, že používá zabezpečenou komunikaci, musí dát vědět všem

zařízením v síti, aby při komunikaci s ním používali také zabezpečenou komunikaci.

2.4.3.4.3 Správa sítě

ZDO musí být v případě routeru nebo koncového zařízení schopno zajistit přihlášení do PAN sítě a při ztrátě spojení jej obnovit (odlišná procedura oproti navázání prvního spojení se sítí). Pokud je zařízení router, nebo koordinátor, je ZDO zodpovědné za výběr kanálu a založení sítě. Mezi úkony spadající pod správu sítě patří také nalezení zařízení v síti a určení základních vlastností sítě.

Dalšími funkcemi ZDO jsou správa spojování a správa uzlu.

2.4.4 Síťová vrstva (NWK layer)

Od síťové vrstvy je požadováno, aby zajistila správné fungování linkové vrstvy (definované standardem IEEE 802.15.4-2003) a poskytovala potřebné rozhraní pro aplikační vrstvu (APL). Vrstva MAC poskytuje pouze single-hop přenos, hlavním úkolem síťové vrstvy je tedy zajištění komunikace se zařízeními přístupnými přes routery (tedy multi-hop). Koncept pro rozhraní s aplikační vrstvou obsahuje dvě služby. Jedná se opět o data entity a management entity (NLDE resp. NLME). Zatímco NLDE má za úkol pouze připravit rámce pro MAC vrstvu přidáním hlavičky a být schopna doručit rámce dalšímu zařízení, NLME toho má na starosti mnohem víc. Je zodpovědná za konfiguraci zařízení např. při přihlášení do sítě, za vytvoření, přihlášení a opuštění sítě, dále za přidělování adres přihlášeným zařízením (v případě koordinátora nebo routeru) a také za objevení dalších zařízení.

2.4.4.1 Obecný formát NWK rámce

Jsou definovány dva formáty NWK rámce. Jedním je datový, tím druhým je

příkazový rámec. Jejich zobecnění je možné znázornit dle tabulky 12.

Byty: 2	2	2	1	1	různý
Frame Control	Destination Address	Source Address	Radius	Sequence Number	Frame Payload
	Routing Fields				
NWK Header					NWK Payload

Tabulka 12: Obecný formát NWK rámce

Datový rámec má stejnou hlavičku (NWK Header), jako je v obecném formátu. Zbytek obsahuje data, která má za úkol (od vyšší vrstvy – tedy APS) NWK odeslat. Příkazový rámec může obsahovat 4 typy příkazů – Route request, Route reply, Route error a Leave. První z je použit při hledání routerů pro cestu k cílovému zařízení. Další dva jsou odpovědí na první příkaz. Pokud chce zařízení opustit síť, použije příkaz Leave.

2.4.5 Způsob zabezpečení ZigBee zařízení

Zabezpečení dat ZigBee zařízení zahrnuje metody pro ustanovení klíče, jeho transport a bezpečnost vysílaných rámců. Úroveň zabezpečení záleží na bezpečném uchovávání klíčů a na správné implementaci kryptografických algoritmů. Zabezpečení je prováděno pouze mezi různými zařízeními, ne v rámci jednoho. Vrstvy a aplikace jednoho zařízení si navzájem „důvěřují“.

Samotné zabezpečení konkrétního zařízení je založeno na linkových a síťových klíčích. Linkové jsou používány při vzájemné komunikaci dvou zařízení, kdežto síťové klíče najdou uplatnění při broadcastu. V obou případech se jedná o 128-bitové klíče. Jejich získání je možné buďto ustanovením během provozu, nebo jsou klíče předinstalovány, například při výrobě.

3. Bluetooth

3.1 Historie

Rozvoj technologie Bluetooth řídí zvláštní zájmová skupina SIG (Special Interest Group). Ta se skládá z 9 velkých společností, kterými jsou Microsoft, Intel, Toshiba, Motorola, 3Com, Ericsson, Nokia, IBM, Lucent Technologies a spousta členských firem. SIG vznikla v polovině roku 1998. První specifikace (verze 1.0) byla vydána v roce 1999. Ve verzi 1.2 je maximální dosah zvětšen na 100 metrů (u verze 1.0 a 1.1 10 metrů), specifikace této verze byla uvolněna v listopadu roku 2003. V současné době funguje nejvíce zařízení na tomto vývojovém stupni standardu. Téměř přesně o rok později byla dopracována zatím nejnovější verze. Ta má název 2.0 + EDR (EDR = Enhanced Data Rate). Disponuje třikrát větší rychlostí než předchozí.

3.2 Technické parametry

Stejně jako ZigBee, i tento standard komunikuje ve volném ISM pásmu 2,4 GHz. Vzhledem k bezlicenčnímu provozu je to naprosto pochopitelné (stejně tak u Wi-Fi). V tomto pásmu definuje 79 komunikačních kanálů. Rychlost přenosu závisí na zvoleném kódování. Verze 2.0 podporuje jak Basic rate (modulace GFSK), tak i nové Enhanced data rate (PSK). Prvně jmenované kódování udržuje zpětnou kompatibilitu z nižšími specifikacemi. Dovoluje komunikovat maximální teoretickou rychlostí 1Mbit/s. Novější z nich dosahuje nejvyšší rychlosti okolo 3Mbit/s. Ovšem pro lepší kvalitu komunikace např. při zhoršených podmínkách je povoleno v rámci EDR používat nižší rychlost (sníží chybovost přenášených dat) 2Mbit/s.

3.3 Popis

Tento standard je určen pro použití při tvorbě WPAN sítí. Jedním z cílů je náhrada kabelů sériové komunikace. Prioritou Bluetooth je tedy rychlejší komunikace se středním dosahem. Bluetooth zařízení používají profily, které definují, jakým způsobem mezi sebou zařízení komunikují (to má se ZigBee společné). O tom, že Bluetooth směřuje hlavně k náhradě kabelů, svědčí už samotné názvy profilů, např. Bluetooth Serial Port, File Transfer, Network Access a další.

Tato technologie je implementována v aplikacích jako mobilní telefony, GPS zařízení, PC periférie, handsfree k mobilnímu telefonu, tiskárny, PDA, fax a samozřejmě PC.

4. Porovnání a vymezení využití

Technologie Bluetooth je na trhu již velice dlouhou dobu a dostala čas se prosadit. V počátcích to ovšem na úspěch nevypadalo. Mnozí odborníci ji zavrhovali argumenty, že je drahá, náročná a neefektivní. V současné době je ale Bluetooth jednou z nejrozšířenějších pro WPAN.

ZigBee je technologie velice mladá. Její rozšíření v praxi je zatím téměř nulové. Mezi největší výrobce ZigBee transceiverů patří Chipcon, Ember, Honeywell, Freescale (Motorola) a Microchip. Tradiční výrobce mikropočítačů Atmel je zatím ve vývoji ZigBee modulů velmi pozadu.

V porovnání lze tedy u Bluetooth vycházet z praktických znalostí, kdežto u ZigBee pouze z technických parametrů a předpokladů.

Nejprve je potřeba vymezit aplikace, pro které se jednotlivé protokoly hodí. Často kladenou otázkou bývá, zda si oba standardy konkurují, nebo se doplňují. Může být použito ZigBee tam, kde úspěšně funguje Bluetooth a naopak?

Shrňme tedy základní vlastnosti obou standardů.

Bluetooth: rychlost přenosu 1-3Mbit/s, systémové zdroje 250kB, 7 uzlů v síti (v 1.x), dosah 100m, objevení jiného zařízení cca 2 sekundy

ZigBee: rychlost přenosu 20-250kb/s, systémové zdroje 30kB, 256/65k uzlů v síti, dosah 100m (i více), objevení jiného zařízení řádově v milisekundách

ZigBee je (oproti Bluetooth) standard, který je značně jednodušší pro implementaci. Systémové nároky na mikrokontroler jsou menší. Berme také v úvahu, že mikroprocesor má za úkol nejen ovládat ZigBee transceiver a zajišťovat funkčnost protokolu, ale také právě řídit aplikaci. Bezdrátový protokol není v aplikaci to hlavní, ale pouze prostředek komunikace. Pro zařízení, které komunikují standardem Bluetooth, je tedy použití

jednodušších, levnějších 8-bitových mikrokontrolerů téměř nemožné. V dnešní době jsou nejlepší „osmibity“ vybaveny pamětí právě 256kB, což stačí akorát na realizaci protokolu. Naproti tomu paměť nutná pro kód při použití ZigBee je velice malá, takže je naopak použití 8-bitových mikrokontrolerů doporučeno.

Maximální přenosová rychlost je opět značně rozlišná. Zatímco se Bluetooth hodí k přenášení větších souborů, například přesouvání obrázků a audio souborů z počítače do mobilního telefonu nebo PDA a odesílání souborů pro tisk, ZigBee má přenosovou rychlost podstatně nižší. Je tedy vhodná pro aplikace, kde nezáleží tolik na rychlosti komunikace, ale na její kvalitě, jednoduchosti hardware a software, nižší prodlevě při přenosu, atd. Při měření rychlosti Bluetooth v praxi jsem dosáhl v průměru 30kB/s (tedy 240kb/s). Test obsahoval kopírování větších souborů (250kB-1MB) z počítače do mobilního telefonu a PDA.

Velice rozdílná je také doba, kterou potřebují oba standardy na to, aby našli nové zařízení a odpověděli na příkaz. Toto není samozřejmě dané hardwarem, ale definovaným protokolem. ZigBee je v tomto ohledu navržen tak, aby dokázal velmi rychle reagovat na různé podněty.

Z uvedených vlastností tedy vyplývají oblasti využití obou technologií.

Bluetooth je standard s robustnějším protokolem, má vysokou přenosovou rychlost. Na druhé straně pomaleji reaguje, je náročnější na vývoj software. Hodí se do takových typů aplikací, kde je vyžadována solidní rychlost komunikace, nevadí prodlevy (např. zásahy uživatele) a nároky na prostor pro protokol. Těmito aplikacemi jsou zařízení, které mají vlastní prostředí, operační systém, nebo i jednodušší zařízení, které ale používají třeba pouze jeden profil. Nasazení Bluetooth lze doporučit u mobilních telefonů, počítačů, PDA, fotoaparátů a veškerých PC periferiích.

ZigBee je velice jednoduchý a má relativně nízkou přenosovou rychlost. Rychle reaguje na požadavky, díky nižším rychlostem nabízí kvalitní komunikaci a snadno se implementuje do naprosto jednoduchých zařízení. Cílovými aplikacemi jsou tedy taková zařízení, která posílají malé množství dat, ale kladou důraz na to, aby data došla včas a v pořádku a mají málo místa pro protokol bezdrátového standardu. Tedy nemají vlastní operační systém ani uživatelské prostředí. Je vhodný pro různé typy senzorů s využitím v domácnostech, továrnách i nemocnicích, také jako náhradu za dálková ovládání a další zařízení komunikující přes infračervený port. Lze jej též použít pro některé PC periferie –

například klávesnice a myši.

Přesto, že si obě technologie mohou v určitých oblastech konkurovat, v zásadě se doplňují.

5. Aplikace ZigBee

V této kapitole podrobně popíší způsob, jakým lze tuto technologii aplikovat do praxe. Nejprve je potřeba si jasně uvědomit všechny aspekty, které budou hrát při vývoji důležitou roli a jejich zanedbání by mohlo vést k neúspěchu. Nezbytná je představa o tom, jak má naše zařízení vypadat a co všechno bude vykonávat. Pokud tedy máme vyřešenou hardwarovou část, můžeme přistoupit k programování software.

5.1 Než začneme

Jako příklad, který nás bude provázet, jsem zvolil zařízení, které sbírá údaje od okolních zařízení vybavených senzory (a samo je vybaveno stejnými) detekujícími teplotu, vlhkost a přítomnost nebezpečných plynů. Samo je připojeno kabelem (sériová linka) k PC, které údaje vyhodnocuje a v případě překročení limitů informuje bezpečnostní středisko. Celý komplet (všechna zařízení) se nachází v rozlehlé hale továrny.

Před začátkem vývoje konkrétní aplikace si musíme uvědomit základní informace, tedy nároky a požadavky, které jsou od našeho produktu očekávány. Naše zařízení má tedy za úkol sbírat údaje od okolních senzorů rozmístěných po hale továrny, přidat k nim svoje a předat data počítači.

5.2 Sestavení hardware

Tato část vývoje je velice důležitá. Pokud bychom opominuli nějaká důležitá fakta a desku plošných spojů (DPS) navrhli s chybami, v lepším případě by zařízení nefungovalo. Ovšem v tom horším by se chovalo chybově a my bychom v další části vývoje přisuzovali chyby softwaru.

Nyní ale přistoupíme k návrhu DPS. Zvolíme hlavní součástky, od kterých se budou odvíjet jednak ostatní nutné součástky (tranzistory, odpory, kondenzátory...) a také rozhraní, kterým budou komponenty komunikovat. Jako řídicí jednotku zvolíme mikrokontroler od firmy Atmel. Tato volba vychází jednak proto, že mám s těmito čipy zkušenost, a také pro dostupnost vývojových nástrojů. Od mikrokontroleru požadujeme paměť pro kód zhruba 64kB. Přibližně 32kB vyžaduje protokol ZigBee, 32kB by tedy měl být dostatečný prostor pro aplikaci.

Pro získání údajů potřebujeme A/D převodník pro převod údajů z čidel. Komunikace s PC probíhá sériovou linkou a s transceiverem přes SPI rozhraní. Takt procesoru záleží i na zvoleném napětí. Na výkon 8MHz je zapotřebí minimálně 2,7V. Protože není nikde uváděn doporučený takt, předpokládáme, že bude dostačující (pokud by při testování docházelo k většímu výpadku příjmu paketů, je možné, že se tak děje z důvodu nedostatečného taktu procesoru). Z vybraných parametrů nám jako nejvhodnější mikrokontroler vychází Atmel ATmega64L.

Nyní musíme zvolit bezdrátovou jednotku. K tomu nám poslouží ZigBee modul CC2420 od firmy Chipcon. Ten je připraven pro standard IEEE 802.15.4-2003 (tedy pro fyzickou a linkovou vrstvu). Může být napájen dvěma způsoby. Buďto použitím externího stabilizátoru napětí (možný rozsah vstupního napětí 1,6-2,0V), nebo využitím integrovaného (rozsah 2,1-3,6V). Vývoj směřujeme tak, aby bylo možné jednu DPS použít v co nejvíce případech. Naše zařízení může sice být napájeno z adaptéru (je v blízkosti PC), ale senzory umístěné po celé hale budou jistě napájeny bateriemi. Proto se snažíme najít kompromis mezi rychlostí mikrokontroleru a napětím obvodu. Je možné taktovat procesor až na 16MHz, ale pouze v případě, že napětí bude mít úroveň 4,5V. Při napájení DPS bateriemi typu AAA (1,2V) to znamená použití 5 takových baterií, protože musíme napětí nejprve stabilizovat (speciální stabilizátor má úbytek proudu 0,4V). Při taktu 8MHz

a napětí 2,7V (plus 0,4V, tedy 3,1V) nám pro provoz vystačí 3 AAA baterie.

Ke spojení mikrokontroleru a PC potřebujeme měnič napěťových úrovní, protože úrovně čipu a sériové linky nejsou kompatibilní. Zde padá volba na integrovaný obvod MAX232.

4kB integrované paměti RAM procesoru nebudou pro aplikaci postačovat, rozšíříme ji tedy externí pamětí 64kB, což je zároveň maximum. Protože se jedná o vývoj a zatím neznáme minimální nebo doporučené hodnoty, vybavíme naše zařízení pamětí o této velikosti, i když to zřejmě bude více, než je potřeba.

Transciever bude pro svou funkčnost potřebovat anténu. Na výběr máme ze dvou (doporučovaných) možností. Můžeme použít externí anténu pro frekvenci 2,4GHz (např. jako na produktech pro Wi-Fi), která má lepší signál a menší nároky na návrh DPS. Daní za tyto výhody jsou její rozměry. Druhou volbou je anténa integrovaná, tedy vyleptaná na DPS. Její vlastnosti jsou opačné než u externí antény, je levnější, ale zabere více místa. Pro ušetření nákladů a názornost zvolíme integrovanou anténu.

Množství součástek, které jsou potřeba pro správné fungování antény, je malé. Pro vytvoření antény musíme znát její rozměry. Ty nalezneme v dokumentaci k výrobkům firmy Chipcon a jedná se o anténu typu inverted-F. Ve vývojovém kitu s ní dle dokumentace dosáhli vývojáři firmy Chipcon komunikace na vzdálenost 170 metrů.

DPS ale neslouží pouze k přenosu dat, ale také ke sběru dat ze senzorů. Ty jsou připojeny na osmikanálový integrovaný 10-bitový A/D převodník.

V našem případě vyvíjíme zařízení s tím, že ještě neznáme dokonale ZigBee. Proto by měla DPS obsahovat tlačítka, LED diody a LCD displej (nebo zpětnou komunikaci z PC) pro zjišťování stavů mikrokontroleru a ZigBee modulu během testování.

Po rozvržení desky plošných spojů a seznamu součástek přejdeme ke konkrétnímu návrhu DPS. K tomuto účelu nám může posloužit řada programů. Jedním z nich je Eagle od firmy Cadsoft. Jeho použití je jednoduché a pro tuto aplikaci postačující. Program je možno v rámci studentské licence používat bez omezení pro návrh DPS do rozměru 8x10 cm.

Jeho knihovny ale neobsahují integrované obvody ATmega64L a CC2420. Je tedy nutné je do knihoven přidat. Dle dokumentace přesně nakreslíme jeho pouzdro a piny. Dále vyrobíme jeho schéma a přiřadíme pinům funkce.

Vzhledem k složitosti zařízení a také proto, že jej chceme minimalizovat, navrhne desku jako vícevrstvou. V materiálech firem, vyrábějících vývojové kity, se objevují (a jsou doporučovány) desky čtyřvrstvé (ve studentské verzi je však možno navrhovat pouze dvouvrstvé). Zařízení, komunikující na takto vysokých frekvencích, jsou velmi citlivá na rozvržení DPS. Někteří výrobci (včetně firmy Chipcon) doporučují při vytváření vlastních desek okopírovat jimi navržené DPS (v oblasti transcieveru). Minimalizujeme tak šum.

Vyrobení takovéto desky plošných spojů je v amatérských podmínkách nemožné. Výrobek si lze nechat zhotovit například od firmy Tesla.

5.3 Programování

Naprogramovat tuto aplikaci znamená nahrát mnohokrát strojový kód do mikrokontroleru a ozkoušet v něm. Je samozřejmě nemyslitelné čip v programátoru nahrát, přemístit do zařízení, otestovat, znovu umístit do programátoru atd. Firma Atmel nabízí pro své produkty dvě varianty.

První, a zároveň jednodušší, je programování pomocí ISP (In System Programming). Toto rozhraní nabízí programování mikročipu přímo v aplikaci. Konektor programátoru je součástí DPS, je připojen přímo k mikročipu. Konektor obsahuje dvě řady po pěti pinech, jejichž rozestup je desetina palce (2,54mm). Toto rozhraní slouží pouze k naprogramování čipu, k testování musí postačit např. LED diody.

Druhou možností je rozhraní JTAG, které je kompatibilní se standardem IEEE 1149.1. V případě jeho použití se jedná o programátor kombinovaný s emulátorem. Atmel nedoporučuje používat klasické emulátory, tedy zařízení, která napodobují mikrokontrolery. V mezních situacích se může emulátor zachovat jinak než čip a velice tak zkomplikovat situaci. Programátor s rozhraním JTAG je podobně jako ISP programátor připojen k mikrokontroleru. V obslužném programu, který běží na PC, můžeme definovat breakpointy, díky čemuž je nám umožněno zjišťovat stavy všech integrovaných periferních jednotek, interní a externí paměti RAM, paměti flash (paměť pro kód) a EEPROM a také hodnoty registrů.

Rozhodnutí, jaká volba je pro nás vhodnější, ovlivňuje i cena. Originální programátor od firmy Atmel s ISP rozhraním seženeme za cenu okolo 1500Kč, JTAG programátor od stejného výrobce je přibližně desetkrát dražší. I v tomto případě oproti klasickým emulátorům výrazně ušetříme.

Dále se musíme rozhodnout, jaký programovací jazyk zvolíme, aby vyhovoval jak použití pro daný mikrokontroler, tak samotnému programátorovi. Při volbě jazyka musíme zvážit tvorbu overhead code. Je to nárůst kódu, který vytvoří kompilátor, aby mohl správně přeložit zdrojový kód do strojového. Platí, že čím vyspělejší a komplexnější jazyk, tím větší overhead code. Pokud napíšeme kód pro stejnou aplikaci v assembleru a v jazyce vyšší úrovně, poměr strojových kódů je právě overhead code.

Ačkoli tedy kompilace assembleru nevytváří žádný kód navíc, programování takovéto aplikace je naprosto nereálné. Musíme tedy zvolit jiný jazyk. Mezi nejrozšířenější ve světě mikrokontrolerů patří C a Pascal, poslední dobou se hlavně u 16-bitových mikročipů rozšiřuje modifikace jazyku C++. Firma Atmel nabízí kompletní vývojové prostředí, které obsahuje textový editor, různé možnosti nastavení a hlavně v novějších verzích i spolupráci s kompilátorem jazyka C (WinAVR) pro čipy Atmel rodiny AVR. Součástí AVR Studio je i vestavěný simulátor. Z těchto důvodů tedy zvolíme jazyk C.

Nyní přistoupíme k samotnému programování. Nejprve si ujasníme schéma, jakým bude software pracovat. Program bude obsahovat dvě části. První je ZigBee protokol, tou druhou je samotná aplikace. Ta se skládá z vyhodnocování údajů ze senzorů, komunikace s PC a ovládání transcieveru resp. užití protokolu. Sběr hodnot ze senzorů bude probíhat A/D konverzí každých 60 vteřin. Časování této procedury je zajištěno hardwarovým časovačem. Po uběhnutí intervalu je spuštěna konverze a uloženy výsledky. Okolní zařízení provádějí měření ve stejném intervalu a data ihned po získání odešlou. Naše zařízení po obdržení dat provede již zmíněné získání údajů a všechna data odešle přes sériovou linku do PC. Nyní není pro naši potřebu důležité, jak bude probíhat odesílání dat do PC a měření vlastních senzorů. Podstatná je pro nás pouze komunikace s protokolem a jeho funkce.

ZigBee standard se skládá se čtyř vrstev. Při tvorbě kódu budeme postupovat od nejnižší směrem nahoru. Vrstva se z koncepčního hlediska skládá z konstant a proměnných atributů, které tvoří informační bázi vrstvy, a také z rozhraní, kterým vrstva komunikuje s tou nadřazenou. Fyzická vrstva bývá označována jako firmware. ZigBee moduly jsou často

vybavovány funkcemi, které nám ušetří práci. Zjistíme tedy, o které (užitečné z hlediska programování fyzické vrstvy) se v případě CC2420 jedná. V první řadě je to hardwarové zabezpečení přenášených dat pomocí AES-128. Nemusíme tedy tento algoritmus implementovat programově. Další usnadnění nabízí automatický výpočet CRC před odesláním dat, stejně jako kontrola přijatých dat. Modul má v paměti dva bloky po 128 bytech pro bufferování dat přijatých a k odeslání.

Rozhraní pro komunikaci s transceiverem není pochopitelně součástí standardu. My ovšem tuto část opominout nemůžeme. Výměna dat mezi ATmega64L a CC2420 probíhá pomocí SPI (Serial Peripheral Interface) kompatibilního rozhraní. To obsahuje dva piny pro plně duplexní komunikaci, pin Chip Select, který ohraničuje celou komunikaci, a pin, určující frekvenci komunikace. Toto rozhraní je synchronní, jedno ze zařízení funguje jako master a určuje frekvenci (tedy ovládá tyto piny). CC2420 se chová jako zařízení typu slave, tedy je ovládané masterem.

Když transceiver přijme data nebo dokončí měření, uvědomí mikroprocesor IRQ (Interrupt Request) pinem. Mikrokontroler poté přečtením speciálního registru (přes SPI rozhraní) zjistí, o jaký typ přerušení se jedná a podle toho se dále zachová. ATmega64L při změně hodnoty na pinu připojeném k IRQ vygeneruje hardwarové přerušení. Mikročip získá data z transceiveru a uvědomí o tom linkovou vrstvu.

Kód celého protokolu uspořádáme následovně. Každá vrstva se bude skládat ze souborů, které obsahují funkce, hlavičku (oba reprezentují rozhraní), definici konstant a proměnných definovaných vrstvou (reprezentuje informační bázi). Soubory s funkcemi a s hlavičkou existují pro každou službu rozhraní zvlášť (většinou obsahují data a management entity).

Fyzické vrstvě navíc náleží komunikační rozhraní se ZigBee modulem. Standardem není postup striktně předepsán, ale vzhledem k náročnosti aplikace se budeme doporučení držet (to je popsáno v dokumentacích k příslušným vrstvám). Pro konkrétní představu vypíši seznam souborů, které by měla fyzická vrstva obsahovat (zkratka sap znamená service access point, tedy bod přístupu ke službě, kterou vrstva poskytuje): `phy_pd_sap.c`, `phy_pd_sap.h`, `phy_plme_sap.c`, `phy_plme_sap.h`, `phy_pib.h` plus soubory s kódem pro řízení transceiveru `cc2420_rf_sap.c`, `cc2420_rf_sap.h`.

Schéma linkové vrstvy (MAC) je velice podobné, takže se bude shodovat i struktura souborů. K linkové vrstvě přistupujeme opět rozhraním, skládajícího se ze dvou

přístupových bodů (management a data), vrstva obsahuje též svoji informační bázi.

Než přejdu ke zbylým vrstvám (definovaných ZigBee) a celé aplikaci, na příkladu ukážu, jak by mohlo vypadat odeslání dat. Postup předvedu pouze v rámci PHY a MAC vrstev, jako ukázka postupu programování to postačí. Na začátku je vrstvě MAC předán požadavek od vyšší vrstvy (tedy NWK, jíž zadá tento úkol APL) k odeslání dat. To můžeme chápat jako provedení funkce datové entity rozhraní MAC vrstvy. Údaje, nutné k vyřízení této žádosti, tedy adresa odesílatele a příjemce, samotná data atd., jsou argumenty této funkce. Nazvěme ji třeba `macde_data_send()`. Ta pak na konci vrátí status odeslání zprávy a její handle. Protože nemůže funkce vrátit víc oddělených hodnot, utvoříme z návratových hodnot strukturu.

Jako první provede `macde_data_send()` nastavení transcieveru na příjem voláním `phyme_set_trx(rx)`, jejíž návratová hodnota udává, zda se jej podařilo nastavit. Nyní je vše připraveno na provedení CCA (zjištění, zda je kanál volný). Takový úkol má na starosti `phyme_do_cca()`. Návratová hodnota této funkce je hodnota znamenající úspěch, nebo neúspěch (kanál je momentálně obsazený). Kanál je volný, takže přistoupíme k odeslání dat. Přepneme ZigBee modul do vysílacího režimu – `phyde_set_trx(tx)`.

Pro předání dat fyzické vrstvě použijeme `phyde_data_send()` (tato funkce je zodpovědná za korektní vyslání dat transcieverem), která opět vrací strukturu obsahující status odeslání a handle zprávy. Odeslali jsme data a přepneme modul na příjem pro případné potvrzení doručení k adresátovi užitím `phy_set_trx(rx)`. Provedli jsme všechny náležitosti potřebné k odeslání zprávy a už jen vrátíme handle zprávy a status doručení znamenající úspěch.

Tento postup programování odeslání paketu samozřejmě není jediný správný, vychází ale ze schématu vrstev a doporučení dokumentace. Názvy funkcí jsem volil tak, aby bylo na první pohled zřetelné, jakému rozhraní náleží a co je jejich úkolem. Začátek názvu značí, zda se jedná o linkovou, nebo fyzickou vrstvu (mac, nebo phy) a službu daného rozhraní (me – management entity, de – data entity). Zbytek potom udává, co přesně konkrétní funkce provádí.

ZigBee ale obsahuje i další vrstvy, jimiž jsou síťová (NWK) a aplikační (APL). Postup programování je i nadále stejný, ovšem vyšší vrstvy vykonávají mnohem více úkolů. Nyní popíši, jakým způsobem naprogramujeme provázání s bezdrátovým protokolem.

Sada senzorů je pohledem ZigBee aplikační objekt. Pokud bychom potřebovali umožnit počítači (připojenému přes sériový port) přímý přístup k protokolu, může i on představovat další aplikační objekt. Naše zařízení je reprezentováno ZigBee Device Object (ZDO), takže hostitelská aplikace je vlastně objekt (ovšem speciální a jeho index je vždy 0). Všechny části aplikace tedy komunikují s protokolem rozhraním aplikační vrstvy. Právě proto jsme při návrhu software přistupovali k programu jako ke dvěma odděleným částem.

Jak bylo uvedeno v kapitole rozebírající standard, objekty si pro svou komunikaci musí vybrat profil. To uvádím z důvodu, že si při návrhu software musíme tento aspekt uvědomit a přizpůsobit tomu strukturu programu.

V popisu profilu je totiž mimo jiné uvedeno, kolik paměti musí mít koordinátor (což je i náš případ) vyhrazeno pro spojovací a routovací tabulku, pro popisovače zařízení používajících sleep mode a také maximální počty zařízení spadajících pod jeden router resp. koordinátor. Při špatném výběru profilu bychom mohli na konci zjistit, že jsme nějaký limit překročili a musíme tuto část přeprogramovat.

6. Závěr

Jak jsem již na jiných místech uvedl, standard ZigBee je jednoduchý na implementaci do aplikací, které jsou řízeny jednoduchými 8-bitovými mikrokontrolery. Tento důvod a volně šířitelný protokol jsou hlavní důvody vedoucí k optimizmu ohledně budoucnosti ZigBee.

Mým úkolem bylo provést strukturou tohoto bezdrátového protokolu a ukázat jeho implementaci ve skutečné aplikaci. Doufám, že jsem toho dosáhl, a že se mé teoretické poznatky rozšíří v budoucnosti o ty praktické.

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky
Akademický rok: 2004/2005

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Zdeněk ŠÍMA
Studijní program: B1802 Aplikovaná informatika
Studijní obor: Výpočetní technika

Název tématu: ZigBee & Bluetooth

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

Se zvyšující se mobilitou všech počítačových systémů roste i úloha bezdrátových sítí. Jejich vhodnost pro jednotlivé aplikace je závislá na teoretických a technických parametrech. Pro použití v experimentální praxi připadají v úvahu hlavně dvě normy starší Bluetooth a zcela nová definice ZigBee.

Úkolem posluchače je porovnat a vymezit vůči sobě protokoly sítí na bázi ZigBee a Bluetooth. Otestovat použitelnost v praktickém experimentálním provozu a sestavit doporučení implementace ve vědecké praxi, případně při řízení experimentů. Dále zpracovat knihovny a návody pro začlenění výše uvedených bezdrátových sítí do vývojového softwaru. Případně přímo sestavit vhodné knihovny či jmenné prostory pro aplikační software.

Rozsah práce: 60

Rozsah příloh:

Forma zpracování bakalářské práce: tištěná

Seznam odborné literatury:

1. Internetový portál www.ZigBee.com
2. Internetový portál www.bluetooth.org
3. Lacko, Luboslav: Programujeme mobilní aplikace ve Visual Studiu.NET, Computer Press a.s.BRNO.2004. ISBN 80-251-0176-2

Vedoucí bakalářské práce:

Ing. Václav Novák, CSc.
Katedra informatiky

Datum zadání bakalářské práce:

22. dubna 2005

Termín odevzdání bakalářské práce:

30. dubna 2006

prof. RNDr. Miroslav Papáček, CSc.

děkan



doc. RNDr. Václav Nýdl, CSc.

vedoucí katedry

V Českých Budějovicích dne 22. dubna 2005