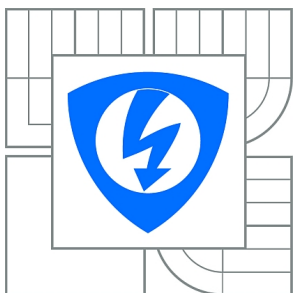


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# **BEZDRÁTOVÝ SENZOR PRO INTELIGENTNÍ BUDOVY**

WIRELESS SENSOR FOR INTELLIGENT BUILDINGS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**TOMÁŠ GREPL**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ONDŘEJ HYNČICA**

BRNO 2011



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
**Automatizační a měřicí technika**

**Student:** Tomáš Grepl

**ID:** 119413

**Ročník:** 3

**Akademický rok:** 2010/2011

**NÁZEV TÉMATU:**

## Bezdrátový senzor pro inteligentní budovy

**POKYNY PRO VYPRACOVÁNÍ:**

Seznamte se s protokolem ZigBee. Navrhněte algoritmus aplikace pro bezdrátový snímač pro sledování prostředí budov. Porovnejte různé způsoby přenosu dat ze snímače a vyhodnoťte energetickou náročnost jednotlivých řešení. V rámci práce implementujte navržený algoritmus a experimentálně ověřte jeho vlastnosti.

**DOPORUČENÁ LITERATURA:**

- [1] CROSS, Pete. Zeroing on ZigBee. Circuit Cellar. [online]. 2005, iss. 175. Dostupné z <<http://www.circuitcellar.com/library/print/0205/Cross175/2.htm>>. ISSN 1528-0608.
- [2] STALLINGS William. Wireless Communications & Networks. 2.vyd. Upper Saddle River (NJ): Pearson Prentice Hall, 2005. ISBN 0-13-191835-4.
- [3] TUTWSN - Wireless Sensor Network. [online] URL: <[http://www.tkt.cs.tut.fi/research/daci/ra\\_tutwsn\\_overview.html](http://www.tkt.cs.tut.fi/research/daci/ra_tutwsn_overview.html)> [cit. 2009-09-15]

**Termín zadání:** 7.2.2011

**Termín odevzdání:** 30.5.2011

**Vedoucí práce:** Ing. Ondřej Hynčica

**prof. Ing. Pavel Jura, 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 porovnáním energetické náročnosti dvou měřících schémat měřící teplotu pro bezdrátové měření v budově. Měření probíhá pomocí bezdrátových modulů ZigBit900 založených na standardu ZigBee. V prvním celku se práce věnuje teoretickému popisu komunikačního standardu. Rozebírá základní problematiku technologie a uvádí tak čtenáře do problematiky. Druhá část popisuje vývojový kit ZigBit 900. Další část uvádí do základní problematiky programovacího balíčku BitCloud, který je určen pro programování tohoto vývojového kitu. Dále je návrh dvou měřících schémat teploty, výběr vhodné aplikace, části změněného kódu a popis měření teploty. v Poslední kapitole jsou zobrazena a vyhodnocena naměřená teplota, vypočítána teoretická hodnota spotřeby pro kmitočtové pásmo 868 MHz a 2,4 GHz. V závěru je diskuze dosažených výsledků.

## KLÍČOVÁ SLOVA

ZigBee, měření teploty, ZigBit900, výpočet spotřeby, snímač v inteligentní budově

## ABSTRACT

The theme of the Bachelor's work is based on the comparison of two measuring schemes measured the temperature for the wireless measurement of the temperature in the building and the comparison of the power consumption of these schemes. Measurement is performed by using wireless modules ZigBit900 which are based on ZigBee standard. The first part of the work deals with the theoretical description of the communication standard. There is analyzed the basic issue of technology and the reader can be indicated to the issue. In the second part there is mentioned the description of the developmental kit ZigBit 900. Another section brings the basic information about the programming package BitCloud, which is used for programming of the developmental kit. In addition, designs of two measuring schemes of temperature are mentioned there. Then the reader find there the selection of appropriate application and the part of emended code and the description of temperature measurement. The last chapter schows the measuring and assessing of the temperature. The calculation of the theoretical value of power consumption for the frequency band 868 MHz and 2,4 GHz is the lastest researched problem in the work. In conclusion of this Bachelor's work is included the discussion of achieved results.

## KEYWORDS

ZigBee, temperature measure, ZigBit900, calculate consumption, sensor for building intelligent

GREPL, Tomáš *Bezdrátový senzor pro inteligentní budovy*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřící techniky, 2011. 44 s. Vedoucí práce byl Ing. Ondřej Hynčiča

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Bezdrátový senzor pro inteligentní budovy“ jsem vypracoval 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 uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno .....

.....

(podpis autora)

Chtěl bych poděkovat všem přátelům, kteří na mě mysleli a podporovali mě při tvorbě práce. Velký dík patří Báře Oravčíkové, za každé dobré slovo, povzbuzení a obrovskou oběť. Také mým rodičům, kteří mi věřili a cítil z jejich strany silnou podporu a důvěru.

Další komu chci poděkovat je Dávid Muzika za pomoc a rady při tvorbě práce. A největší dík chci směřovat mému vedoucímu práce Ing. Ondřeji Hynčicovi za veškerou pomoc, velmi cenné rady, nadhled nad problematikou a velkou dávku trpělivosti.

# OBSAH

<b>Úvod</b>	<b>10</b>
<b>1 ZigBee</b>	<b>11</b>
1.1 Standard ZigBee . . . . .	12
1.1.1 Frekvenční pásma . . . . .	12
1.1.2 Síťová zařízení . . . . .	12
1.1.3 Topologie sítě . . . . .	13
1.1.4 ZigBee PRO . . . . .	14
1.2 Model ZigBee . . . . .	14
<b>2 MeshNetics ZigBit 900 kit</b>	<b>16</b>
2.1 Základní charakteristika kitu . . . . .	16
2.2 Parametry ZigBit 900 . . . . .	16
<b>3 BitCloud ZigBee PRO 1.11.0</b>	<b>18</b>
3.1 Výběr programu . . . . .	18
3.2 Popis dema Lowpower . . . . .	18
3.2.1 Adresace v síti . . . . .	18
3.2.2 Funkce aplikace . . . . .	18
3.2.3 Nastavení Lowpoweru pro měření teploty. . . . .	20
3.3 Nastavení sítě pro ZigBee . . . . .	20
3.3.1 Makefile . . . . .	20
3.3.2 configuration.h . . . . .	20
3.3.3 Nastavení sítě pro měřicí aplikaci . . . . .	20
<b>4 Řešení studentské práce</b>	<b>22</b>
4.1 Periodické sledování teploty . . . . .	22
4.1.1 Popis řešení . . . . .	22
4.2 Triggerované sledování teploty . . . . .	22
4.2.1 Popis řešení . . . . .	22
4.3 Vývojové diagramy řešení . . . . .	22
4.4 Implementace . . . . .	23
4.4.1 Universal Synchronous Receiver and Transmitter . . . . .	23
4.5 Měření teploty - měřicí schéma . . . . .	24
4.5.1 Zpracování hodnot . . . . .	24

<b>5</b>	<b>Měření a výpočty</b>	<b>26</b>
5.1	Měření . . . . .	26
5.2	Výpočet spotřeby . . . . .	27
5.2.1	Frekvenční pásmo 868 MHz . . . . .	27
5.2.2	Frekvenční pásmo 2,4 GHz . . . . .	27
5.3	Měření delšího teplotního děje - Varianta A . . . . .	28
5.3.1	Odchylka periodického a triggerovaného průběhu od referenčního teplotního průběhu . . . . .	29
5.3.2	Spotřeba . . . . .	29
5.4	Měření kratšího teplotního děje - Varianta B . . . . .	30
5.4.1	Odchylka periodického a triggerovaného průběhu od referenčního průběhu teploty . . . . .	31
5.4.2	Spotřeba . . . . .	32
5.5	Vyhodnocení . . . . .	32
<b>6</b>	<b>Závěr</b>	<b>34</b>
	<b>Literatura</b>	<b>36</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>38</b>
	<b>Seznam příloh</b>	<b>39</b>
<b>A</b>	<b>Vývojové diagramy měřících metod</b>	<b>40</b>
A.1	Periodické měření . . . . .	40
A.2	Trigger . . . . .	40
<b>B</b>	<b>Ukázky kódu</b>	<b>41</b>
B.1	Funkce pro triggerování . . . . .	41
B.2	USART . . . . .	42
B.2.1	Proměnné, přetypování, volání . . . . .	42
B.2.2	Funkce . . . . .	42
B.2.3	usartDescriptor . . . . .	42
<b>C</b>	<b>Relativní odchylky měřených průběhů od skutečné teploty</b>	<b>43</b>
C.1	Varianta A . . . . .	43
C.2	Varianta B . . . . .	44

## SEZNAM OBRÁZKŮ

1.1	Topologie star . . . . .	13
1.2	Topologie tree . . . . .	13
1.3	Topologie mesh . . . . .	14
1.4	ISO model protokolu ZigBee [3]. . . . .	15
2.1	Modul ZigBit900[12]. . . . .	16
5.1	Schéma zapojení při měření teploty . . . . .	26
5.2	Graf průběhu teploty při měření Variantou A . . . . .	28
5.3	Varianta A - spotřeba uzlů při kmitočtovém pásmu 868 MHz a 2,4 GHz	30
5.4	Graf průběhu teploty při měření Variantou B . . . . .	31
5.5	Varianta B - spotřeba uzlů při kmitočtovém pásmu 868 MHz a 2,4 GHz	32
A.1	Vývojový diagram algoritmu měřícího teplotu periodicky . . . . .	40
A.2	Vývojový diagram algoritmu triggrovaného měření teploty . . . . .	40
C.1	Odchytky od skutečné hodnoty? . . . . .	43
C.2	Odchytky od skutečné hodnoty? . . . . .	44

## SEZNAM TABULEK

1.1	Porovnání ZigBee a Bluetooth[2]. . . . .	11
1.2	Frekvenční pásma pro ZigBee[3]. . . . .	12
3.1	Nastavení DIP přepínačů k získání krátké adresy. . . . .	19
5.1	Počet měření a bezdrátových přenosů koncových zařízení . . . . .	29
5.2	Součet čtverců a průměr relativních odchylek pro Variantu A . . . . .	29
5.3	Počet měření a bezdrátových přenosů koncových zařízení . . . . .	31
5.4	Součet čtverců a průměr relativních odchylek pro Variantu B . . . . .	31

# ÚVOD

Bakalářská práce se zabývá metodami návrhu a realizací aplikace pro bezdrátový snímač, určený pro sledování prostředí budov, s ohledem na energetickou náročnost zvolených řešení. Snímač má komunikovat přes protokol ZigBee. Jedna základní myšlenka při vyvíjení této technologie byla, že bude nízkoenergetická, možnost napájet síťové uzly bateriově s co nejdelší životností (řádově měsíce až jednotky let) baterií, a dosáhnout tak plného oddělení od kabelových rozvodů. Z tohoto důvodu si kladu za cíl bakalářské práce porovnat měřicí metody dle energetické náročnosti řešení v reálném prostředí a dosažené přesnosti přenosu dat ze snímače do nadřazeného systému. Výsledek práce může být aplikován v praxi.

V rámci bakalářské práce byla vytvořena aplikace řešící snímání teploty v inteligentní budově a její následný bezdrátový přenos pomocí protokolu ZigBee. Samotný algoritmus byl implementován a odladěn na bezdrátových modulech ZigBit900 od firmy MeshNetics, které mohou být napájeny bateriově. Pomocí těchto modulů byla experimentálně naměřena teplota. Měření probíhalo pomocí dvou přístupových metod. První metoda periodicky sleduje a odesílá teplotu a druhá metodu teplotu periodicky měří, porovná ji s naposledy odeslanou teplotou a pokud se liší, či neliší o určenou hodnotu, nastane nebo nenastane bezdrátový přenos. Kmitočtové pásmo pro přenos bylo nastaveno na 868 MHz, a byla použita modulace Binary-Phase Shift Keying (BPSK).

V teoretické části práce je seznámení se ZigBee, popsána technologie a vývojový kit, s kterým jsem pracoval, uvedena nastavení protokolu. V další kapitole je popsán návrh řešení a jeho implementace. Závěrečná kapitola se věnuje zpracování výsledků, výpočtu odchylky od skutečného teplotního průběhu, přibližnému teoretickému výpočtu spotřeby uzlů a jejich relativnímu porovnání. V této kapitole je přidán i teoretický výpočet přenosu stejně dlouhého paketu, na otevřeném kmitočtovém pásmu 2,4 GHz.

# 1 ZIGBEE

ZigBee je bezdrátová komunikační technologie. Při prvním pohledu na parametry technologie může připomínat standard Bluetooth, který v době zakládání základních kamenů ZigBee prožíval obrovský boom. V podstatě tyto dvě technologie spolu souvisí. Jsou zahrnuty do kategorie IEEE 802.15. Obě spojují nízkovýkonová zařízení na malé vzdálenosti (desítky metrů) v sítích PAN. Bluetooth v době svého vydání na trh představovalo v bezdrátové komunikaci krok dopředu. Ale našlo uplatnění převážně ve spotřební elektronice. Pro některé aplikace nebylo příliš vhodné. A proto v roce 2002 vznikla ZigBee Alliance za níž stojí přední světové firmy zabývající se automatizací a koncem roku 2004 vydávají standart ZigBee, který je nízkoenergetický, umožňuje mít v síti až tisíce uzlů a umožňuje, aby zařízení v síti mohlo tzv. spát, tedy být v režimu, ve kterém nekomunikuje s okolím ani nevykonává žádnou činnost, kdy má naprosto minimální spotřebu (jednotky  $\mu$ Ampér). To jsou nejzákladnější vlastnosti ZigBee. V Tab. 1.1 jsou porovnány nejrozšířenější bezdrátové technologie[2][4].

Tab. 1.1: Porovnání ZigBee a Bluetooth[2].

Vlastnost	ZigBee	Bluetooth
nominální dosah (bez zesilovače)	30 m	10 m
maximální přenosová rychlost	250 kb/s	700 kb/s
maximální počet zařízení v jednom segmentu	255	8
maximální počet zařízení v síti	4000	není omezeno
komplexnost protokolu (velikost firmwaru)	30 kB	250 kB
vyhledání nového zařízení	typicky 30 ms	do 3 s

Příklady uplatnění standardu ZigBee[4]:

- automatizace, průmyslová automatizace - komunikace senzorů, řízení regulátorů, motorů
- spotřební elektronika - bezdrátové ovládání a sběr dat spotřebičů
- počítačové periferie - myši, klávesnice, herní ovladače
- zdravotnictví - patientské monitory
- zabezpečovací systémy - senzory, klávesnice, čtečky přístupových čipových karet/kódů
- inteligentní budovy - sensorická i řídicí část

## 1.1 Standard ZigBee

### 1.1.1 Frekvenční pásma

Pro bezdrátovou komunikaci na standardu Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 bylo vymezeno několik rádiových pásem. To z důvodu použití v různých geografických polohách, kde se liší předpisy a organizace pásem. Hlavním rozdílem je organizace v Evropě a Americe. Pro použití v různých lokalitách jsou definována pásma zobrazena v Tab. 1.2 [3].

Tab. 1.2: Frekvenční pásma pro ZigBee[3].

Lokace	Pásmo	Další specifikace
Globální	ISM 2,4 GHz	16 kanálů, rychlost 250 kb/s
Amerika a Austrálie	915 MHz	10 kanálů, rychlost 40 kb/s
Evropa	868 MHz	1 kanál, rychlost 20 kb/s

K implementaci navrženého algoritmu bylo použito evropské frekvenční pásmo 868 MHz.

### 1.1.2 Síťová zařízení

ZigBee definuje tři síťové prvky:

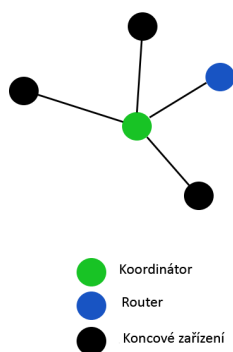
- **Koordinátor** - základní (páteří) prvek sítě. Toto zařízení musí obsahovat každá ZigBee síť (příp. v síti peer-to-peer jsou obě zařízení koordinátory). Koordinátor přiděluje síťové adresy ostatním zařízením. Zajišťuje topologii sítě, její zabezpečení. Jako jediný prvek sítě je neustále zapnutý, proto není zvykem, že by byl bateriově napájený.
- **Router** - směřovač, v hierarchii sítě je druhé nejvyšší zařízení. Vytváří topologii bezdrátové sítě.
- **Koncové zařízení** - end device, nevytváří dál síťovou strukturu a komunikuje pouze s jedním nadřazeným zařízením. Zajišťuje vykonávání úkonů - sběr dat, řízení procesů.

Na zařízení v síti lze pohlížet i z pohledu, zda-li mají v sobě implementován protokolový rámec, či nikoliv. Zařízení, které v sobě má protokolový rámec se nazývá Full Function Device (FFD) a může zajišťovat veškeré síťové služby, může tvořit strukturu sítě. Dle předešlého rozdělení zde patří koordinátor a router. Namísto Reduced Functionality Device (RFD) zařízení má v sobě implementovány jen ty nezbytnější protokolové knihovny, aby byla omezena hardwarová náročnost a spotřeba elektrické energie zařízení. Takové zařízení může pracovat pouze jako koncové - end device[1].

### 1.1.3 Topologie sítě

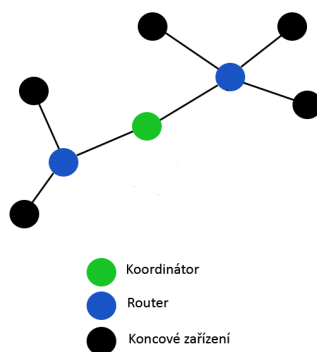
ZigBee využívá k adresaci jednotlivých zařízení binární adresovací kódy, které jsou dlouhé (64 bit) nebo zkrácené (16 bit). Každá bezdrátová síť je označena 16 bitovým identifikátorem Personal Area Network (PAN), který slouží k rozpoznání více sítí, které se v jednom prostoru překrývají. Takto vytvořená bezdrátová síť může vytvořit jednu ze tří základních topologií [3]:

- Star - vždy definováno jedno zařízení, přebírající funkci koordinátoru a ostatní působí jako koncová zařízení. topologie je zobrazena na obrázku 1.1.



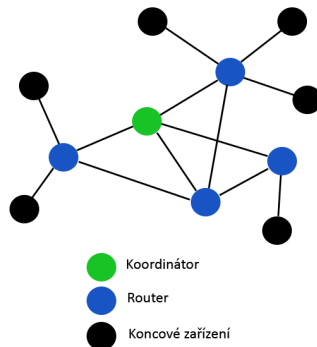
Obr. 1.1: Topologie star

- Tree - jedno zařízení slouží jako koordinátor a ostatní jako koncová zařízení. Na rozdíl od star nemusejí všechna zařízení komunikovat přímo s koordinátorem, ale mohou využít jiné zařízení v síti, které má implementováno FFD. Díky tomu umožňuje tato topologie zvyšovat dosah bezdrátové sítě. Jak topologie vypadá, je znázorněno na obrázku 1.2.



Obr. 1.2: Topologie tree

- Mesh - tato topologie kombinuje obě předchozí, a to náhodným způsobem, dokáže tedy vytvořit rozličné spojení mezi zařízeními v síti. Propojení topologie je znázorněno na obrázku 1.3.



Obr. 1.3: Topologie mesh

V práci je použita topologie mesh, protože prostředí BitCloud pracuje se ZigBee PRO, které podporuje pouze topologii mesh. Pomocí této struktury lze vytvořit i předchozí dvě topologie, tedy použitá topologie je typu mesh, vypadající jako star.

### 1.1.4 ZigBee PRO

Od uplynutí zavedení původního standardu v roce 2004 vznikly jeho používáním nové požadavky a proto v roce 2008 vychází ZigBee PRO s několika změnami. ZigBee PRO je částečně zpětně kompatibilní [5].

Změny v ZigBee PRO[5]:

- Adresování - ZigBee PRO používá stochastické přidělování adres s mechanismem detekce kolizí adres. Používá pouze strukturu mesh.
- Proměnná vysílací frekvence - při provozování ZigBee v Industrial, scientific and medical (ISM) pásmu, může být rušeno od jiných aplikací, které jsou používány v domácnostech i průmyslu a mohou způsobit zhoršenou kvalitu přenosu. Proměnná vysílací frekvence může zmírnit dopady kolizí z důvodu rušení.
- Zabezpečení přenosu - vytváření celosíťového šifrovacího klíče.

## 1.2 Model ZigBee

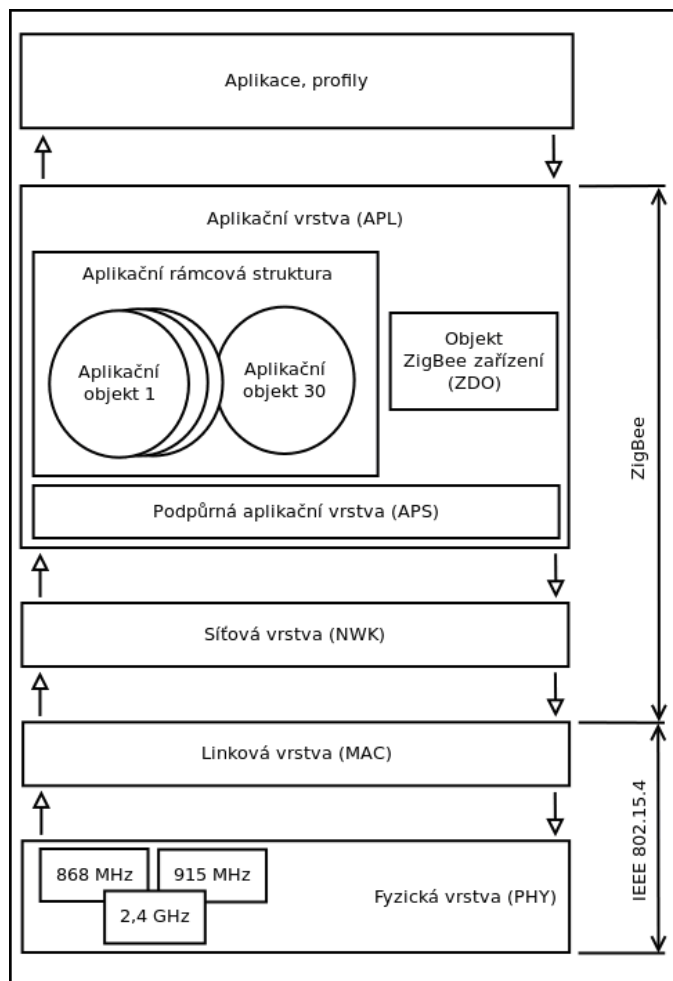
Struktura ZigBee je zobrazena na obrázku 1.4. Základ tvoří fyzická vrstva, která obstarává rádiový přenos a linková vrstva, která definuje samotný komunikační pro-

tokol.

Nad těmito vrstvami je vlastním standardu definovaná ZigBee síťová vrstva a struktura pro aplikační vrstvu. Síťová vrstva připojuje a odpojuje k síti, zabezpečuje a směruje pakety. U koordinátoru startuje síť a přiřazuje adresy zařízením v síti.

Aplikační vrstva se skládá z pomocné aplikační vrstvy, která udržuje párovací tabulky, objektů ZigBee, který definuje roli objektu v rámci sítě a uživatelských aplikačních objektů.

Nad těmito vrstvami je samotná aplikace, které k horním vrstvám přistupuje[3].

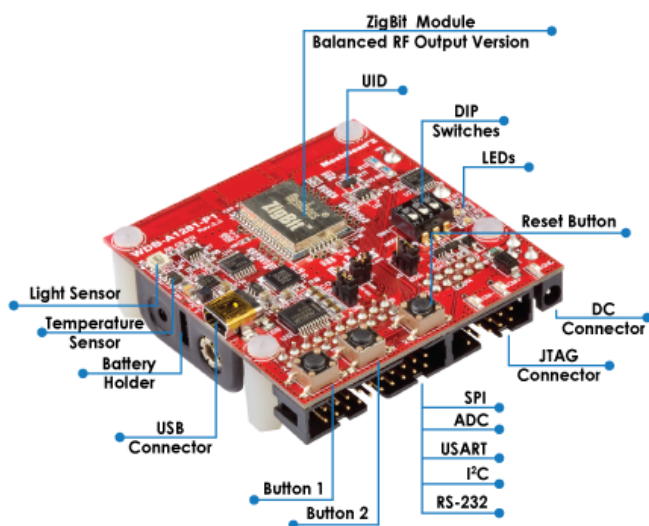


Obr. 1.4: ISO model protokolu ZigBee [3].

## 2 MESHNETICS ZIGBIT 900 KIT

### 2.1 Základní charakteristika kitu

Vývojový kit, určený především pro vývojáře. Kit obsahuje 3 moduly, které komunikují na standardu ZigBee, mohou být bateriově napájené. Celý modul řídí mikroprocesor ATmega1281 a bezdrátovou komunikaci zajišťuje čip AT86Rf212. Přímou na desce plošného spoje se nachází snímač teploty, světla a stavu baterií, dvě ovládací tlačítka SW1 a SW2, 3 pinový DIP přepínač, mini USB konektor, RESET tlačítko, 3 LED a dva porty pro připojení externích snímačů, či komunikačních sběrnic. Celý modul s popisem jeho částí je na obrázku 2.1.



Obr. 2.1: Modul ZigBit900[12].

### 2.2 Parametry ZigBit 900

- Rozměry 18,8 x 13,5 mm
- Vysoká receive (RX) citlivost (-110 dBm)
- Více jak 10 dBm výstupní výkon
- Malý energetický odběr (<6  $\mu$ Ah)
- 128 KB flash paměti, 8 KB random-access memory (RAM), 4 KB Electrically Erasable Programmable Read-Only Memory (EEPROM)
- dvě modulace BPSK, Quadrature Phase Shift Keying (Q-PSK)
- možná rozhraní: Universal Synchronous Receiver and Transmitter (USART), Inter-Integrated Circuit (I2C), Serial Peripheral Interface bus (SPI) a další

- 868 / 915 MHz pásmo [7]

## 3 BITCLOUD ZIGBEE PRO 1.11.0

Atmel BitCloud je soubor softwaru, knihoven, manuálů pro podporu aplikací běžících na bezdrátových platformách Atmel a je zcela v souladu se ZigBee PRO platformou. Obsahuje snadno použitelné funkce pro vývojáře, demo aplikací, manuály pro tvorbu vlastní aplikace a soubor knihoven a funkcí pro obsluhu ZigBee a podporovaného hardwaru. V současné době je nejnovější verzí verze 1.11, která vyšla v březnu letošního roku. BitCloud obsahuje také Bootloader, pomocí něž se nahrává zkompilovaný program do hardwaru.

### 3.1 Výběr programu

BitCloud nabízí několik demo aplikací, kdy se každá z aplikací snaží demonstrovat určitý okruh podporovaných funkcí. Všechny demoaplikace nejsou určeny pro jakýkoliv hardware, který BitCloud podporuje[6]. Pro ZigBit jsou všechny demoaplikace podporovány, proto mohl být proveden výběr podle její funkčnosti s ohledem na co největší podobnost s řešenou úlohou. Popis funkčnosti demoaplikací a možnost použití na určeném hardwaru je v manuálu `AVR2052_BitCloud_Quick_Start_Guide.pdf`. Potřebám periodického měření teploty a triggerování se nejvíce blížilo demo Low-power.

### 3.2 Popis demo Lowpower

Aplikace, demonstruje malou nízkoenergetickou síť, ve které je jeden koordinátor a až 7 koncových zařízení. Data tedy sbírá koordinátor. Minimální velikost sítě je jeden koordinátor a jedno koncové zařízení.

#### 3.2.1 Adresace v síti

Každé zařízení v síti musí mít svou unikátní krátkou adresu. Adresa se volí DIP přepínači na ZigBitu. Koordinátoru odpovídá adresa 0x0000 (všechny DIP v poloze OFF). Třetí DIP přepínač reprezentuje nejvýznamnější bit krátké adresy. Tabulka 3.1 ukazuje nastavení DIP přepínačů odpovídající jednotlivým adresám.

#### 3.2.2 Funkce aplikace

Po nahrání aplikace do zařízení ZigBit a nastavení krátké síťové adresy na DIP přepínačích může být spuštěna síť. Síť se spouští tlačítkem SW1 na každém zařízení zvláště. Při spuštění sítě na koordinátoru, spustí síť, to je indikováno rozsvícením

Tab. 3.1: Nastavení DIP přepínačů k získání krátké adresy.

Adresa uzlu	SW4:1	SW4:2	SW4:3
0	OFF	OFF	OFF
1	OFF	OFF	OFF
2	ON	ON	OFF
3	OFF	ON	OFF
4	ON	OFF	ON
5	OFF	OFF	ON
6	ON	ON	ON
7	OFF	ON	ON

zelené Light-Emitting Diode (LED). Pokud koordinátor síť vytvoří, zelená LED svítí neustále. Při spuštění koncového zařízení se snaží najít síť, do které by se připojil - to je indikováno blikáním zelené LED s periodou 500 ms. Při nálezu sítě se do ní přihlásí, to je indikováno rozsvícením zelené LED, pokud je zařízení v síti, může přejít do režimu spánku, zelenou LED vypne.

Všechny zařízení v síti indikují rádiové vysílání/příjem (TX/RX) stejně. TX mód je indikován rozsvícením červené LED po dobu přenosu a RX mód je indikován oranžovou LED. Výpadek koordinátoru z již vytvořené sítě je indikován na koncových zařízeních opět blikáním zelené LED - hledají novou síť.

Bezdrátový přenos z koncových zařízení se provádí se zadanou periodou (parametr CS\_END\_DEVICE\_SLEEP\_PERIOD, zadaná hodnota 10000 odpovídá době 10 s). K vyžádání přenosu mimo tuto periodu slouží tlačítko SW2 na koncových zařízeních. Čítač s periodou se použitím tohoto tlačítka, respektive přenosu vyresetuje a běží od začátku. Tlačítko SW2 na koordinátoru slouží ke změně snímačů na všech koncových zařízeních v síti. Je možnost přepínat (v tomto pořadí) mezi teploměrem -> stav baterií -> snímačem osvětlení. Při stisku SW2 na koordinátoru bude informace od koncových zařízení předána při jejich dalším probuzení a přenosu směrem ke koordinátoru.

Koordinátor každou přijatou hodnotu odesílá přes USART. Pokud je připojen k PC přes USB kabel, lze hodnoty sledovat v příslušném softwaru. Odesílá adresu koncového zařízení, které mu hodnotu poslalo, senzor, který byl k měření použit a příslušná hodnota [9]

### 3.2.3 Nastavení Lowpoweru pro měření teploty.

Pro měření byl využit koordinátor spojený s dvěma koncovými zařízeními, které měly krátké síťové adresy 0x0001 a 0x0002. Byl využit pouze teplotní senzor. Napájení USB kabel, který byl použit i k USART přenosu. Zařízení se probouzela periodicky, tedy nebyla spouštěna tlačítkem SW2.

## 3.3 Nastavení sítě pro ZigBee

### 3.3.1 Makefile

Pro správnou funkci všech funkcí s Hardware (HW) musí být v Makefilu Vybrán správný HW. V případě použití všech uzlů v síti ZigBit 900, musí být odkomentováno:

```
PROJECT_NAME = MeshBean_900
CONFIG_NAME = All_ZigBit_Atmega1281_Rf212_8Mhz_Gcc
```

### 3.3.2 configuration.h

V tomto souboru se nastavují základní síťové vlastnosti. V této podkapitole jsou obsaženy základní definice.

- `#define APP_NWK_NODE_ADDRESS 0x0000` nastavení krátké síťové adresy na 0x0000
- `#define BSP_ENABLE_RS232_CONTROL 1` povolení sériové komunikace
- `#define CS_CHANNEL_PAGE 0` definuje typ modulace. Při kmitočtovém pásmu 2,4 GHz je ignorován. U 868/912 MHz jsou akceptovány pouze hodnoty 0 (BPSK) a 2 (Q-PSK)
- `#define CS_CHANNEL_MASK 32` bitové pole, které určuje frekvenční pásmo uzlů. Pět nejvýznamnějších bitů (B31...B27) musejí být vždy 0. Zbylých 27 bitů (B26...B0) ukazují dostupnost jednotlivých kanálů. Pro frekvenci 868 MHz nastaveno na `1L<0x00`.
- `#define CS_EXT_PANID 0xAAAAAAAAAAAAAAAAALL` rozšířená adresa PAN
- `#define CS_END_DEVICE_SLEEP_PERIOD` perioda, se kterou se koncová zařízení probouzí
- `#define CS_RF_TX_POWER 3` nastavení výkonu na 3 dBm [8]

### 3.3.3 Nastavení sítě pro měřicí aplikaci

Bezdrátová síť byla nastavena na evropské pásmo, tedy 868 MHz, rychlost přenosu tedy 20 kb/s a modulace BPSK. Při relativním porovnání spotřeby je proveden i

výpočet pro frekvenci 2,4 GHz s přenosovou rychlostí 250 kb/s.

## 4 ŘEŠENÍ STUDENTSKÉ PRÁCE

Pro porovnání dat snímaných snímačem koncového zařízení přenesených do koordinátoru různými způsoby jejich přenosu byla zvolena dvě různá řešení měření a přenosu dat - periodické sledování teploty a triggerované sledování teploty. Obě řešení byla implementována, experimentálně ověřeny jejich vlastnosti a vliv na spotřebu koncových zařízení.

### 4.1 Periodické sledování teploty

#### 4.1.1 Popis řešení

Koncové zařízení má pevně určenou periodu, kdy se probudí. Po probuzení změří teplotu, kterou ihned odesílá koordinátoru. Po odeslání hodnoty přechází opět do režimu spánku. Teplota je tedy sledována a odesílána s pevnou periodou. Algoritmus tohoto řešení je zobrazen ve vývojovém diagramu v příloze A.1.

### 4.2 Triggerované sledování teploty

#### 4.2.1 Popis řešení

Koncové zařízení má pevně stanovenou periodu, se kterou se probouzí a měří teplotu. Naměřenou hodnotu porovná s poslední odeslanou hodnotou a odešle ji jen v případě, že se liší od předchozí naměřené hodnoty o určené  $\Delta$ . Algoritmus je zobrazen ve stavovém diagramu v příloze A.2. Při realizaci triggerování byla  $\Delta = 1$  °C.

### 4.3 Vývojové diagramy řešení

Stavový diagram graficky zobrazuje algoritmus programu obsluhující koncové zařízení. Oba diagramy jsou obsaženy v příloze A.

Popis vybraných bloků vývojových diagramů:

- START - zapnutí koncového zařízení tlačítkem SW1
- INICIALIZACE OD SÍTĚ - přihlášení do sítě, získání adresy, navázání komunikace s koordinátorem. V případě nezdaru inicializace se koncové zařízení snaží inicializaci v náhodných časových intervalech opakovat. A to až do té doby než je odpojen od napájení, či v případě bateriového napájení se mu

vybijí baterie. Toto řešení není ideálním způsobem řešení nepodařené inicializace, bylo však implementováno v demo programu Lowpower a tato část kódu nebyla upravována.

- ČEKÁ - zařízení je v režimu spánku, po určenou periodu

Vývojové diagramy neobsahují konec algoritmu, neboť úkolem koncových zařízení v reálném použití je měřit teplotu stále dokola, dokud se v nich nevybijí baterie. Poté je musí zodpovědná osoba vyměnit, pokud se chce pokračovat v měření. Při měření byl algoritmus ukončen odpojením napájení z USB portu.

## 4.4 Implementace

Periodické odesílání dat je vyřešeno v aplikaci Lowpower. Proto se podkapitola zabývá pouze změnou funkce, která čte hodnotu ze snímače. Funkce postupuje dle algoritmu z vývojového diagramu A.2. Její ukázka je v příloze B.1.

Aktuálně naměřená hodnota je uložena v proměnné `value`. V proměnné `oldSentValue` je uchována hodnota teploty předchozího rádiového přenosu. První podmínka testuje, zda je koncové zařízení opravdu ve stavu pro měření hodnoty. Druhá podmínka povolí odeslání naměřené hodnoty, pokud byla hodnota změřena správně a při měření nenastala nějaká chyba. A třetí podmínka je podmínka triggeru, testuje, zda se liší aktuálně naměřená hodnota od předchozí hodnoty o méně než zadané `delta`, či nikoliv. V případě pozitivního výsledku podmínky, tedy aktuální naměřená hodnota se neliší o více než `delta`, se zařízení opět uspí a testovat bude opět při dalším probuzení. V případě negativního výsledku podmínky je aktuálně naměřená hodnota odeslána koordinátoru a koncové zařízení přechází do spánkového režimu až po rádiovém přenosu. Po nastavení režimu koncového zařízení na spánkový (`END_DEVICE_STATE_SLEEP_PREPARE`), musí proběhnout zhasnutí indikačních LED pomocí funkce `appCloseLeds()`, protože zelená LED zůstávala v režimu přenosu, tedy zapnutá. Lepším řešením spánku by bylo přepnutí do režimu `END_DEVICE_STATE_PERIPHERY_CLOSING`, který se do spánkového režimu přepne sám. Tento režim nejprve zavře všechna používaná zařízení na desce. Této možnosti ale nebylo využito, neboť ihned zavřel i USART, který byl použit pro ladění programu. Nevypnutí ostatních zařízení na desce plošného spoje není nikterak na překážku, jelikož byla měřena pouze teplota a spotřeba koncového zařízení je pouze vypočítaná.

### 4.4.1 Universal Synchronous Receiver and Transmitter

V demu Lowpower je již implementováno rozhraní USART a demo ho používá k výpisu přijatých hodnot z koordinátoru. Avšak při ladění programu byla potřeba

odesílat znaky, řetězec z různých míst a zdrojových souborů. Proto musela vzniknout ještě jedna funkce, přijímající jako parametr dvojitý ukazatel na přetypované pole hodnot. Při volání funkce z potřebné části programu muselo prvně dojít k přetypování a až následně k zavolání funkce pro zápis na USART. Ukázka s použitými proměnnými, přetypováním a voláním funkce je v příloze B.2.1. Důležité při implementaci je správné přetypování hodnoty, která má být odeslána.

Samotná funkce, implementovaná do zdrojového kódu souboru `boardabstraction.c` pak vypadala takto B.2.2. Funkce je samozřejmě globální.

Nastavení parametrů pro USART přenos je v `usartDescriptoru` B.2.3, který musí být správně nastaven. Také musí být povolena sériová linka (v demu Lowpower již povolena byla kvůli zobrazování dat z koordinátoru). `UsartDescriptor` je tedy strukturou nastavení. Tento strukturovaný datový typ je definován v `BitCloudu - HAL_UsartDescriptor_t`.

Díky `BitCloudu`, který má funkci pro odesílání na USART již implementovanou, bylo vlastní odeslání na USART již jednoduché. Důležitý je správný formát a datový typ odesílaných dat..

## 4.5 Měření teploty - měřicí schéma

Měření teploty bylo provedeno ve dvou různých cyklech, lišících se od sebe.

První varianta měření (nazvaná A) měřila dlouhý teplotní děj. Tedy simulovala měření teploty v budově. Samotná teplota byla sledována s periodou 2 minuty a jednotlivé uzly, měřící a odesílající teplotu metodou periodického měření a triggerování měřily teplotu s periodou 10 minut. Celé měření trvalo 9 hodin. Pro takovýto delší teplotní děj jsem využil nočního poklesu venkovní teploty a měřil jsem od půlnoci do ranních hodin. Hodnoty jsou zobrazeny v grafu 5.2. Poloha měřících uzlů byla východním směrem, proto jsou hodnoty po východu slunce tak vysoké, poněvadž na koncová zařízení svítilo slunce.

Druhá varianta řešení (varianta B) měřila kratší teplotní děje, které byly simulované. Senzory byly náhodně zahřívány/ochlazovány. Perioda měření teploty byla 30 sekund a koncová zařízení odesílala, respektive měřila, teplotu s periodou 2 minuty. Samotné měření probíhalo 1 hodinu a 50 minut. Naměřené hodnoty jsou zobrazeny v grafu 5.4 v následující kapitole.

### 4.5.1 Zpracování hodnot

Pro relativní porovnání experimentálně naměřených hodnot, jsou tyto hodnoty v následující kapitole zpracovány. Je vypočtena přibližná spotřeba koncových zařízení

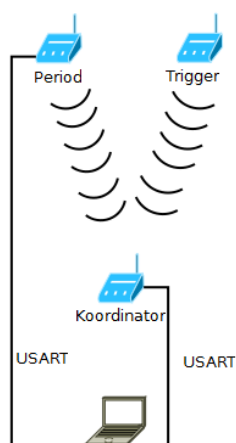
pokud jen měří teplotu nebo měří i vysílají v jednom kroku, při jednom probuzení. Hodnota spotřeby je vypočítána pro frekvenční pásmo, na kterém běžel experimentální pokus a pro frekvenční pásmo 2,4 GHz pro porovnání spotřeby. Dále byly vypočítány odchylky jednotlivých měřících metod od skutečného teplotního průběhu. Výpočty většího množství dat jsou v rámci udržení přehlednosti práce uvedeny pouze graficky. Také naměřené průběhy jsou uvedeny jen graficky. Hodnoty uvedené v tabulce jsou na příloženém DVD.

## 5 MĚŘENÍ A VÝPOČTY

### 5.1 Měření

Při měření bylo potřeba zjistit přesný průběh teploty, aby mohly být zobrazeny a vyhodnoceny odchylky periodického a triggerovaného získávání teploty. Při měření tohoto přesného průběhu teploty bylo využito možnosti napájet koncová zařízení přes USB port. To bylo možno, protože byla měřena jen teplota. Spotřeba měřena nebyla, ale je přibližně dopočítána pro naměřený teplotní průběh. Díky tomuto napájení byla využita komunikace přes rozhraní USART. Celé zapojení je zobrazeno na obrázku 5.1. Tedy jedno z koncových zařízení se probouzelo častěji než jen pro měření a přenos rádiiem, ale probouzelo se v krátké periodě a měřilo teplotu, jejíž hodnotu ihned odesílalo do PC přes rozhraní USART. Poté se přepnulo do režimu spánku. Podle varianty řešení se u tohoto uzlu každou 4. nebo 5. měřící periodou naměřená hodnota zpracovala k bezdrátovému přenosu.

Zařízením, které provádělo toto referenční měření teploty byl modul měřící teplotu periodicky. Tedy každou naměřenou teplotu poslal přes rozhraní USART, měření probíhalo s krátkou periodou rovnou 4násobku (respektive 5násobku) periody pro bezdrátové měření a tedy každou příslušnou periodu odeslal teplotu i bezdrátově. Koordinátor teplotu přijal a poslal přes rozhraní USART do PC, kde byla zaznamenána. K záznamu teploty, odesílané zařízeními přes USART byl použit free software 232Analyzer. Umožňuje přepnutí přijatého znaku z DEC na HEXa i ASCI, čehož bylo využito. Hodnoty byly zpracovány v tabulkového procesoru.



Obr. 5.1: Schéma zapojení při měření teploty

## 5.2 Výpočet spotřeby

### 5.2.1 Frekvenční pásmo 868 MHz

Výpočet spotřeby je orientační a slouží k relativnímu porovnání obou měřících metod. Přibližná doba měření teploty koncovým zařízením je 10 ms a procesor ATmega 1281 má spotřebu 5mA [10]. Spotřeba je vypočtena ve vztahu 5.1.

$$E_{1281} = 5e^{-3} * 10e^{-3} = 50\mu Ah \quad (5.1)$$

Tomuto frekvenčnímu pásmu odpovídá přenosová rychlost 20 kb/s. Z této hodnoty lze vypočítat, dobu vysílání paketu. Je-li vzata v úvahu průměrná délka paketu 64B (maximální je 127B). Pak lze napsat 5.2.  $\text{Paket} = 64B = 512 \text{ b}$   $\text{Rychlost} = 20 \text{ kb/s}$

$$t = \frac{512}{20000} = 25,6ms. \quad (5.2)$$

Spotřeba AT86Rf212 je při TX režimu 17mA [11]. A při délce přenosu 25,6 ms je spotřeba rádiového čipu 5.3

$$E_{rf212} = 17e^{-3} * 25e^{-3} = 425\mu Ah \quad (5.3)$$

A celková spotřeba koncového zařízení, při měření teploty a jejím odesláním je 5.4

$$E_{868MHz} = E_{1281} + E_{rf212} = 50 + 425 = 475\mu Ah \quad (5.4)$$

Stavů při přenosu paketu je více, než jen TX. Ale ten má největší vliv.

### 5.2.2 Frekvenční pásmo 2,4 GHz

Při použití kmitočtového pásma 2,4 GHz je maximální přenosová rychlost 250 kb/s (pozn. pro použití tohoto frekvenčního pásma by musel být použit rádiový čip AT86Rf230). Doba vysílání je tedy 5.5

$$t = \frac{512}{250000} = 2,02ms. \quad (5.5)$$

Spotřeba při kmitočtovém pásmu v TX režimu je 16,5 mA. Z toho vyplývající spotřeba rádiového čipu 5.6 [11].

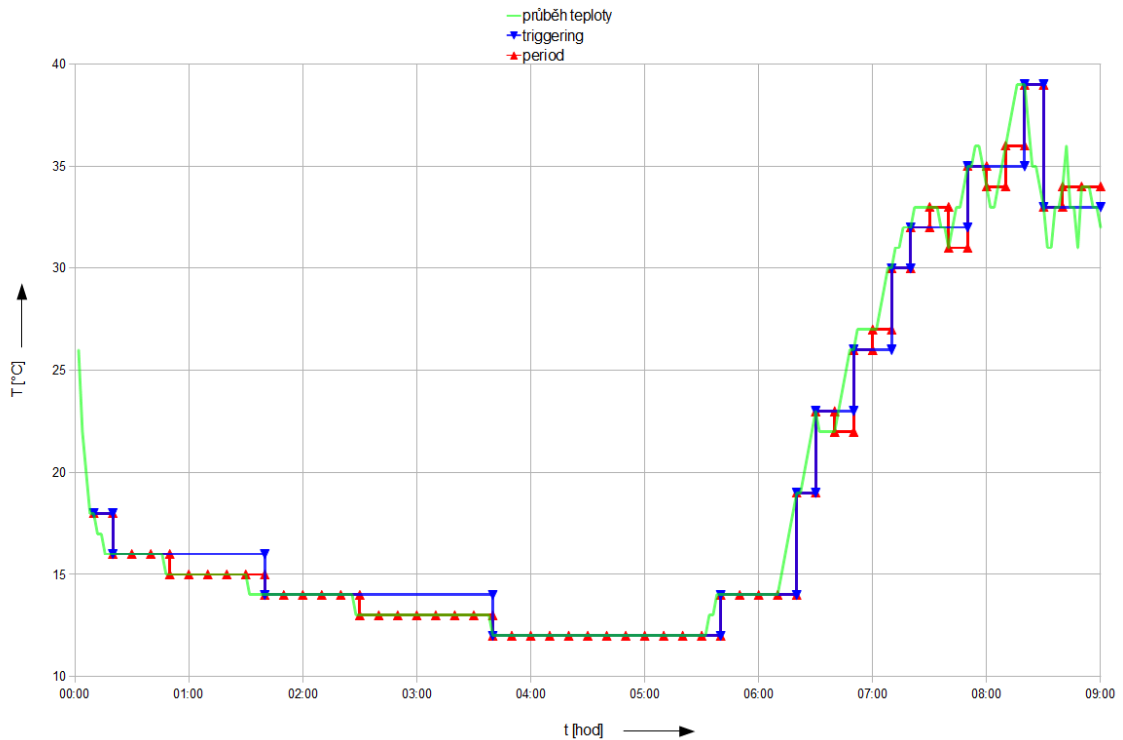
$$E_{rf230} = 2,2e^{-3} * 16,5e^{-3} = 36,3\mu Ah \quad (5.6)$$

A celková spotřeba koncového zařízení pracujícího ve frekvenčním pásmu 2,4 GHz vyjde 5.7

$$E_{2,4GHz} = E_{1281} + E_{rf230} = 50 + 36,3 = 86,3\mu Ah \quad (5.7)$$

### 5.3 Měření delšího teplotního děje - Varianta A

Při měření Varianty A byl naměřen průběh zobrazený v grafu 5.2. Počet bezdrátových přenosů, u triggeru počet měření teploty a počet rádiových přenosů jsou zobrazeny v tabulce 5.1. Koncové zařízení měřící teplotu periodicky, po každém měření teploty také vysílalo. Proto je počet měření a vysílání stejný.



Obr. 5.2: Graf průběhu teploty při měření Variantou A

Tab. 5.1: Počet měření a bezdrátových přenosů koncových zařízení

Řešení přenosu	Počet měření	Počet bezdrátových přenosů	Celková spotřeba
Trigger	54	13	8,23 mAh
Period	54	54	25,65 mAh

### 5.3.1 Odchylka periodického a triggerovaného průběhu od referenčního teplotního průběhu

Pro porovnání přesnosti obou realizačních metod byla vypočtena odchylka nejmenších čtverců změřených průběhů od průběhu skutečné teploty. Tedy byla spočtena odchylka v každém bodě, kdy koncová zařízení měřila, od skutečného teplotního průběhu a z každé této odchylky byl spočítán obsah čtverce, který má délku strany příslušnou odchylce. Sečtením těchto čtverců byla zjištěna pro změřený teplotní průběh celková odchylka měřící metody. Jejich porovnání je uvedeno v tabulce 5.2, kde je také uvedena průměrná hodnota relativní odchylky obou metod. Průběh relativní odchylky pro variantu A je v příloze C.1.

Tab. 5.2: Součet čtverců a průměr relativních odchylek pro Variantu A

	Průměrná relativní odchylka [%]	Součet čtverců odchylek
Period	1,78	266
Trigger	3,51	357

### 5.3.2 Spotřeba

Na základě naměřených výsledků, počtu rádiových přenosů u periodického odesílání a počtu měření a rádiových přenosů u triggerování byla vypočtena spotřeba obou měřících metod.

Výpočet energie spotřebované za celé měření Varianty A vychází z počtu rádiových vysílání a měření zobrazených v tabulce 5.1 a hodnot vypočtených pro jeden rádiový přenos.

Při periodickém měření teploty je celková spotřeba 5.8.

$$E_{period,868MHz} = E_{868MHz} * 54 = 475e^{-6} * 54 = 25,65mAh \quad (5.8)$$

Při triggerovaném měření teploty je celková spotřeba 5.9

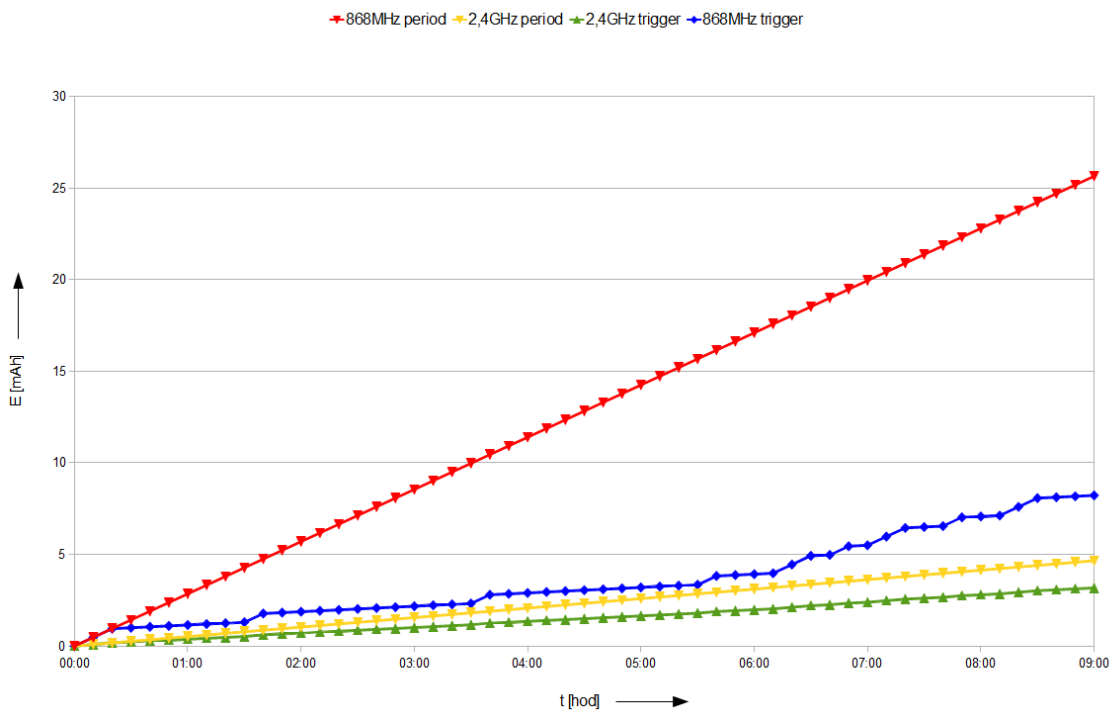
$$E_{trigger,868MHz} = E_{868MHz} * 13 + E_{1281} * 41 = 475e^{-6} * 13 + 50e^{-6} * 41 = 8,23mAh \quad (5.9)$$

Teoretické výpočty spotřeby, při uvažovaném kmitočtovém pásmu 2,4 GHz pro periodické odesílání 5.10 a pro triggerování 5.11.

$$E_{period,2,4GHz} = E_{2,4GHz} * 54 = 86,3e^{-6} * 54 = 4,66mAh \quad (5.10)$$

$$E_{trigger,2,4GHz} = E_{2,4GHz} * 13 + E_{1281} * 41 = 86,3e^{-6} * 13 + 50e^{-6} * 41 = 3,17mAh \quad (5.11)$$

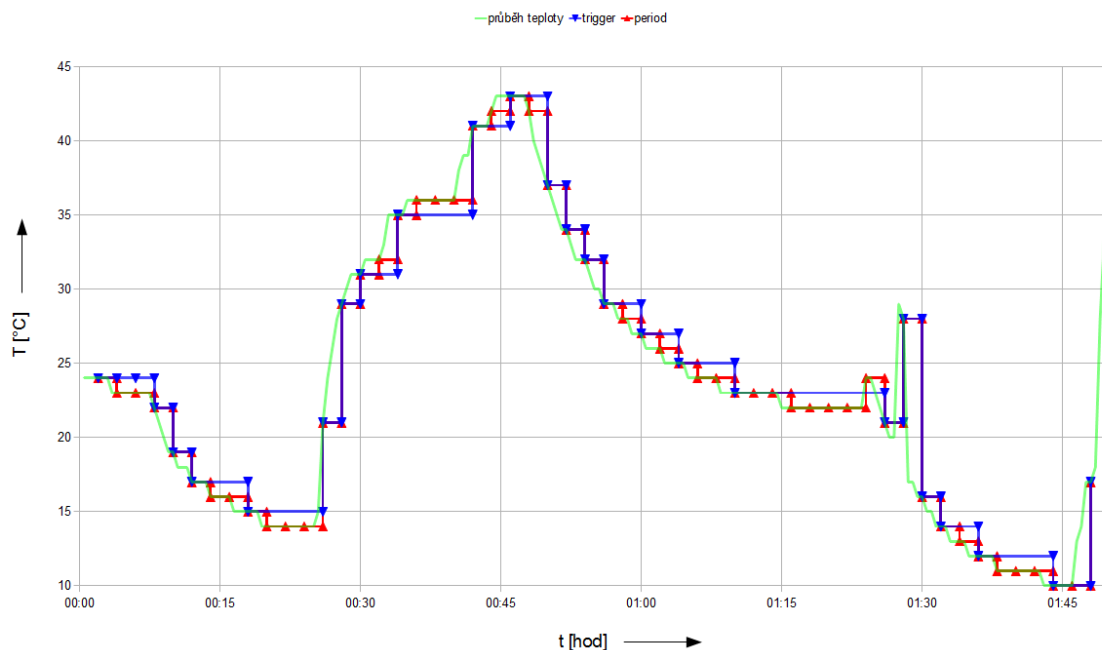
Výsledky průběžných výpočtů spotřeby jsou zobrazeny v grafu 5.3. Celková spotřeba uzlů je uvedena výše v tabulce (jen pro kmitočtové pásmo 868 MHz) 5.1.



Obr. 5.3: Varianta A - spotřeba uzlů při kmitočtovém pásmu 868 MHz a 2,4 GHz

## 5.4 Měření kratšího teplotního děje - Varianta B

Při měření Varianty B, byl naměřen teplotní průběh, který je zobrazen v grafu 5.4. Celkový počet rádiových přenosů u periodického odesílání hodnoty a počet rádiových přenosů a měření teploty u triggerování jsou zobrazeny v tabulce 5.3. Pro stejný teplotní průběh byl proveden výpočet spotřeby při kmitočtovém pásmu 2,4 GHz. Výsledky výpočtu jsou zobrazeny v grafu společně s kmitočtovým pásmem 868 MHz.



Obr. 5.4: Graf průběhu teploty při měření Variantou B

Tab. 5.3: Počet měření a bezdrátových přenosů koncových zařízení

Řešení přenosu	Počet měření	Počet bezdrátových přenosů	Celková spotřeba
Trigger	54	25	11,68 mAh
Period	54	54	25,65 mAh

### 5.4.1 Odchylka periodického a triggerovaného průběhu od referenčního průběhu teploty

Pro porovnání přesnosti obou realizačních metod pro Variantu B, byla vypočtena odchylka nejmenších čtverců změřených průběhů od průběhu skutečné teploty. Sečtením čtverců byly vypočteny pro změřený teplotní průběh celkové odchylky měřících metod. Jejich porovnání je uvedeno v tabulce 5.4, ve které je uvedena i průměrná hodnota relativní odchylky obou metod od skutečného teplotního průběhu. Průběh relativních odchylek v čase je zobrazen v příloze C.2.

Tab. 5.4: Součet čtverců a průměr relativních odchylek pro Variantu B

	Průměrná relativní odchylka [%]	Součet čtverců odchylek
Period	3,93	1129
Trigger	6,4	1375

## 5.4.2 Spotřeba

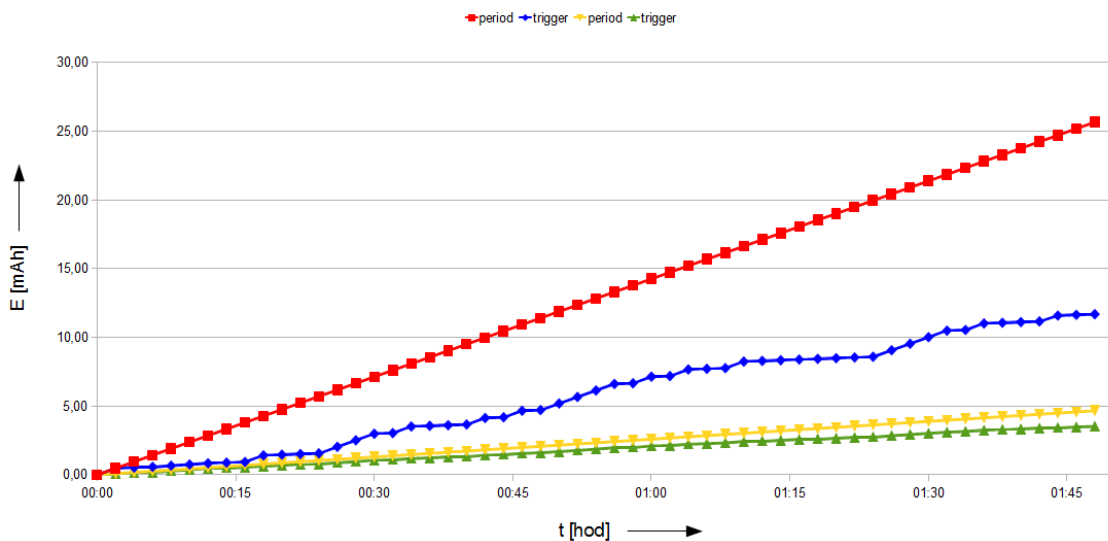
Výpočet spotřeby byl proveden pro obě měřící metody, na frekvenčním pásmu 868 MHz. Jejich průběh je zobrazen v grafu 5.5. Celková spotřeba koncových zařízení za celou dobu měření je výše v tabulce 5.3. Byl proveden i teoretický výpočet pro frekvenční pásmo 2,4 GHz, uvažované při stejném teplotním průběhu. Výsledky výpočtu jsou zobrazeny v grafu také.

$$E_{period,868MHz} = E_{868MHz} * 54 = 475e^{-6} * 54 = 25,65mAh \quad (5.12)$$

$$E_{trigger,868MHz} = E_{868MHz} * 25 + E_{1281} * 29 = 475e^{-6} * 13 + 50e^{-6} * 41 = 11,68mAh \quad (5.13)$$

$$E_{period,2,4GHz} = E_{2,4GHz} * 54 = 86,3e^{-6} * 54 = 4,66mAh \quad (5.14)$$

$$E_{trigger,2,4GHz} = E_{2,4GHz} * 25 + E_{1281} * 29 = 86,3e^{-6} * 13 + 50e^{-6} * 41 = 3,52mAh \quad (5.15)$$



Obr. 5.5: Varianta B - spotřeba uzlů při kmitočtovém pásmu 868 MHz a 2,4 GHz

## 5.5 Vyhodnocení

Pro oba teplotní průběhy byla nejvyšší spotřeba u periodického odesílání teploty. Spotřeba triggerovaného průběhu se liší při Variantě A a B. Je to způsoben častějším

vysíláním triggeru při častěji se měnící teplotě. Z průběhu spotřeby triggeru při Variantě A je vidět, že jeho spotřeba je více jak 3x nižší než periodického odesílání. Správnou volbou periody měření a  $\Delta$  pro konkrétní budovu, kde by byl bezdrátový uzel umístěn by šla spotřeba ještě více snížit. Zjednodušeně by se dalo říci, snažit se o to, aby trigger vysílal co nejméně.

Při porovnání průběhů spotřeby pro kmitočtové pásmo 2,4 GHz je spotřeba daleko nižší oproti kmitočtovému pásmu 868 MHz. Je to způsobeno zkrácenou dobou vysílání paketu, díky vyšší přenosové rychlosti. Vzhledem k tomu, že u bezdrátových modulů, které jsou napájeny bateriově je žádoucí, aby měly co nejmenší spotřebu energie a největší výdrž na jednu sadu baterií je toto kmitočtové pásmo pro realizaci nízkoenergetického zařízení vhodnější.

## 6 ZÁVĚR

V rámci předložené bakalářské práce měl být čtenář seznámen s bezdrátovým protokolem ZigBee. Následně bylo v plánu navrhnout algoritmus pro bezdrátový snímač, který sledoval prostředí v budově. Porovnat různé přenosy dat a vyhodnotit energetickou náročnost jednotlivých řešení a experimentálně ověřit jeho vlastnosti.

V úvodu bakalářské práce je charakterizována technologie ZigBee, hardware ZigBit 900 od firmy MeshNetics a vývojové prostředí BitCloud pro ZigBit verze 1.11 firmy ATMEL. Ve spojitosti se zkoumaným problémem byly navrženy dva algoritmy měřící teplotu a přenášející ji z koncových uzlů k síťovému koordinátoru. Algoritmus s názvem Period, který teplotu sledoval periodicky, tedy periodicky měřil a ihned odesílal. A druhý algoritmus nazvaný Trigger, který periodicky změřenou teplotu porovnával s předchozí odeslanou a pokud byla větší než určený rozdíl, odeslala se, ovšem pokud byla naměřená hodnota teploty menší, zařízení přešlo do režimu spánku. Oba tyto algoritmy se podařilo implementovat do hardwaru ZigBit 900, za pomoci demo aplikace Lowpower z ATMEL BitCloudu.

Pomocí obou metod - period i trigger - byly naměřeny dva teplotní průběhy. První nazvaný Varianta A měřil pomalý teplotní průběh, který simuloval teplotu v inteligentní budově. Druhý nazvaný Varianta B změřil rychlejší teplotní průběh. Naměřené hodnoty byly zpracovány a jsou zobrazeny v grafech. Z důvodu vysokého počtu naměřených dat, který by způsobil nepřehlednost práce, je v číselné formě uvádím pouze na přiloženém DVD. Dále byl proveden orientační výpočet spotřeby koncových zařízení pro kmitočtové pásmo 868 MHz, na kterém zařízení měřila a pro kmitočtové pásmo 2,4 GHz. Všechny výpočty byly relativně porovnány.

Při porovnání naměřených spotřeb při Variantě A a Variantě B je vidět zvýšená spotřeba Triggeru při častější změně měřené hodnoty. Proto se dá říci, že čím bude teplotní průběh konstantnější, tím více energie trigger uspoří. Naopak, pro měnící se průběh, se jeho spotřeba bude blížit periodickému řešení. Životnost baterií by v takovém triggerovacím zařízení mohla být přes 4 měsíce. Pro doladění spotřeby by se dala upravit i perioda, se kterou se koncové zařízení probouzí a měří.

Zajímavé je i porovnání odchylek obou metod od skutečného průběhu. Kdy u Varianty A byla průměrná relativní odchylka periodického měření menší než 2 %, u Varianty B odchylka vzrostla na hodnotu vyšší než měl trigger při Variantě A. Chyba triggeru při rychlejším průběhu Varianty B vzrostla na hodnotu 6,4 %. Při měření v budovách jsou takové hodnoty odchylek plně dostačující.

Porovnáním frekvenčních pásem 868 MHz a 2,4 GHz je zřejmý pozitivní dopad 12,5x vyšší přenosové rychlosti (u 2,4 GHz) na spotřebu. Toto porovnání svádí k diskuzi, zda by zvyšování přenosové rychlosti přineslo nižší spotřebu.

Práci lze rozšířit několika směry. Teoretický směr, kdy by se přesně vypočítala

hodnota spotřeby uzlu pro jeden rádiový přenos. Toto je velmi pracné řešení a možná až zbytečné, protože při nemožnosti spojení, neobdržení paketu apod. rádio čeká náhodně zvolenou dobu a opakuje pokus o komunikaci, což prodlužuje dobu vysílání paketu o neznámou dobu (pro výpočet se musí použít pravděpodobnost). Další možností rozšíření by bylo měřit přímo hodnoty spotřeby koncových zařízení elektrickým měřicím přístrojem. Spotřeba by šla také měřit tak, že jako zdroj energie by byl použit akumulátor nebo kondenzátor a měřil by se čas, za který zařízení spotřebuje všechnu známou energii v článku. Nevýhodou akumulátoru je, že lze uloženou energii zjistit velmi těžce, kdežto u kondenzátoru lze uložená energie zjistit velmi přesně.

# LITERATURA

## Elektronické články

- [1] CROSS, Pete. *Zeroing on ZigBee. Circuit Cellar.* [online]. 2005, iss. 175, [cit. 2011-05-25]. Dostupný z URL: <<http://www.circuitcellar.com/library/print/0205/Cross175/2.htm>>. ISSN 1528-0608.
- [2] BRADÁČ, Zdeněk; FIEDLER, Petr; KAČMÁŘ, Milan. *Bezdrátové komunikace v automatizační praxi VII: další standardy. AUTOMA* [online]. 2004, 7, [cit. 2011-05-25]. Dostupný z URL: <[http://www.odbornecasopisy.cz/index.php?id\\_document=32437](http://www.odbornecasopisy.cz/index.php?id_document=32437)>.
- [3] BRADÁČ, Zdeněk. *Bezdrátový komunikační standard ZigBee. Automatizace* [online]. Duben 2005, 48, č. 4, [cit. 2011-05-25]. Dostupný z URL: <<http://www.automatizace.cz/article.php?a=638>>.
- [4] *ZigBee. In Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 9. 7. 2005, 12:33, last modified on 28. 3. 2009, 07:36 [cit. 2011-05-25]. Dostupný z URL: <<http://cs.wikipedia.org/wiki/ZigBee>>.
- [5] VOJÁČEK, Antonín. *ZigBee PRO - nová vylepšená verze bezdrátové komunikace ZigBee. Rady a poslední novinky z oboru automatizace* [online]. 2 listopad 2008, [cit. 2011-05-25]. Dostupný z URL: <<http://automatizace.hw.cz/zigbee-pro-nova-vylepsena-verze-bezdratove-komunikace-zigbee>>.

## Manuály, datasheety

- [6] *AVR2052: BitCloud Quick Start Guide* [online]. 03/2011 [cit. 2011-05-25]. 5.2.1 Reference Applications, s. . Dostupný z URL: <[http://www.atmel.com/forms/bitcloud\\_rzraven.asp?category\\_id=163&family\\_id=676&subfamily\\_id=2124&fn=dl\\_BitCloud\\_ZIGBIT\\_1\\_11\\_0.zip](http://www.atmel.com/forms/bitcloud_rzraven.asp?category_id=163&family_id=676&subfamily_id=2124&fn=dl_BitCloud_ZIGBIT_1_11_0.zip)>.
- [7] *ZigBit 900 OEM Module Product Datasheet* [online]. 1,2. 9/2008 3s [cit. 2011-05-25]. Dostupný z URL: <[http://www.meshnetics.com/netcat\\_files/Image/M-251%28ZigBit%20900%20OEM%20Module%20Product%20Datasheet%29.pdf](http://www.meshnetics.com/netcat_files/Image/M-251%28ZigBit%20900%20OEM%20Module%20Product%20Datasheet%29.pdf)>. ZigBit900OEDOC.M-251 06V.1.2.
- [8] *AVR2050\_BitCloud\_Developers\_Guide* [online]. 03/2011 14s [cit. 2011-05-25]. Dostupný z URL:

<[http://www.atmel.com/forms/bitcloud\\_rzraven.asp?category\\_id=163&family\\_id=676&subfamily\\_id=2124&fn=dl\\_BitCloud\\_ZIGBIT\\_1\\_11\\_0.zip](http://www.atmel.com/forms/bitcloud_rzraven.asp?category_id=163&family_id=676&subfamily_id=2124&fn=dl_BitCloud_ZIGBIT_1_11_0.zip)>.

[9] HELP *BitCloud\_Stack\_API\_Reference* [online]. 03/2011 [cit. 2011-05-25]. Dostupný z URL: <[http://www.atmel.com/forms/bitcloud\\_rzraven.asp?category\\_id=163&family\\_id=676&subfamily\\_id=2124&fn=dl\\_BitCloud\\_ZIGBIT\\_1\\_11\\_0.zip](http://www.atmel.com/forms/bitcloud_rzraven.asp?category_id=163&family_id=676&subfamily_id=2124&fn=dl_BitCloud_ZIGBIT_1_11_0.zip)>.

[10] *ATmega640/1280/1281/2560/2561 Datasheet* [online]. 05/2011 [cit. 2011-05-25]. Dostupný z URL: <[http://www.atmel.com/dyn/resources/prod\\_documents/doc2549.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2549.pdf)>.

[11] *AT86RF212 Datasheet* [online]. 2/2010 [cit. 2011-05-25]. Dostupný z URL: <[http://www.atmel.com/dyn/resources/prod\\_documents/doc8168.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc8168.pdf)>.

### **Obrázky dostupné na internetu**

[12] *Modul ZigBit900*

[cit. 2011-05-25]. Dostupný z URL:

<[http://www.meshnetics.com/images/mb/p1\\_big.png](http://www.meshnetics.com/images/mb/p1_big.png)>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

USART Universal Synchronous Receiver and Transmitter

FFD Full Function Device

RFD Reduced Functionality Device

PAN Personal Area Network

ISM Industrial, scientific and medical

EEPROM Electrically Erasable Programmable Read-Only Memory

RAM random-access memory

HW Hardware

SW Software

BPSK Binary-Phase Shift Keying

Q-PSK Quadrature Phase Shift Keying

RX receive

I2C Inter-Integrated Circuit

IEEE Institute of Electrical and Electronics Engineers

SPI Serial Peripheral Interface bus

IEEE Institute of Electrical and Electronics Engineers

IEEE Institute of Electrical and Electronics Engineers

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance

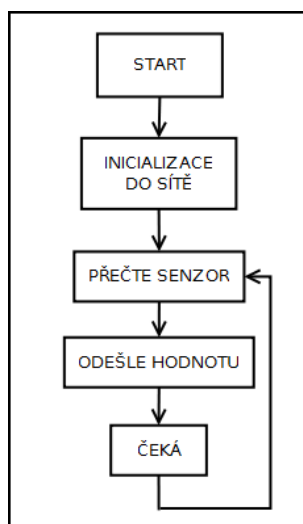
LED Light-Emitting Diode

# SEZNAM PŘÍLOH

<b>A</b>	<b>Vývojové diagramy měřících metod</b>	<b>40</b>
A.1	Periodické měření . . . . .	40
A.2	Trigger . . . . .	40
<b>B</b>	<b>Ukázky kódu</b>	<b>41</b>
B.1	Funkce pro triggerování . . . . .	41
B.2	USART . . . . .	42
B.2.1	Proměnné, přetypování, volání . . . . .	42
B.2.2	Funkce . . . . .	42
B.2.3	usartDescriptor . . . . .	42
<b>C</b>	<b>Relativní odchylky měřených průběhů od skutečné teploty</b>	<b>43</b>
C.1	Varianta A . . . . .	43
C.2	Varianta B . . . . .	44

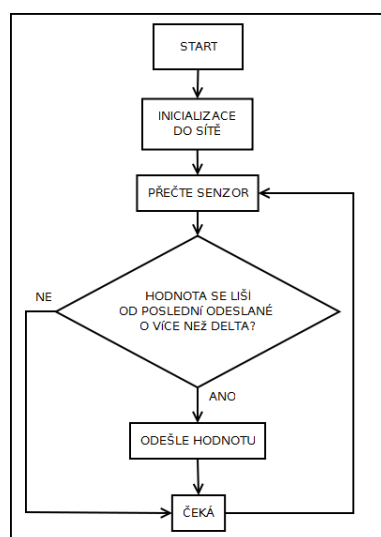
# A VÝVOJOVÉ DIAGRAMY MĚŘÍCÍCH METOD

## A.1 Periodické měření



Obr. A.1: Vývojový diagram algoritmu měřícího teplotu periodicky

## A.2 Trigger



Obr. A.2: Vývojový diagram algoritmu triggrovaného měření teploty

## B UKÁZKY KÓDU

### B.1 Funkce pro triggerování

```
static void sensorHandler(bool result, int16_t value)
{
    static int16_t delta = 1;    //variable of delta triggering of previous

    if (endDeviceState != END_DEVICE_STATE_MEASURING)
        return;

    // Temperature measured will be sent as an application message
    if (true == result)
    {
        // condition for trigger
        if (((oldSentValue-delta)<=value)&&(value<=(oldSentValue+delta)))
        {
            appMessageBuffer.msg.report.value = value;

            endDeviceState = END_DEVICE_STATE_SLEEP_PREPARE;
            appCloseLeds();
        }
        else
        {
            appMessageBuffer.msg.report.value = value;
            oldSentValue = value;

            endDeviceState = END_DEVICE_STATE_MESSAGE_SENDING;
        }
    }
    else
    {
        appMessageBuffer.msg.report.value = 0;
        endDeviceState = END_DEVICE_STATE_MESSAGE_SENDING;
    }

    SYS_PostTask(APL_TASK_ID);
}
```

## B.2 USART

### B.2.1 Proměnné, přetypování, volání

```
// variables for USART
static uint8_t msg[40] = "";
static uint8_t strLength = 0;

strLength = sprintf((char*)msg, "Retezec");
WriteUsartMessage(msg, strLength);
```

### B.2.2 Funkce

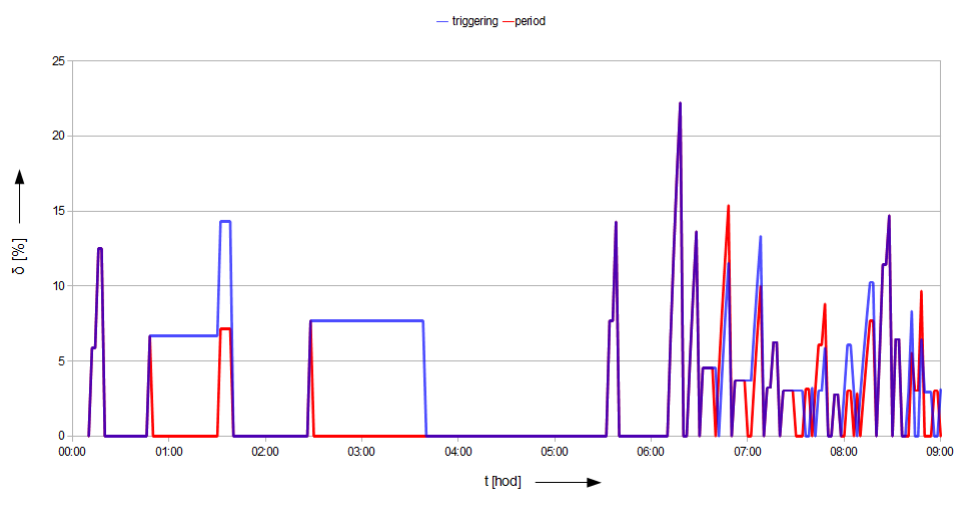
```
void WriteUsartMessage(uint8_t* uData, uint8_t uLength)
{
    USART_Write(&usartDescriptor, (void*)uData, uLength);
}
```

### B.2.3 usartDescriptor

```
usartDescriptor =
{
    .tty           = APP_USART_CHANNEL,
    .mode          = USART_MODE_ASYNC,
    .baudrate      = USART_BAUDRATE_38400,
    .dataLength    = USART_DATA8,
    .parity        = USART_PARITY_NONE,
    .stopbits     = USART_STOPBIT_1,
    .rxBuffer      = NULL,
    .rxBufferLength = 0,
    .txBuffer      = usartTxBuffer,
    .txBufferLength = APP_USART_TX_BUFFER_SIZE,
    .rxCallback    = NULL,
    .txCallback    = NULL,
    .flowControl   = USART_FLOW_CONTROL_NONE,
};
```

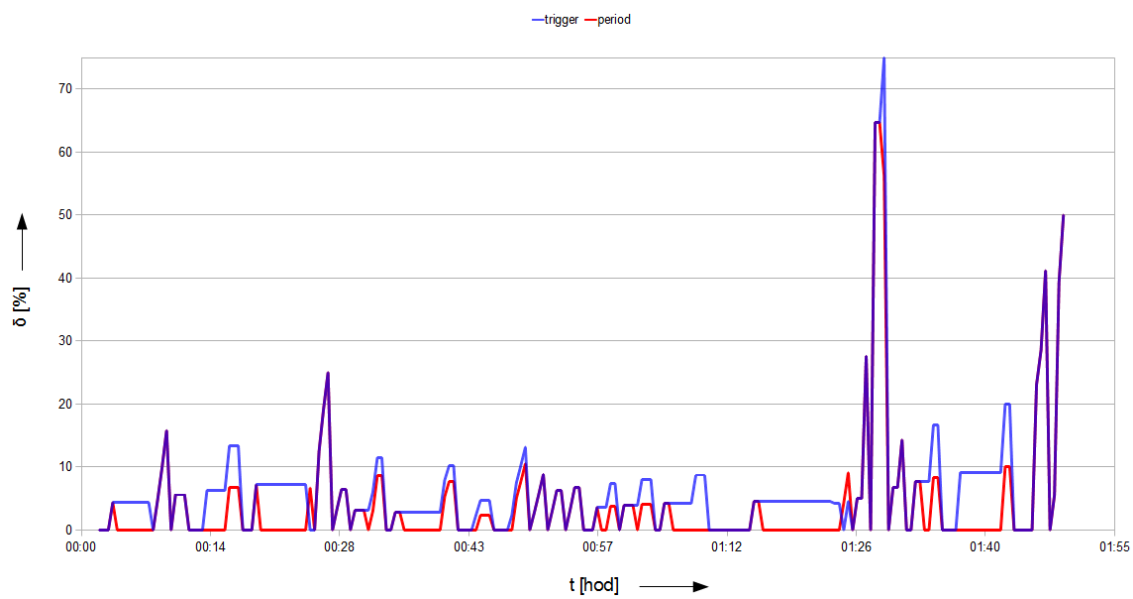
# C RELATIVNÍ ODCHYLKY MĚŘENÝCH PRŮ- BĚHŮ OD SKUTEČNÉ TEPLoty

## C.1 Varianta A



Obr. C.1: Odchylky od skutečné hodnoty?

## C.2 Varianta B



Obr. C.2: Odchylky od skutečné hodnoty?