

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE PRO VYČÍTÁNÍ DAT Z CHYTRÝCH MĚŘIČŮ POMOCÍ BEZDRÁTOVÉ TECHNOLOGIE ZIGBEE

APPLICATION FOR SMART METER DATA READOUT USING ZIGBEE TECHNOLOGY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jonáš Bräuer

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Kryštof Zeman

BRNO 2017



Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Jonáš Bräuer

ID: 162522

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Aplikace pro vyčítání dat z chytrých měřičů pomocí bezdrátové technologie ZigBee

POKYNY PRO VYPRACOVÁNÍ:

Na poli chytrých měřičů, které jsou nedílnou součástí Internetu věcí, stále probíhá vývoj. Jednou z jeho částí je i vývoj komunikačních jednotek a jejich testování. Cílem práce bude seznámení se s možností využití technologie ZigBee pro přenos dat od chytrých měřičů do koncentrátoru. V praktické části se student zaměří na sestavení komunikace mezi Zigbee zařízením a chytrým měřičem. Přes toto spojení bude následně vyčítat data, případně ovládat daný měřič.

DOPORUČENÁ LITERATURA:

[1] GISLASON, Drew. Zigbee wireless networking. Amsterdam: Elsevier / Newnes, c2008. ISBN 978-0-75-6-8597-9.

[2] FLOERKEMEIER, Christian. (ed.). The Internet of things: first International Conference, IOT 2008 : Zurich, Switzerland, March 26-28, 2008 : proceedings. Berlin: Springer, c2008. Lecture notes in computer science, 4952. ISBN 978-3-540-78730-3.

Termín zadání: 1.2.2017

Termín odevzdání: 8.6.2017

Vedoucí práce: Ing. Kryštof Zeman

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Bakalářská práce se zabývá komunikací bezdrátových zařízení v ZigBee síti. V teoretické části je protokol ZigBee srovnán s dalšími možnými řešeními sensorových sítí (WPAN a LPWAN) a je rozebrána jeho struktura. V praktické části byla vytvořena Java aplikace na vzdálenou konfiguraci a vyčítání dat.

KLÍČOVÁ SLOVA

ZigBee, XBee, IEEE 802.15.4, Java, IoT, M2M

ABSTRACT

Bachelor thesis deals with communication of wireless devices using ZigBee technology. In the theoretical part, the ZigBee protocol is compared to other technological solutions of sensor networks (WPAN and LPWAN) and its structure is being explained. In the practical part, a Java application for remote configuration and data collection has been created.

KEYWORDS

ZigBee, XBee, IEEE 802.15.4, Java, IoT, M2M

BRÄUER, Jonáš *Aplikace pro vyčítání dat z chytrých měřičů pomocí bezdrátové technologie ZigBee (IEEE 802.15.4)* : bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 46 s. Vedoucí práce byl Ing. Kryštof Zeman

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Aplikace pro vyčítání dat z chytrých měřičů pomocí bezdrátové technologie ZigBee (IEEE 802.15.4)“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Kryštofu Zemanovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora(-ky)



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

PODĚKOVÁNÍ

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

Brno

.....

podpis autora(-ky)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	11
1 IoT	12
1.1 Historie	12
1.2 Oblasti využití IoT	12
2 M2M	13
2.1 Historie	13
2.2 Dostupné technologie	13
3 ZigBee	15
3.1 Struktura dat	15
3.1.1 Struktura paketu na síťové vrstvě	16
3.1.2 Struktura rámce na linkové vrstvě	17
3.1.3 Struktura rámce na fyzické vrstvě	18
3.2 Struktura ZigBee	18
3.2.1 Fyzická vrstva	18
3.2.2 Linková vrstva	19
3.2.3 Síťová vrstva	21
3.2.4 Aplikační vrstva	24
3.3 Princip komunikace	24
3.3.1 ZigBee síť	24
3.3.2 Adresace	25
3.3.3 Adresace v rámci uzlu	26
4 Praktická realizace	28
4.1 Vybrané řešení	29
4.1.1 Digi XBee	29
4.1.2 Crowduino deska	31
4.1.3 RaspBerry Pi	32
4.1.4 Java RxTx	33
4.1.5 XBee Java Library	33
4.2 Modelový scénář	33
4.3 Výstup práce	34
4.3.1 Příprava	34
4.3.2 Arduino program	35
4.3.3 Java aplikace	36
4.4 Příklad praktického využití aplikace	38

5 Závěr	40
Literatura	41
Seznam symbolů, veličin a zkratk	43
Seznam příloh	45
A Obsah přiloženého CD	46

SEZNAM OBRÁZKŮ

3.1	Obecná struktura rámce/paketu napříč vrstvami protokolu	16
3.2	Architektura protokolu ZigBee	19
3.3	Struktura super-rámce	21
3.4	Podporované topologie	22
3.5	Topologie a alokace adres pro Tree routing	22
3.6	Route discovery	23
3.7	Schéma formace a připojení se do sítě [11]	25
3.8	Schéma koncového bodu a v něm obsažených clusterů	28
4.1	Schéma modulu XBee S2C	29
4.2	Printscreen aplikace X-CTU v síťovém módu s modelovým scénářem	31
4.3	Deska Crowduino s naznačeným přepínačem USB-MCU	32
4.4	Struktura použitých rámců	35
4.5	Printscreen vytvořené aplikace	37
4.6	Příklad konfiguračního souboru	38
4.7	Graf naměřených teplot	39

SEZNAM TABULEK

2.1	Přehled základních vlastností jednotlivých protokolů	15
3.1	Funkcionalita RFD	20
3.2	příklady veřejných aplikačních profilů	27
4.1	Technické specifikace modulů XBee S2B a S2C	30
4.2	Některé důležité parametry XBee modulu	34

ÚVOD

Hlavním tématem této bakalářské práce je technologie ZigBee. Jedná se o relativně mladou technologii (První specifikace byla schválena v roce 2004), která se snaží prosadit v oblasti sítí WPAN (Wireless Personal Area Network). Tato technologie byla vyvinuta ke komunikaci na krátké vzdálenosti s velmi nízkou spotřebou energie. ZigBee a podobné “úsporné“ technologie jsou velmi vhodné pro využití v senzorových sítích. Tyto sítě mají velké uplatnění nejen v průmyslové oblasti, kde jsou již několik desetiletí úspěšně využívány, ale začínají získávat na popularitě i v soukromé sféře. Nejčastěji jako sítě pro domácí automatizaci. ZigBee protokol je svými specifikacemi více než vhodný k použití v takovýchto sítích.

V teoretické části této práce bude blíže představena technologie ZigBee a s ní související pojmy. Bude srovnána s jinými technologiemi/standardsy sítí WPAN, popsána její struktura dle vrstev referenčního modelu ISO/OSI a princip jak spolu zařízení v síti komunikují.

Praktická část je poté zaměřena na propojení samotného protokolu ZigBee s aplikací, schopnou vyčítat data z chytého měřiče. Bude popsáno vybrané HW řešení, jak bylo sestaveno spojení mezi počítačem a bezdrátovým modulem, a poté bude popsána i vytvořená aplikace.

1 IOT

IoT (Internet of Things), česky „Internet Věcí“ je pojem označující propojení fyzických objektů do sítě a následné sbírání a zpracování dat. V případě IoT jsou tyto objekty připojeny, jak již název vypovídá, do Internetu, což jim umožňuje nejen komunikovat mezi sebou, ale i přistupovat k Internetovým službám, či přímo interagovat s lidmi. Již dnes můžeme pozorovat, jak na trhu přibývají každodenní chytré zařízení opatřené síťovým rozhraním, jako: chytré hodinky, fitness náramky, zubní kartáčky apod. Takovéto vylepšení všedních objektů umožňuje zaznamenávat a využívat data v dříve nemyslitelných rozměrech.

1.1 Historie

I přesto, že pojem Internet věcí je poměrně nová záležitost (tento termín údajně jako první použil K.Ashton, jeden ze zakladatelů výzkumné skupiny Auto-ID Labs - průkopníka IoT, v názvu své prezentace v roce 1999), začíná velmi rychle nabývat významu. Podle statistik, každoročně vydávaných společností Cisco, připadlo v roce 2003 zhruba 0,08 připojených zařízení na osobu. Již v roce 2004 překonal počet připojených zařízení počet lidí na planetě a v roce 2015 pak na každého člověka připadlo téměř 3,5 připojených zařízení.[2]

1.2 Oblasti využití IoT

Tento vzestupný trend umožňuje vyžití chytrých řešení téměř ve všech aspektech lidské činnosti a výrazně tím zvyšovat efektivitu a bezpečnost procesů.[3]

1. zdraví - E-Health
 - detekce pádu pro starší lidi či postižené
 - kontrola vitálních funkcí
 - sledování pacientů v nemocnicích
2. chytré domy - Home Automation
 - měření a správa spotřeby energií
 - chytré domácí bezpečnostní systémy
3. chytrá města - Smart Cities
 - chytré parkování, optimalizace městského provozu
 - monitorování stavu budov a jiných staveb
 - chytré osvětlení

4. chytré měřiče - Smart Metering
 - měření spotřeby a správa elektrické energie - SmartGrid
 - měření tlaku ve vodovodních potrubích
5. průmysl/logistika - Industrial Control
 - kontrola a správa výrobních procesů
 - systémy pro správu vozového parku
6. zemědělství - Smart Agriculture
 - adaptivní zavlažování
 - monitorování zdraví a změn v chování chovných zvířat

2 M2M

Komunikace mezi přístroji (M2M, Machine to Machine) tvoří samotné jádro myšlenky Internetu věcí. Tento pojem označuje komunikaci (ať už drátovou či bezdrátovou) mezi dvěma a více přístroji bez přímé interakce člověka (narozdíl od sítí H2H, Human to Human, ve kterých zařízení pouze zprostředkují komunikaci, ale koncový uživatel je vždy člověk).

2.1 Historie

Koncept M2M se začal rozvíjet již od druhé poloviny 20. století. Jako první myšlenka M2M se považuje vynález „chytrého“ telefonu Theodora G. Paraskevakose, na základě které začal v roce 1968 pracovat na technologii Caller-ID - posílání údajů o volaném čísle spolu s hlasovými daty po telefonní lince.[4] Sítě M2M jsou dnes velmi rozšířené převážně v oblasti průmyslu, kde umožňují efektivní administrativu procesů a přináší značné finanční úspory.

2.2 Dostupné technologie

Vzhledem k vzrůstajícímu zájmu o M2M komunikaci, dochází k vývoji celé řady technologií pro bezdrátový přenos. Mezi rozšířenější protokoly pro WPAN (Wireless Personal Area Network) patří Bluetooth Smart/BLE (Bluetooth Low Energy), ZigBee, Z-Wave nebo Thread. V oblasti LPWAN (Low-Power Wide-Area Network) poté SigFox a LoRaWAN. Všechny z výše uvedených technologií se zaměřují na nízkoenergetický přenos malých množství dat.

- **Bluetooth** je nejrozšířenější WPAN technologie využívaná převážně v oblasti mobilních zařízení (přenos dat, bezdrátová sluchátka, periferie, ...). Nová specifikace, Bluetooth Smart/Low Energy, z roku 2010 přizpůsobuje Bluetooth

ještě více pro oblast, kde je vyžadována nízká spotřeba. Narozdíl od klasického Bluetooth mají zařízení s BLE možnost uvedení do hlubokého spánku a data vysílat jen v krátkých intervalech. Ani nejnovější specifikace ale zatím neumožňuje tvorbu mesh sítí.[6]

- **Z-Wave** je velmi rozšířená proprietární technologie vyvinutá společností ZenSys, která se specializuje hlavně na aplikaci v chytré domácnosti. Protokol nevyužívá technik rozprostření spektra a signál je klíčován frekvenčním posuvem. Toto řešení je jednodušší na implementaci, ale neposkytuje tak dobré přenosové vlastnosti (odstup signál-šum) ani odolnost proti rušení jako technologie stavějící na IEEE 802.15.4.[12]
- **Thread** je otevřený standard oznámený v roce 2014. Standard je spravován Thread Aliancí (jedním ze zakladatelů je společnost Nest Labs). Stejně jako ZigBee staví na IEEE 802.15.4, ale pro svou síťovou vrstvu využívá IPv6 adresaci (konkrétně technologii 6LoWPAN) a k přenosu protokol UDP (User Datagram Protocol).[10]
- **SigFox** spadá do sítí LPWAN a nabízí alternativu k celulárním sítím. Jedná se také o proprietární řešení, kde společnost udržuje svoji vlastní síťovou infrastrukturu. Technologie využívá ISM (industrial, scientific and medical) sub-GHz pásmo a samotná komunikace využívá velmi úzkých kanálů (Ultra Narrow Band), což umožňuje použití méně kvalitních krystalů ve vysílačích koncových zařízení, tudíž jejich nižší cenu. Topologie SigFox sítě je tvaru hvězdy, kde koncové prvky komunikují s bránou (podobně jako v případě klasických celulárních sítí). Komunikace může probíhat obousměrně, je ovšem značně asymetrická a směr downlink je omezen na pár zpráv za den (maximálně 140).[9]
- **LoRaWAN** je protokol, vyvíjený LoRa aliancí, který využívá k modulaci signálu technologii LoRa, proprietární modulační schéma odvozené od CSS (Chirp Spread Spectrum). LoRaWan je přímou konkurencí SigFox, jedná se ale o otevřenou specifikaci. Jako u SigFox je síťová topologie tvaru hvězdy a komunikace probíhá od koncových prvků ke koncentrátoru (bráně). Narozdíl od technologie SigFox ale nelimituje směr downlink, a proto je vhodnějším kandidátem pro aplikace vyžadující obousměrnou komunikaci ve větší míře.[8]

Protokol	Pásmo	Dosah	Přenosová rychlost	Počet zařízení v síti
Bluetooth [6]	2,4 GHz	100 m	100 kbit/s (BLE)	8
ZigBee [11]	2,4 GHz	10-100 m	20-250 kbit/s	65536
Z-Wave[12]	868 MHz (EU)	30 m	10-100 kbit/s	32768
Thread [10]	2,4 GHz	30 m	250 kbit/s	250
SigFox [9]	868 MHz (EU)	3-50 km	až 1 kbit/s	-
LoRaWAN [8]	868 MHz (EU)	2-22 km	0,3-50 kbit/s	-

Tab. 2.1: Přehled základních vlastností jednotlivých protokolů

3 ZIGBEE

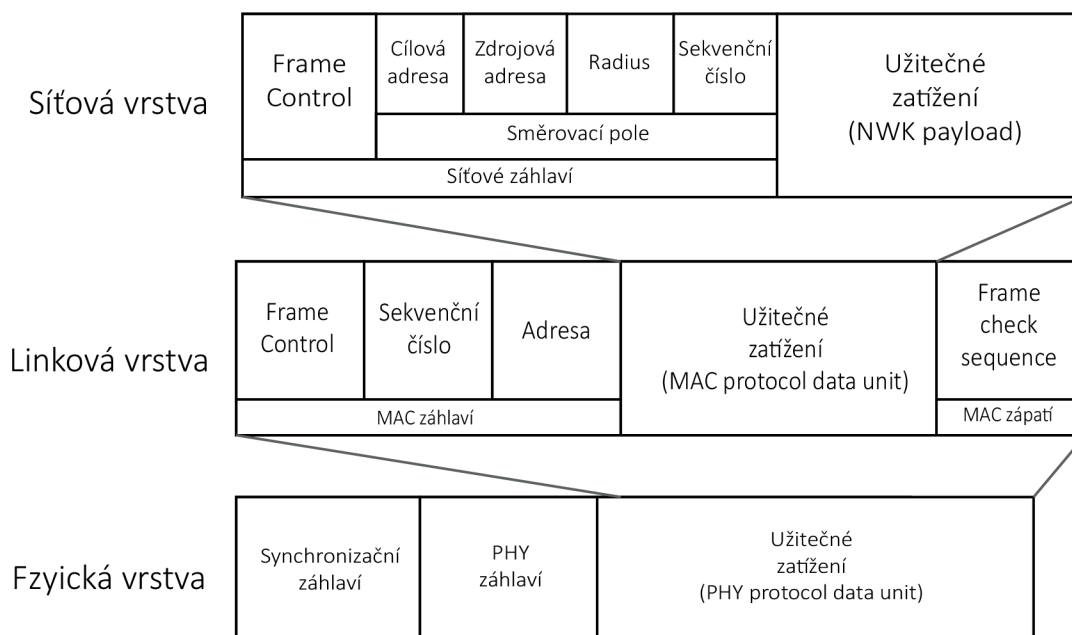
Technologie ZigBee je technologie o jejíž vývoj a standardizaci se stará ZigBee Alliance. První dvě vrstvy (fyzická a linková) dle ISO/OSI modelu jsou převzaté ze standardu IEEE 802.15.4 a Zigbee standard samotný poté na tyto základy přidává síťovou a aplikační vrstvu.

ZigBee je otevřený standard, kdokoliv (kdo se stane členem ZigBee Alliance) má tudíž možnost technologii využít a je zaručená kompatibilita mezi zařízeními různých výrobců (OEM - Original Equipment Manufacturer).

Technologie ZigBee umožňuje tvorbu mesh sítí (možnost i hvězdicové nebo stromové topologie) s celkovým počtem až 65 536 zařízení. Dosah signálu se pohybuje mezi deseti až sto metry, v závislosti na lokaci a okolních podmínkách. ZigBee umožňuje každé zařízení v síti uvést do hlubokého spánku, tedy značnou úsporu energie. Na jednu baterii je zařízení schopno pracovat až několik let.

3.1 Struktura dat

Jako většina dnešních sítí i sítě ZigBee komunikují přepojováním paketů. Data jsou rozdělena aplikací na malé bloky, ty jsou postupně v každé vrstvě protokolu zapouzdřovány metadatami (tento proces se nazývá encapsulation) a poté jsou nezávisle na sobě odeslány sítí. Na úrovni síťové vrstvy datové bloky nazýváme **pakety**, na linkové a fyzické potom **rámce**. Jak již bylo řečeno, pakety/rámce se skládají kromě z dat samotných, i z dat řídicích (metadat), potřebných ke správnému doručení. Bity se přidávají na začátek (záhlaví - header) i na konec (zápatí - trailer). Strukturu jednotlivých paketů/rámců nyní přiblížíme dle vrstev protokolu (od síťové vrstvy směrem k fyzické).



Obr. 3.1: Obecná struktura rámce/paketu napříč vrstvy protokolu

3.1.1 Struktura paketu na síťové vrstvě

Obecně se paket na síťové vrstvě (NPDU, Network Protocol Data Unit) skládá ze záhlaví (header) a přenášených dat (payload). Přenášená data se liší podle typu paketu, ale struktura záhlaví je vždy pevně daná. Povinné pole záhlaví jsou:

1. Frame control (2 byty), obsahuje tyto pole:
 - Typ paketu – určuje zdali se jedná o datový nebo příkazový paket.
 - Verze protokolu – určuje verzi ZigBee protokolu, která je zrovna využívána.
 - Route discovery – lze volit mezi potlačením, povolením nebo vynucením algoritmu route discovery (vysvětleno níže).
2. Cílová adresa (2 byty) – 16bitová ZigBee adresa.
3. Adresa zdroje (2 byty) – 16bitová ZigBee adresa.
4. Radius (1 byte) – jedná se o celé číslo určující kolikrát může být paket směrován (počet skoků), než je zahozen. Předchází se tak zacyklení paketu v síti.
5. Sequence number (1 byte) – číslo určující pořadí paketu. S každým odeslaným paketem se o jedno navýší.

Kromě těchto povinných polí může paket obsahovat ještě pole s 64bitovými IEEE adresami zdroje i cíle (slouží k obnovování záznamů o zařízeních a odhalování konfliktů adres v síti) a pole multicast control (určuje, zdali má být paket směrován i pomocí zařízení, které nepatří do cílové multicastové skupiny).

Rozlišujeme 2 druhy paketů:

1. **Datový paket** (Data frame) – musí obsahovat pole frame control (s hodnotou 00 v poli typ paketu) a kombinaci směrovacích informací potřebnou ke správnému doručení. Jako užitečná informace je vložena sekvence bytů žádaná další vyšší vrstvou k odeslání.
2. **Příkazový paket** (Command frame) – musí obsahovat pole frame control (s hodnotou 01 v poli typ paketu a hodnotou 00 – potlačení – v poli route discovery). Užitečnou informaci tvoří pole: identifikátor příkazu (command identifier) a samotná hodnota příkazu (command payload). Mezi základní příkazy patří:
 - (a) Vyžádání nalezení cesty (Route request) – je využíváno v případě, že není známá cílová adresa a zařízení se rozhodne použít algoritmus Route discovery (vysvětleno níže).
 - (b) Odpověď na Route Request (Route reply) – slouží jako odpověď na Route request v případě, že cílové zařízení bylo nalezeno.
 - (c) Opuštění sítě (Leave) – složí k informování ostatních zařízení nebo k žádání o opuštění sítě.

3.1.2 Struktura rámce na linkové vrstvě

Na úrovni linkové vrstvy je celý síťový paket zapouzdřen mezi MAC záhlaví a zápatí. MAC záhlaví se skládá z polí:

1. Frame control (2 byty), obsahuje mimo jiné tyto pole:
 - Typ rámce – určuje o jaký typ rámce se jedná (beacon rámeček - 000, data - 001, potvrzení přijetí (acknowledgement) - 010, příkaz - 011).
 - Vyžádání potvrzení přijetí (Acknowledgment Request).
 - Mód adresace (zdroje i cíle) – určuje jaká informace je obsažena v polích určených k adresaci (krátká 16bitová adresa nebo rozšířená 64bitová adresa). V případě, že adresa ani PAN (Personal Area Network) identifikátor nejsou obsaženy je hodnota 0 a rámeček je určen pro, nebo pochází od, PAN koordinátora.
 - Verze rámce – indikuje s jakou verzí IEEE 802.15.4 je rámeček kompatibilní.
2. Sekvenční číslo – princip je stejný jako na úrovni síťové vrstvy.
3. PAN identifikátor zdroje/cíle (2 byty) – určuje do jaké sítě, jednoznačně označené PAN identifikátorem, odesílatel nebo příjemce patří.
4. Adresa zdroje/cíle (2 nebo 8 bytů) – krátká nebo rozšířená adresa, v závislosti na módu adresace ve Frame control poli.

Celé MAC zápatí obsahuje vygenerovaný kontrolní kód (FCS – Frame Check Sequence).

Podle funkce rozlišujeme 4 druhy rámců:

1. Beacon rámeček.
2. Data rámeček.
3. Potvrzovací rámeček (Acknowledgement).
4. Příkazový rámeček (Command frame). Mezi základní patří:
 - (a) Vyžádání asociace (Association request) - zařízení doposud nepřihlášené do sítě pomocí tohoto rámcu žádá o asociaci. PAN koordinátor sdělí zařízení výsledek pomocí Association response rámcu.
 - (b) Požadavek na data (Data request).
 - (c) Požadavek na Beacon rámeček (Beacon request) – slouží k vyzvání PAN koordinátora o zaslání informací o síti ve formě Beacon rámcu.

3.1.3 Struktura rámcu na fyzické vrstvě

Na fyzické vrstvě je před celý MAC rámeček předřazeno synchronizační a PHY (Physical Layer) záhlaví, toto záhlaví obsahuje pouze:

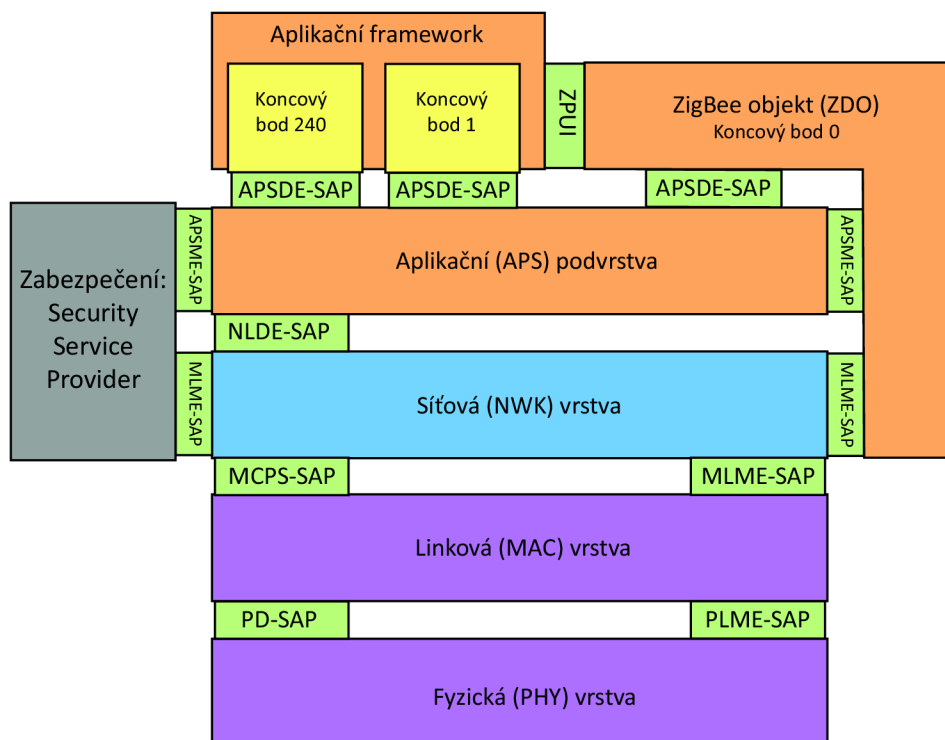
1. Synchronizační záhlaví – skládá se z preamble (v případě O-QPSK, Orthogonal Quadrature Phase Shift Keying modulace se jedná o 4 byty samých nul).
2. Oddělení – SFD Start Frame Delimiter, pevně daná sekvence bitů.
3. Délka užitečného zatížení (maximálně 7 bitů).

3.2 Struktura ZigBee

ZigBee se skládá ze 4 vrstev. Ze standardu IEEE 802.15.4 využívá vrstvu fyzickou (PHY) a linkovou (MAC) a na ty přidává vrstvu síťovou (zkráceně NWK – network) a aplikační (zkráceně APL – application). Mezi vrstvami jsou umístěny přístupové body, tzv. Service Access Point (SAP), které slouží jako abstrakce pro vrstvy nad a pod. Stejně jako v případě IEEE 802.15.4 i ZigBee využívá dvou SAP na vrstvu. Jeden pro správu a druhý pro data samotná.

3.2.1 Fyzická vrstva

Fyzická vrstva se stará o kódování a dekódování dat a jejich přenos fyzickým prostředím. Z pásem, pro které je standard IEEE 802.15.4 definován, ZigBee využívá ISM pásma 868 MHz s přenosovou rychlostí 20 kbit/s (pro Evropu), 915 MHz se 40 kbit/s (pro Austrálii a Ameriku) a celosvětově 2,4 GHz (až 250 kbit/s). Je využito přímého rozprostření spektra (Direct Sequence Spread Spectrum – DSSS), kde je takto rozprostřený signál modulován v sub-GHz pásmech pomocí BPSK (Binary Phase Shift Keying) a v pásmu 2,4 GHz pomocí O-QPSK.



Obr. 3.2: Architektura protokolu ZigBee

Vrstva dále obsahuje část nazývanou Physical Layer Management Entity – PLME, která svým SAP rozhraním umožňuje volat funkce pro správu. Také je zde udržována databáze atributů (PIB – PHY PAN Information Base) fyzické vrstvy.

3.2.2 Linková vrstva

Linková – MAC (Media Access Control) vrstva má několik funkcí. Mezi ty hlavní patří:

1. **Řízení přístupu k médiu** – v případě IEEE 802.12.4 (tedy i ZigBee) je použita metoda CSMA/CA (Carrier Sense Multiple Access/Colision Avoidance). Tato metoda spočívá v ověření, zdali je radiové prostředí volné a až poté začne zařízení vysílat. Pokud je prostředí obsazené, čeká náhodnou dobu (nazývanou Backoff-period), než se pokusí znovu vysílat.
2. **Generování beacon rámců** – v případě, že zařízení je koordinátor, má na starost informovat ostatní zařízení o síti, kterou řídí. To se provádí pomocí tzv. beacon rámců, které jsou zasílány na vyžádání (Beacon request) nebo periodicky (v beacon-enabled sítích).
3. **Synchronizace s beacon rámcí** – pokud se jedná o beacon-enabled síť,

zařízení se musejí přizpůsobit koordinátorovi a přistupovat k médiu v k tomu určených timeslotech, ohraničených beacon rámci.

4. **PAN asociace/disociace** – proces přiřazení zařízení do sítě.

Dle IEEE 802.15.4 rozlišujeme v rámci MAC vrstvy zařízení na:

- FFD (Fully Function Device): zařízení využívající MAC funkce v plném rozsahu.
- RFD (Reduced Function Device): zařízení, které mají pouze základní funkce (viz Tab. 3.1) a mohou komunikovat pouze s FFD.

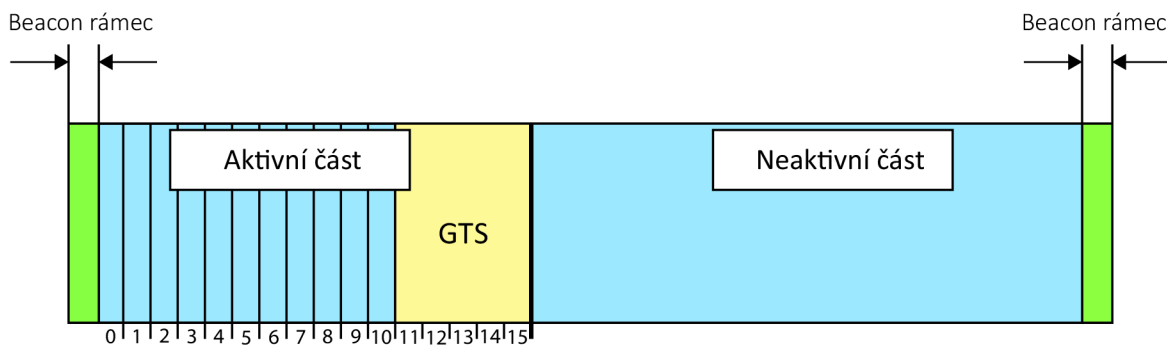
Příkaz	Vysílání	Příjem
Association request		X
Association response		X
Disassociation notification	X	X
Data request	X	
PAN ID conflict notification	X	
Beacon request		
Coordinator realignment		X
GTS request		

Tab. 3.1: Funkcionalita RFD

Beacon-enabled/nonbeacon-enabled

Z pohledu beacon rámců rozlišujeme síť:

- **Beacon-enabled** – beacon rámce jsou vysílány v pravidelných intervalech a komunikace je strukturalizována tzv. super-rámcem (není podporováno v případě mesh topologie). Super-rámec (Obr.3.3) je ohraničen beacon rámci a PAN koordinátor jej může libovolně rozdělit na aktivní a neaktivní část. Aktivní část je vždy rozdělena na 16 timeslotů, ve kterých mohou zařízení komunikovat pomocí CSMA/CA (zde nazýváno „slotted“ CSMA/CA). V rámci aktivní části může PAN koordinátor vyhradit prostor pro zařízení vyžadující přednostní komunikaci. Tyto časové úseky se nazývají Guaranteed Time Slot (GTS).



Obr. 3.3: Struktura super-rámce

- **Nonbeacon-enabled** – beacon rámce jsou vysílány jen při vyžádání (Beacon request) a zařízení v síti komunikují tehdy, kdy potřebují. V těchto sítích aktivní prvky sítě (směrovače) stále naslouchají a přístup k médiu je řešen pomocí standardní (zde „unslotted“) CSMA/CA.

MAC vrstva také obsahuje část pro management, nazývanou MAC subLayer Management Entity (MLME), která umožňuje volání funkcí a správu atributů udržovaných v tzv. MAC sublayer PIB.

3.2.3 Síťová vrstva

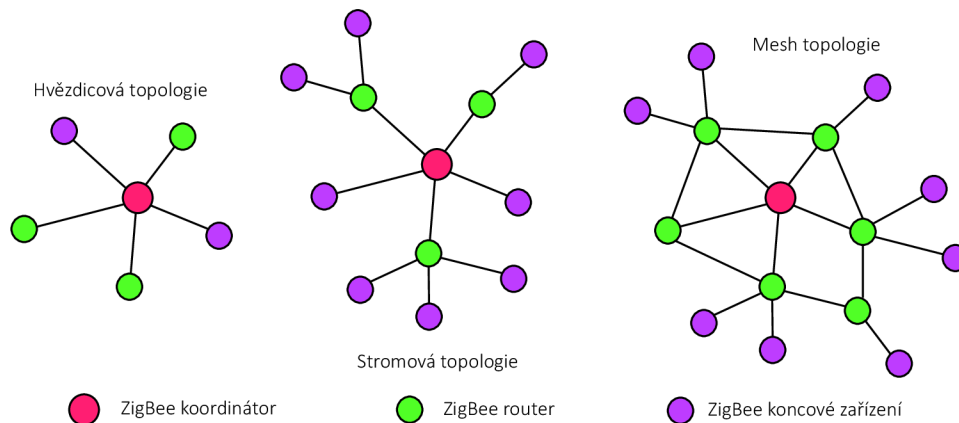
Síťová vrstva (NWK) je specifická pro standard ZigBee. Vrstva má několik funkcí, mezi ty hlavní patří:

- Zakládání sítě.
- Připojování a odpojování se z/do sítě.
- Adresaci a schopnost koordinátora přiřazovat adresy.
- Směrování a udržování směrovací tabulky.
- Odhalování sousedních zařízení.

V rámci ZigBee sítí rozlišujeme tři zařízení:

- ZigBee End Device (ZED) – koncové zařízení, může být FFD nebo RFD.
- ZigBee Router (ZR) – FFD se schopností směrování.
- ZigBee Coordinator (ZC) – ZigBee koordinátor (FFD) - spravuje celou síť.

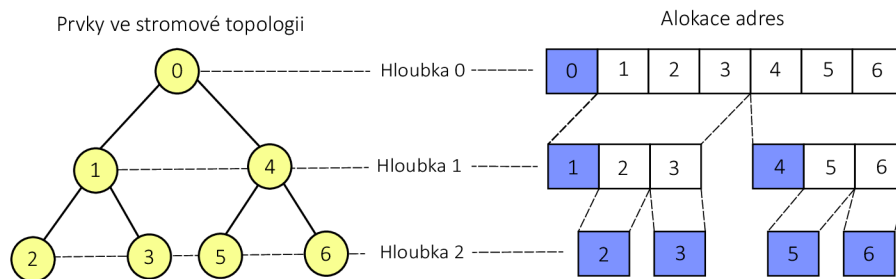
ZigBee podporuje kromě topologie mesh také topologie tvaru hvězdy nebo stromu (viz Obr.3.4). V každé síti musí být jeden ZigBee koordinátor, který se o síť stará.



Obr. 3.4: Podporované topologie

Směrování v ZigBee sítích probíhá v závislosti na topologii. V případě stromové topologie je využíván velmi prostý Tree Routing algoritmus. V případě meshové topologie je směrování prováděno na základě záznamů ve směrovací tabulce, popřípadě pomocí route discovery. Tyto metody budou dále vysvětleny.

Tree Routing

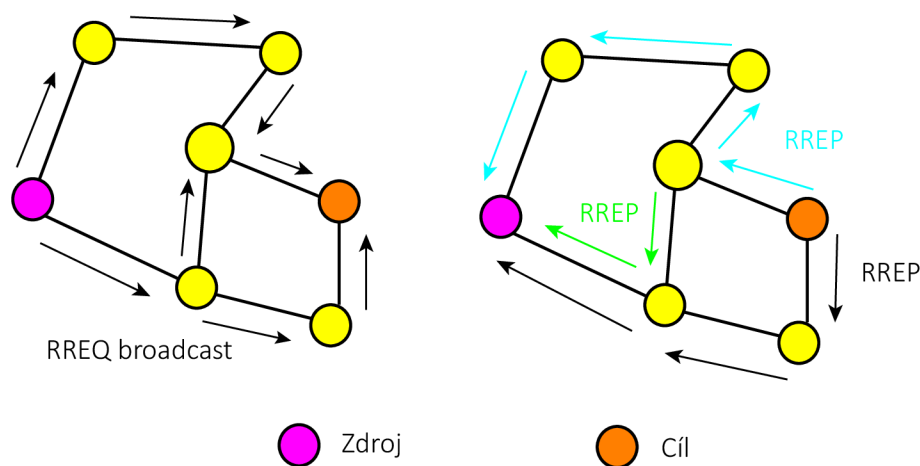


Obr. 3.5: Topologie a alokace adres pro Tree routing

Tree Routing algoritmus spočívá v přednastavení klíčových parametrů (počet směrovačů a jejich podřízených uzlů v síti a hloubku zanoření sítě) koordinátorem. Z těchto parametrů je poté vytvořena mapa sítě a síťové adresy jsou přiřazovány v závislosti na poloze ve stromě (viz Obr.3.5). V této pevné topologii poté uzel podle adresy ví, kde přesně ve stromě cílové zařízení hledat a paket směřuje buď směrem nahoru nebo dolů. Výhodou je nenáročnost směrování, tudíž možnost implementace i do zařízení s omezenou kapacitou paměti. Také pokud je vyžadováno mít síť v beacon-enabled stavu, jinak než použitím stromové nebo hvězdicové topologie to možné není (mesh topologie beacon-enabled síť nepodporuje). Velkou nevýhodou je neflexibilita sítě

(adresy jsou statické a síť nereaguje na změny v topologii) a fakt, že při výpadku koordinátora dojde k selhání celé sítě.

Směrování v mesh síti – route discovery



Obr. 3.6: Route discovery

Každé zařízení ve funkci ZR nebo ZC udržuje v paměti záznamy o možných destinacích (uchovává se jak 64bitová IEEE adresa tak 16bitová síťová adresa). Pouze v případě že cílová adresa není zařízení známá je použit AODV (Ad hoc On-Demand Distance Vector Routing) algoritmus, neboli route discovery. V prvním kroku zařízení rozešle všesměrový RREQ (Route Request) rámeček. Do tabulky uloží záznamy o všech zařízeních, na které rámeček vyslalo. Tyto rámce se dále šíří sítí (viz Obr.3.6). Každé další zařízení nejdříve porovná svoji adresu s adresou uvedenou v RREQ rámečku, pokud se neshoduje vyšle všesměrový RREQ dál a vytvoří stejným způsobem záznamy v tabulce spolu se záznamem o adrese zařízení od kterého RREQ obdrželo. Když cílové zařízení zachytí RREQ rámeček a adresy se budou shodovat, vyšle zpět odpověď – RREP (Route Reply). Tento rámeček se, díky záznamům vytvořeným dříve, šíří zpět (viz Obr.3.6), až dorazí do místa odkud původní RREQ vzešel. Po určité době vyprší platnost záznamů neúspěšných RREQ a ty jsou smazány.

Mesh sítě jsou silnou stránkou ZigBee a díky route discovery algoritmu je směrování v takovýchto sítích velmi spolehlivé. Mesh sítě tak narozdíl od stromových topologií dokáží rychle reagovat na změny a nemají žádný jeden bod selhání. I při poruše koordinátora routery pokračují v komunikaci a route discovery zaručuje, že nová cesta je vždy nalezena (pokud existuje).

3.2.4 Aplikační vrstva

Aplikační vrstva se v ZigBee skládá ze tří hlavních částí:

- **Aplikační subvrstva** (Application Support Sublayer - APS) je rozhraní mezi aplikací a sítovou vrstvou a poskytuje obecné funkce pro zacházení s daty.
- **Aplikační framework**, který je multiplexorem pro koncové body.
- **Koncové body** (endpoints), ve kterých jsou usazené aplikační objekty se svými aplikačními profily. Aplikační objekt s indexem 0 je nazýván ZigBee Device Object (ZDO) a implementuje ZigBee Device profil. Funkce obsažené v tomto profilu umožňují provádět obecné síťové operace (network address request/response, node descriptor request, network discovery request).

3.3 Princip komunikace

V této části bude přiblížena komunikace v ZigBee sítích. Budou popsány principy adresace a funkce aplikační vrstvy ZigBee protokolu rozšiřující funkcionalitu uzlů samotných.

3.3.1 ZigBee síť

Komunikace mezi jednotlivými zařízeními probíhá vždy v rámci jedné ZigBee sítě. Pro každou ZigBee síť jsou charakteristické její PAN ID a kanál, na kterém bude komunikace probíhat. Vytvořit síť je schopen pouze koordinátor (ZC). PAN ID je 16bitové číslo (pro ZigBee síť, nalezl od IEEE 802.15.4, je rozsah od 0x0000 do 0x3fff), které umožňuje koexistenci až 16 000 sítí [5]. Kromě 16bitového PAN ID je pro správné určení sítě využíváno také tzv. rozšířeného (extended) 64bitového PAN ID. Tento údaj je obsažen pouze v těle beacon rámce, aby mohlo zařízení určit přesněji do které sítě se připojí.

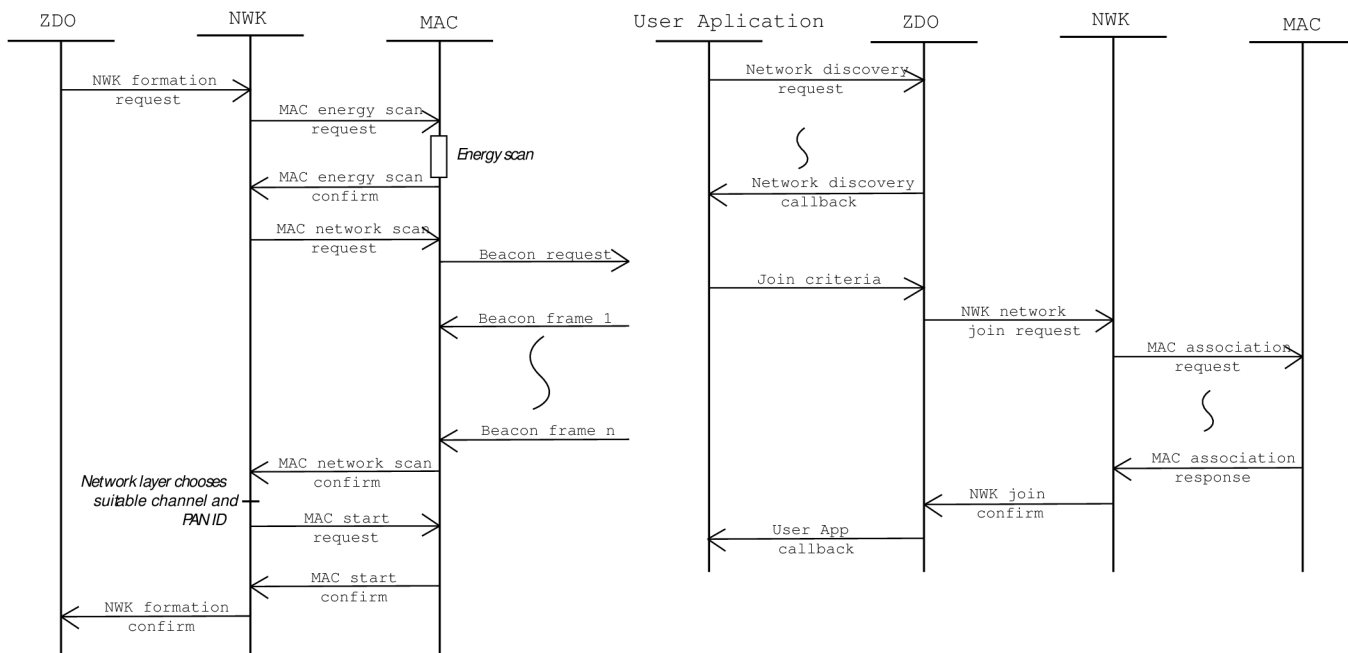
Pro správné fungování jsou důležité 2 úkony: formace sítě PAN koordinátorem a proces připojení zařízení do sítě.

Formace sítě

Formace sítě (viz Obr.3.7) započne zaslaním požadavku od ZDO síťové vrstvě na vytvoření sítě. Po provedení skenování okolí (pomocí beacon request rámce a příkazů MAC energy/network scan request) síťová vrstva zvolí neduplicitní PAN ID a vhodný kanál a příkazem MAC start request vytvoří síť. Veškeré důležité informace (informace o síti, PAN ID, verze používaného protokolu nebo počet zařízení, které se mohou připojit) jsou obsaženy v těle beacon rámce.

Formation Sequence Diagram

Join Sequence Diagram



Obr. 3.7: Schéma formace a připojení se do sítě [11]

Připojení do sítě

Stejně jako v případě vytváření sítě proběhne nejdříve sken okolí (MAC network scan request). Z přichozích beacon rámců jsou vyčteny důležité informace o dostupných sítích a ty jsou dále uloženy a prezentovány ZDO a aplikaci. Na základě výběru sítě je poté zaslán požadavek NWK join a MAC association request. Na straně příjemce je uložen záznam o sousedním uzlu, je mu přiřazena 16bitová síťová adresa a poté je zaslána odpověď interpretována v původním zařízení jako MAC association confirm a NWK join confirm (viz Obr.3.7). Po dokončení této procedury je zařízení připojeno k síti [11].

3.3.2 Adresace

Způsobů, jak adresovat koncový prvek je v ZigBee sítích několik. Adresovat lze jediný uzel (Unicast), všechny uzly v síti (Broadcast) nebo jen některé (Groupcast). Kromě adresace uzlu samotného lze rozlišovat až 240 koncových prvků uvnitř uzlu [11].

Unicast

Pokud chceme zaslat paket na jeden uzel v síti, musíme ho nějakým způsobem identifikovat. K tomu slouží síťová a MAC adresa. **Síťová adresa**, jinak nazývaná krátká

adresa nebo adresa uzlu (NwkAddr, short address, node address) je 16bitové číslo, které slouží k jednoznačné identifikaci uzlu v síti. Vyhrazená adresa 0x0000 patří vždy koordinátorovi sítě. Síťová adresa je přiřazena zařízení vždy, když se připojuje do sítě. Adresa je, kromě případu stromové topologie, kde je adresa přiřazována podle pořadí ve kterém se zařízení do sítě připojila, přiřazována náhodně koordinátorem. **MAC adresa**, jinak nazývaná dlouhá adresa (IEEE address, long address, extended address) je 64bitové číslo, které jednoznačně identifikuje každé zařízení na světě. 24 vyšších bitů je většinou tvořeno tzv. OUI (Organizational Unique Identifier) identifikátorem, který je přidělen výrobcí IEEE aliancí. MAC adresa by měla být uložena ve flash paměti každého zařízení a není nijak spjatá se síťovou adresou (ta se může s každým novým připojením do sítě změnit).

Broadcast

Jak již bylo řečeno, v ZigBee sítích je možné rámce zasílat jako unicast, broadcast nebo groupcast. V případě broadcastu lze dále ještě volit zdali chceme adresovat všechny uzly v síti (adresa 0xffff), všechny nespící uzly v síti (adresa 0xfffd) nebo pouze routery (adresa 0xfffc). Druhý důležitý parameter při využívání broadcastu je jeho radius, kterým se dá funkce broadcastu značně upravit (např. Broadcast vysílání s radiem 1 bude přijat pouze sousedními uzly).

Groupcast

Kromě komunikace unicast nebo broadcast podporují ZigBee sítě i multicast, nazývané groupcast. Je k tomu využívána skupinová adresa (group address), 16bitové číslo uložené v záhlaví APS paketu. Aplikační vrstva díky ní určuje zdali jsou příchozí rámce určeny pro konkrétní uzel a nežádoucí rámce zahazuje.

3.3.3 Adresace v rámci uzlu

Uzel lze lokalizovat v síti (definovanou PAN ID) pomocí síťové nebo rozšířené IEEE adresy, popřípadě podle skupinové adresy. ZigBee ale tyto síťové koncepty rozšiřuje a přidává funkcionalitu i do uzlů samotných. Aplikační vrstva přidává koncové body (Endpoints) a ty umožňuje efektivně využívat pomocí předdefinovaných profilů a clusterů. [11]

Aplikační profily

Aplikační profily slouží k zahrnutí podobných aplikací a zařízení do jednoho společného logického celku (domény). Každý aplikační profil má své ID (App Profile ID), 16bitové číslo s rozsahem 0x0000 – 0x7fff pro veřejné profily (definované ZigBee

aliancí) a rozsahem 0xbf00 – 0xffff pro soukromé profily (definované samotnými výrobci). V rámci daného profilu jsou vždy definovaná (pomocí clusterů) zařízení, která se zde očekávají. Například v profilu domácí automatizace jsou obsaženy zařízení jako: vypínač, dveřní zámek, rolety, termostat, apod. Implementace takto předdefinovaných zařízení značně usnadňují vývoj konkrétních systémů.

Číselné označení (App Profile ID)	Profil
0101	Monitorování průmyslových budov
0104	Domácí automatizace
0105	Automatizace komerčních budov
0107	Telekomunikační aplikace
0108	Domácnost a nemocniční péče

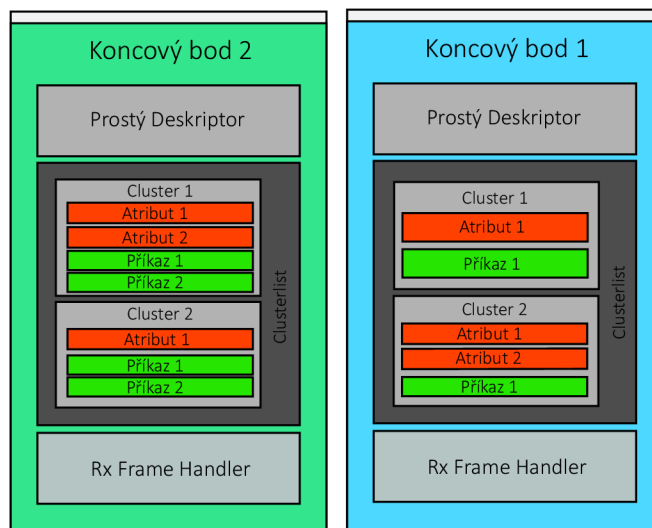
Tab. 3.2: příklady veřejných aplikačních profilů

Clustery

Clustery jsou obsaženy v knihovně ZCL (ZigBee Cluster Library) a pomáhají definovat význam jednotlivých koncových bodů. Obsahují příkazy a atributy, které slouží jako makra k očekávaným úkonům daného zařízení. Například *level control cluster* obsahuje atribut *current level* (momentální úroveň např. napětí) a příkaz *move to level* (nastavení určité úrovně) [11].

Koncové body

Jak již bylo řečeno v každém uzlu je k dispozici v rámci Aplikačního frameworku 240 koncových bodů (Endpoints). Koncové body umožňují koexistenci více aplikačních profilů nebo zařízení (např. různých senzorů) uvnitř jednoho síťového uzlu. Každý koncový bod lze adresovat jednotlivě pomocí 8bitového čísla (Destination endpoint). Analogicky k síťovým adresám je i zde vyhrazen jeden koncový bod pro účely broadcastu (0xff) a příchozí paket adresován pro tento koncový bod obdrží všechny koncové body v daném uzlu. Jak lze vidět na Obr.3.8, funkcionality koncového bodu je definovaná právě pomocí clusterů. Každý cluster má své atributy a příkazy. Všechny clustery využívané v daném koncovém bodu jsou shrnuty v seznamu (Device Cluster List). Veškeré definice koncového bodu jsou poté svázány pomocí tzv. prostého deskriptoru (simple descriptor), ve kterém je uveden název koncového bodu, aplikační profil, a veškeré využívané clustery [11].



Obr. 3.8: Schéma koncového bodu a v něm obsažených clusterů

4 PRAKTICKÁ REALIZACE

Cílem bakalářské práce bylo vytvořit Java aplikaci na vyčítání dat z bezdrátových ZigBee senzorů. V semestrální práci bylo dosaženo úspěšného propojení XBee modulu s PC i Raspberry Pi a nalezení potřebných knihoven a nástrojů.

V navazující části byla nejdříve odzkoušena komunikace 2 XBee modulů mezi sebou (koodrinátor – router). Pro umožnění implementace logiky do koncového bodu byla zvoleno použití Arduino desky, tudíž není k dispozici už jen pár pinů na modulu samotném a funkce protololu ZigBee, ale rozsah pinů celé desky a navíc možnost dopsat v jazyce C jakoukoliv funkcionalitu, kterou si uživatel bude přát. V rámci práce byl do desky nahrán program umožňující přečtení přijmaných zpráv a následné ovládání LED diody, vyčítání hodnot z potenciometru a vyčítání teploty. Na straně koordinátora poté byla vytvořena a odzkoušena aplikace samotná.

Aplikace byla inspirována programem X-CTU a umožňuje tedy správu lokálně připojeného zařízení i všech ostatních zařízení v síti. Možné je měnit parametry, pomocí skenu zobrazovat zařízení ve stejné síti a vyčítat i měnit jejich parametry. V poslední řadě je možné kontrolovat správné fungování odesíláním všesměrových rámců a zobrazováním odpovědí od zařízení v síti.

Celá aplikace je vytvořena ve formě textového grafického rozhraní, tím je zaručena možnost zobrazení i bez grafického výstupu v případě SSH spojení po síti. V případě nedostatečného výkonu hostitelského zařízení je k dispozici i čistě konzolová verze (spouštěná s parametrem *cli*). Třetí komponenta vytvořené aplikace je utilita na čtení samotných dat (spouštěná s parametrem *data*). Podle konfiguračního souboru program plánuje kdy, z jakého zařízení a jaká data budou vyčítána a ty poté

ukládá do souboru ve formátu CSV.

4.1 Vybrané řešení

Vzhledem k tomu, že ZigBee je otevřená specifikace, velké množství výrobců (OEM) vyvíjí své vlastní řešení. Mezi nejznámější výrobce samotných čipů (ZigBee silicon providers) patří: Atmel, Ember, Freescale, Integration Associates, Jennic, Microchip, NEC, Oki, Radio Pulse, Renesas, ST a Texas Instruments.

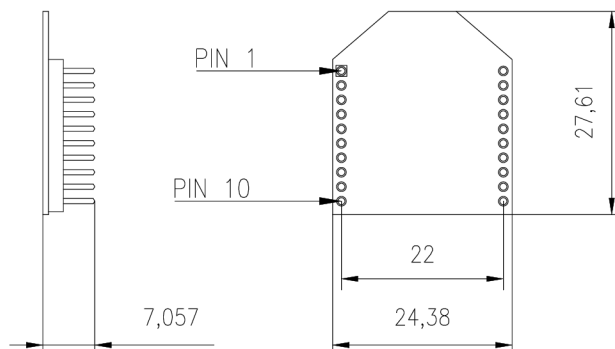
Pro práci byl zvolen výrobce konkrétních modulů Digi International s moduly XBee. V této části bude popsáno konkrétní ZigBee řešení zvolené pro semestrální práci. Přiblížen bude jak hardware, tak použité knihovny a softwarové nástroje.

4.1.1 Digi XBee

Digi International je společnost zabývající se již více jak 30 let technologiemi spojenými s M2M. Vyvíjejí celou řadu zařízení, zejména XBee/RF moduly, celulární řešení, vestavěné systémy seriálové servery a seriálové rozbočovače. Digi vyrábí RF moduly s podporou různých technologií. Jejich moduly umí využívat nejen 802.15.4 PHY a MAC samotné ale i stacky ZigBee, ZNet, DigiMesh nebo i WiFi.

XBee modul

Moduly vybrané pro bakalářskou práci jsou XBee S2C TH, konkrétně verze s konektorem U.FL. a starší S2B, model s vestavěnou anténou (viz Obr.4.1) Oba moduly pracují v pásmu 2,4 GHz a podporuje IEEE 802.15.4 samotné, DigiMesh i ZigBee. Oba moduly jsou, až na drobné rozdíly, totožné.



Obr. 4.1: Schéma modulu XBee S2C

Parametr	S2B	S2C
Dosah (městská oblast)	až 60 m	až 60 m
Dosah (volné prostranství)	až 300 m	až 300 m
Vysílací výkon	10 mW	6,3 mW (boost mód), 3,1 mW
Přijímací citlivost	-102 dBm	-102 dBm (boost mód), -100 dBm
Proudový odběr (vysílání)	170 mA	45 mA (boost mód), 33 mA
Proudový odběr (příjem)	45 mA	31 mA (boost mód), 28 mA

Tab. 4.1: Technické specifikace modulů XBee S2B a S2C

Transparentní a API mód

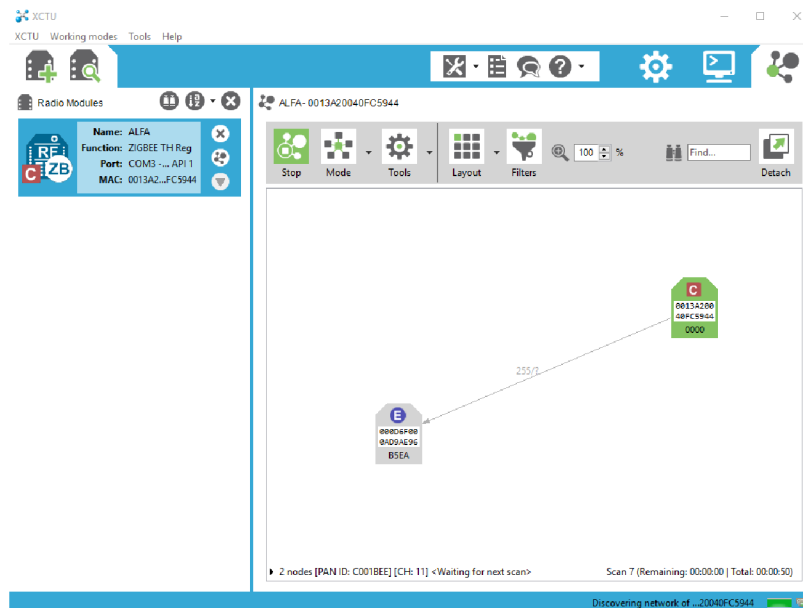
Každý XBee modul má možnost fungovat ve 2 módech. V transparentním a API (Application Programming Interface) módu. Tento mód je určen AT parametrem AP (0 pro transparentní, 1 pro API).

Jednodušší, transparentní, mód je určen jen pro základní aplikace. Zařízení pouze přeposílá vstup na jiné, momentálně připojené, zařízení. Pouze v případě vložení speciální sekvence znaků (+++) přejde do stavu, ve kterém je možné vyčítat a měnit parametry pomocí AT příkazů. Více ale transparentní mód nenabízí, navíc lze spojit vždy pouze 2 zařízení najednou.

Opravdový potenciál XBee modulů lze rozvinout pouze v API módu. Umožňuje komunikaci pomocí generovaných paketů, tvorbu mesh sítí a zároveň zařízení v tomto módu mohou být ovládána pomocí XBee Java Library.

X-CTU

Pro správu modulů vyvíjí společnost Digi aplikaci s názvem X-CTU. Prostřednictvím této aplikace je možné v přehledném grafickém prostředí spravovat a nastavovat jednotlivé moduly a mít přehled nad celou sítí. První modul je potřeba propojit s PC seriálovou linkou (USB) a pomocí tohoto zařízení je poté možné provádět skeny okolí a k nalezeným modulům se tak připojit a spravovat je vzdáleně.



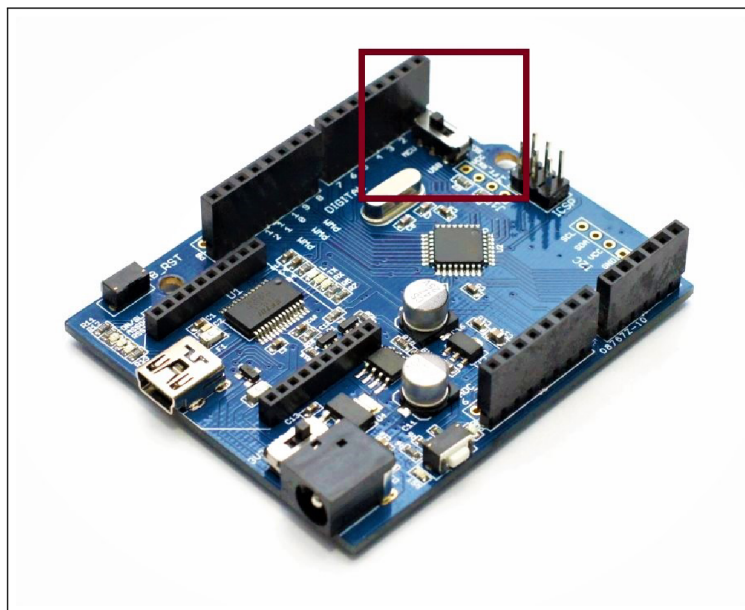
Obr. 4.2: Printscren aplikace X-CTU v síťovém módu s modelovým scénářem

X-CTU má 3 módy. V konfiguračním módu je možné zobrazit veškeré důležité parametry modulu a snadno je přepsat. Také je zde možné spravovat protokoly, které modul podporuje, a jejich verze. V konzolovém módu je možné otevřít komunikaci na seriové lince a se zařízením komunikovat přímo pomocí AT příkazů. Je-li v zařízení povolen API mód (parametr AP = 1), je možné se zařízením komunikovat pomocí vygenerovaných paketů (k tomu je v prostředí určen generátor paketů). Pokud je zařízení v API módu lze je využít síťový mód. V něm je možné zobrazit celou síť a mít tak přehled o všech síťových prvcích v topologii.

4.1.2 Crowduino deska

Pro rozšíření možností na straně koncového zařízení bylo zvoleno řešení od společnosti Elecrow – deska Crowduino v1.0, plně kompatibilní s Arduino. Deska má již vestavěn shield na XBee modul a taktéž s ním zaručuje plnou kompatibilitu.

S osazeným XBee modulem lze komunikovat pomocí vestavěného USB konektoru nebo použít sběrnici I2C. Pro správné fungování je třeba dbát na správnou pozici přepínače USB–MCU (viz Obr.4.3). V pozici USB je sériová komunikace přesměrována přímo do mikrokontroléru a je možné tedy nahrávat programy z prostředí Arduino (jako deska „Arduino Deumilanove or Diecimila“). V této pozici je ale obcházen XBee modul a pro jeho správnou funkci je nutné po nahrání programu přepnout do pozice MCU. Deska má 14 digitálních a 6 analogových output/input pinů.



Obr. 4.3: Deska Crowduino s naznačeným přepínačem USB–MCU

4.1.3 RaspBerry Pi

Díky multiplatformní povaze Javy je možné výslednou aplikaci spustit na kterémkoliv přístroji hostující JVM (Java Virtual Machine). Pro tento projekt bylo zvoleno zařízení Raspberry Pi. Na Zařízení byl nainstalován systém Raspbian 8.0. Zařízení bylo připojeno do sítě a přístup k němu byl uskutečněn pomocí SSH (Secure Shell) připojení. Kromě SSH je možné k zařízení přistupovat skrze vzdálenou plochu (u Raspberry Pi možné řešit pomocí aplikace VNC Viewer). Přenos souborů do zařízení byl uskutečněn pomocí protokolu SFTP (SSH File Transfer Protocol). Pro správné fungování java aplikací musí být na hostitelském zařízení přítomné prostředí JRE (Java Runtime Environment) nebo produkt JDK (Java Development Kit), obsahující JRE. Na systému Raspbian je JDK nainstalováno automaticky, jinak lze snadno doinstalovat pomocí příkazu `apt-get instal librxtx-java8-jdk` (pro verzi 1.8). Dále je také potřeba provést instalaci Java RxTx doplňující knihovny (viz níže), pomocí příkazu `apt-get instal librxtx-java`.

Po těchto krocích je možné spouštět java aplikace komunikující přes seriovou linku prostřednictvím JDK příkazu `java`. Aplikace byla napsána v IDE prostředí Eclipse a byla vyexportována jako spustitelný `.jar` soubor. Tento soubor byl přenesen pomocí SFTP do Raspberry Pi, a tam byl spuštěn příkazem `java` s parametrem `-jar`.

4.1.4 Java RxTx

Z důvodů umožnění portability výsledné aplikace byla zvolena pro její tvorbu Java platforma. Klíčovou součástí aplikace je možnost komunikovat s připojeným zařízením přímo prostřednictvím seriového spojení. Možností jak komunikovat přes seriovou linku pomocí Javy je několik. Pro semestrální práci byla zvolena open-source knihovna Java RxTx. Jedná se o volně dostupné řešení ¹, které je zároveň podporováno na velkém množství platform. Celá knihovna je k dispozici ve formátu *.jar* a stačí pouze připojit k projektu v IDE (Integrated Developer Environment) nebo manuálně zadat cestu k těmto knihovnám přímo při kompilaci. Na systémech Linux je možné knihovnu RxTx stáhnout prostřednictvím balíčkovacího systému a příkazu *apt-get instal librxtx-java*. V případě systému Windows či Linuxs pro architekturu x64 nebo x86 je navíc potřeba stáhnout doplňující knihovny *.dll* nebo *.so* ². Tyto doplňující knihovny je potřeba umístit do adresáře dle přiloženého návodu.

4.1.5 XBee Java Library

Vzhledem k volbě XBee modulu jako přímo připojeného zařízení bylo možné využít Java knihovny vytvořené přímo společností Digi – XBee Java Library. Součástí této knihovny je i Java RxTx, které je zde využito při seriové komunikaci. Při práci na aplikaci samotné tedy nebylo třeba využívat jen knihovny RxTx a na ní stavět, ale bylo možné využít celé XBee knihovny a práci na aplikaci si tak výrazně zjednodušit. Knihovny jsou k dispozici na stránkách Digi³. Staženou knihovnu s názvem *xbjlib-x.y.z.jar* (podle verze) opět stačí pouze připojit k projektu a je možné využívat v plném rozsahu.

4.2 Modelový scénář

V modelovém scénáři probíhá komunikace mezi dvěma zařízeními. Modul XBee S2C je koordinátorem sítě a modul S2B je koncové zařízení (funkce end device nebo router). Pomocí vytvořené aplikace je možné spravovat jak koordinátora (skrze USB), tak koncové zařízení vzdáleně. Je možné měnit parametry (viz Tab.4.2) obou těchto zařízení a vyčítat data měřená koncovým zařízením.

¹<http://rxtx.qbang.org>

²<http://fizzed.com/oss/rxtx-for-java>

³<http://docs.digi.com/display/XBJLIB/XBee+Java+Library>

Označení parametru	Význam
ID	PAN ID
ZS	Verze ZigBee stacku
CE	Coordinator enable – zařízení bude koordinátorem
SH	Horních 32 bitů MAC adresy
SL	Dolních 32 bitů MAC adresy
MY	16bitová síťová adresa
EE	Povolení šifrování
AP	API nebo transparentní mód
SM	Nastavení spánku
AI	Indikace poslední asociace – diagnostika

Tab. 4.2: Některé důležité parametry XBee modulu

4.3 Výstup práce

Tato část se věnuje praktickému výstupu bakalářské práce. Jsou zde popsány nezbytné kroky k uvedení použitých zařízení do správného stavu, je vysvětleno jak bylo vyřešeno čtení zpráv v Arduino programu a popsána je i funkcionality aplikace samotné.

4.3.1 Příprava

Aby aplikace správně fungovala a komunikace probíhala tak, jak je žádoucí, byl použitý hardware uveden do následujícího stavu:

XBee moduly

Použité moduly byly připojeny k osobnímu počítači a byly do nich nahrány nejnovější verze firmwaru. Pro koordinátor (modul S2C) se jedná o firmware ZIGBEE TH Reg verze 405F. Komunikace s koncovým zařízením (modul S2B) bylo otestováno s funkčními sety ZigBee Router API a End Device API (obě verze 23A7).

Po sjednocení PAN ID koordinátor sítě ihned koncové zařízení zaregistruje a je přidáno do sítě. Po nastavení těchto parametrů je možné zařízení zapojit dle modelového scénáře a komunikace skrze aplikaci probíhá bez problému.

Raspberry Pi

Raspberry Pi bylo zvoleno jako testovací zařízení reprezentující skupinu embedded zařízení a bylo tudíž dokázáno, že aplikaci lze spouštět nezávisle na architektuře

procesoru. Jediná podmínka je, že zařízení musí být kompatibilní s JRE (alespoň verze 6, z důvodů využití knihovny *Java.util.concurrent*) a zároveň musí být kompatibilní s použitou knihovnou pro přístup k seriovému portu – Java RxTx.

Pro testování byly k dispozici byly dvě zařízení. Rapsberry Pi model 1 a výkonnější model 3. Na těchto zařízeních byl spouštěn systém Raspbian 8.0 s nainstalovaným JDK 1.8 a balíčkem *java rxtx* (viz část 4.1.3).

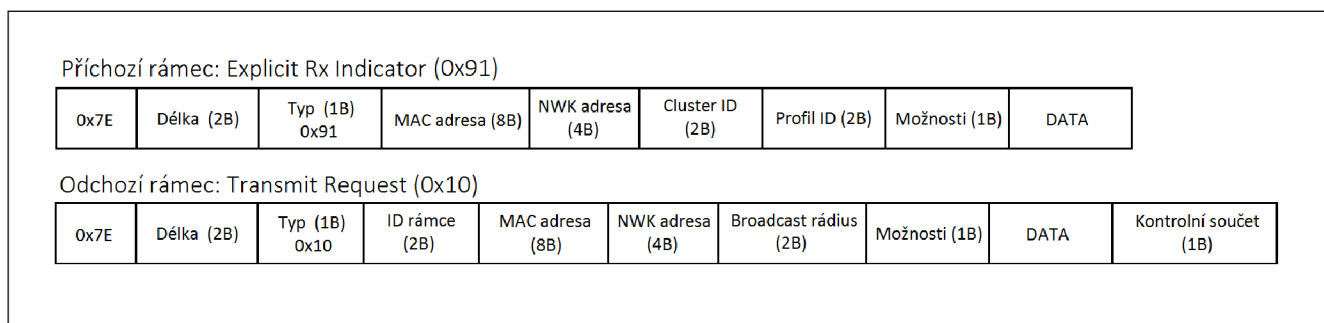
Crowduino

Deska Crowduino byla pouze osazena modulem XBee a v prostředí Arduino IDE byl nahrán vypracovaný program. Pro zaručení správné komunikace s PC je nutné nastavit typ desky jako *arduino duemilanove or diecimila*.

4.3.2 Arduino program

Aby bylo možné správně přečíst informaci v příchozím paketu/rámci bylo nutné napsat jednoduchý analyzátor/parser. Tento analyzátor rozkládá příchozí paket a extrahuje z něj potřebná data.

Funkce tohoto parseru je založená na známe struktuře rámce. Pro jednoduchost jsou zpracovávány pouze rámce nesoucí data (rámce typu *Transmit request* a *Explicit Rx Indicator*, viz Obr.4.4)



Obr. 4.4: Struktura použitých rámců

Crowduino deska s XBee modulem komunikuje skrze seriovou linku, stačí tedy v programu naslouchat a data z něj přímo číst nebo do něj zapisovat. V hlavní smyčce programu je podmínka, která čeká na delimiter rámce (vždy 0x7E). Když jej zaznamená jsou přečteny další 2 byty (celková délka rámce). Jelikož se další data na seriovém rozhraní neobjeví ihned, je potřeba program na chvíli pozastavit. Pozorováním bylo zjištěno že doba, kterou rozhraní potřebuje na 1 B je zhruba 1 ms. Je proto použita hodnota délky ze záhlaví rámce a po uběhnutí doby (která je pro každý rámec jinak dlouhá) jsou dočteny zbývající data. Pro vykonání úkonu jsou

uloženy data z těla rámce a pro následnou odpověď adresy (jak MAC, tak i síťová). Zbylé informace nejsou podstatné, mohou tedy být zahozeny.

Po provedení úkonu (zablikání, přečtení hodnoty z potenciometru, teploměru, ...) je na dříve uloženou adresu odeslána odpověď. Opět stačí pouze zapisovat byty dle známe struktury (Obr. 4.4). Na rozdíl od pouhého čtení, v případě příchozího rámce, je při odesílání potřeba na konec spočítat kontrolní součet (checksum). Při zapsání nesprávné hodnoty modul nepřijme řetězec jako rámeček a nedojde tedy k jeho odeslání.

4.3.3 Java aplikace

Jak již bylo zmíněno aplikaci lze spouštět ve více režimech. Bez udání parametru je aplikace spuštěna ve výchozím grafickém módu, parametr *cli* spustí čistě konzolovou verzi aplikace a parametr *data* spustí režim pro čtení samotných dat. Čistě konzolová verze je přítomna pouze pro případ, že nedostatečný výkon hostitelského zařízení znemožní použití grafické verze, funkcionality je ale stejná. Tato verze dále přiblížena tedy nebude.

Grafické rozhraní

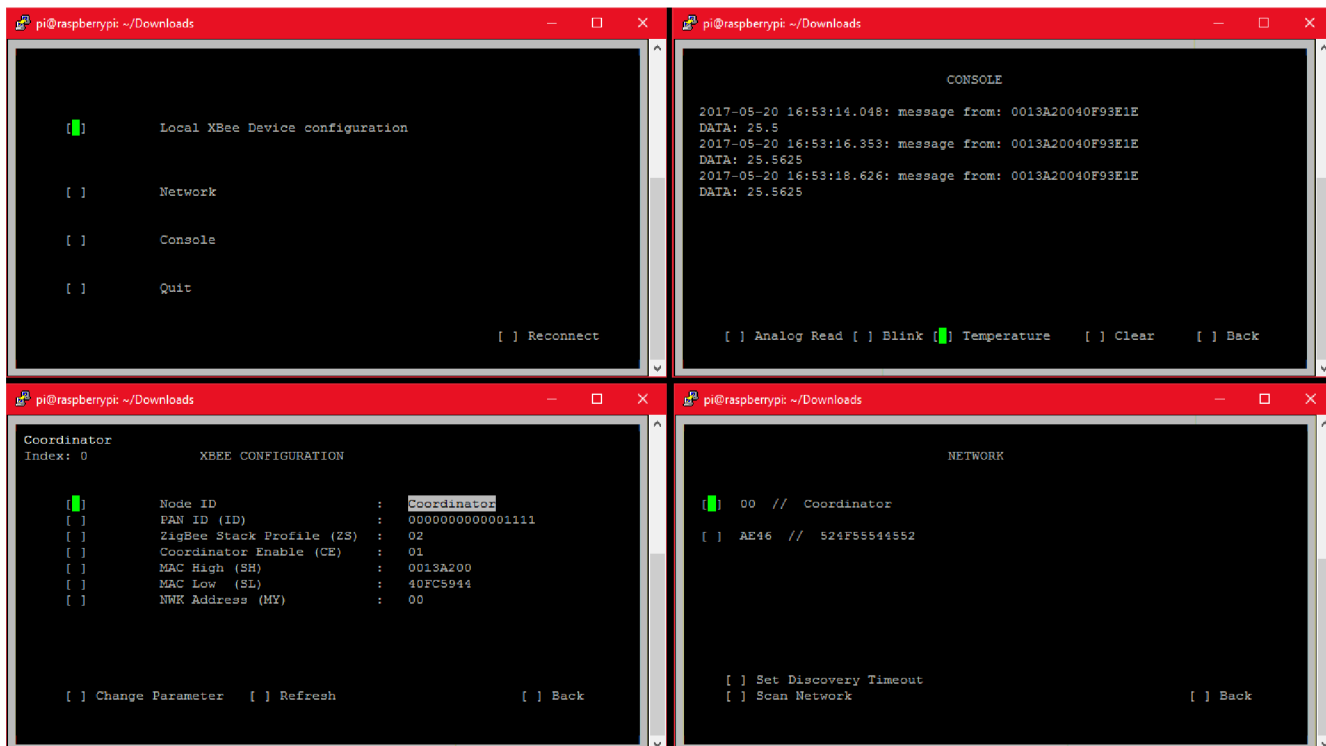
Pro větší přehlednost a intuitivnost aplikace je jednoznačně lepší použít určitou formu grafického rozhraní. Sofistikované grafické frameworky jsou ale většinou vázány na platformu a jejich použití v nevýkonných embedded zařízeních většinou není možné. Z těchto důvodů bylo pro aplikaci zvoleno grafické rozhraní na textové bázi. Je zachována jednoduchost ovládání díky možnosti posouvat kurzor, výpočetní náročnost je velmi malá a zároveň lze takovéto rozhraní zobrazovat čistě v terminálu, k využívání tedy není zapotřebí monitor či jiné zobrazovací zařízení.

TUI (Text User Interface) je dříve velmi využívanou formou rozhraní. Většina knihoven či nástrojů pro jejich tvorbu jsou ale v jazyce C a jsou velmi úzce přizpůsobeny na určitou platformu. V případě Javy jsou možnosti (pokud nechceme použít nativní knihovny a převádět pomocí JNI – Java Native Interface) značně omezené. Jedna z variant (využita k tvorbě aplikace) je opensource knihovna s názvem Lanterna ⁴.

Tato knihovna umožňuje tvorbu rozhraní čistě textových (umožňuje libovolné vyreslování znaků do konzole a ovládání pozice kurzoru) i rozhraní semi-grafických s okny (ve stylu BIOS nastavení). Aby zbytečně nebylo plýtváno výkonem byla zvolena „úspornější“ první varianta.

⁴<https://github.com/mabe02/lanterna>

Funkcionalita aplikace byla částečně inspirována výše popsaným programem X-CTU od společnosti Digi. Jedná se tedy spíše o konfigurační utilitu. V hlavním menu (viz Obr. 4.5) je možné vstoupit do části konfigurace přímo připojeného modulu, části shrnutí sítě a nebo do konzole, ve které lze odesílat pakety zařízením v síti a pozorovat odezvy. V síťové části lze na libovolné zařízení kliknout a vstoupit do konfigurace daného zařízení stejně jako jako je tomu u lokálně připojeného zařízení. Parametry lze odtud vzdáleně také měnit.



Obr. 4.5: Printscreens vytvořené aplikace

Režim čtení dat

Tento režim je přístupný v případě spuštění aplikace s parametrem *data*. Není zde využito knihovny Lanterna, uživatel je pouze v konzoli dotázán na cestu k zařízení, cestu kam ukládat vygenerovaný soubor s posbíranými daty a cestu ke konfiguračnímu souboru. Poté už aplikace běží sama až do chvíle kdy uživatel zadá příkaz *exit*. Tím je smyčka ukončena.

Způsob, jakým jsou data ze zařízení v síti vyčítána, lze nastavit pomocí konfiguračního souboru. Takový soubor musí být textový s n-řádky. Každý řádek musí mít následující parametry oddělené dvojtečkou: doba s jakou má být akce prováděna (v sekundách), příkaz pro koncové zařízení (hexadecimálně) a seznam MAC adres na které má být paket zaslán (nebo klíčové slovo *broadcast*). Příklad viz Obr. 4.6.

```
10:34:0013A20040F93E1E ,0013A20040AC15E4 ,0013A212B0213C1E
60:31:broadcast
120:33:0013A20040F93E1E
```

Obr. 4.6: Příklad konfiguračního souboru

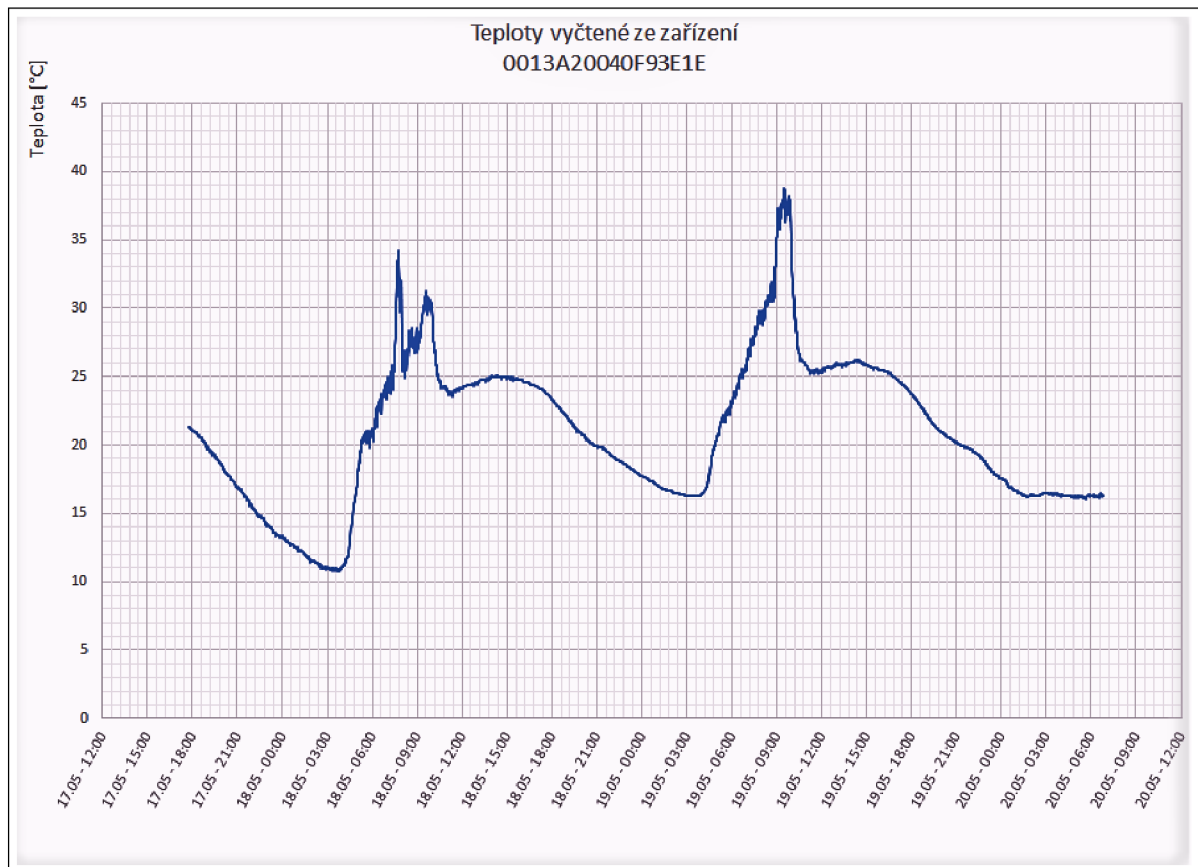
Při běhu programu je poté tento soubor parsován a pro každý řádek je vytvořeno samostatné vlákno. Spouštění těchto vláken je poté plánováno instancí *ScheduledExecutorService* z knihovny *concurrent*. V případě zadání *default* místo cesty ke konfiguračnímu souboru, je program spuštěn s předdefinovaným nastavením: všesměrové vysílání každou minutu.

V případě, že zařízení vyslalo dotaz na data a dostane odpověď, odpověď je zapisována do souboru. Soubor je ve formátu CSV – Comma Separated Values, přesně tedy datum a přesný čas vyčtení hodnoty, MAC a síťová adresa zařízení a hodnota samotná (buď jako číslo s plovoucí desetinnou čárkou nebo v hexadecimálním tvaru). Tyto hodnoty jsou oddělené čárkou.

4.4 Příklad praktického využití aplikace

Jako příklad možnosti použití aplikace bylo zvoleno prosté vyčítání venkovní teploty. Hodnota byla vyčítána každou minutu po dobu přibližně dvou dnů. Po sesbírání dat bylo cca 3200 hodnot vloženo do tabulkového editoru a byl vykreslen graf závislosti teploty na čase (viz Obr. 4.7).

Zařízení bylo připojeno kabelem USB k Raspberry Pi a program byl spuštěn vzdáleně pomocí SSH klienta. Vzhledem k tomu, že program je spouštěný v systému na bázi Linux a nejedná se o službu běžící na pozadí, aby bylo možné uzavřít SSH spojení a neukončit tím běžící proces, bylo nutné odpojit proces od terminálu. Způsobů jak tohoto docílit je několik. V tomto případě byl použit program *screen*. Tento program umožňuje v rámci jedné relace spustit více instancí terminálu a libovolně mezi nimi přepínat. Jedna z jeho funkcionalit, pro tento případ velmi užitečná, je možnost program po stisku **Ctrl-A + D** „minimalizovat“ a terminál tím od něj odpojit. Je možné tedy SSH relaci ukončit a aplikace spolu se čtením dat běží dále. Po libovolné době je možné se k „minimalizovanému“ *screen* programu vrátit jeho spuštěním s parametrem *-x*.



Obr. 4.7: Graf naměřených teplot

Tento příklad je pouze ilustrační. Vzhledem k téměř neomezeným možnostem Arduino desky je možné naimplementovat do programu jakoukoliv funkcionalitu. Pro zaručení správné komunikace stačí poté pouze připsat další případ do výrazu switch-case reagující na požadovanou hodnotu příkazu v příchozím paketu. Díky vypracovanému parseru tento bude příkaz přečten a požadovaná činnost bude provedena.

5 ZÁVĚR

V teoretické části práce byly prostudovány principy ZigBee sítí. Byla popsána struktura protokolu napříč vrstvami a struktura rámců na jednotlivých vrstvách. Dále byla popsána základní konstrukce aplikační části protokolu a způsoby adresace a komunikace zařízení v síti.

V praktické části bylo nejprve dosaženo propojení s osobním počítačem přes sériovou linku. Hlavní část práce poté spočívala ve vytvoření Java aplikace. Toho bylo docíleno s využitím knihoven XBee Java Library, Java RxTx a Lanterna. Tato aplikace umožňuje skrze grafické rozhraní zobrazovat připojené zařízení a konfigurovat jejich parametry. Pokud je aplikace spuštěna s parametrem *data* spustí se utilita na plánovanou komunikaci se zařízeními. Ta lze využít například k vyčítání dat. Spolu s Java aplikací byl napsán program v jazyce C, který umožňuje použití Arduino desky spolu s modulem XBee.

V praxi může být aplikace velmi užitečná. Uživatel může nastavit senzorovou síť a koordinátora sítě připojit pouze na malé nenáročné zařízení (např. Raspberry Pi), které bude sloužit jako brána. Díky vytvoření aplikaci bude možné i na tomto jinak nepodporovaném zařízení vzdáleně monitorovat stav zařízení v síti a za běhu měnit některé vlastnosti sítě. To vše bez nutnosti fyzicky přijít a na jiném podporovaném zařízení (PC) konfiguraci provést.

Jako další rozšíření práce by bylo možné doimplementovat několik drobných funkcionalit. V první řadě v konfigurační části aplikace by bylo dobré ze statického seznamu udělat posuvný a umístit na něj všechny dostupné parametry modulu. To stejné v seznamu síťových zařízení, v případě rozsáhlejších sítí. V části konzolové dále pak, podobně jako v X-CTU, implementovat generátor paketů, aby bylo možné odesílat libovolné ZigBee rámce. Také by byla potřeba aplikaci řádně otestovat, odstranit chyby a ověřit kompatibilitu s širším spektrem zařízení, jak hostitelského tak bezdrátových modulů v síti (i jiných výrobců ZigBee modulů než XBee), které v době psaní bakalářské práce bohužel nebyla k dispozici.

LITERATURA

- [1] BOSWARTHICK, David, Omar ELLOUMI a Olivier HERSENT *M2M Communications: A Systems Approach* [online]. 8th. United States: John Wiley & Sons, 2012 [cit. 2016-12-11]. ISBN 9781119994756.
- [2] EVANS, Dave *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything* [online]. [cit. 2016-12-14]. Dostupné z: [online]. [cit. 2016-12-14]. Dostupné z: http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
- [3] Libelium: *50 Sensor Applications for a Smarter World. Libelium* [online]. [cit. 2017-05-31]. Dostupné z: http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/
- [4] Wikipedia: *Theodore Paraskevakos* [online]. [cit. 2017-05-31]. Dostupné z: https://en.wikipedia.org/wiki/Theodore_Paraskevakos
- [5] GISLASON, Drew. *Zigbee Wireless Networking* Newnes.Newnes, 2008. 448 s. ISBN 978-0750685979.
- [6] GOMEZ, Carles, Joaquim OLLER a Josep PARADELLS. *Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology* [online]. 2012 [cit. 2016-11-06]. DOI: 10.3390/s120911734. Dostupné z: <http://www.mdpi.com/1424-8220/12/9/11734/html>
- [7] GOMEZ, Carles a Josep PARADELLS. *Wireless home automation networks: a survey of architectures and technologies* IEEE Communications Magazine [online]. 2010, 48(6), 92-101 [cit. 2016-11-06]. DOI: 10.1109/MCOM.2010.5473869. Dostupné z: <http://dl.acm.org/citation.cfm?id=1824639>
- [8] LinkLabs: *What Is LoRaWAN?* [online]. [cit. 2016-12-14]. Dostupné z: <http://www.link-labs.com/what-is-lorawan/>
- [9] LinkLabs: *What Is SigFox?* [online]. [cit. 2016-12-14]. Dostupné z: <http://www.link-labs.com/what-is-sigfox/>
- [10] Thread Group: *Oficiální webová stránka Thread Group* [online]. [cit. 2016-11-06]. Dostupné z: <http://threadgroup.org/>
- [11] ZigBee Aliance *ZigBee Specification Overview* [online]. [cit. 2011-12-03] Dostupné z URL: <http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeeopro/>

- [12] Z-Wave: *Oficiální webová stránka Z-Wave aliance* [online]. [cit. 2016-11-06]. Dostupné z: <http://www.z-wave.com/>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

WPAN	Wireless Personal Area Network
IoT	Internet of Things
ISO/OSI	International Organization for Standardization/Open Systems Interconnection
M2M	Machine to Machine
H2H	Human to Human
IEEE	Institute of Electrical and Electronics Engineers
OEM	Original Equipment Manufacturer
ISM	industrial, scientific and medical
LPWAN	Low-Power Wide-Area Network
BLE	Bluetooth Low Energy
UDP	User Datagram Protocol
NPDU	Network Protocol Data Unit
PAN	Personal Area Network
FCS	Frame Check Sequence
SFD	Start Frame Delimiter
CSS	Chirp Spread Spectrum
MAC	Media Access Control
BPSK	Binary Phase Shift Keying
O-QPSK	Orthogonal Quadrature Phase Shift Keying
DSSS	Direct Sequence Spread Spectrum
PLME	Physical Layer Management Entity
PIB	PHY PAN Information Base
SAP	Service Access Point

FFD	Fully Function Device
RFD	Reduced Function Device
CSMA/CA	Carrier Sense Multiple Access/Colision Avoidance
GTS	Guaranteed Time Slot
MLME	MAC subLayer Management Entity
APL	Application Layer
ZED	ZigBee End Device
ZR	ZigBee Router
ZC	ZigBee Coordinator
NLME	Network Layer Management Entity
AODV	Ad hoc On-Demand Distance Vector Routing
APS	Aplication Support Sublayer
ZDO	ZigBee Device Object
ZCL	ZigBee Cluster Library
API	Application Programming Interface
IDE	Integrated Developer Environment
JVM	Java Virtual Machine
SSH	Secure Shell
JRE	Java Runtime Environment
JDK	Java Development Kit
SFTP	SSH File Transfer Protocol
TUI	Text User Interface
JNI	Java Native Interface
CSV	Comma Separated Values

SEZNAM PŘÍLOH

A Obsah přiloženého CD

46

A OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje elektronickou verzi bakalářské práce, zdrojové kódy Java aplikace i Arduino aplikace a vyexportovaný spustitelný *.jar* soubor.

Soubor „/ArduinoAplikace/ArduinoAplikace.ino“ obsahuje arduino program ve kterém je možné implementovat libovolnou funkcionalitu.

Aplikace byla otestována s Java SE 8 na PC x64 s operačními systémy Microsoft Windows 10 a Ubuntu 16.04 a na Raspberry Pi 1 a 3 s operačním systémem Raspbian 8.0. Aplikace byla vždy spouštěna vzdáleně pomocí SSH klienta Putty.