

BRNO UNIVERSITY OF TECHNICKÉ V BRNĚ



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ ÚSTAV TELEKOMUNIKACÍ FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION DEPARTMENT OF TELECOMMUNICATIONS

ENERGETICKY EFEKTIVNÍ ZPRACOVÁNÍ DAT NA UZLECH BEZDRÁTOVÉ SENZOROVÉ SÍTĚ ENERGY AWARE DATA PROCESSING ON WIRELESS SENSOR NODES

DIZERTAČNÍ PRÁCE DOCTORAL THESIS

AUTOR PRÁCE AUTHOR Ing. VLADIMÍR ČERVENKA

VEDOUCÍ PRÁCE SUPERVISOR

doc. Ing. DAN KOMOSNÝ, Ph.D.

BRNO 2014

ABSTRAKT

Tato práce se soustředí na energetickou efektivitu dílčích aspektů při zpracování dat na uzlech bezdrátové senzorové sítě (WSN) a snaží se nalézt cesty pro další snížení spotřeby energie pro výpočetně náročné aplikace WSN. Práce nabízí detailní analýzu energeticky efektivních mesh protokolů, společně s přehledem současně dostupných hardwarových platforem. Na základě vlastního výzkumu je pak prezentována výpočetně výkonná senzorická jednotka, na níž je demonstrován navržený systém autonomního přenosu dat pro WSN. V neposlední řadě je pozornost věnována systému řízení, resp. implementaci operačního systému reálného času (RTOS), jenž by zohlednil přísné energetické nároky a podpořil svou funkcí energetické efektivity WSN systémů vede přes holistický přístup při návrhu softwarové a hardwarové architektury a tedy těsné součinnosti hardwaru a softwaru/ firmwaru. Výsledky uvedeného výzkumu nachází přímé uplatnění v průmyslu i akademické sféře, kde lze hardwarovou platformu společně s operačním systéme bez dalších úprav či znalosti fungování WSN využít. Systém autonomního přenosu dat pak nabízí cestu pro další vývoj snižování energetické náročnosti bezdrátových systémů.

KLÍČOVÁ SLOVA

Bezdrátové senzorové sítě, personální sítě, přenos dat, RTOS.

ABSTRACT

This thesis focuses on energy efficiency of particular aspects of data processing on wireless sensor nodes and deals with methods aiming to decrease energy consumption of wireless sensor network (WSN) applications requiring high processing power. The work presents comprehensive analysis of mesh protocols and available hardware platforms in terms of energy efficiency. A new energy efficient sensor node with high processing capability is presented and utilized to evaluate the proposed system for autonomous data transmission in WSN. Finally, an implementation of real-time operation system supporting mesh communication and very strict energy requirements through energy profiles is also presented. A valuable finding is that further increase of energy efficiency is only possible through a holistic approach in software and hardware architecture design, so that hardware and software/ firmware are tightly coupled. The output of this research might be applied in industry or academy field without necessity of any change or prior knowledge of WSN. The autonomous system of data transmission opens new research possibilities to decrease energy requirements of WNS.

KEYWORDS

Wireless sensor networks, personal networks, data transmission, RTOS.

ČERVENKA, Vladimír *Energeticky efektivní zpracování dat na uzlech bezdrátové senzorové sítě*: dizertační práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 123 s. Vedoucí práce byl doc. Ing. Dan Komosný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou doktorskou práci na téma "Energeticky efektivní zpracování dat na uzlech bezdrátové senzorové sítě" jsem vypracoval samostatně pod vedením vedoucího doktorské 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 doktorské práce dále prohlašuji, že v souvislosti s vytvořením této doktorské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom 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)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu disertační práce panu doc. Ing. Danu Komosnému Ph.D. za odborné vedení, konzultace a pomoc při vypracování této práce.

Brno

(podpis autora)



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 popsaný v této doktorské 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)





EVROPSKÁ UNIE EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Ú	vod		13
	Mot	ivace pro zvýšení výkonu senzorových uzlů	14
	Ene	rgetická efektivita	15
1	Sou	ičasný stav poznání	16
	1.1	Komponenty hardwaru senzorových uzlů	16
		1.1.1 Komunikační subsystém	17
		1.1.2 Výpočetní subsystém	18
		1.1.3 Senzorický subsystém	18
		1.1.4 Napájecí subsystém	19
	1.2	Hardwarové platformy	20
	1.3	Softwarová řešení pro WSN	20
		1.3.1 Operační systémy	21
		1.3.2 Komerční protokolové sady	23
	1.4	Distribuované výpočty na uzlech WSN	23
2	Cíle	e práce	27
3	Exp	perimentální analýza protokolových sad s podporou mesh komu-	
	nika	ace	29
	3.1	Efektivnost přenosu dat	30
	3.2	Způsob měření	30
		3.2.1 Vyhodnocení dat	32
	3.3	Výsledky měření	34
		3.3.1 Propustnost dat na aplikační vrstvě	34
		3.3.2 Zpoždění přeposíláním paketů	35
		3.3.3 Samoopravná schopnost sítě	38
	3.4	Závěry analýzy	40

4	Ene	ergeticky efektivní hardwarová platforma	43	
	4.1	Požadavky na hardwarovou platformu	43	
	4.2 Dostupná řešení			
	4.3	Vlastní návrh hardwarové platformy	52	
		4.3.1 Návrh výpočetního subsystému	52	
		4.3.2 Návrh zabezpečení	55	
		4.3.3 Komunikační subsystém	57	
		4.3.4 Návrh senzorického subsystému	57	
		4.3.5 Návrh napájecího subsystému	58	
	4.4	Hardwarová architektura vyvinuté platformy	59	
		4.4.1 Autonomní systém příjmu dat	60	
		4.4.2 Popis konvenčního způsobu	62	
		4.4.3 Demonstrace autonomního systému příjmu dat	64	
5	\mathbf{RT}	OS s podporou mesh komunikace a řízením spotřeby energie	66	
	5.1	Podpora RTOS na EE-WSN	67	
		5.1.1 CMSIS	67	
		5.1.2 Exkluzivní přístup a MPU	67	
	5.2	Výběr jádra	68	
		5.2.1 Modifikované jádro	68	
	5.3	Komunikační protokolová sada	69	
		5.3.1 Modifikace komunikační protokolové sady	70	
	5.4	Implementace jádra a protokolové sady	72	
6	Exp	perimentální ověření hardwarové platformy EE-WSN	74	
	6.1	Ověření výkonu	74	
	6.2	Spotřeba energie v režimu spánku	76	
	6.3	Přímé zpracování dat na uzlu WSN	79	
	6.4	Autonomní systém příjmu dat	81	
7	Exp	perimentální ověření systému EE-RTOS	83	
8	Dis	kuse	86	
9	Záv	τĂr	89	
-			60	
Li	terat	tura	91	
Se	znar	m zkratek 1	104	
Se	eznar	n příloh 1	107	

Α	Přehled dostupných hardwarových platforem	108
в	Elektrotechnické schéma EE-WSN B.1 Seznam součástek	110 . 114
\mathbf{C}	Ukázka analytického frameworku	115
Ρu	iblikace autora	119
Cι	ırriculum Vitæ	121

SEZNAM OBRÁZKŮ

1.1	Jednoduchý orientovaný graf pro algoritmus CLBD FFT, tzv. motýlek.	24
1.2	Schéma komunikace pro výpočet FFT ($N = 16$) algoritmu CLBD FFT.	25
1.3	Jednoduchý orientovaný graf pro algoritmus UPAD FFT, tzv. motýlek.	26
3.1	Srovnání spotřeby energie pro odeslání 1Byte dat	31
3.2	Struktura rámce IEEE 802.15.4	31
3.3	Proces měření	32
3.4	Záznam radiového prostředí v době měření	33
3.5	a) Topologie pro měření propustnosti. b) Způsob výpočtu a topologie	
	pro měření zpoždění při směrování. c) Topologie pro měření samoo-	
	pravné schopnosti sítě	35
3.6	Propustnost dat na aplikační vrstvě bez zabezpečení. Červené tóny	
	reprezentují BitCloud a $modré t \acute{o}ny$ pak Lightweight Mesh. Čsílo v le-	
	gendě indikuje velikost uživatelských dat v Byte a písmen o ${\cal A}$ značí	
	aktivní AppACK	36
3.7	Propustnost dat na aplikační vrstvě se zabezpečením. Červené tóny	
	reprezentují BitCloud a $modré t \acute{o}ny$ pak Lightweight Mesh. Čsílo v le-	
	gendě indikuje velikost uživatelských dat v Byte a písmen o ${\cal A}$ značí	
	aktivní AppACK	37
3.8	Zpoždění při směrování. Císlo na ose x představuje velikost uživatel-	
	ských dat v Byte a písmeno S značí aktivované zabezpečení	38
3.9	Srovnání datové propustností síťové a aplikační vrstvy. Císlo na ose	
	x představuje velikost uživatelských dat v Byte a písmeno S značí	
	aktivované zabezpečení.	39
3.10	Zpoždění samoopravného procesu. Horní část prezentuje zpoždění pro	
	nalezení alternativní cesty a spodní část celkový čas nutný k zotavení	
	sítě	40

3.11	Režie samoopravného procesu. Horní část prezentuje režii pro nalezení
	alternativní cesty a spodní část celkovou režii potřebnou k zotavení
	sítě
4.1	Přehled spotřeby energie v aktivním režimu
4.2	Přehled spotřeby energie v úsporných režimech
4.3	Srovnání výkonu DSP algoritmů v počtu hodinových cyklů CPU 54
4.4	Autentizační přístupový port AAP
4.5	Zapojení napájecí části
4.6	Hardwarová architektura vyvinuté platformy
4.7	Sběrnicový systém mikrokontroléru
4.8	Diagram uspořádání bloku DMA
4.9	Peripheral Reflex System
4.10	Časový průběh konvenčního přenosu dat
4.11	Časový průběh autonomního příjmu dat
5.1	Srovnání běhu RTOS bez tickless módu (horní část) a s tickless mó-
	dem (spodní část).
5.2	Sekvenční diagram funkce SYS_TaskHandler
5.3	Modifikovaný systém příjmu dat
5.4	Implementovaný systém autonomního příjmu dat
6.1	Prototypové jednotky ARM Cortex-M4F (horní část) a Cortex-M3
	(spodní část)
6.2	Srovnání spotřeby energie pro různé parametry FFT
6.3	Srovnání el. příkonu pro úsporný mód A
6.4	Srovnání el. příkonu pro úsporný mód B
6.5	Průběh spotřeby energie pro konvenční příjem dat
6.6	Průběh spotřeby energie pro autonomní příjem dat
7.1	Průběh spotřeby energie pro FFT aplikaci EE-RTOS
7.2	Průběh spotřeby energie pro FFT aplikaci EE-LWM
C.1	Analytický framework. Test propustnosti dat na aplikační vrstvě 116
C.2	Analytický framework. Test zpoždění směrováním
C.3	Analytický framework. Test samo opravné schopnosti sítě

SEZNAM TABULEK

3.1	Výchozí konfigurace LWM	32
4.1	Přehled komerčně dostupných výkonných platforem	45
4.2	Spotřebu energie a doba výpočtu algoritmu FIR	46
4.3	Spotřebu energie a doba výpočtu pro Dhrystone benchmark	46
4.4	Spotřebu energie a doba výpočtu pro Whetstone benchmark	47
4.5	Přehled RF transceiverů	49
4.6	Přehled současně využívaných výkonných platforem	50
4.7	Rychlosti výpočtů FPU jednotky.	55
4.8	Vlastnosti transceiveru AT86RF233	57
4.9	Elektrický příkon mikrokontroléru EE-WSN	63
6.1	Srovnání spotřeby energie pro různé parametry FFT	76
6.2	Náročnost algoritmu UPAD FFT, jeden hop	79
6.3	Náročnost algoritmu UPAD FFT, multi-hop	79
6.4	Náročnost operací UPAD FFT na senzorovém uzlu EE-WSN	80
6.5	Spotřeba energie pro výpočet FFT na uzlu EE-WSN	81
A.1	Přehled dostupných hardwarových platforem.	109

ÚVOD

Bezdrátové senzorové sítě (WSN, Wireless Sensor Network) jsou kategorií bezdrátových sítí, které jsou charakteristické nízkou přenosovou rychlostí, avšak dlouhým bezobslužným provozem (měsíce až roky). WSN síť je složena ze spolupracujících a velmi omezených zařízeních, které zaznamenávají, zpracovávají, uchovávají a odesílají data. Na rozdíl od WLAN (Wireless Local Area Network) jsou WSN sítě typicky samo-organizující (bez nutnosti konfigurace) a striktně optimalizovány pro konkrétní aplikaci. Nejčastěji využívají ke komunikaci radiových vln a jejich dosah může být od centimetrů až po kilometry [87]. Tato práce se soustředí výhradně na senzorové sítě založené na standardu IEEE 802.15.4, který je mj. nejrozšířenějším v této oblasti.

Samotná myšlenka využití vzájemně komunikujících uzlů sahá až k počátkům Internetu. Rozvoj tohoto vědního oboru však dovolil až příchod prvních hardwarových platforem jako je Mica¹. Od té doby prošly senzorické sítě bouřlivým vývojem, přičemž ty nejúspěšnější projekty jako Smart Dust [159] (později Dust Networks), či Contiki [152] (později Thingsquare) se rychle vymezily od akademické sféry a přešly do soukromé sféry. Právě tento rok, 2014, by se měl stát podle řady průmyslových expertů a firem (Silicon Labs, Atmel) rokem IoT (Internet of Things), kde hrají WSN systémy, společně s WLAN a DLNA [106], klíčovou roli.

WSN přitahují pozornost vědců napříč celého akademického spektra, protože dovolují uplatnit nejrozmanitější myšlenky takřka bez limitů. WSN má velmi bohaté uplatnění. Rozsáhlý přehled aplikací WSN je k dispozici v [115], reálnému nasazení jednotlivých WSN systémů se pak věnuje [143]. Jedná se především o geolokaci, snímání fyzikálních veličin v průmyslu a zemědělství, sledování namáhání konstrukcí staveb, komunikaci mezi automobily (VANET) a inteligentní domácnosti. Speciálním případem je zdravotnictví WBAN (Wireless Body Area Network) nebo také WBSN (Wireless Body Sensor Network), které je pro své vysoké nároky náročnou

¹Rok 2001, University of California, Berkeley.

oblastí. Úspěšný návrh hardwarové platformy je pak podmíněn znalostí z oborů bezdrátové komunikace, zpracování signálů, kryptografie, senzorických systémů aj. [53].

Tato práce se soustředí na energetickou efektivitu dílčích aspektů při zpracování dat na uzlech bezdrátové senzorové sítě a snaží se nalézt cesty pro další snížení spotřeby energie pro výpočetně náročné aplikace WSN. Po úvodu do problematiky v kap. 1 jsou definovány cíle této práce (kap. 2). Následně je přistoupeno k detailní analýze komunikačních protokolů, zajištujících mesh komunikaci (kap. 3). Pro účely demonstrace a ověření navržených řešení je v kap. 4 prezentována hardwarová platforma s vysokým výpočetním výkonem společně s návrhem systému autonomního přenosu dat. Kap. 5 je věnována systému řízení, resp. návrhu implementace operační systému, jenž by zohlednil přísné energetické nároky a podpořil svou funkcí energetickou efektivitu celého systému. Kap. 6 a 7 pak experimentálně ověřuje navržené řešení předchozích kapitol, přičemž kap. 8 nabízí diskusi těchto výsledků.

Motivace pro zvýšení výkonu senzorových uzlů

Bohužel současné senzorické uzly nejsou pro výpočetně náročné aplikace vhodné [84], a proto je pro ně díky omezeným výpočetním a paměťovým prostředkům charakteristické střídání provádění intenzivních výpočtů a běžné funkce. Nedostatek výpočetního výkonu současných WSN platforem se stává hlavním limitujícím faktorem [3, 74, 85]. Jedinou možností je tak využití některé z výpočetně výkonných hardwarových platforem (Imote2 [111], Sun SPOT [117] či MedusaMK-2 [136]) na úkor spotřeby energie a pořizovací ceny [84, 165].

Mezi WSN oblasti typicky vyžadující vysoký výpočetní výkon a zároveň velmi nízkou spotřebu energie, z důvodu co nejdelšího bezobslužného provozu, patří oblast lékařství (WBSN), zabezpečení a perzistentního monitoringu prostředí a konstrukcí staveb. Zde je potřeba signálového zpracování s vysokým rozlišením (komprese, filtrování snímaného signálu) či náročných výpočtů asymetrické kryptografie. Kupříkladu u algoritmů pro detekci srdeční činnosti se využívá dolní propusti realizované FIR (Finite Impulse Response) filtry a u respiračních algoritmů pak pásmové propusti realizované IIR (Infinite Impulse Response) filtry [99]. Zvýšené paměťové a výpočetní nároky mají i aplikace využívající standard 6LoWPAN [69] s podporou RPL [67], nebo klasicky brány WSN sítí.

Zabezpečení a energetická efektivnost jsou přitom nejdůležitějšími stavebními kameny reálně nasazených WSN aplikací [79]. Vysoký výpočetní výkon a nízké energetické požadavky jsou ale při návrhu platformy zcela protichůdné vlastnosti [127], a proto musí být nutně zvoleno kompromisní řešení. Vysoký výpočetní výkon však znamená rychlejší zpracování dané úlohy a systém tak může více času strávit v úsporném režimu, což přímo ovlivňuje jeho energetickou efektivitu.

Energetická efektivita

Tento pojem není v technickém pojetí přesně vymezen a je proto potřeba jej pro tuto práci definovat. Vždy však záleží, v jakém kontextu bude energetická efektivita posuzována, protože na rozdíl od účinnosti záleží i na postoji posuzovatele.

Obecně lze říci, že energetická efektivita je poměrné vyjádření energie využité na užitečnou práci a celkové energie vstupující do systému. Např. v [158] definovali energetickou efektivitu ϕ určitého procesu P jako vztah mezi zesílením G a cenou C:

$$\phi_P = \frac{G_P}{C_P}; \quad G \in \aleph; C, \phi \in \Re; C \ge 1; 0 \le \phi_P \le 1.$$
(1)

Nicméně tato definice není pro účely této práce příliš vhodná. V případě srovnání RF (Radio Frequency) transceiverů definujeme energetickou efektivitu pro aktivní stav a režim spánku. První definice vyjadřuje poměr užitečně vyzářené energie o daném výkonu (0 dBm) k celkově vynaložené energii. Vzhledem k způsobu využití RF transceiverů ve WSN je nutné uvažovat i spotřebu energie v režimech spánku, čímž lze dosáhnout uceleného obrazu energetické efektivity. Přesná střída obou režimů je však závislá na konkrétní aplikaci.

Pro srovnání energetické efektivity mikrokontrolerů je sledována spotřeba energie na vykonání jednotkové aktivity. I zde je pro úplnost nutné spotřebu v aktivním režimu vztáhnout ke spotřebě v režimu spánku a to v poměru střídy konkrétní aplikace.

1 SOUČASNÝ STAV POZNÁNÍ

Vzhledem k tomu, že senzorické uzly musí být levné a malých rozměrů [154], není možné je vybavit velkým zdrojem elektrické energie. Situace je dále ztížena skutečností, že jednotlivé uzly musí pracovat bezobslužně z důvodu často obtížného přístupu či velkému počtu zařízení. Ve výsledku pak zařízení musí pracovat za pomoci alternativních zdrojů energie (solární články, vibrace, rádiové vlny, atp. [42]) a vlastní baterie řádově roky. Z uvedeného vyplývá, že energetické požadavky tvoří klíčový parametr návrhu celé sítě. Spotřeba elektrické energie proto byla ustanovena jako největší omezení pří návrhu bezdrátové senzorové sítě [121].

Jak bude dále popsáno, největší úspory energie je dosaženo za pomoci úsporných MAC (Medium Access Control) protokolů [89], které mohou snížit aktivitu hardwaru pod 1 % celkové provozní doby uzlu [87]. Senzorické uzly tak pracují jen během velmi krátkých časových intervalů, přičemž se snaží ve velké míře využít dostupných režimů spánku. Proto je v oblasti WSN důležitá i minimalizace spotřeby během režimu spánku. Ta je u aplikací s nízkým datovým tokem dominantní složkou spotřeby a hlavním faktorem limitujícím dobu životnosti uzlů [87].

1.1 Komponenty hardwaru senzorových uzlů

Základními požadavky na hardware senzorických uzlů je malá velikost, nízká cena a schopnost pracovat řádově roky na malý akumulátor (standardně 2 NiMh AA baterie). Díky pokroku technologií je možné integrovat potřebné elektrické obvody do samostatných čipů a radikálně tak snížit velikost i cenu uzlů. Architekturu hardwaru pro WSN lze rozdělit do čtyř samostatných subsystémů:

- a) komunikační subsystém,
- b) výpočetní subsystém,
- c) senzorický subsystém,

d) napájecí subsystém.

1.1.1 Komunikační subsystém

Komunikační subsystém je složen z rádio-transceiveru a antény, zajišťující bezdrátovou komunikaci s ostatními uzly. Zároveň se jedná o subsystém s největšími energetickými nároky, neboť komunikace má největší podíl na celkové spotřebě [52, 156]. Samotnou komunikaci lze uskutečnit v několika frekvenčních spektrech. Krom rádiových vln (RF) je přenos možné realizovat pomocí optických [16] či akustických vln [13]. Přestože optické transceivery dosahují vyšší energetické účinnosti [87], vyžadují přímou viditelnost, což je pro oblast WSN s více uzly takřka nemožné. V případě přenosu pomocí akustických vln se vývoj soustředí především na podvodní komunikaci [51]. Zde je limitující především šířka frekvenčního pásma, vysoká chybovost a zpoždění šířené vlny [70]. Navzdory vyšší spotřebě el. energie je díky všesměrovému přenosu a vyspělosti RF technologie přenos radiovými vlnami výhodnější. Charakter šíření a vznik interferencí je pak determinován pracovní frekvencí systému.

Standard IEEE 802.15.4. je nejrozšířenějším standardem bezdrátových senzorových sítí, jehož využívají další standardy jako ZigBee [167], MiWi [105], nebo 6LoW-PAN [69]. Standard IEEE 802.15.4-2003 obecně definuje vrstvu přístupu k médiu a několik fyzických vrstev, jejichž rozšíření dále popisují standardy IEEE 802.15.4a [57], IEEE 802.15.4c [59]a IEEE 802.15.4d [60]. Rozšíření označované jako IEEE 802.15.4-2006 (IEEE 802.15.4b [58]) pak nabízí mj. zvýšení přenosové rychlosti, nová modulační schémata a definici metody Parallel Sequence Spread Spectrum (PSSS). Poslední rozšířením je pak IEEE 802.15.4e-2012 [61]. IEEE 802.15.4 definuje v rámci bezlicenčních ISM (Industrial Scientific, Medicine) pásem dvě fyzické vrstvy využitelné v ČR. Jedná se o pásmo v oblasti 868 MHz umožňující přenos 20 kb/s (BPSK) a frekvenční pásmo 2,4 GHz, kde je maximální přenosová rychlost definována na 250 kb/s [58].

Rádiové transceivery mohou pracovat v jednom ze čtyř módů: příjem, vysílání, stav nečinnosti a režim spánku. Nejvyšší spotřeba je během příjmu a vysílání. V době nečinnosti je pak většina obvodů vypnuta, až na hlavní oscilátor. Ten se vypíná až v režimu spánku. Přehled současně využívaných nízko příkonových rádiotransceiverů je společně s jejich parametry uveden v tab. 4.5.

1.1.2 Výpočetní subsystém

Pro výpočetní subsystém se v oblasti WSN tradičně využívá nízkopříkonových mikrokontrolerů (MCU, Micro Controller Unit), které nabízí přijatelný výkon pro většinu aplikací, velmi nízkou spotřebu a především nízkou cenou ve srovnání s běžnými architekturami (FPGA, DSP, x86-64). Současná programovatelná hradlová pole (FPGA, Field Programmable Gate Array) jsou navíc svou spotřebou energie pro oblast WSN nevhodná. Efektivním řešením mohou být tzv. ASIC (Application Specific Integrated Circuits) obvody, které jsou zhotoveny pro specifickou aplikaci a dosahují tak vysoké výpočetní a energetické efektivity. Jejich návrh a produkce jsou však cenově výhodné až pro velkoobjemové produkce. Jelikož mezi základní požadavky bezdrátových senzorických sítí patří nízká pořizovací cena, bude pozornost věnována právě MCU.

Mikrokontroléry obsahují procesor, paměti RAM a ROM, časovače/ čítače a vstupně výstupní periferie. Nejčastěji se jedná o 8 až 16-ti bitové MCU s omezeným počtem instrukcí (27 instrukcí pro MSP430 [121]) a jen několika kB SRAM paměti. Těmi nejvyužívanějšími mikrokontroléry v oblasti WSN jsou ATmega128L [9] (Atmel Corporation), MSP430 [150] (Texas Instrument), 8051 [65] (Intel) a PIC18 [104] (Microchip). Nutno podotknout, že výkon je ovlivněn mimo jiné i konkrétní instrukční sadou.

V posledních několika letech se v průmyslu ve zvýšené míře uplatňuje myšlenka využití více procesorů, potažmo využití více jádrových procesorů. Některé vestavěné systémy dokonce využívají heterogenních jader, jejichž příkladem může být optimalizační technologie ARM big.LITTLE. Ta zjednodušeně využívá synergie vysokovýkonného procesoru (např. ARM Cortex-A17) s energeticky efektivním procesorem (např. ARM Cortex-A7). Tato kombinace pak dokáže ušetřit až 75% energie CPU a navýšit výkon o až 40% u aplikací s vysokým počtem vláken [5]. Technologie ARM big.LITTLE se s výhodou využívá v chytrých telefonech nebo tabletech (např. Samsung Galaxy S4). Při zachování stejné filozofie pak např. výrobce NXP vytvořil v oblasti výkonově omezených zařízení asymetrický dvou-jádrový mikrokontrolér LPC4300 [114], kombinující jádra ARM Cortex-M4F a ARM Cortex-M0. Více je výběru vhodného MCU věnováno v kapitole 4.3.1.

1.1.3 Senzorický subsystém

Účelem senzorického subsystému je interakce s okolím. Může jít o sledování určitého fyzikálního jevu, popř. o přímé ovlivnění svého okolí. K dispozici je široký výběr senzorů, které se dělí na analogové a digitální. Analogové senzory vyžadují AD převodník, mají větší fyzické rozměry a vyšší spotřebu. Oproti tomu digitální senzory jsou malé, energeticky efektivní, jsou schopny dodávat data přímo v patřičných jednotkách pomocí datové sběrnice, přičemž jsou již z výroby předkalibrovány. Z důvodu úspory energie jsou tyto senzory po většinu času také v režimu spánku a zapínají se jen v okamžiku měření.

1.1.4 Napájecí subsystém

Napájecí subsystém je zodpovědný za zajištění a regulaci potřebné energie. Vzhledem k přísným energetickým nárokům WSN musí být subsystém vysoce efektivní, dokázat dodávat krátkodobě špičkový proud desítky miliampér a přitom mít minimální klidový proud [87].

Tento subsystém je typicky složen z vhodného akumulátoru (alkalický, Li-Pol, Li-ion), regulátoru napětí a volitelně může obsahovat i obvod pro dolování energie (EH, Energy Harvesting) z okolních zdrojů energie.

Oblast EH v posledních letech koncentruje pozornost řady výzkumných projektů [42, 100, 132, 135]. Nejvíce využívanými zdroji energie pro obvody EH je světlo, vibrace, pohyb a proudění vzduchu. Solární články patří tradičně k nejvyužívanějším, díky velmi nízké ceně a relativní vyspělosti technologie [156]. Piezoelektrická konverze patří mezi nejúčinnější metody získávání energie z vibrací, přičemž maximální příkon se dnes pohybuje kolem 60 µW [135]. Rapidní vývoj v oblasti superkapacitorů umožnil jejich využití i pro uzly WSN. Účelem je překlenout období, kdy alternativní zdroje nemohou dodávat el. energii (oblačnost nebo noc, v případě solárních článků). Jejich hlavní výhodou oproti klasickým akumulátorům je relativní teplotní nezávislost, tedy malá změna napětí s teplotou, jak bylo dokázáno v [132]. Na druhou stranu se potýkají s únikem energie a vysokou cenou. Díky stále nízké účinnosti konverze energie však lze EH využít jen za cenu větších rozměrů a vysoké pořizovací ceny.

Na druhou stranu je potřeba říci, že ačkoliv je WSN jednou z nejslibnějších aplikací EH, nejde o samostatnou oblast vývoje WSN, ale spadá spíše do vývoje analogové techniky. Přesto populární články prezentují "Energeticky soběstačné uzly" [75] a "Bez-bateriové senzorové uzly" [120, 164], které pouze využívají specializované obvody, tzv. energy harvestery, pro získávání alternativní energie, ale neřeší podstatu problému – energetickou efektivitu bezdrátového systému. Tedy jen změnu zdroje napájení. Lze se tak setkat se zcela energeticky neefektivními senzorovými uzly jako Mica2 doplněnými o solární panel [47] a podobně.

1.2 Hardwarové platformy

Přestože preferovaná oblast využití výstupů této práce je WBSN, pozornost je v této práci věnována tzv. generickým hardwarovým platformám. Vzhledem k šíři aplikační domény WSN je dokonce nemožné porovnávat platformy vyvinuté pro různé specifické aplikace. Tyto univerzální jednotky jsou vyvíjeny převážně pro vývoj v oblasti sítí a pro monitoring životního prostředí. Většinou bývají doplněny o základní senzory např. teploty, světla, tlaku, atp. nebo konektorem pro rozšiřující desky. Výhodou takovýchto řešení je finanční efektivita výroby i nákupu, neboť je možné stejný senzorický modul využít v rozdílných aplikačních oblastech. Přehled populárních platforem společně s jejich vlastnostmi je uveden v tab. A.1 v příloze. Je zřejmé, že mikrokontroléry ATmega128x a TI MSP430 patří v oblsti WSN mezi ty nejvyužívanější. Rozsáhlý přehled současných senzorových uzlů lze nalézt např. v [15, 45, 77], komerčně dostupné v [162]. Přehled platforem využívaných v průmyslu je uveden např. v [96].

1.3 Softwarová řešení pro WSN

Bezesporu nejvíce pozornosti je ve vývoji WSN věnováno softwarovým řešením. Mezi hardwarovými platformami nebývaly valné rozdíly a 8/16-ti bitové mikrokontroléry dosahují srovnatelného výkonu i energetické náročnosti. Proto se vývoj soustředí na softwarové techniky, kde hraje nalezení maximální energetické efektivity WSN systému primární roli. Zlepšení energetické efektivity lze přitom pomocí softwaru¹ dosáhnout na všech vrstvách WSN modelu, počínaje linkovou.

Jelikož komunikační subsystém spotřebovává největší část energie celého uzlu, je přirozené, že největší energetické úspory lze dosáhnout využitím energeticky efektivních MAC protokolů. Jde o mechanismy, které aktivují radiovou část jen po dobu nezbytně nutnou. Po dobu deaktivace je však znemožněn příjmem zpráv od okolních uzlů a efektivnost přenosu klesá. Na výběr je dnes nepřeberné množství návrhů a několik desítek funkčních implementací [12, 55, 80, 82, 147].

V zásadě se MAC protokoly rozdělují na synchronní, asynchronní, přičemž u asynchronních metod se dále rozlišuje, zda je přenos dat iniciován odesílatelem (LPL, Low Power Listening), nebo příjemcem (LPP, Low Power Polling/ Probing). Mezi nejstarší implementace asynchronních LPL metod patří B-MAC [129], na jehož základech staví např. X-MAC[18], Contiki-MAC [144], CMAC [94] nebo WiseMAC [34].

 $^{^1}$ Označení software a firmware jsou v rámci usnadnění výkladu zaměnitelná.

Mezi populární asynchronní LPP metody patří RI-MAC [146], PW-MAC [149], RC-MAC [54] či A-MAC [32]. Synchronní metody tradičně reprezentuje S-MAC [163], T-MAC [28], z nichž vychází např. RMAC [31], DW-MAC [145] a DMAC [98], nebo komerčně úspěšný TSMP [128].

Je vhodné zmínit, že přes velké množství navržených směrovacích protokolů pro WSN, jejichž přehled je dostupný v [83, 95, 118] se dnes prakticky využívá převážně AODV (Ad hoc On-Demand Distance Vector) [123] a RPL [67]. AODV má v oblasti WSN díky ZigBee Pro poměrně dlouhou historii, přičemž RPL byl definován IETF (Internet Engineering Task Force) teprve v roce 2010. AODV patří mezi tzv. reaktivní protokoly, jejichž cesty jsou vytvářeny pouze na vyžádání a využívá k tomu tradičních směrovacích tabulek s časově řízenou údržbou. RPL vznikl za součinnosti průmyslu i akademické sféry v rámci pracovní skupiny ROLL, jako primární směrovací protokol pro IPv6 a je favorizovaným řešením, jenž se dočkalo několika vylepšení, např. [2, 134, 138]. Jednotnost ve výběru směrovacího protokolu přinesou až závěry nově vytvořené pracovní skupiny IEEE 802.15.10 L2R (TG10) [62].

Z pohledu aplikační vrstvy lze zajistit zvýšení efektivity systému tradičně pomocí kompresních [24, 78, 131, 161] a agregačních technik [17, 73, 166]. Tyto techniky ale nelze implementovat plošně. Je třeba tyto zvážit dle individuálních možností konkrétní aplikace.

1.3.1 Operační systémy

V minulosti byly pro uzly WSN vytvářeny velmi jednoduché aplikace, např. pro sběr dat, které byly v rámci celé sítě shodné. Sítové protokoly a aplikace byly významně spjaty a společně s protokoly nižších vrstev tvořily monolitický celek. Takový software byl využitelný pouze pro specifické aplikace bez možnosti přidání nové funkcionality.

Díky stále rostoucí energetické efektivitě WSN je možné provozovat tyto sítě po relativně dlouhý čas – řádově roky. Dnešním trendem je naopak využití senzorové sítě pro širou škálu aplikací [30]. To však vyžaduje použití vyspělejšího přístupu v podobě operačních systémů (OS).

Úlohou každého operačního systému je spolehlivá podpora běhu aplikací a poskytnutím pohodlné a bezpečné abstrakci hardwarových prostředků. Operační systémy pro uzly WSN jsou samozřejmě méně komplexní než více účelové operační systémy vestavěných zařízení. Postrádají jakékoliv uživatelské prostředí a jejich funkce jsou většinou omezovány pouze na jedinou aplikaci. Důvodem jsou specifické požadavky na běh aplikací a omezené výpočetní a energetické možnosti. Operační systémy WSN se typicky dělí na *událostně řízené* (event-driven) a *více vláknové* (multithreaded). Jejich vlastnosti jsou následující.

Událostně řízený model: Každá akce je vyvolána určitou událostí. Kritické události nejsou zpracovány okamžitě. Vývoj aplikace je velmi náročný a vyžaduje propracovaný návrh, přičemž je využíváno stavových automatů, popř. tzv. super smyček. Pro velmi jednoduché aplikace má však tento model stále své opodstatnění.

Více vláknový model: Nabízí významné vylepšení redukcí složitosti kódu. Koncepčně jsou úlohy zpracovány paralelně. Změna kontextu sice vyžaduje větší paměťové nároky, nicméně umožňuje kritickým událostem zpracování v reálném čase. Navíc významně usnadňuje programování aplikací a jejich souběh.

Dnes nejrozšířenější operační systémy pro bezdrátové senzorové sítě jsou TinyOS [91] a Contiki OS [152] [133]. Oba operační systémy jsou představiteli událostně řízeného modelu a jsou využívány převážně na akademické půdě a v oblasti vývoje. Jejich kód je veřejně přístupný, což dovoluje snadné úpravy a implementaci nových protokolů. Mezi populární více vláknové systémy pro WSN patří např. Mantis [14], Nano-RK [88] či LiteOS [22].

TinyOS patří mezi monolitické systémy a byl vytvořen pro výzkumné účely v oblasti WSN, avšak jeho tvůrci již vývoj ukončili. Přesto patří mezi nejvyžívanější a existuje celá řada protokolů pro tento OS, včetně µIP a základní podporou µIP6. Je však implementován proprietárním jazykem nesC, což stěžuje práci s tímto OS. Jelikož jde o událostně řízený systém, aplikace obsahuje obslužné rutiny událostí a jednotlivé úlohy jsou sémanticky vykonávány až do jejich konce.

Contiki OS využívá modulární architektury a stejného událostně řízeného modelu jako TinyOS, ale navíc nabízí tzv. lehkou implementaci vláken – protothreads. Tedy jakousi abstrakci více vláknového programovacího modelu. Vlákna jsou pak spouštěna na základě příchozí události. Dále ContikiOS nabízí souborový systém Coffee a plnou podporu IPv4 a IPv6 přes µIP a µIP6 stack [152].

Nano-RK obsahuje preemptivní plánovač založený na prioritách. Jádro využívá monolitické architektury a statického časového frameworku [43], což znamená, že ověření časových limitů je prováděno off-line. Přesto patří jako jediný z uvedených operačních systémů do skupiny RTOS (Real-Time Operating System). Nano-RK nabízí několik implementací MAC protokolů (B-MAC, PCF TDMA, RT-Link), nicméně používá jen statickou správu paměti.

Operační systém Mantis také obsahuje preemptivní plánovač založený na prioritách a pro interní komunikaci nabízí podporu semaforů a mutexů. Na rozdíl od Nano-RK využívá vrstvové architektury a dynamické alokace paměti, nicméně nedisponuje podporou ochrany paměti, což může vést k fatálnímu selhání systému.

Největší nevýhodou zmíněných systémů, společně např. s operačním systémem SOS je jejich krátký životní cyklus, neboť byly vyvíjeny na akademické půdě a jejich vývojáři již graduovali. Proto nejsou ani nové hardwarové architektury podporovány.

1.3.2 Komerční protokolové sady

V komerční sféře je nejvíce využíváno ZigBee Pro [167] technologie, která představuje ucelené řešení, zajištující sestavení, popř. zotavení sítě, směrování mezi uzly a nabízí snadné řešení pro tvorbu uživatelských aplikací. Zdrojové kódy jednotlivých implementací jsou však nepřístupné. Běžně tak uživatelské aplikace využívají pouze API a konfiguračních souborů. Z těchto důvodů je protokolová sada ZigBee Pro nevhodná pro další vývoj protokolů, nicméně nabízí stabilní a ověřenou technologii a především vzájemnou kompatibilitu mezi různými výrobci skrze standardizované profily.

Novinkou je proprietární protokolová sada Lightweight Mesh (LWM) [11] společnosti Atmel. LWM je velmi jednoduchá protokolová sada, specificky vyvinutá pro výpočetně a paměťově omezená zařízení. Typická aplikace LWM tak vyžaduje přibližně 8 kB Flash a 4 kB RAM paměti [11]. Současná verze 1.2.1 již podporuje vedle proprietárního způsobu i AODV směrování. Oproti protokolu ZigBee Pro však není nutné pro start sítě vyhradit konkrétní uzel, nejsou zde periodicky rozesílány služební informace a síť není formována vztahy rodičů a potomků.

1.4 Distribuované výpočty na uzlech WSN

Z důvodu zmíněných limitujících výpočetních a paměťových prostředků dostupných platforem jsou výzkumníci nuceni hledat cesty, jak výpočetně náročné operace vykonat na nízko-příkonových procesorech. Nutno podotknout, že tato omezení dala za vznik nesmírně efektivním řešením. Z nich jmenujme například výpočty eliptických křivek na osmi bitových MCU ATmega128 [81], implementace klasifikační metody pomocí kondenzačních k-dimenzionálních stromů [72].

Postupem času se vytvořila samostatná oblast vývoje WSN, zaměřující se na takzvané *distribuované výpočty* (Distributed Computing), nebo také paralelní výpočty. Tradičně jsou algoritmy signálového zpracování v oblasti WSN realizovány centralizovaným způsobem, kde jsou snímaná data z jednotlivých uzlů sítě shromážděna na centrálním prvku (např. PC) na němž jsou dále zpracovány. Dokladem je například sledování stavu indukčních motorů pomocí frekvenční analýzy rychlou



Obr. 1.1: Jednoduchý orientovaný graf pro algoritmus CLBD FFT, tzv. motýlek.

Fourierovou transformací (FFT, Fast Fourier Transform) snímaných signálů [86, 97]. V obou příkladech jsou surová data odesílána na centrální počítač a analýza byla provedena v programu Matlab. Distribuované metody naopak rozprostírají výpočty mezi všechny zúčastněné uzly. Symbolickým příkladem mohou být např. implementace umělých neuronových sítí, kde bezdrátové senzorové uzly tvoří distribuovanou výpočetní platformu [137]. Autoři [126] hovoří o návrhu sítě s 6144 neurony, v [48] informovali dokonce o implementaci 10.000 neuronů. Hybridním řešením jsou pak např. DPF (Distributed Particle Filtering) algoritmy, které zasílají lokální statistiky ke zpracování na centrální jednotku [27, 46].

Jednou z oblastí distribuovaných výpočtů jsou algoritmy distribuovaného signálového zpracování DDSP (Distributed Digital Signal Processing) [25]. Vyjděme z předpokladu, že běžný distribuovaný FFT algoritmus typu DIT (Decimation in Time), např. Conventional Load Balanced Distributed FFT [20], jednorozměrného signálu o N prvcích vyžaduje $N \log_2 N$ operací komplexního násobení, $N \log_2 N$ sčítání a je rozdělen do $\log_2 N$ stupňů. Mezi každým stupněm pak dvojce senzorových uzlů svou komunikací formují tzv. motýlky, které vycházejí z grafu signálových toků. Způsob komunikace jednotlivých uzlů zachycuje obr. 1.1, kde W_N^r představuje činitel $e^{-j\frac{2\pi}{N}r}$, tedy

$$W_N^r = e^{-j\frac{2\pi}{N}r}. (1.1)$$

V každém z n stupňů tak každý senzor odesílá a přijímá jeden paket. Schéma komunikace pro výpočet 16-ti bodové FFT pomocí 16-ti senzorových uzlů algoritmu Conventional Load Balanced Distributed (CLBD) FFT opět vychází z grafu signálových toků a je uvedeno na obr. 1.2. Jde vidět, že popsaný algoritmus vyžaduje $N \log_2 N$ přenesených paketů.

Vylepšením tohoto algoritmu je např.[1], nebo Unbalanced Power-Aware Distributed (UPAD) FFT [25], kde je odstraněna redundance na úkor nerovnoměrného rozložení zátěže napříč sítí. Provedená změna je patrná z obr. 1.3. Opět se zde přenáší dva pakety, na obou uzlech dochází k sčítání, ale pouze jeden uzel z páru (p2 na obr. 1.3) provádí komplexní násobení. Schéma komunikace pro výpočet 16-ti bodové FFT pomocí 16-ti senzorových uzlů je stejné jako na obr. 1.2. Uvedený algoritmus



Obr. 1.2: Schéma komunikace pro výpočet FFT (${\cal N}=16)$ algoritmu CLBD FFT.



Obr. 1.3: Jednoduchý orientovaný graf pro algoritmus UPAD FFT, tzv. motýlek.

tedy opět vyžaduje $N \log_2 N$ sčítání, $N \log_2 N$ přenesených paketů, ale jen $\frac{N}{2} \log_2 N$ operací komplexního násobení.

Dalším vylepšením je pak například algoritmus Power-Time Efficient Distributed FFT [20], který odstraňuje nevýhodu předcházejícího UPAD FFT, tedy nerovnoměrného rozložení zátěže mezi uzly. Pro prvních $\frac{N \log_2 N}{2}$ stupňů jde principiálně o algoritmus UPAD FFT, ale poté jsou data distribuována takovým způsobem, že uzly, které v předchozích stupních provedli více násobení se párují s uzly, které provedli méně násobení. Tím se docílí rozložení zátěže do celé sítě.

Situace se však komplikuje, pokud všechny uzly nejsou ve vzájemném dosahu – což je v praxi skoro vždy. Uvažujme nyní případ multi-hop sítě. Nechť h značí maximální počet uzlů v komunikačním dosahu libovolného uzlu a k číslo stupně. Potom počet potřebných přenosů paketů stupně k je roven $\left[\frac{2^{(k-1)}}{h}\right]$. Celkový počet přenesených paketů je tak pro UPAD FFT algoritmus v multi-hop síti dán vztahem [20]:

$$N \times \sum_{k=1}^{n} \left[\frac{2^{k-1}}{h} \right] \approx \frac{N(N-1)}{h}.$$
(1.2)

Tyto informace slouží jako úvod do problematiky distribuovaných výpočtů a později, v kap. 6.3, budou uvedené techniky využity ke srovnání s přímým přístupem výpočtů na uzlech WSN, konkrétně na nově vyvinuté platformě EE-WSN.

2 CÍLE PRÁCE

Bezdrátové senzorové sítě se charakterizují využíváním levných a velmi omezených senzorových uzlů. Především z pohledu výpočetního výkonu, paměťového prostoru a výdrže baterie. Typicky je tak z důvodu nízké spotřeby energie senzorový uzel vybaven 8 až 16-ti bitovým mikrokontrolérem s taktovací frekvencí do 8 MHz. Samozřejmě dnes existují i platformy disponující dostatečným výpočetním výkonem jako iMote2 [111], Medusa MK-2 [136], Sun SPOT [117] či Lotus [101], ale jejich spotřeba energie v aktivním i úsporném režimu je pro potřeby WSN příliš vysoká a mohou tak pracovat typicky jen po velmi omezený čas¹. Cílem této práce je mj. dokázat, že senzorové uzly mohou disponovat dostatečně vysokým výpočetním potenciálem, s hardwarovou podporou DSP algoritmů, výpočtů s čísly s pohyblivou řádovou čárkou a přitom minimálně zachovat energetické a finanční požadavky stávajících systémů.

Jestliže budou jednotlivé uzly bezdrátové senzorové sítě disponovat dostatečným výpočetním výkonem, může být upuštěno od distribuovaných a centralizovaných výpočtů, např. [1, 20, 21, 25] a [86, 97]. Myšlenka přímého zpracování dat přímo na uzlech senzorové sítě není nová, nicméně vždy chyběly prostředky pro její uplatnění v plném rozsahu. Dalším cílem je tak dokázat, že bezdrátové uzly mohou fungovat plně autonomním způsobem, což má přímý dopad na energetické nároky a stabilitu celé sítě, zvláště při výpadku centrálního logického prvku. Zvýší se také podpora mobilních ad-hoc sítí a duty-cycle MAC protokolů. Mimo to se dramaticky zvýší energetická efektivita celého systému, protože nebude nutné bezdrátově přenášet velké objemy dat, ale pouze samotné výsledky, či změny stavů.

Dnes nejrozšířenější protokolovou sadou bezdrátových senzorových sítí je ZigBee Pro [167], k čemuž také přispívá masivní podpora ZigBee Aliance, která sdružuje přední výrobce elektroniky, národní regulační úřady, university a další. Stále častěji se však hovoří o reálném příchodu IoT, který by měl využívat revoluční 6LoWPAN.

¹Např. platforma Sun SPOT může setrvat v aktivním stavu jen 23 hodin [116].

Dále je k dispozici řada proprietárních řešení. Výběr správné protokolové sady je přitom pro energeticky efektivní mesh sítě zcela esenciální. Bude proto provedena analýza současných řešení a vybrán vhodný kandidát k vlastní implementaci.

Dosavadní vývoj v oblasti MAC protokolů přiblížil efektivnost řízení spotřeby RF transceiverů k fyzickému maximu a lze předpokládat, že další vývoj bude přinášet jen zlepšení v řádu procent. Změníme-li stávající architekturu (HW/SW) bezdrátových senzorových systémů, můžeme nalézt další možnosti snížení potřeby energie a docílit tak skokového zvýšení energetické efektivity. Mezi hlavní cíl tedy patří návrh a ověření nového energeticky efektivního přístupu přenosu dat v bezdrátových senzorových sítích.

K plné podpoře přímého zpracování dat na senzorovém uzlu chybí systém řízení, resp. operační systém, jenž by zohlednil přísné energetické nároky a podpořil svou funkcí energetickou efektivitu celého systému. Posledním z cílů této práce je tak implementace operačního systému reálného času (RTOS) s podporou bezdrátových mesh sítí a řízením spotřeby energie. Vzhledem k novým možnostem vyvinuté platformy EE-WSN již provoz RTOS nepředstavuje problém [35] a lze využít cenná vylepšení jako determiničnost jednotlivých úloh, podporu přesné synchronizace, snadnou přenositelnost aplikací a snadnost vytváření aplikací bez nutné znalosti spodních vrstev systému.

Cíle této práce lze shrnout do následujících bodů:

- a) Analýza současných standardů a protokolových sad podporujících mesh komunikaci.
- b) Nalezení nového energeticky efektivního přístupu pro přenos dat v bezdrátových senzorových sítích.
- c) Implementace operačního systému reálného času (RTOS) pro bezdrátové mesh sítě s řízením spotřeby energie.
- d) Navrženou hardwarovou platformou překonat současné paradigma, že uzly bezdrátové senzorové sítě jsou na úkor energie výpočetně omezená zařízení. Tedy dokázat, že senzorové uzly nutně nemusí disponovat výpočetně a pamětově omezenými mikrokontroléry, aby dosáhly nízké spotřeby energie, a to v aktivním i úsporném režimu.
- e) Potvrzení hypotézy H₁ a překonat tak současné paradigma využití distribuovaných a centralizovaných výpočtů na uzlech bezdrátové senzorové sítě. Hypotéza H₁: Uzly bezdrátové senzorové sítě mohou být výpočetně nezávislé a jsou tedy schopny provádět výpočty, jako např. 512-bodá FFT, přímo na uzlech s nižší energetickou náročností, než s pomocí distribuovaných výpočtů.

3 EXPERIMENTÁLNÍ ANALÝZA PROTOKOLOVÝCH SAD S PODPOROU MESH KOMUNIKACE

Výběr vhodné komunikační protokolové sady je pro energeticky efektivní bezdrátové mesh sítě zcela esenciální. V dnešní době však neexistuje obecný konsensus ani standard definující protokolovou sadu [148]. Standardy IEEE 802.15.4 a IEEE 802.15.5 definují pouze doporučení pro první dvě vrstvy WSN modelu (fyzickou a linkovou). Standardy upravující vyšší vrstvy jsou např. ZigBee Pro a 6LoWPAN. IETF standard 6LoWPAN [69] definuje rozhraní mezi druhou a třetí vrstvou, které umožnuje spolupráci senzorových uzlů se zařízeními podporujících IPv6 [69], a to bez nutnosti translační brány. Pro směrování pak IETF doporučuje RPL protokol [67] a z aplikačních protokolů byl pro aplikace 6LoWPAN vyvinut COAP [68] protokol. Jediným uceleným řešením je tak stále ZigBee protokol, dnes ve verzi ZigBee Pro. Novinkou mezi mesh protokoly je pak proprietární protokolová sada Lightweight Mesh (LWM).

Vzhledem k tomu, že drtivá většina článků i komerčních analýz protokolových sad vychází z počítačových simulací [49, 92, 93, 113, 160], rozhodl se řešitel s kolegy provést detailní analýzu komunikačních sad na reálných zařízeních v reálném prostředí [41, 109]. Přestože bylo publikováno několik studií vycházejících z praktických měření, jejich analýza je zaměřena na měření specifických parametrů individuální protokolové sady [19, 76], mají za cíl hodnotit odlišné parametry jako RSSI (Received Signal Strength Indication) [125], poměr přijatých paketů [44], nebo jsou výsledky prezentovány v takové formě, že je nemožné je srovnat s jinými [90]. Navíc mají výsledky těchto článků jen omezenou vypovídací hodnotu, neboť v nich chybí např. bližší informace o charakteru přenosového kanálu či statistické vlastnosti prezentovaných výsledků.

V zájmu nalezení vhodné implementace energeticky efektivní komunikační sady, pro splnění prvního z vytyčených cílů, autor provedl sérii následujících měření. Nejprve byly výše uvedené standardy podrobeny testu energetické efektivity (kap. 3.1) a s vybranými kandidáty pak byla provedena detailní analýza, rozdělená do tří scénářů, zaměřujících se na:

- a) propustnost dat na aplikační vrstvě (kap. 3.3.1),
- b) zpoždění při směrování (kap. 3.3.2),
- c) samoopravnou schopnost sítě (kap. 3.3.3).

Vybrané scénáře by měly postihnout nejkritičtější vlastnosti WSN sítí pro reálný provoz a tedy podat dostatečné informace k nalezení vhodné implementace pro potřeby této práce. Podrobnostem o způsobu měření je věnována kap. 3.2.

3.1 Efektivnost přenosu dat

Jak již bylo řečeno, v prvním kroku byly výše uvedené standardy podrobeny testu energetické efektivity na RF transceiveru Atmel AT86RF233 s rychlostí přenosu 250 kb/s, tedy symbolovou délkou 32 µs. Graf na obrázku 3.1 vyjadřuje potřebné množství energie pro odeslání 1 Byte uživatelských dat. Hodnoty zohledňují veškerou datovou režii spojenou s odesíláním zprávy a vychází z výrobcem uvedených hodnot spotřeby pro jednotlivé operace [10]. Je tedy zahrnut i čas pro odsílání SHR (Synchronization header) a PHR (PHY header) (viz obr. 3.2). Režie vyšších protokolů je pak přenášena v MSDU (MAC Service Data Unit).

Ve srovnání (obr. 3.1) je zahrnut i samotný standard IEEE 802.15.4, označen jako MAC, nicméně ten samozřejmě nepodporuje mesh komunikaci a je zde uveden jako referenční maximum energetické efektivity. Z grafu je pak patrné, že nejlepší energetické efektivity, v rámci mesh protokolů, dosahuje proprietální protokol Lightweight Mesh.

Ačkoliv je standard 6LoWPAN velice atraktivní a výrazně se podílí na popularizaci senzorových sítí, z pohledu energetické efektivity jde o zcela nevhodný koncept. Podrobně byl tento standard popsán v [36].

3.2 Způsob měření

Z důvodu zpětného srovnání výsledků s jinými pracemi byla veškerá měření provedena na dobře známé hardwarové platformě Iris [103]. Jednotka Iris je vybavena



Obr. 3.1: Srovnání spotřeby energie pro odeslání 1Byte dat.

8-bitovým MCU Atmel ATmega 1281 a RF transceiverem Atmel AT86RF230. Měření tak byla provedena v pásmu 2,4 GHz.

Pro měření byly finálně vybrány implementace LWM verze 1.0.1 a BitCloud verze 1.11.0, reprezentující ZigBee Pro. Obě verze protokolových sad byly v době měření nejnovější dostupné. V obou případech bylo nutné provést portaci platformy Iris, což obnášelo především úpravy HAL (Hardware Abstraction Layer) vrstvy, a také přípravu řady aplikací pro každý scénář vlastního měření. Výsledky měření pro ZigBee Pro pochází z [109] a LWM z [41], kde lze případně nalézt více podrobností o jednotlivých měřeních.

Základní konfigurace LWM je uvedena v tab. 3.1, přičemž bylo využito nativního způsobu směrování. Každý z dílčích scénářů byl hodnocen pro dvě velikosti uživatelských dat. Minimální velikost 1 Byte a maximální velikost 77 Byte. Přestože LWM podporuje přenos až 109 Byte uživatelských dat, byla tato hodnota limitována BitCloud implementací, jejíž maximum je právě 77 Byte.



Obr. 3.2: Struktura rámce IEEE 802.15.4.

Tab.	3.1:	Výchozí	konfigurace	LWM.
------	------	---------	-------------	------

parametr	hodnota
APP_CHANNEL	0x1A
SYS_SECURITY_MODE	1
NWK_BUFFERS_AMOUNT	5
NWK_MAX_ENDPOINTS_AMOUNT	3
NWK_DUPLICATE_REJECTION_TABLE_SIZE	20
NWK_DUPLICATE_REJECTION_TTL	1000
NWK_ROUTE_TABLE_SIZE	20
NWK_ROUTE_DEFAULT_SCORE	3
NWK_ACK_WAIT_TIME	1000

Proces měření vystihuje obr. 3.3. V závislosti na měřeném scénáři byl do jednotek Iris nahrán patřičný firmware a poté byly jednotky rozmístěny do příslušné topologie. Po celou dobu měření byl využívaný kanál 16 monitorován spektrálním analyzátorem MetaGeek Wi-Spy 2.4x Chanalyzer¹ a v případě zvýšeného pozadí či akumulace rušení z postranního kanálu by došlo k zahození výsledků měření. Otisk radiového prostředí je patrný z obr. 3.4. Je patrné, že pozadí nepřesáhlo hodnotu -90 dBm. Radiový provoz byl neinvazivním způsobem zaznamenáván pomocí vyvinutého analyzátoru Open ZigBee Sniffer [108, 110] s disektory pro Wireshark. Tento analyzátor je schopen záznamu s časovou přesností 10^{-5} s, nicméně zaznamenané časy byly zaokrouhleny na 10^{-4} s.



Obr. 3.3: Proces měření.

3.2.1 Vyhodnocení dat

Zaznamenaná data byla nejprve testována grafickými a kvantitativními statistickými testy. Jednalo se o vyhotovení histogramu, Q-Q plot a autokorelačního testu. Protože mají výsledky velmi malou směrodatnou odchylku, jsou zde prezentovány jejich mediány a pro grafické zobrazení jsou použity sloupcové grafy, namísto kufrových. Každý dílčí scénář pak byl třicet krát opakován, přičemž každé měření se skládá minimálně z tisíce odeslaných paketů pro jeden uzel.

¹Chanalyzer: Dostupný na: http://www.metageek.net/.



Obr. 3.4: Záznam radiového prostředí v době měření.

Ačkoliv lze surová data z programu Wireshark exportovat do csv souboru, neexistuje vhodný nástroj pro jejich analýzu. Z tohoto důvodu byl v jazyku VBA vytvořen analytický framework, který emuluje funkci RF transceiveru (IEEE 802.15.4) a LWM protokolu. Vzhledem ke zpětné analýze zachycených dat, však musí framework postupovat reverzním způsobem. Při zpracovávání paketu je např. prozkoumáno okolí v rozsahu 54-ti symbolových period (864 µs) a vyhledáno příslušné potvrzení (ACK). Pakliže není nalezeno, je vyhledán duplicitní přenos paketu atd. Framework lze díky VBA jazyku implementovat do prostředí Microsoft Excel, kde lze spuštěním makra pohodlně načíst csv soubor a spustit automatizovanou analýzu dat. Skripty dokáží samy detekovat inicializační i finalizační událost, např. sestavení komunikace pro měření propustnosti, či změnu mezilehlého uzlu u testu samo zotavení sítě.

Framework obsahuje skripty pro analýzu:

- a) Datové propustnosti na aplikační vrstvě. Lze zobrazit i režii linkové vrstvy, společně s počtem opakovaných přenosů (retransmission) a to jak počtem událostí, tak počtem paketů. Je možné zobrazit statistiky přístupu konkrétních uzlů k médiu, což lze využít i pro ladění hodnot konkurenčního přístupu, tzv. CSMA (Carrier Sense Multiple Access) backoff exponentu, již instalované sítě. Pro případ manuální analýzy je pak možné zapnout vizuální podporu, kdy jsou pakety obarveny dle jejich významu i původce. Příklad je uveden na obr. C.1.
- b) Zpoždění při směrování. Příklad je uveden na obr. C.2.

c) Samoopravné schopnosti sítě. Zde je vyhodnoceno časové zpoždění od okamžiku první nedoručené zprávy až po úspěšně doručnou zprávu přes nově nalezenou cestu. Dále je zde uvedena i informace o celkové režii. Příklad je uveden na obr. C.3.

3.3 Výsledky měření

3.3.1 Propustnost dat na aplikační vrstvě

Tento scénář se snaží vyhodnotit maximální možné přenosové možnosti bezdrátových linek. Přestože většina WSN aplikací nevyžaduje vysoké přenosové rychlosti, řádově jen desítky kilobitů za sekundu, existují situace, kdy je potřeba přenést mezi uzly velké množství dat v krátkém čase. Takové situace vznikají například při aktualizaci firmwaru, nebo jako reakce více uzlů na sledovanou událost.

Standard IEEE 802.15.4 sice definuje pro pásmo 2,4 GHz maximální přenosovou rychlost linek na 250 kb/s, nicméně aplikační rychlost se v praxi značně liší od uvedené hodnoty. Redukci přenosové rychlosti ovlivňuje režie použité protokolové sady, způsob potvrzení o přijetí ACK² (Acknowledgement), způsob přístupu ke kanálu, ale i hardwarově implementované funkce jako automatické potvrzení o přijetí, automatické opětovné odeslání paketu, výpočet cyklického redundantního součtu (CRC, Cyclic redundancy check) či AES (Advanced Encryption Standard) [71].

Topologie sítě byla složena z cílového uzlu – brány a jednoho až tří konkurenčně vysílajících uzlů, komunikujících ve vzájemném radiovém dosahu (viz obr. 3.5a). Každý z vysílajících uzlů byl naprogramován k odeslání 1000 paketů bráně, přičemž pakety byly odesílány ihned po obdržení ACK předcházejícího paketu. Vlastní měření (analýza) začínalo vždy po naplnění směrovacích tabulek, aby nebylo zpoždění při hledání cesty zohledněno ve výsledcích.

Výsledná přenosová rychlost byla spočítána jako:

$$R = \frac{N \times L_{app_data} \times 8}{T_{end} - T_{start}},$$
(3.1)

kde R značí přenosovou rychlost aplikační vrstvy v bitech za sekundu, N je počet odeslaných paketů, L_{app} je velikost uživatelských dat v Byte, T_{end} je časové razítko posledně potvrzeného paketu a T_{start} je časové razítko prvního odeslaného paketu. Výsledky prvního scénáře jsou dostupné na obr. 3.6 a 3.7. Zobrazené hodnoty reprezentují medián ze série třiceti testů. Vlivem softwarové chyby LWM zde

 $^{^2 \}mathrm{na}$ linkové či aplikační vrstvě



Obr. 3.5: a) Topologie pro měření propustnosti. b) Způsob výpočtu a topologie pro měření zpoždění při směrování. c) Topologie pro měření samoopravné schopnosti sítě.

bohužel chybí výsledky měření pro více odesílajících uzlů a aktivovaném zabezpečení a potvrzení na aplikační vrstvě (AppACK).

Obr. 3.6 zachycuje dva grafy pro rozdílnou velikost uživatelských dat. Je zřejmé, že horní část, pro minimální velikost paketu, ukazuje nižší propustnost dat, vlivem vysoké režie s respektem k objemu zasílaných dat. Dále je vidět, že režie ZigBee protokolu způsobuje ještě nižší propustnost a to pro oba případy aktivovaného i neaktivovaného aplikačního potvrzení. Přenosová rychlost LWM je tak na volném kanálu až třikrát vyšší. Jak lze předpokládat, nejvyšší rychlosti je dosaženo pro maximální velikost uživatelských dat (spodní část obr. 3.6). V extrémním případě – tři konkurenčně odesílající uzly s maximální velikostí datové části a aktivním AppACK dosahuje LWM takřka dvojnásobně lepších výsledků.

Obrázek 3.7 zobrazuje opět dvojici grafů s rozdílnou velikostí uživatelských dat, jen s rozdílem že data jsou nyní šifrována. Zde jsou již rozdíly propustností průtokových sad markantnější. LWM dosahuje dokonce třináct krát vyšší propustnosti oproti BitCloud implementaci. Je třeba poznamenat, že obě protokolové sady využívají rozdílných šifrovacích algoritmů (BitCloud AES a LWM protokol XTEA), nicméně dle měření [56] dosahuje AES na stejném hardwaru (ATmega 1281) až dvakrát rychlejšího výpočtu, a proto za razantní snížení výkonu BitCloud není zodpovědný samotný šifrovací algoritmus, ale jiná výpočetní režie.

3.3.2 Zpoždění přeposíláním paketů

Mezitímco předchozí scénář hodnotil propustnost linky přes jeden skok (bez mezilehlého uzlu), tento scénář se zaměřuje na měření latence mezi uzly v multi hop topologii. Odesílající uzel zde komunikoval s bránou přes dva mezilehlé uzly, zajištující směrování. Způsob měření je společně s topologií patrný z obr. 3.5b. Nejtěžší částí bylo zajištění a kontrola požadované komunikace, která musela být fyzicky



Obr. 3.6: Propustnost dat na aplikační vrstvě bez zabezpečení. *Červené tóny* reprezentují BitCloud a *modré tóny* pak Lightweight Mesh. Čsílo v legendě indikuje velikost uživatelských dat v Byte a písmeno A značí aktivní AppACK.

zajištěna. Pro každý ze série testů bylo přeneseno přes 30.000 paketů, jejichž výsledky jsou formou mediánu prezentovány v obr. 3.8. Poznamenejme, že uvedený čas zahrnuje i čas přenosu paketů.

Ačkoliv výsledky vypovídají o přibližně srovnatelném zpoždění u obou protokolových sad pro nezabezpečenou komunikaci, je třeba upozornit, že BitCloud využívá směrovacího protokolu AODV (Ad hoc On-Demand Distance Vector). V případě zabezpečené komunikace se výsledky LWM takřka nezměnily a jejich mírné zvýšení plyne z navýšení velikosti paketu o 4 Byte. Nicméně u BitCloud implementace je nárůst zpoždění pro minimální velikost paketu 18 krát a pro maximální velikost dokonce 28 krát.


Obr. 3.7: Propustnost dat na aplikační vrstvě se zabezpečením. *Červené tóny* reprezentují BitCloud a *modré tóny* pak Lightweight Mesh. Čsílo v legendě indikuje velikost uživatelských dat v Byte a písmeno A značí aktivní AppACK.

Vzhledem k tomu, že mezilehlé uzly nemusí z pohledu vrstvového modelu zpracovávat pakety až k aplikační vrstvě, ale přeposílají je již na třetí vrstvě, lze předpokládat, že budou vykazovat vyšší výkon, než koncové uzly. Tento předpoklad, společně s vyššími nároky na zpracování potvrzuje pro BitCloud následující srovnání. Výkon LWM zůstává pro případ bez zabezpečení komunikace takřka beze změny. Obr. 3.9 srovnává přepočítané zpoždění a datovou propustnost koncových uzlů ze scénáře části 3.3.1.

Přestože zpracování paketů na koncových uzlech představuje pro BitCloud znatelnou redukci výkonu, propustnost u LWM zůstala pro nešifrovaný přenos téměř beze změny. Na druhou stranu výpočet šifrování přináší u LWM předpokládaný po-



Obr. 3.8: Zpoždění při směrování. Číslo na ose x představuje velikost uživatelských dat v Byte a písmeno S značí aktivované zabezpečení.

kles propustnosti, nicméně u BitCloud implementace se další redukce propustnosti neprojevuje. Jak už bylo řečeno v [109], výpočet šifrování je pravděpodobně natolik náročný, že další režie spojená s vrstvovým zpracováním již s respektem k šifrování nepředstavuje významné zpoždění.

3.3.3 Samoopravná schopnost sítě

Tento scénář se snaží vyhodnotit schopnost bezdrátové senzorové sítě autonomně reagovat na náhlý výpadek mezilehlého uzlu. Vzhledem k charakteru přenosového média jde o zcela zásadní vlastnost, neboť jde o velmi dynamické a nespolehlivé prostředí. Pro testování byly využity čtyři senzorové uzly v topologii diamantu (obr. 3.5c), přičemž byla měřena doba potřebná k zotavení sítě – nalezení alternativní cesty k bráně a datová režie – celkový objem vysílaných dat. Odesílající uzel byl samozřejmě mimo radiový dosah brány, která tak byla dostupná přes mezilehlé uzly. Tyto mezilehlé uzly pak byly alternativně zapínány a vypínány.

Vyhodnoceny byly celkem dva parametry. Zpoždění pro nalezení nové cesty, počítané jako:

$$T_{presmerovani} = T_{nova_cesta} - T_{zadost_o_presmerovani},$$
(3.2)

kde T_{nova_cesta} představuje časové razítko prvního paketu, využívajícího nové cesty a $T_{zadost_o_presmerovani}$ je časové razítko paketu, indikujícího žádost o nalezení nové cesty.



Obr. 3.9: Srovnání datové propustností síťové a aplikační vrstvy. Číslo na ose x představuje velikost uživatelských dat v Byte a písmeno S značí aktivované zabezpečení.

Druhým parametrem byl čas zotavení sítě:

$$T_{zotaveni} = T_{nova_cesta} - T_{prvni_nedoruceny_paket},$$
(3.3)

kde $T_{prvni_nedoruceny_paket}$ je časové razítko prvního nedoručeného paketu původní cesty.

Výsledky jsou prezentovány na obr. 3.10 pomocí dvojice grafů, z nichž horní část prezentuje zpoždění pro nalezení alternativní cesty a spodní graf pak celkový čas nutný k zotavení sítě. Je zajímavé, že LWM dokáže celý proces nalezení nové cesty dokončit v kratším čase než BitCloud, ale vyžaduje k tomu odeslání většího objemu dat. Přehled o potřebné režii k nalezení alternativní cesty přináší obr. 3.11 (horní



část), na jehož spodní části je pak uvedena souhrnná režie pro celý proces obnovy sítě.

Obr. 3.10: Zpoždění samoopravného procesu. Horní část prezentuje zpoždění pro nalezení alternativní cesty a spodní část celkový čas nutný k zotavení sítě.

3.4 Závěry analýzy

Dle zde prezentovaných výsledků, je zřejmé, že z pohledu rychlosti a efektivnosti přenosu dat vychází protokolová sada LWM nejlépe. Největšího rozdílu propustnosti dat bylo dosaženo s aktivovaným šifrováním, kde LWM předčilo ZigBee Pro dokonce **13 krát**. Přeposílání paketů se směrováním pak představuje pro ZigBee až **30 krát** větší zpoždění než pro LWM. Ze srovnání aplikační a linkové propustnosti



Obr. 3.11: Režie samoopravného procesu. Horní část prezentuje režii pro nalezení alternativní cesty a spodní část celkovou režii potřebnou k zotavení sítě.

je navíc patrná zvýšené režie zpracování dat na koncových uzlech ZigBee. Výsledky také poukazují na skutečnost, že robustnost ZigBee Pro technologie se negativně podepisuje na výkonových nárocích na hardware. LWM však nenabízí takové možnosti jako ZigBee Pro. ZigBee Pro představuje funkční a stabilní řešení s širokou podporou aplikačních profilů a především velmi snadný způsob vytváření aplikací. Vždy tak samozřejmě záleží na konkrétních požadavcích na finální aplikaci.

Pro výběr vhodné implementace je kladen důraz na energetickou efektivitu a podporu mesh komunikace. Z měření vyplývá, že oba kandidáti nabízí stabilní řešení pro zajištění mesh komunikace, ovšem LWM s menší datovou režií a rychlejším zpracováním na témže hardwaru. Jelikož LWM předčilo BitCloud ve všech scénářích (a až c), rozhodl se autor pro využití LWM. Pro dosažení vyššího zabezpečení komunikace a energeticky efektivnějšího přenosu dat však bude potřeba systém LWM modifikovat. Více je úpravám LWM věnováno v kap. 5.3.1.

4 ENERGETICKY EFEKTIVNÍ HARDWAROVÁ PLATFORMA

Nutnost vývoje vlastní platformy vyvstala převážně z faktu, že jak na komerčním trhu, tak v současné literatuře není dostupná HW platforma, splňující níže uvedené požadavky. Poznamenejme, že nejde o hlavní cíl práce, nýbrž o nutný nástroj k demonstraci a ověření cílů *d* a *e*. Tedy dokázat, že senzorové uzly nutně nemusí disponovat výpočetně a paměťově omezenými mikrokontroléry, aby dosáhly nízké spotřeby energie v aktivním i úsporném režimu (viz str. 28) a zároveň, že senzorové uzly mohou být výpočetně nezávislé a dosáhnout nižší energetické náročnosti, než s pomocí distribuovaných výpočtů. Hardwarová platforma je navíc vyvinuta pro těsnou součinnost s později prezentovaným softwarovým řešením, díky čemuž lze platformu využít i k demonstraci splnění cíle *b*. Tedy k ověření energetické efektivity navrženého systému přenosu dat v oblasti, prezentovaného v kap. 4.4.1. Vlastní návrh platformy, začíná kapitolou 4.3.

4.1 Požadavky na hardwarovou platformu

Kritickým bodem jsou samotné požadavky na platformu, neboť jde o zcela protichůdné vlastnosti.

- a) Výpočetní výkon: Hlavním požadavkem je dostatečně vysoký výpočetní výkon a zároveň však extrémně nízká energetická spotřeba. Platforma musí umožnit signálové zpracování snímaných dat s takovou rychlostí, aby nedocházelo k jejich ztrátě i při vysoké vzorkovací rychlosti (> 500 Hz).
- b) Fyzické vlastnosti: Mezi další vlastnosti pak patří malá velikost, která by umožnila využití platformy v oblasti WBSN, malá váha a přitom dlouhá výdrž zdroje energie.
- c) Podpora bezdrátovýchc mesh sítí.

d) Bezpečnost. Platforma musí umožnit zabezpečení bezdrátové komunikace, vyhovující aktuálním doporučením NIST (National Institute of Standards and Technology) [112], s minimálním potenciálem k prolomení bezpečnosti např. odposlechem sběrnic, či vyčtením obsahu pamětí.

U všech uvedených vlastností bude hrát roli energetická efektivita řešení jednotlivých subsystémů. Výběr požadavků odráží současné potřeby trhu i výzkumu (viz motivace na str. 14).

4.2 Dostupná řešení

Z uvedeného přehledu kap. 1.2 vyplívá, že hardwarových platforem v oblasti WSN s vysokým výpočetním výkonem není mnoho. Aktuální srovnání dostupných WSN platforem je provedeno také v [45]. Platformy obsahující mikrokontroléry ATmega128x či TI MSP430 lze pro svůj nízký výpočetní výkon a omezenou RAM paměť vyřadit (viz tab. A.1). Je jasné, že tyto platformy nebyly vyvinuty pro výpočetně náročné aplikace, ale pro levné a velmi jednoduché snímání fyzikálních vlastností ve svém okolí.

Mezi komerční platformy specificky vyvinuté pro aplikace vyžadující vysoký výpočetní výkon patří Sun SPOT [117], Intel Mote2 (iMote2) [111], STARGATE [102], LOTUS [101] a Preon32 [157]. Podrobné informace o těchto platformách jsou uvedeny v následující tabulce 4.1.

Výše uvedené platformy nebyly k dispozici pro změření přesných údajů spotřeby. Údaje proto vychází z dostupných informací jednotlivých výrobců mikrokontrolerů. Ti však neuvádí kompletní přehled spotřeby pro všechny rychlosti procesoru a za jednotných podmínek, a proto není srovnání zcela přesné. Pro představu o energetické náročnosti jsou však údaje dostačující.

Komerční platformy iMote2 a STARGATE jsou založeny na procesorech ARM9s instrukční sadou ARMv5 a Thumb 2 a podporou DSP instrukcí. Sun SPOT využívá starší architektury s instrukční sadou ARMv4 a Thumb první verze s výkonem 200 MIPS (při 180 MHz). Sun SPOT proto dosahuje menšího výkonu a horší energetické efektivity [6]. Poznamenejme, že platforma Sun SPOT prošla celkem osmi revizemi a existují proto verze s odlišnými MCU. Přestože architektury ARM9a ARMv7-M nelze zcela přesně srovnat, výsledky výkonnostních testů Dhrystone 2.1, Feedback-Hammerstein a Gompertz algoritmů ukazují na vyšší výpočetní výkon procesorů ARM9 [29]. Architektura ARMv7-M však dosahuje podstatně lepší energetické efektivity [6] a výrazně lepší podpory pro WSN aplikace (např. pomocí vlastního řadiče přerušení (NVIC, Nested Vectored Interrupt Controller), determi-

Transceiver		TI CC2420	TI CC2420	(MICA2 node)	AT86RF231	AT86RF231
Spotřeba MCU^a		$\begin{array}{l} 570 \ \mathrm{mW}/\ 416 \ \mathrm{MHz}\ (1,35 \ \mathrm{V})^c\\ 116 \ \mathrm{mW}/\ 104 \ \mathrm{MHz}\ (0,9 \ \mathrm{V})^c\\ 44,2 \ \mathrm{mW}/13\mathrm{MHz}\ (0,85 \ \mathrm{V})^c\\ \mathrm{standby}\ \mathrm{mode}^d\ 1,72 \ \mathrm{mW} \end{array}$	80,5 mW/ 180 MHz c standby mode e 520 $\mu A~(1,72$ mW)	$\begin{array}{l} 411 \ \mathrm{mW}/\ 400 \ \mathrm{MHz} \\ 283 \ \mathrm{mW}/\ 300 \ \mathrm{MHz} \\ 178 \ \mathrm{mW}/\ 200 \ \mathrm{MHz} \\ \mathrm{idle} \ \mathrm{mode}^{f} \ 63 \ \mathrm{mW}\ /200 \ \mathrm{MHz} \\ \mathrm{idle} \ \mathrm{mode}^{f} \ 45 \ \mathrm{mW}\ /33 \ \mathrm{MHz} \end{array}$	139 mW/ 100 MHz 23,1 mW/ 12 MHz power down mode ^{g} 0,102 mW	$93.4 \text{ mW}/72 \text{ MHz}^c$ $12,2 \text{ mW}/8 \text{ MHz}^c$ stop mode 0,086 mW
Pamět Flash	[kB]	32768	128	32768	512	256
Paměť RAM	[kB]	$256~(+32\mathrm{MB}^b)$	16	$64 \ (+64 \mathrm{MB}^b)$	64	64
Max. takt	procesoru [MHz]	416	180	400	100	72
Mikrokontrolér		Intel XScale PXA271 (ARM9, ARMv5TE)	Atmel AT91RM9200 (ARM7, ARMv4T)	Intel Xscale PXA255 (ARM9, ARMv5TE)	NXP LPC1758 (ARMv7-M)	ST STM32 N/A (ARMv7-M)
Výrobce		Crossbow	Oracle	Crossbow	MEMSIC	Virtenio
Název uzlu		iMote2	Sun SPOT	STARGATE	LOTUS	Preon32

Tab. 4.1: Přehled komerčně dostupných výkonných platforem.

^aPakliže není uvedeno jinak, hodnoty spotřeby zahrnují spotřebu pouze MCU a vychází z informací od výrobce. Platí pro spuštěný program while(1) {} z flash paměti s externím zdrojem taktu, vypnutými periferiemi, zdroj napět
í $3,3\,\mathrm{V}$ a okolní teplotou 25 °C.

 b Externí paměť.

 $^c\mathrm{Bl}$ íže nespecifikované podmínky běhu programu.

 d Veškerá aktivita procesoru i PLL je pozastavena. Běží jen RTC a management spotřeby. Zdroj napětí 1,1 V

^eVeškerá aktivita procesoru i PLL je pozastavena. Běží jen RTC.

fVeškerá aktivita procesoru je pozastavena. Běží RTC, hlavní zdroj taktu a management spotřeby.

⁹Zdroj taktu pro MCU a Flash paměť je vypnut s možností obnovení stavu pamětí a registrů. 32 kHz RC oscilátor je aktivní. Vyžaduje 100 µs ke spuštění Flash paměti.

Architektura MCU	Doba výpočtu	Doba výpočtu -O	Spotřeba energie
	[cyklů]	$[{ m cykl}$ ů $]$	$[\mu J]$
ARM7TDMI (LPC 2148)	28370	24897	86,27
Cortex-M0 (LPC 1114)	57610	54214	38,46
Cortex-M1 Actel Fusion AFS1500)	57610	54214	$291,\!58$
Cortex-M3 (LPC 1343)	36676	32292	$30,\!37$
ATmega128 (AT $1284p$)	249368	240696	317,72
MSP430 (MSP430F5438)	113919	109138	$122,\!45$

Tab. 4.2: Spotřebu energie a doba výpočtu algoritmu FIR.

Kompilováno v IAR Systems V6.30. O- optimalizováno pro rychlost. Převzato z [124].

Architektura MCU Doba výpočtu Doba výpočtu -O Spotřeba energie [cyklů] [cyklů] [µJ] ARM7TDMI (LPC 2148) 62877 17277 59,86 Cortex-M0 (LPC 1114) 97135 16959 12,03Cortex-M1 Actel Fusion AFS1500) 97135 16959 159,44 Cortex-M3 (LPC 1343) 2545823,9494147 ATmega128 (AT 1284p) 237037 162380 214,34 MSP430 (MSP430F5438) 168235 103109 115,69

Tab. 4.3: Spotřebu energie a doba výpočtu pro Dhrystone benchmark.

Kompilováno v IAR Systems V6.30. O- optimalizováno pro rychlost. Převzato z [124].

nistické latence přerušení, systémového časovače). Z tohoto pohledu tak nejlépe vychází platformy LOTUS a Preon32.

Pro lepší představu o výkonových vlastnostech jednotlivých architektur jsou zde uvedeny tabulky 4.2, 4.3 a 4.4. Tabulka 4.2 uvádí spotřebu energie a dobu výpočtu filtrace signálu z 51 16-ti bitových vzorků filtrem s konečnou impulzní odezvou (FIR) sedmnáctého řádu.

Tabulka 4.3 pak prezentuje výsledky referenčního testu Dhrystone benchmark, který měří výkon procesoru při práci s celými čísly nad ukazateli, strukturami a řetězci.

Obdobně tabulka 4.4 uvádí výsledky referenčního testu Whetstone benchmark, který se zaměřuje na výkon procesoru při práci s čísly s plovoucí desetinnou čárkou při výpočtech matematických funkcí jako sinus, kosinus a logaritmus.

Pro své velké rozměry jsou platformy Sun SPOT (41 x 70 mm), STARGATE (95 x 63 mm) i Lotus (76 x 34 mm) nevyhovující. Naopak Preon32 (28 x 19 mm) má velmi příznivé rozměry i pro využití v oblasti WBSN.

Z pohledu spotřeby energie jsou úsporné módy platforem iMote2, Sun SPOT

Architektura MCU	Doba výpočtu	Doba výpočtu -O	Spotřeba energie
	[cyklů]	[cyklů]	$[\mu J]$
ARM7TDMI (LPC 2148)	43433	42300	146,57
Cortex-M0 (LPC 1114)	88725	87155	61,84
Cortex-M1 Actel Fusion AFS1500)	88725	87155	468,83
Cortex-M3 (LPC 1343)	58758	56692	$53,\!32$
ATmega128 (AT $1284p$)	175892	174316	230,1
MSP430 (MSP430F5438)	115406	112151	125,83

Tab. 4.4: Spotřebu energie a doba výpočtu pro Whetstone benchmark.

Kompilováno v IAR Systems V6.30. O- optimalizováno pro rychlost. Převzato z [124].

a STARGATE pro potřeby WSN aplikací nevyhovující, přičemž nejlepších hodnot dosahuje platforma Preon32 ve stop režimu (0,086 mW). Zde je zdroj taktu pro MCU i Flash pamět je vypnut s možností obnovení stavu pamětí a registrů. Aktivní je jen 32 kHz RC oscilátor.

Ze srovnání RF transceiverů (tab. 4.5) je evidentní, že platformy LOTUS a Preon32 předčí své konkurenty v citlivosti i energetické efektivitě pro příjem a vysílání. Vzhledem k energetické efektivitě a požadavkům na malou velikost platformy se tato práce soustředí výhradně na RF transceivery v pásmu 2,4 GHz. V tomto pásmu dovoluje standard IEEE 802.15.4 využít vysoké rychlosti (250 kb/s) a při využití menšího činitele rozprostření až 2 MB/s. Transceivery proto mají v tomto pásmu větší energetickou efektivitu a dovolují využít fyzicky malých antén. Využití srovnatelně malých antén pro nižší pásma by vedlo ke snížení citlivosti, potažmo spolehlivosti. Další nespornou výhodou je možnost využití 16-ti kanálů v pásmu 2,4 GHz, čímž lze na rozdíl od pásma 868 MHz eliminovat případné interference s dalšími zařízeními.

Poznamenejme, že někteří autoři (např. [87]) prezentují energetickou efektivitu transceiverů, jako poměr spotřeby energie při vysílání či příjmu a rychlosti přenosu. Výsledkem je pak energie na jeden bit E/bit. Při využití standardu 802.15.4 je ale přenosová rychlost u všech transceiverů fixní (250 kb/s) a proto je uvedení samotné spotřeby energie dostatečné.

Bohužel ani jedna z uvedených platforem neposkytuje podporu zabezpečení bezdrátové komunikace. Hardwarové AES akcelerátory RF transceiverů (např. u AT-86RF231) jsou díky nezabezpečené komunikaci přes SPI sběrnici snadnou snadno překonatelné.

V posledních letech se v odborné literatuře objevilo několik návrhů hardwarových platforem, založených na výkonných mikrokontrolérech *ARM7TDMI-S* a *ARMv7-M*. Konkrétně jde o platformy NetBrick [3], Opal [79], Egs [85], AVS-Extrem [33],

NTU-NSC WSN M3-SuperNode [66] a [74]. Jejich parametry jsou opět shrnuty v přehledové tabulce 4.6. Senzorové uzly využívající heterogenních procesorů, jako Medusa MK-2 [136], zde nejsou uvedeny, neboť jsou z logiky věci dražší a nutně zvyšují klidový proud celého systému. To má za následek nižší energetickou efektivitu takovýchto řešení.

Platforma AVS-Extrem využívá již zastaralé 32-bitové ARM architektury ARM7TDMI-S, která byla nahrazena např. procesory Cortex–M0 či Cortex-M3. Cortex-M3 nabízí výrazně vyšší výkon, energetickou efektivitu, a vylepšené režimy spánku [4]. Cortex-M0 pak nabízí mimo vyšší výkon i trojnásobnou úsporu energie a třetinovou plochu čipu. S rozměry 30 x 100 mm a primárním zdrojem energie čtyř D-monočlánků (16 Ah) nepaří mezi malé platformy. Společně s absencí podpory kryptografického zabezpečení komunikace tak AVS-Extrem nevyhovuje především bodům b) a d) z požadavků kap. 4.1.

Bezdrátová platforma Egs byla vyvinuta zejména pro lékařské aplikace. Její rozměry cca 85 x 60 mm však vylučují využití pro oblast WBSN. Využití dvou transceiverů navíc zvyšuje velikost klidového proudu a tím pádem snižuje energetickou efektivnost celé platformy. Vzhledem k chybějící kryptografické podpoře tak platforma opět nevyhovuje především bodům b) a d).

Motivací pro vývoj platformy NetBrick je taktéž neexistence dostatečně výkonných WSN uzlů. Autoři vychází ze své praxe s reálnými aplikacemi (monitoring vodního prostředí a predikce sesuvu kamenů), kde jsou senzory jako MicaZ výkonově nevyhovující. Tato platforma byla vyvinuta s obdobnými požadavky na vysoký výkon a nízkou spotřebu. Její rozměry jsou 170 x 107 mm a je napájena 12V olověnou baterií o kapacitě 18 Ah. Požadavky bodů b a d tak zde nejsou splněny.

Platforma Opal byla vyvinuta zejména k využití pro perzistentní monitorování okolí s důrazem na energetickou efektivitu a zabezpečení komunikace. Pro zajištění bezpečnosti využívá platforma dedikovaného TPM (Trusted Platform Module) obvodu AT97SC3203S. Ten zajišťuje hardwarově akcelerovaný výpočet RSA podpisů do velikosti 2048 bitů, SHA-1 a obsahuje i generátor náhodných čísel, vyhovující FIPS 140-2. Když pomineme nezabezpečenou komunikaci TPM modulu přes TWI (Two-wire Interface) sběrnici, chybí zde podpora symetrické kryptografie. Zvláštností je využití dvou transceiverů, přičemž každý z nich může využívat dvou antén pro zajištění podpory MIMO (Multiple Input/ Multiple Output). I zde však využití dvou transceiverů vede ke zvýšení klidového proudu a snížení energetické efektivity. Celkové rozměry cca 100 x 70 mm jsou také neuspokojivé. Bodům b) a d) z kap.4.1 není zcela vyhověno ani zde.

Název transceiveru/ Vlastnosti	AT86RF231	NXP JN5148	TI CC2420	TI CC2520	UBEC UZ2400	Xbee $Pro S1$
Spotřeba energie – příjem [mW]	36,9	$57, 75^{a}$	62,04	55,5	59,4	181,5
Spotřeba energie – vysílání [mW]	$42/+3~\mathrm{dBm}$	ı	·	$100,8/~+5~\mathrm{dBm}$	$72,6/~+3~\mathrm{dBm}$	$495/~+10~\mathrm{dBm}$
Spotřeba energie – vysílání [mW]	$34.8/~0~\mathrm{dBm}$	$49,5^b$	$57,42/~0~\mathrm{dBm}$	$77,4/~0~\mathrm{dBm}$	ı	I
Spotřeba energie – vysílání [mW]	22,2/ -17 dBm	I	32,67/ -15 dBm	$48,\!6/$ -18 dBm	ı	I
Citlivost [dBm]	-101	-95	-95	-98	-95	-100
Spotřeba energie – úsporný režim [nW]	60	396	60	90	6600	33000^{c}

transceiverů.
RF
Přehled
4.5:
Tab.

^aHodnota nezahrnuje spotřebu integrovaného MCU. ^bHodnota výstupního výkonu není definována. ^cProdleva při probuzení 13,2 ms.

49

Název uzlu	Mikrokontrolér	Max. takt procesoru [MHz]	Paměť Ram [kB]	Pamět Flash [kB]	Spotřeba energie MCU ^a	Transceiver
AVS-Extrem	NXP LPC2387 (ARMv4T, ARM7TDMI-S)	72	64	512	63 mA/ 72 MHz (208 mW) 15 mA/ 10 MHz (49,5 mW) power down mode ^b c 113 μ A (0,373 mW) 106 mA/ 72 MHz (350 mW) ^f	TI CC1101
Egs	Atmel SAM3U2C (ARMv7-M)	96	36	$128 (+1MB^d)$	48.3 mA/ 96 MHz (159 mW) 39.7 mA/ 72 MHz (131 mW) 30.5 mA/ 48 MHz (101 mW) 10.1 mA/ 12 MHz (33.3 mW) 7.54 mA/ 8 MHz (24.9 mW) wait mode ^{be} 20 μA (0,066 mW)	TI CC2520, Mitsumi WML-C46 (Class 2 Bluetooth)
NetBrick	ST STM32F103 (ARMv7-M)	72	$64~(+1 \mathrm{MB}^d)$	512 $(+128 \mathrm{MB}^d)$	30,5 mA/ 72 MHz (101 mW) 2.0,7 mA/ 48 MHz (68,3 mW) 8.2 mA/ 16 MHz (15,1 mW) 5 mA/ 8 MHz (16,5 mV) stop mode ^b 35 µA (0,116 mW) 69 mA/ 72MHz (227,7 mW) ^f	NXP JN5148
Intel-NTU NSC WSN M3-SuperNode	N/A (ARMv7-M)	72	48	256	70 mA/ 72 MHz (231 mW) ^g 6,5 mA/ 8 MHz (21,5 mW)	UBEC UZ2400
Opal	Atmel SAM3U4E (ARMv7-M)	96	52	$256~(+2\mathrm{GB}^d)$	48,3 mA/ 96 MHz (159 mW) 39,7 mA/ 72 MHz (131 mW) 30,5 mA/ 48 MHz (101 mW) 10,1 mA/ 12 MHz (33,3 mW) 7,54 mA/ 8 MHz (24,9 mW) wait mode ^{be} 20 μA (0,066 mW)	AT86RF231, AT86RF212
[74]	ST STM32F407 (ARMv7E-M)	168	196	$1024 (+32MB^d)$	46 mA/ 168 MHz (151,8 mW) 30 mA/ 90 MHz (99 mW) 25 mA/ 72 MHz (82,5 mW) ^h 22 mA/ 60 MHz (72,6 mW) 18 mA/ 48 MHz (59,4 mW) ^h 12 mA/ 30 MHz (59,4 mW) ^h 12 mA/ 30 MHz (13,2 mW) 4 mA/ 8 MHz (13,2 mW) 5 top mode ^b 450 μ A (1,485 mW) 100 mA/ 168MHz (330 mW) ^f	Digi XBEE Pro S1

Tab. 4.6: Přehled současně využívaných výkonných platforem.

^aPakliže není uvedeno jinak, hodnoty spotřeby zahrnují spotřebu pouze MCU a vychází z informací od výrobce. Platí pro spuštěný program while(1) {} z flash paměti s externím zdrojem taktu, vypnutými periferiemi, zdroj napětí 3,3 V a okolní teplotou 25 °C.

^bZdroj taktu pro MCU a Flash paměť je vypnut s možností obnovení stavu pamětí a registrů. 32 kHz RC oscilátor je aktivní.

 $^{C}_{\rm Vyžaduje 100}$ µ
s ke spuštění Flash paměti.

 $d_{\mathrm{Extern}()}$ paměť.

^eVyžaduje 10 µs k probuzení systému se zdrojem taktu 4/8/12 MHz z RC oscilátoru. Prodleva startu PLL není započtena.

 $f_{
m Reálné měření na platformě.}$

 $g_{
m Zapnuté}$ periferie. hHodnoty byly získány pomocí polynomiální regrese, polynomem třetího řádu, podle y = 1E-06x3 - 0,0011x2 + 0,4112x + 0,7266.

Nejméně informací je dostupných k platformě M3-SuperNode. Ta byla vyvinuta jako Energy-Harvesting Sensor Module, tedy senzorický uzel s podporou získávání energie z okolí a rovněž využívá mikrokontrolér ARM Cortex-M3. Důraz na malou velikost, či zabezpečení není kladen ani zde a požadavky bodů b a d nejsou opět splněny.

Platforma [74] byla vyvinuta pro výpočetně náročné aplikace, zvláště pak pro monitorování dopravy inteligentních měst, s velmi nízkou spotřebou energie. Platforma využívá mikrokontrolér architektury ARMv7E-M a LiFePO4 akumulátor s kapacitou 8 Ah. S rozměry cca 80 x 60 mm a absencí kryptografického zabezpečení je s respektem k požadavkům z bodů b a d i tato platforma nevyhovující.

Jak již bylo řečeno platformy Egs a Opal využívají více RF transceiverů, čímž zvyšují velikost klidového proudu při jejich nevyužití a tím pádem snižují energetickou efektivnost celé platformy. Jako zcela nevhodné, z pohledu energetické efektivity, lze považovat transceivery NXP JN5148 a Xbee Pro S1, protože obsahují plnohodnotné mikrokontroléry. To se nutně odráží jak ve zvýšené spotřebě při provozu, tak v režimu spánku. V rámci komunikačního subsystému tak vychází nejlépe platforma Opal, nicméně obsahuje více RF transceiverů. Mezi platformami s jedním transceiverem pak M3-SuperNode.

Z uvedeného vyplívá, že platforma AVS-Extrem poskytuje nejnižší výkon mezi uvedenými kandidáty. Všechny ostatní využívají architektury *ARMv7-M*, přičemž platforma [74] pak *ARMv7E-M*. Proto zde bude hrát roli zejména návrh a způsob optimalizace daného mikrokontroléru od výrobce. Pro lepší představu o spotřebě energie v aktivním režimu za různých taktovacích frekvencí je zde přiložen obr. 4.1, vycházející z hodnot tab. 4.6. Z ní je jednoznačně patrné, že nejvyšší energetické efektivity v aktivním režimu dosahuje platforma [74], následovaná platformou NetBrick.

Na obr. 4.2 pak lze nalézt přehled spotřeby energie ve srovnatelných úsporných režimech. Tyto hodnoty opět vychází z tab. 4.6. Zde je bohužel patrná výrazná slabina platformy [74]. Přes vysokou en. efektivitu v aktivním režimu je díky vysoké spotřebě v režimu spánku zcela nevyhovující. V tomto ohledu nejlépe obstály platformy Egs a Opal, následované opět platformou NetBrick.

Je zřejmé, že každá z HW platforem, jak publikovaných, tak komerčních má silné i slabé stránky. Jako nejlepší možnost pro výpočetně náročné aplikace byla například v [45] vybrána platforma Sun SPOT. S respektem k výše uvedeným požadavkům kap. 4.1 a energetické efektivitě výpočetního a komunikačního subsystému však nejvíce vyhovuje platforma Preon32. Ta ale, stejně jako všechny ostatní, nenabízí žádnou kryptografickou podporu.



Obr. 4.1: Přehled spotřeby energie v aktivním režimu.



Obr. 4.2: Přehled spotřeby energie v úsporných režimech.

4.3 Vlastní návrh hardwarové platformy

V následujících kapitolách je popsán návrh vlastní hardwarové platformy, pojmenované EE-WSN (Energy Efficient Wireless Sensor Node).

4.3.1 Návrh výpočetního subsystému

Kritickým bodem návrhu energeticky efektivní platformy je bezesporu výběr vhodné architektury mikrokontroléru, potažmo konkrétní implementace určitého výrobce.

V předcházející kapitole 4.2 byla předvedena řada výkonných platforem vystavěných na architektuře jádra ARMv7-M, čímž by se mohlo zdát, že ty procesory našli své uplatnění i v oblasti WSN. Nicméně z uvedených zdrojů energie jednotlivých platforem (např. 16 Ah/AVS-Extrem, 18 Ah/NetBrick) lze tušit, že vysoký výpočetní výkon je tak vykoupen zvýšenou spotřebou a že ve srovnání s klasickými mikrokontroléry jako ATmega128 a TI MSP430, nedosahují v režimu spánku tak nízké spotřeby. WSN aplikace jsou charakteristické velmi krátkou aktivitou a většinu času (běžně 99%) zůstávají v úsporných režimech. Právě tam jsou mikrokontroléry současných výkonných platforem neefektivní. Ideou je, aby hardware co nejrychleji provedl náročný výpočet a ihned přešel do režimu spánku. V závislosti na střídě aktivní doby se pak může stát spotřeba v úsporném režimu dominantní složkou.

Patrně nejdůkladnější analýzu vhodnosti architektury ARMv7-M pro WSN aplikace byla provedena v [84]. Rozsáhlá studie srovnává tuto 32-bitovu architekturu se současně nejvyužívanějším mikrokontrolérem ve WSN oblasti TI MSP430 z pohledu výpočetního výkonu a energetické efektivity. Jednoznačným závěrem je obhájení dominantní pozice 16-ti bitové platformy (MCU MSP430) díky výborné energetické efektivitě. Studie potvrzuje vyšší výpočetní výkonu procesorů ARM Cortex-M3 ovšem za cenu přibližně dvojnásobné spotřeby energie celého systému při běhu aplikací pod operačním systémem TinyOS.

Přes výše uvedená negativa a stále kontroverznímu postavení procesorů Cortex-M3 v oblasti WSN se autor této práce rozhodl využít nejvýkonnější architektury z rodiny procesoru Cortex-M a sice Cortex-M4F. Ten je obecně pokládán za příliš výkonný a nevhodný pro WSN aplikace [155]. O ojedinělosti využití Cortex-M4 architektur pro WSN aplikace svědčí i fakt, že v současné literatuře existuje pouze jediný návrh hardwarové platformy s tímto procesorem [74]. Ze srovnání spotřeby energie v úsporných režimech (obr. 4.2) pak vyplývá, že tato platforma dosahuje jednoznačně nejvyšší spotřeby. Autor však pevně věří, že pečlivým návrhem hardwarové platformy, využitím plné funkcionality jádra MCU ve spojení s vhodně řešeným firmwarem dosáhne vytyčených požadavků kap. 4.1 a to s ještě nižšími energetickými požadavky. To by mělo v důsledku přispět ke splnění cílů d a e.

ARM Cortex-M4

Procesory Cortex-M4 jsou vylepšenou verzí Cortex-M3, a proto přebírají veškeré jeho vlastnosti a doplňují je o řadu rozšíření. Podobně je tomu i v případě srovnání instrukčních sad, kde je instrukční sada Cortex-M3 (74 instrukcí) podmnožinou Cortex-M4 (137 instrukcí) [7]. Zmíněná rozšíření Cortex-M4 se týkají především podpory DSP (Digitální Signálový Procesor) algoritmů, jako je FFT (Fast Fourier Transform), IIR a FIR filtrů a urychlených matematických výpočtů pomocí hard-



Obr. 4.3: Srovnání výkonu DSP algoritmů v počtu hodinových cyklů CPU [153].

warové MAC_{math} jednotky (Multiply Accumulate), nově podporující jedno cyklové výpočty nad 32-bitovými slovy. Cortex-M4 tak kombinuje vlastnosti MCU a výkonných DSP. SIMD (Single Instruction Multiple Data) instrukce pak umožňují paralelní výpočet několika celočíselných aritmetických operací v jediném cyklu, což vede k dalšímu zvýšení výkonu DSP operací. Srovnání výkonu běžných celočíselných DSP operací pro jádra Cortex-M3 a Cortex-M4 je na obr. 4.3.

Cortex-M4 jsou založeny na velmi výkonném 32-bitovém RISIC (Reduced Instruction Set Computer) procesoru, kde je většina instrukcí provedena v jediném cyklu. K tomu přispívá i třístupňové zřetězení paralelního zpracování instrukcí, které je na rozdíl od *ARM7TDMI* procesoru doplněno o predikci skoků, čímž nedochází k degradaci výkonu vlivem nuceného vyprázdnění a znovunačtení instrukcí z operační paměti. Tím dokáže procesor zajistit trvalý výkon 1,25 DMIPS/MHz¹ [7]. Přestože oba procesory ARM Cortex-M4 i Cortex-M3 nabízí srovnatelný výkon 1,25 DMIPS/MHZ, výsledky na obr. 4.3 dokazují výrazně lepší podporu matematických operací jádra Cortex-M4.

Jak již bylo řečeno, Cortex-M3/M4 poskytují výbornou podporu aplikací WSN skrze integrovaný řadič přerušení (NVIC) a deterministické latence přerušení ve 12-ti cyklech. Dále je to velmi efektivní bitová manipulace, kde procesory Cortex-M využívají mapování jednotlivých bitů RAM paměti a registrů periferií na alias adresy [153]. To ve výsledku znamená menší velikost kódu a rychlejší provedení změn.

Dvaceti čtyř bitový systémový časovač pak nabízí zdroj taktu pro operační sys-

 $^1\mathrm{A}$ ž 1,91 DMIPS/MHz při simultánní (multi-file) kompilaci.

1
14
1
3
14

Tab. 4.7: Rychlosti výpočtů FPU jednotky.

Informace z [8].

témy reálného času (RTOS), které dosahují díky rozdílnému zpracování instrukční sady a zcela odlišnému zpracování přerušení Cortex-M3/M4 až dvojnásobně výkonnější než na procesorech *ARM7TDMI* [119]. Jednou z největších výhod také možnost využití MPU jednotky (Memory Protection Unit), která umožňuje definovat tzv. privelegované a neprivilegované oblasti adresního prostoru zahrnujícího samotný kód, RAM i periferie. MPU jednotkou však disponují jen některé Cortex-M3/M4 mikrokontroléry.

Speciálním případem je vybraný mikrokontrolér EFM32WG332F256, ARM Cortex-M4F, který dále rozšiřuje Cortex-M4 o 32 instrukcí integrací hardwarové FPU jednotky (Floating Point Unit). Tím je dosaženo rychlých výpočtů i s čísly s plovoucí desetinnou čárkou i v jediném cyklu. FPU jednotka podporuje aritmetické operace dle standardu IEEE 754. Přehled rychlosti výpočtů FPU jednotky je uveden v následující tabulce 4.7. Samozřejmostí je přetypování mezi celočíselnými formáty a čísly s plovoucí desetinnou čárkou v jediném cyklu.

Z pohledu výkonu tak procesory Cortex-M4F nemají v oblasti WSN konkurenci. Prozatím však nebyly řešeny energetické požadavky. Ty jsou společně se srovnáním s ostatními platformami předmětem kap. 6.1. Reálnému měření spotřeby energie v režimech spánku je pak věnována kapitola 6.2.

4.3.2 Návrh zabezpečení

Kryptografický návrh by měl ideálně zajistit utajení a integritu dat, nejlépe i autentizaci jednotek.

Pro zajištění utajení a integrity dat je využito algoritmu symetrické kryptografie AES, tedy služeb definovaných standardem IEEE 802.15.4. Konkrétně AES-CCM-32 s režií pouze 4 B. Nicméně standard IEEE 802.15.4 už nedefinuje proces distribuce klíčů a vzájemné autentizace. Pro zajištění šifrovacích klíčů, popř. vzájemné autentizace se dnes nabízí využití asymetrické kryptografie v podobě např. eliptických křivek nebo pomocí hybridního řešení, popsaného v autorově článku [37]. Přestože procesor ARM Cortex-M4 je dostatečně výpočetně výkonný pro implementaci náročných protokolů asymetrické kryptografie, navržený systém k autentizaci využívá opět symetrické kryptografie, konkrétně algoritmu SHA-256 a sdíleného tajemství. Podrobný popis je však nad rámec této práce.

Uvedený výpočet AES-128 či AES-256 přitom může být na navržené hardwarové platformě realizován bez intervence CPU. Procesor tak může být v době výpočtu v režimu spánku, čímž se dále zvyšuje energetická efektivita navrženého řešení. Dle autorových měření je tímto způsobem dosažena redukce spotřeby energie více než stonásobná [38]. Díky integraci akcelerátoru v MCU je možné využít šifrování dat, aniž by data opustila MCU a je tedy nemožné data v otevřené podobě odposlechnout skrze komunikační sběrnici².

Mikrokontrolér EFM32WG332 sice neobsahuje kryptograficky zabezpečenou část paměti, nicméně data uložená v interních pamětech (Flash, SRAM) lze uzamknout proti vyčtení. Nulováním slova DLW (Debug Lock Word) je přístup k jádru Cortex-M4 a systémové sběrnici přes paměťový přístupový port (AHB-AP) kontrolován autentizačním modulem AAP (Authentication Access Port), jak naznačuje obr. 4.4. Po této změně není debugger schopen přistupovat k jádru nebo systémovým sběrnicím, ale pouze k AAP registrům. Obnova přístupu přes AHB-AP ovšem způsobí, že obsah SRAM a Flash paměti je smazán.

Více je problematika bezpečnosti na úrovní softwaru probrána v kap. 5.3.1, případně v autorových článcích [38] a [37].



Obr. 4.4: Autentizační přístupový port AAP. Inspirováno [139].

 $^{2}\mathrm{SPI}$ nbeo UART

4.3.3 Komunikační subsystém

Pro zajištění bezdrátové mesh komunikace byl vybrán RF transceiver AT86RF233, komunikující v ISM pásmu 2,4GHz. RF transceiver je kompatibilní se standardem IEEE 802.15.4 a disponuje řadou HW akcelerovaných MAC operací (FCS, CSMA-CA, atd.). Mimo standard pak dovoluje využít přenosové rychlosti až 2000 kb/s. S parametry uvedenými v tab. 4.8 dosahuje tento transceiver aktuálně nejlepších hodnot spotřeby energie pro všechny sledované stavy - příjem, vysílání a režim spánku. Hodnoty tab. 4.8 vychází z katalogového listu výrobce [10].

Vlastnost	Hodnota
Spotřeba energie – příjem $[mW]$	$35,\!4$
Spotřeba energie – vysílání [mW]	41,4/ +4 dBm
Spotřeba energie – vysílání [mW]	35,4/ 0 dBm
Spotřeba energie – vysílání [mW]	21,6/ -17 dBm
Citlivost [dBm]	-101
Spotřeba energie – úsporný režim $[nW]$	60

Tab. 4.8: Vlastnosti transceiveru AT86RF233.

4.3.4 Návrh senzorického subsystému

Navrhovaný senzorový uzel je vyvíjen jako generická platforma pro široké spektrum aplikací. Všestranně využitelná platforma rozšiřuje oblast zájemců a v důsledku tak snižuje výrobní náklady. Jednoúčelové senzorické systémy se naopak vyznačují vysokou cenou, která je dána mj. maloobjemovou výrobou. Příkladem mohou být sněhoměrné váhy [64].

Digitální i analogové senzory je možné k EE-WSN připojit pomocí deseti pinového rozhraní, jenž může fungovat v roli digitální sběrnice USART (Universal Synchronous/ Asynchronous Receiver and Transmitter) či I²C (Inter-Integrated Circuit Interface), nabo jako 7 vstupů pro ADC (Analog to Digital Converter), 2 výstupy DAC (Digital to Analog Converter), 2 analogové komparátory, 2 čítače impulsů či jako operační zesilovač. Tímto způsobem je možné platformu rychle adaptovat k požadovaným účelům.

Pro demonstrační účely je využita rozšiřující deska obsahující foto tranzistor TEMT6200FX01 pro měření intenzity světla.



Obr. 4.5: Zapojení napájecí části.

4.3.5 Návrh napájecího subsystému

Srdcem napájecího subsystému je synchronní DC-DC Buck konvertor TI TPS62730, který je speciálně navržen pro nízko-příkonové RF transceivery. Tento obvod se vyznačuje vysokou účinností konverze napětí (až 95%) a velmi malým výstupním zvlněním. Unikátností je tzv. bypass režim, kdy je přes interní přepínač připojeno vstupní napětí na výstupní svorky obvodu a napájecí subsystém má v tomto režimu spotřebu proudu pouhých 30 nA. Bypass režim tak řeší hlavní nevýhodu DC-DC konvertorů, účinnost při nízkém odběru proudu, a lze tedy dosáhnout vysoké účinnosti se zátěží i bez. Tento režim je proto ideální pro podporu režimů spánku MCU a RF transceiveru.

Buck konvertor pomáhá snížit napětí v případě, že napětí baterie je vyšší než systém vyžaduje a tímto způsobem dokáže, dle měření na MSP430 [151], snížit spotřebu proudu až o 30%.

Aby mohl být operační systém informován o stavu napětí baterie, je přes odporový dělič připojena baterie k analogově digitálnímu převodníku MCU. Tímto způsobem by však docházelo ke ztrátám energie v době, kdy systém napětí neměří. Proto je zde využito spínače TS5A3159, který má v rozpojeném stavu spotřebu pouhé 2 nA a reakční čas 15 ns.

4.4 Hardwarová architektura vyvinuté platformy

Pro lepší představu o celkové architektuře vyvinuté platformy EE-WSN je zde uveden obr. 4.6. Detailní elektrotechnické schéma je přiloženo v příloze B. Jednotlivé části systému pak budou detailně popsány v následujících kapitolách.



Obr. 4.6: Hardwarová architektura vyvinuté platformy.

Je třeba zdůraznit, že precizní návrh architektury hardwaru je zcela zásadní. Výsledné výkonové a především energetické vlastnosti platformy jsou totiž přímo závislé na kvalitě návrhu. Je proto běžné, že i při využití stejných komponent lze dosáhnout velmi rozdílných výsledků. Jako příklad uveďme měření doby výpočtu AES (128b) algoritmu [50] na platformách MicaZ a TmoteSKY. Přestože obě platformy využívají k výpočtu shodný hardware, Chipcon CC2420, MicaZ je přibližně 15 krát rychlejší. Do určité míry jde samozřejmě i o kvalitní návrh firmware, o němž je zmíněno v následující kapitole 5.

Neméně důležitým aspektem jsou také pořizovací náklady, které jsou obecně pro jádra ARM Cortex-M příznivější, než pro 8/16-ti bitové MCU. Překvapivě je tak EE-WSN s výkonným mikrokontrolérem ARM Cortex-M4F více než čtyřikrát levnější ve srovnání s jedením z nejvyužívanějších MCU v oblasti WSN ATmega128L (využitý v platformách Cricket, BTnode, Mica2, Mica2Dot, MicaZ, Medusa MK-2). Konkrétně EFM32WG332F256-QFP64 stojí $3,5 \in$, přičemž ATmega128L-8AU $14,51^3 \in$.

 $^{^{3}\}mathrm{Cena}$ platná při odběru 10 ks, 13.7.2014 u distributora Mouser Electronics, www.mouser.com.



Obr. 4.7: Sběrnicový systém mikrokontroléru. Inspirováno [139].

4.4.1 Autonomní systém příjmu dat

Hlavní výhodou platformy EE-WSN je především velké množství hardwarových akcelerátorů a řady autonomních modulů (DMA, MPU), které jsou nejen v oblasti WSN unikátní. Vnitřní struktura sběrnic navíc umožňuje paralelní funkci nezávislých (nadřazených) sběrnic. Mikrokontroléry Cortex-M4 jsou vybaveny několika paralelními sběrnicemi, tvořícími matici AHB (Advanced High-performance Bus) sběrnic, které dovolují MCU využívat několik nadřazených sběrnic. A právě této vlastnosti je využito k návrhu nového energeticky efektivního přístupu pro přenos dat v bezdrátových senzorových sítích. Podle dostupných informací jde o inovativní a nepublikovaný způsob přenosu dat s výhledem na razantní snížení spotřeby energie, respektive nalezení zcela nové cesty. Jde o softwarové řešení, které se opírá o hardwarové možnosti navržené platformy EE-WSN. Dodejme, že popsané řešení není možné realizovat na žádné jiné z hardwarových platforem uvedených v této práci.

Systém sběrnic mikrokontroléru EFM32WG332 využívá systému AMBA (Advanced Microcontroller Bus Architecture) a tvoří vícevrstvou matici AHB sběrnic, které umožňují komunikaci nadřazených sběrnic s jejich protějšky (AHB slave). Výhodou je možný simultánní přístup k více podřízeným (AHB slave) modulům současně. Periferie jsou pak s AHB sběrnicovou maticí propojeny přes AHB/ APB (Advanced Peripheral Bus) můstek. Sběrnicový systém mikrokontroléru EE-WSN je vyobrazen na obr. 4.7.

Ačkoliv využití DMA (Direct Memory Access) modulu k přenosu dat není v oblasti vestavěných systémů novinkou, jde o ojedinělou vlastnost mezi uzly WSN. Především z důvodu striktních požadavků na spotřebu energie se v této oblasti využívá velmi jednoduchých MCU. Dedikované moduly jako DSP, AES či DMA jsou



Obr. 4.8: Diagram uspořádání bloku DMA. Inspirováno [139].

výhradou především vyšších architektur mikrokontrolerů, které se zároveň vyznačují vysokou spotřebou. Přesto se v minulosti objevil např. nízko příkonový mikrokontrolér MSP430F1611 s integrovaným DMA kontrolérem. Hardwarové platformy rodiny Telos, využívajících tohoto MCU, tak byly revolučním produktem. MSP430F1611 umožňuje blokový přenos dat a to z ADC registrů či I²C modulu do libovolného umístění, nebo naopak do DAC registrů. Jde tedy o ideální nástroj pro přenos snímaných dat do paměti MCU.

Autonomní systém příjmu dat však využívá DMA modulu odlišným způsobem. Maticová struktura AHB sběrnice platformy EE-WSN umožňuje při vhodném seskupení paměťových objektů využít paralelního přenosu dat s pomocí DMA modulu ve funkci mastera sběrnice. Je využito vhodného uspořádání periferií s bloky SRAM paměti tak, aby mohla být data mezi dvěma paměťovými místy přenášet bez intervence CPU. Konkrétně jde o nezávislý přenos dat mezi SRAM pamětí a RF transceiverem. To bohužel např. u MSP430F1611 není možné. Diagram uspořádání bloku DMA platformy EE-WSN je dostupný na obr. 4.8.

Navržený systém dále prohlubuje myšlenku využití autonomních periferií ve smyslu nezávislého řízení režimů RF transceiveru a to v přesně definovaných intervalech pomocí pomaluběžného časovače. Tyto intervaly samozřejmě vychází z využitého MAC protokolu, přičemž je předpokladem využití např. LPL či LPP technik. Při využití přesného pomaluběžného krystalu lze autonomní systém příjmu dat s výhodou využít i pro synchronní MAC protokoly.

Celý proces je navíc schopen pracovat naprosto samostatně, přičemž stav RF transceiveru je řízen taktéž autonomním způsobem, tedy bez generování přerušení, probuzení CPU a následné obsluhy přerušení. Tím se výrazně šetří energie a zároveň je docíleno vysoké časové přesnosti. Popsaný způsob řešení je závislý na systému PRS (Peripheral Reflex System), který umožňuje nezávislou vzájemnou komunikaci periferií až ve dvanácti kanálech. Periferie jsou rozděleny na tzv. producenty a konzumenty, viz obr. 4.9. Poznamenejme, že PRS komunikuje přes APB sběrnici, jejíž napojení je patrné na obr. 4.7.

I přes extrémně rychlé vyvolání obsluhy přerušení u mikrokontrolerů Cortex-M4, ve 12-ti hodinových CPU cyklech, je možné, že přerušení pro řízení RF transceiveru bude maskováno přerušením s vyšší prioritou a tudíž dojde k časovému posunu okna pro příjem dat od okolních uzlů. Takovéto chování nutně vedle k selhání deterministických procesů MAC protokolu a desynchronizaci celé sítě. Autonomní systém příjmu dat však využívá systému sběrnic dovolující nezávislé řízení RF transceiveru s rychlostí odezvy dvou hodinových cyklů a to i v případě, že CPU vykonává nemaskovatelné přerušení.



Obr. 4.9: Peripheral Reflex System. Inspirováno [139].

4.4.2 Popis konvenčního způsobu

Než bude přistoupeno k detailnímu popisu autonomního systému příjmu dat, bude na obr. 4.10 pro srovnání popsán průběh současného (konvenčního) způsobu přenosu dat na uzlu WSN. Horní osa vyjadřuje spotřebu energie MCU a y je rozdělena na tři energetické hladiny EM0 - EM2. Hladina EM0 značí aktivní režim CPU a všechny periferie mohou být aktivní. V režimu EM1 se nachází CPU v režimu

Tab. 4.9: Elektrický příkon mikrokontroléru EE-WSN.

Stav	Příkon $[\mu W]$
$\mathrm{EM0}^{a}$	32371
$\mathrm{EM1}^{a}$	9115
EM2	2,85
a HFXO = 48 MH	Z

spánku, paměťový systém je aktivní a veškeré periferie mohou být taktéž. EM2 pak značí režim spánku, kde je mimo CPU deaktivován i vysoko rychlostní zdroj taktu (HFXO/HFRCO) a veškeré na něm závislé periferie. Nízkofrekvenční oscilátor (LFXO/LFRCO) však zůstává aktivní, a proto je RTC a nízko příkonový časovač stále dostupný. Hladiny EM0 - EM2 vyjadřují jen stav systému a nikoliv spotřebu. Elektrický příkon v příslušných stavech MCU platformy EE-WSN je uveden v tabulce 4.9. Spodní osa y pak vyjadřuje spotřebu energie RF transceiveru, přičemž stav RF ON značí aktivní režim transceiveru a RF OFF jeho režim spánku.



Obr. 4.10: Časový průběh konvenčního přenosu dat.

Následuje popis jednotlivých akcí:

- Celý průběh začíná prvotní inicializací, při které dojde mj. ke konfiguraci protokolové sady a konfiguraci RF transceiveru, po které přechází radiová část do režimu spánku a nakonec se uspí i MCU (*EM2*).
- 2) V čase t_1 dochází k vypršení časovače a generaci přerušení, které probudí CPU (*EM0*). Obsluha přerušení probudí RF transceiver a celý systém čeká čas δ , zda nedojde k příjmu zprávy pro MAC adresu RF transceiveru. Po vypršení druhého komparátoru časovače nastává přerušení, po kterém dojde k deaktivaci RF transceiveru (přechod od režimu spánku) a MCU přechází do režimu spánku (*EM2*). Tento proces se neustále opakuje se střídou definovanou MAC protokolem.
- 3) V čase t_2 se opět na základě vypršení časovače probudí CPU (*EM0*) a RF

transceiver. Ovšem v čase t_3 přichází z RF transceiveru přerušení informující o příchozí zprávě. CPU zahájí přenos zprávy z vyrovnávací paměti RF transceiveru do RAM paměti MCU, přičemž přerušení od časovače je po dobu přenosu maskováno. Po přenosu zprávy dochází k vyvolání opožděné obsluhy přerušení, které uvede radiovou části i MCU do režimu spánku (*EM2*).

4) V čase t_3 je vyvoláno přerušení systémového časovače, které probudí CPU (*EM0*) a dochází ke zpracování úloh operačního systému. Průběh zpracování je přerušen přerušením, jehož postup je shodný s druhým bodem. Po zpracování všech systémových úloh přechází MCU do režimu spánku (*EM2*).

4.4.3 Demonstrace autonomního systému příjmu dat

Popis autonomního systému příjmu dat je předveden na stejném příkladu funkce WSN uzlu, jako v předchozí kapitole 4.4.2 a pro popis průběhu z obr. 4.11 tedy platí stejné podmínky pro spotřebu energie.



Obr. 4.11: Časový průběh autonomního příjmu dat.

Jednotlivé kroky jsou následující:

- 1) Průběh inicializace je shodný, a proto po konfiguraci protokolové sady a konfiguraci RF transceiveru přechází radiová část i MCU do režimu spánku (*EM2*).
- 2) V tomto bodě nastává zásadní rozdíl ve zpracování. V čase t_1 sice také dochází k vypršení časovače, avšak díky vnitřní struktuře sběrnic nedochází ke generování přerušení, ale je generován impuls do jednoho z kanálů systému PRS. Ten zajistí probuzení RF transceiveru, přičemž po vypršení druhého komparátoru časovače je generován další impuls, který zajistí přechod RF transceiveru do režimu spánku. Po celou dobu příjmu zpráv je CPU v režimu spánku (*EM2*).

- 3) V čase t_2 se opakuje obdobná posloupnost, jako v předchozím bodu. Po příjmu zprávy a následném generování přerušení RF transceiverem, v čase t_3 , je iniciován přenos zprávy z vyrovnávací paměti RF transceiveru do RAM paměti MCU pomocí DMA kontroléru (*EM1*). Zpráva je tak kopírována do paměti MCU bez intervence CPU. Po dobu kopírování jsou potlačena další přerušení, přičemž po dokončení přenosu zprávy jsou přerušení obnovena. Opožděná obsluha přerušení pak uvede radiovou části i MCU do režimu spánku (*EM2*).
- 4) Zpracování úloh operačního systému pak nastává opět na základě přerušení systémového časovače (t_4) .

Je zřejmé, že navržený systém má potenciál pro další razantní snížení spotřeby energie pro příjem dat v oblasti bezdrátových senzorových sítí. Srovnání výsledků reálného měření spotřeby tohoto a konvenčního systému je dostupné v kapitole 6.4.

5 RTOS S PODPOROU MESH KOMUNIKACE A ŘÍZENÍM SPOTŘEBY ENERGIE

Pro podporu přímého zpracování dat na senzorických uzlech je potřeba systém řízení, resp. operační systém, jenž by zohlednil přísné energetické nároky a podpořil svou funkcí energetickou efektivitu celého systému. Tato kapitola je věnována naplnění třetího z cílů práce, a tedy implementaci operačního systému reálného času (RTOS) pro bezdrátové mesh sítě s řízením spotřeby energie. RTOS reprezentuje výrazně sofistikovanější přístup než systémy využívajících stavových automatů a super smyček. RTOS ze své podstaty zajištuje strukturovaný kód aplikace, jež je dále podpořen přes RTOS API. Uplatněním struktury RTOS lze využít objektově orientovaného přístupu a přitom stále vyžívat prostého jazyka *C*. Společně s více vláknovou podporou tak umožňují RTOS systémy lepší správu a aktualizaci jednotlivých komponent, opětovné využití částí kódu a vylepšené testování. Na druhou stranu RTOS vyžadují zvýšené pamětové nároky, které ale pro platformu EE-WSN nepředstavují žádné omezení.

RTOS se skládá z plánovače, který standardně podporuje round-robin, preemptivní a kooperativní multitasking jednotlivých vláken. Komunikace mezi vlákny je pak zajištěna pomocí RTOS objektů, zahrnující signalizaci, semafory, mutexy či poštovní schránky (mailbox).

Základním stavebním článkem jsou vlákna, která jsou obdobou procedur jazyka *C*, avšak s několika rozdíly. Mezitím co procedury vždy končí, řekněme návratovou hodnotou, vlákna RTOS tvoří nekonečné smyčky. Tyto nezávislé smyčky tvoří kompletní aplikaci a jsou spravovány (spouštěny) plánovačem RTOS. Plánovač pak využívá systémového časovače MCU pro generování periodického přerušení, sloužící jako elementární časová jednotka operačního systému. Přepínání vláken je doprovázeno tzv. kontextovou změnou, tedy uložením stavu všech proměnných vlákna do příslušného zásobníku, společně s uložením informací o běhu vlákna do řídícího bloku vlákna a načtením stavu dalšího vlákna. Tento kritický proces je samozřejmě závislý na konkrétním jádře operačního systému a hardwaru.

Přestože kap. 1.3.1 informuje o dostupných operačních systémech pro WSN, žádný z nich nepodporuje architekturu ARM Cortex-M ani RF transceiver AT86RF233. Mimo to, žádný z dostupných operačních systémů pro WSN nenabízí správu řízení spotřeby ani podporu energetických režimů ARM Cortex-M procesorů. Z těchto důvodů bylo přistoupeno k vývoji vlastní implementace operačního systému pro WSN.

5.1 Podpora RTOS na EE-WSN

Mimo již zmíněných výhod platformy EE-WSN pro RTOS systémy ze str. 54 jsou zde popsány další přednosti.

5.1.1 CMSIS

Bezesporu největší výhodou využití jádra ARM Cortex-M je z pohledu softwaru možnost využití standardu CMSIS (Cortex Microcontroller Software Interface Standard). Ten za účelem snadné portace a opětovného využití kódu definovala samotná společnost ARM. CMSIS abízí soubor funkcí a maker pro přístup klíčových registrů a konfiguraci Cortex-M procesoru. Díky CMSIS je tedy možné vytvářet aplikace nezávislé na výrobci hardwaru a stejně tak využívat části kódů odlišných projektů, což samozřejmě značně urychluje vývoj aplikací. CMSIS standard je tvořen pěti specifikacemi: CMSIS core, CMSIS RTOS, CMSIS DSP, CMSIS SVD a CMSIS DAP. Standard tak nabízí i uniformní podporu RTOS systémů pomocí specifikace CMSIS RTOS.

5.1.2 Exkluzivní přístup a MPU

Klíčovou vlastností RTOS systémů je podpora multitaskingu, což dovoluje vyvíjet kód aplikace v nezávislých vláknech, které koncepčně běží paralelně. Běžně jednotlivá vlákna využívají sdílených prostředků MCU (periferie, RAM), a proto RTOS systémy nabízí pro řízení přístupu mechanizmy jako semafory či mutexy. Přestože jde podobné mechanizmy na úrovni operačního systému uplatnit u většiny MCU, procesory Cortex-M4 nabízí soubor instrukcí pro explicitní optimalizaci exkluzivního přístupu.

Mimo to je možné využívat hardwarové MPU jednotky, pro garantování privilegovaného přístupu k vybraným oblastem adresovatelného prostoru (4GB). Tím lze zcela diferencovat přístup operačního systému a uživatelské aplikace, což eliminuje pád celého systému, v případě selhání uživatelské aplikace. S výhodou se dá MPU využít i při vzdáleném upgradu systému, tzv. OTAU (Over The Air Upgrade), kde opět posiluje stabilitu systému.

5.2 Výběr jádra

Pro vývoj vlastní implementace operačního systému WSN byl mezi FreeRTOS, Keil RTX a Micrium uC/OS-II vybrán FreeRTOS [130]. Jde o systém velmi jednoduchý a pamětově nenáročný, avšak robustní systém, který je možné pro nekomerční účely využívat zdarma. Po operačních systémech Android a Ubuntu jde o nejrozšířenější operačních systém pro vestavěné systémy [26] a nabízí podporu 34 hardwarových architektur včetně ARM Cortex-M. Díky tomu FreeRTOS pojímá početnou komunitu vývojářů.

Jádro FreeRTOS nabízí dále preemptivní, kooperativní a hybridní multitasking jednotlivých vláken, podporu MPU na MCU ARM Cortex-M4, detekce přetečení aj. Výraznou výhodou je od verze 7.3.0 možnost využití tzv. *tickless módu*. Tickless mód řeší problém vysoké spotřeby energie u periodického generování přerušení i v případě, kdy se systém nachází ve stavu nečinnosti a nejsou tedy žádné úlohy ke zpracování. Řešením je potlačení periodických přerušení po dobu nečinnosti, respektive využití časovače pro probuzení systému za čas, kdy bude potřeba zpracovat další úlohu. Ilustrace běhu RTOS s a bez tickless módu ve stavu nečinnosti je zachycena na obr. 5.1.

5.2.1 Modifikované jádro

Pro implementaci je využito jádra modifikovaného Silicon Labs [140] s podporou energetických režimů mikrokontroléru platformy EE-WSN. Jádro je postavené na implementaci vPortSuppressTicksAndSleep() a jako zdroj času lze pro tickless mód využít místo systémových hodin RTC modul, který může být řízen RC oscilátorem. Tím je dosaženo extrémně nízké spotřeby a navíc je možné ponechat procesor i periferie v režimu spánku (EM2, viz kap. 4.4.2), aniž by systém ztratil pojem o čase. Klíčový je výpočet ulReloadValue, jež slouží jako vstupní parametr



Obr. 5.1: Srovnání běhu RTOS bez tickless módu (horní část) a s tickless módem (spodní část).

funkce RTC_CompareSet. Ten přímo ovlivňuje délku spánku MCU a je limitován maximální hodnotou definovanou makrem. Dojde-li v době spánku k vyvolání přerušení, přepočte se pomocí hodnoty RTC (ulRemainingCounter) aktuální čas systému.

V tomto kroku jádro RTOS obsahuje vysoce efektivní správu energie s podporou energetických režimů nízké spotřeby, avšak FreeRTOS postrádá jakoukoliv podporu komunikace v rámci WSN, a proto je nutné tuto funkcionalitu implementovat. Tomuto tématu je věnována následující kapitola.

5.3 Komunikační protokolová sada

Ačkoliv FreeRTOS nabízí IP komunikaci pomocí µIP či lwIP sady, díky absenci nižších vrstev tohoto nelze využít ke komunikaci mezi uzly WSN. Výběr vhodné komunikační sady se zde opírá o výsledky kap. 3, přičemž detailní návrh vhodné komunikační architektury je popsán v autorových pracech [40, 63, 107]. Stěžejní je podpora mesh komunikace, malé nároky na výpočetní a paměťový systém a efektivnost datového přenosu. Pro adaptaci tak vychází nejlépe protokolová sada LWM. Ta ovšem není ve stávající podobě pro vytyčené cíle příliš vhodná, a proto byla autorem modifikována.

5.3.1 Modifikace komunikační protokolové sady

V prvním kroku bylo nutné provést portaci LWM na vyvinutou platformu EE-WSN, což si díky nové hardwarové architektuře vyžádalo řadu úprav. Mimo definic pinů pro SPI komunikaci bylo nutné zprovoznit časovač, jenž by využíval hardwarové podpory EE-WSN. V souvislosti se změnou taktu procesoru dochází k ovlivnění vysokofrekvenčního zdroje časování periferií, a proto je konfigurace časovače prováděna pomocí před děličky dynamicky. Následuje výčet inovativních úprav. V rámci přehlednosti bude modifikovaná verze LWM označována jako EE-LWM.

Systém příjmu dat

Nejvýznamnější modifikací protokolové sady LWM je kompletní změna systému příjmu dat. V současné verzi 1.2.1 je prováděna opakovaná kontrola příchozích zpráv pomocí obslužné rutiny PHY_TaskHandler(), která je jednou ze tří funkcí, volaných v rámci SYS_TaskHandler(). Úlohou PHY_TaskHandler() je mj. vyčtení IRQ_STATUS registru RF transceiveru a kontrola třetího bitu, jež by indikoval dokončený příjem zprávy. Jednotlivé obslužné rutiny SYS_TaskHandler() se pak dále větví viz obr. 5.2. Jelikož jsou jednotlivé obslužné rutiny volány v rámci nekonečné smyčky, nutno dodat, že volání PHY_TaskHandler() není nikterak deterministické a funkce je volána až po obsluze všech následujících, respektive předcházejících funkcí. Když uvážíme, že samotná rutina SYS_TaskHandler() je také jedna z několika rutin nadřazené super smyčky, v níž je i např. APP_TaskHandler(), může se stát, že přijatá zpráva ve vyrovnávací paměti RF transceiveru bude přepsána nově příchozí zprávou, aniž by byly vyšší vrstvy informovány.

Modifikovaná verze EE-LWM využívá pro zpracování příchozích zpráv systému přerušení. Jakmile RF transceiver indikuje dokončení procesu přijetí zprávy, vygeneruje přerušení, které probudí MCU z režimu spánku a v rámci obsluhy přerušení IRQHandler() jsou data vyčtena z RF transceiveru do RAM paměti MCU. Po probuzení jádra EE-LWM pak dojde k naplnění struktury PHY_DataInd_t a běh protokolové sady pokračuje stejně, jako v původní verzi LWM. Systém přerušení je ilustrován na na následujícím obrázku 5.3.

Systém autonomního příjmu dat

Verze zahrnující systém autonomního příjmu dat z kap.4.4.1 pak k výše popsanému přenosu dat z RF transceiveru využívá DMA. Implementace byla doplněna o dvojici přerušení volaných nízko příkonovým časovačem, jak je patrné z obr.5.4. V této souvislosti musela být věnována zvýšená pozornost prioritám přerušení v rámci NVIC kontroléru EE-WSN.



Obr. 5.2: Sekvenční diagram funkce SYS_TaskHandler.



Obr. 5.3: Modifikovaný systém příjmu dat.

Zabezpečení komunikace

Původní verze LWM umožňuje využít k šifrování softwarovou implementaci algoritmu XTEA, nebo, umožňuje-li to RF transceiver, hardwarové podpory AES. Jak již ale bylo řečeno v kap. 4.3.2, data společně s heslem prochází přes SPI sběrnici nezabezpečeně a je velmi snadné takovéto zabezpečení prolomit. Algoritmus XTEA je z hlediska bezpečnosti i výpočetní náročnosti velmi dobrou volbou, nicméně navržená platforma EE-WSN obsahuje hardwarový AES akcelerátor s podporou délky klíče až 256 bitů. Jelikož je tento akcelerátor integrován do MCU, lze zamezit odposlechu hesla či dat přes komunikační sběrnici a navíc může být procesor po dobu výpočtu v režimu spánku. Společně s rychlostí výpočtu v 54 hodinových cyklech jde o významné zvýšení energetické efektivity.



Obr. 5.4: Implementovaný systém autonomního příjmu dat.

5.4 Implementace jádra a protokolové sady

Kritickým bodem je návrh architektury operačního systému, resp. způsob implementace komunikační protokolové sady. Hledání optimální varianty z pohledu energetické náročnosti bylo doprovázeno sérií reálných měření výkonu, počtu hodinových cyklů procesoru, pro periodické zpracování příchozích a odchozích zpráv, společně se zpracováním snímaných dat pomocí FFT. Více je ke způsobu a obsahu měření věnováno v kap. 7.

Optimální variantou je, dle provedených měření, rozdělení operačního systému do pěti následujících vláken:

- Řídící vlákno,
- Phy_TaskHandler,
- Sec_TaskHandler,
- Nwk_TaskHandler,
- App_TaskHandler.

Přičemž v závislosti na potřebách aplikace může být systém řízení času ponechán na interní implementaci FreeRTOS časovače, nebo lze využít i hardwarové podpory EE-WSN. Řídící vlákno využívá struktury ukazatelů funkcí obslužných rutin a mutex kontejneru. Úlohy jsou tedy zpracovány až na základě získání mutexu. Vlákno Phy_TaskHandler obsluhuje RF transciever, Sec_TaskHandler řeší šifrování pomocí integrovaného AES modulu, Nwk_TaskHandler má na starost směrování a přípravu dat k odeslání. Vlákno App_TaskHandler je pak určeno pro samotnou uživatelskou aplikaci.

Velmi efektivním je pak způsob implementace vyrovnávací paměti pro příchozí data. Ten pro potřeby FIFO fronty využívá sofistikované struktury systému poš-
tovních schránek (mailbox system), podporované CMSIS-RTOS. Alokovaná paměť je rozdělena do jednotlivých poštovních slotů, odpovídajících struktuře přijímaných zpráv. Po umístění příchozích dat do struktury je poštovní box uzamčen až do doby jeho vyčtení zpracovávajícím procesem. Ten vyčkává až na pokyn v podobě předání signálu a periodicky tak neblokuje čas procesoru. Navíc se tímto způsobem eliminují případy, kdy se proces spustí a data, např. vlivem přerušení s vyšší prioritou, ještě nejsou k dispozici.

Jak již bylo řečeno, demonstraci a ověření vlastností implementovaného operačního systému EE-RTOS je věnována kap. 7.

6 EXPERIMENTÁLNÍ OVĚŘENÍ HARDWAROVÉ PLATFORMY EE-WSN

V této kapitole jsou k dispozici detailní výsledky provedených měření výkonu a energetické spotřeby na platformě EE-WSN za reálných podmínek. Kapitola 6.1 je věnována zejména ověření splnění bodu d z vytyčených cílů této práce. Jde tedy o ověření energetické náročnosti výpočtů na vyvinuté platformě EE-WSN. Kapitola 6.3 se pak věnuje naplnění posledního z cílů, bodu e, a tady důkazu že senzorové uzly mohou být výpočetně nezávislé a dosáhnout nižší energetické náročnosti, než s pomocí distribuovaných výpočtů. Zejména pro bod d jsou dále důležité i energetické vlastnosti platformy v úsporných režimech, kterým je věnována kap 6.2. Poslední část, kap 6.4, prezentuje výsledky reálných měření spotřeby energie pro navržený systém autonomního příjmu, sloužící jako důkaz pro splnění cíle b, této práce.

6.1 Ověření výkonu

První představu o výkonu, potažmo energetické efektivitě ARM Cortex-M4 přináší graf na obr. 4.3, kap. 4.3.1. Přestože oba procesory ARM Cortex-M4 i Cortex-M3 nabízí srovnatelný výkon 1,25 DMIPS/ MHZ, výsledky v grafu dokazují výrazně lepší podporu matematických operací jádra Cortex-M4. To je dáno mimo jiné i výkonnější MAC_{math} jednotkou, schopnou pracovat s 32-bitovými slovy, jejichž výsledkem je 64-bitové číslo. Mimo to Cortex-M4 nabízí i skupinu SIMD instrukcí, DSP a FPU modul.

Pro demonstraci výkonu a srovnání energetické efektivity vezměme například výpočet rychlé Fourierovy transformace, respektive přímé diskrétní Fourierovy transformace (DFT, Discrete Fourier Transform) posloupnosti délky N, definováné vztahem [141]:

$$S[k] = \sum_{n=0}^{N-1} s[n] e^{-jk\frac{2\pi}{N}n}, \qquad k = 0, 1, ..., N-1.$$
(6.1)

V této kapitole jsou k dispozici výsledky výkonu prototypových jednotek (viz obr. 6.1) se zmíněnými procesory ARM Cortex-M4F (EFM32WG990) a ARM Cortex-M3 (EFM32G890) a zástupce 8/16-ti bitových platforem Iris, obsahující MCU Atmel ATmega 1281. Algoritmy zpracování signálů patří typicky mezi výpočetně náročné a jsou proto ideální pro demonstraci nových možností platformy EE-WSN.



Obr. 6.1: Prototypové jednotky ARM Cortex-M4F (horní část) a Cortex-M3 (spodní část).

Pro výpočty FFT na procesorech ARM Cortex-M4F a ARM Cortex-M3 bylo využito knihovny CMSIS-DSP. Konkrétně šlo o algoritmy CFFT (Complex Fast Fourier Transform) _q15, jež využívají pro své výpočty kombinace *radix-2* a *radix-4* algoritmů. Z označení algoritmů je patrné, že výpočty jsou prováděny nad vstupními daty s pohyblivou řádovou čárkou. Jako vstupní signál byla využita data testovacího souboru arm_fft_bin_data.c knihovny CMSIS-DSP. Data byla převedena do formátu Q15 s dvojkovou čárkou. V případě procesoru ATmega 1281 byl využit algoritmus [23], který je silně optimalizován pro rychlost, a proto jsou jeho části psány v assambleru. Vstupem mohou být pouze data s pevnou desetinnou čárkou o šířce 16 bitů. Vstupní signál byl uložen pomocí struktury z dvojice int_16t reprezentující komplexní tvar. Je zřejmé, že díky operacím nad daty s pevnou desetinnou čárkou jsou výpočty na ATmega 1281 méně náročné.

Srovnání spotřeby pro různé parametry FFT (počet bodů) je k dispozici v tabulce 6.1. Pro lepší orientaci jsou výsledky prezentovány i formou grafu na obr. 6.2. Prezentované výsledky nejsou zcela překvapivé, protože ATmega 1281 není výkonově orientovaným procesorem a řada instrukcí je v případě ARM Cortex-M procesorů vykonána v jediném cyklu. Až dvojnásobný rozdíl ve výkonu mezi ARM Cortex-M4F a ARM Cortex-M3 je způsoben sadou SIMD instrukcí, které, jak již bylo zmíněno, umožňují ARM Cortex-M4 paralelní zpracování dvojice 16-ti bitových dat. Díky lepší energetické efektivitě jádra EFM32G890, na měřené frakvenci 16 MHz, však rozdíl ve spotřebě energie není natolik rozdílný.

FFT	ARM Cortex-M4 a	ARM Cortex-M 3^a	ATmega 1281^b
	Energie $[\mu J]$	Energie $[\mu J]$	Energie [µJ]
N = 64	2,96	6,64	357
N = 128	$6,\!10$	$13,\!68$	767
N=256	12,81	28,73	1670
N = 512	27,68	62,06	3591

Tab. 6.1: Srovnání spotřeby energie pro různé parametry FFT.

 a HFXO = 16 MHz, 3 V. Kompilováno pomocí GNU ARM v
4.7.3.

 b HFXO = 16 MHz, 5 V. Kompilováno pomocí AVR/GNU GCC v4.3.3.

Jak bylo možné předpokládat, procesor ARM Cortex-M4F nabízí z měřených architektur nejvyšší výpočetní výkon a v rámci provedených výpočtů i nejvyšší energetickou efektivitu. Ve srovnání s ATmega 1281 dosahuje ARM Cortex-M4F až **130 krát** nižší spotřeby energie pro výpočet 256-ti bodové FFT. Vzhledem k rychlosti výpočtů tak zůstává kritickým parametrem spotřeba energie v režimu spánku.

6.2 Spotřeba energie v režimu spánku

Z principu fungování bezdrátových senzorových sítí (viz kap. 1) plyne důležitost minimální spotřeby uzlů v úsporných módech, neboť zde mohou, v závislosti na potřebách aplikace, setrvat více jak 99% provozní doby uzlu[143]. Časový průběh aplikací se zpracováním signálů, či jinou výpočetně náročnou operací, je však spe-



Obr. 6.2: Srovnání spotřeby energie pro různé parametry FFT.

cifický. Jelikož výpočty na uzlech WSN trvají i několik sekund¹, senzorický uzel by musel trávit před zpracováním další dávky dat i desítky minut v úsporném režimu. To ale aplikace často nedovolují, protože vyrovnávací paměť uzlu pro snímané hodnoty nedokáže pojmout velké množství dat. Navýšení jak externí, tak i interní paměti však nutně vede ke zvýšeným energetickým nárokům. Proto musí výpočty probíhat buď nepřetržitě, nebo jen s velmi malou střídou. Je-li ale bezdrátový uzel dostatečně výkonný, může přes vysokou výpočetní zátěž využívat režimů spánku.

Elektrický příkon pro režimy spánku tří srovnávaných architektur z kap. 6.1 je uveden na obr. 6.3 a obr. 6.4. *Úsporný mód A* představuje *Sleep* mód ARM Cortex-M3/4 a přibližně odpovídá *Idle* módu ATmega 1281, kde je procesor deaktivován, ale paměťový systém a veškeré periferie jsou dostupné. *Idle* mód u ATmega 1281 však zahrnuje i deaktivaci Flash paměti. Tohoto režimu EE-WSN využívá při přenosu dat mezi MCU a RF transceiverem.

Úsporný režim B pak reprezentuje Deep Sleep mód ARM Cortex-M3/4 a ekvivalentní ADCNRM mód ATmega 1281. Pro ADCNRM však chybí bližší specifikace, a proto je v tabulce uveden nižší režim – Standby mód. V Deep Sleep režimu může EE-WSN díky stále aktivnímu časovači setrvat přesně definovaný čas a využívá jej jako režim spánku.

Z uvedených výsledků jednoznačně vyplývá, že prototypové jednotky EE-WSN, at už ve variantě ARM Cortex-M3 či ARM Cortex-M4, dokáží v režimech spánku směle konkurovat "ultra low-power" mikrokontrolerům. V rámci *úsporného módu A*

¹Např. výpočet privátního klíče RSA-2048 na ATmega128 (7,37 MHz) trvá 83,26 sekund [122].



Obr. 6.3: Srovnání el. příkonu pro úsporný mód A.



Obr. 6.4: Srovnání el. příkonu pro úsporný mód B.

nabízí EE-WSN jednotky až **11 krát** nižší příkon a pro variantu *úsporného módu B* dokonce **24 krát** nižší příkon oproti představiteli 8-bitových řešení – jednotce Iris. Nízký el. příkon v úsporných módech byl přitom vždy výluční doménou 8/16-ti bitových MCU a díky tomu si své postavení udrželi dodnes. Těmito výsledky mj. popíráme závěry rozsáhlé studie [84], která jako zásadní a nepřekonanou výhodu nízko příkonových 8/16-ti bitových MCU, oproti ARM Cortex M3, uvádí právě nízký příkon v režimu spánku. Společně s výsledky předchozí kap. 6.1 lze potvrdit, že senzorové uzly nutně nemusí disponovat výpočetně a paměťově omezenými mikrokontroléry, aby dosáhly nízké spotřeby energie, a to v aktivním i úsporném režimu. Totiž, v obou zmíněných režimech platforma EE-WSN dosahuje výrazně lepších výsledků, než současné senzorové uzly. Přitom nabízí dostatečný výkon pro

zpracování např. FFT, FIR, IIR algoritmů s podporou DSP instrukcí a FPU jednotky, podporující výpočty nad čísly s plovoucí řádovou čárkou.

Dodejme, že jednotka Iris obsahuje inovovaný mikrokontrolér ATmega 1281, který vychází z ATmega128L, ale nabízí přibližně o třetinu nižší spotřebu energie v aktivním režimu (16 MHz / 5 V) a třikrát nižší spotřebu v úsporných režimech (*Idle a Power-down*).

Na druhou stranu nízké hodnoty spotřeby energie v úsporných módech platformy EE-WSN nejsou vymezovány samotnou architekturou ARM Cortex-M4, neboť jiné implementace dosahují výrazně odlišných výsledků. Při využití např. STM32F4 [142] by se příkon v režimu spánku (*Stop mode*) zvýšil více než **490**² krát.

6.3 Přímé zpracování dat na uzlu WSN

Jak již bylo naznačeno, v této části jsou srovnány techniky uvedené v kap. 1.4 a přímým přístupem, blíže specifikovaným v kapitole 6.1. Přestože myšlenka přímého zpracování dat na uzlech senzorové sítě není nová, lze předpokládat, že platforma EE-WSN umožní lokální zpracování s nižšími energetickými nároky, než v případě distribuovaných výpočtů. Pro srovnání je tedy opět využito FFT algoritmů, konkrétně algoritmu UPAD (viz kap. 1.4), jehož vyčíslení náročnosti, pro několik různých případů, je uvedeno v tab. 6.2 a tab. 6.3.

Tab. 6.2: Náročnost algoritmu UPAD FFT, jeden hop.

\mathbf{FFT}	počet přenesených paketů	počet sčítání	počet součinů
N = 64	384	384	192
N = 128	896	896	448
N=256	2048	2048	1024
N = 512	4608	4608	2304

Tab. 6.3: Náročnost algoritmu UPAD FFT, multi-hop.

FFT	počet přenesených paketů	počet sčítání	počet součinů
N = 64	1344	1344	672
N = 128	5419	5419	2709
N = 256	21760	21760	10880
N = 512	87211	87211	43605
h=3			

 2 Napětový regulátor v režimu Low Power, Flash paměť v režimu Stop mode.

Tab. 6.4: Náročnost operací UPAD FFT na senzorovém uzlu EE-WSN.

Operace	Spotřeba energie $[\mu J]$
Přenos paketu (MAC protokol, 4B, L2 ACK, 0 dBm)	41377,50
Sčítání či odečítání 16 b kompl. čísel (MCLK = $48 \rm MHz)$	$0,\!67$
Násobení 16 b kompl. čísel (MCLK = $48 \rm MHz)$	$0,\!67$

Aby byly ve vyjádření spotřeby energie algoritmu UPAD FFT omezeny režijní ztráty, lze vycházet z energetické náročnosti jednotlivých operací a celkovou spotřebu získat jejich součtem:

$$E = E_{prenosu} + E_{scitani} + E_{soucinu}.$$
(6.2)

Spotřeba energie těchto operací je samozřejmě platformě závislá a pro srovnání je využito reálného měření operací na vyvinuté platformě EE-WSN. Vzhledem k extrémně krátkému času výpočtů jsou hodnoty spotřebované energie operací komplexního násobení, sčítání a odečítání odvozeny jako podíl více shodných výpočtů v čase. Energetická náročnost jednotlivých operací UPAD FFT je vyjádřena v následující tabulce 6.4. Výsledné hodnoty celkové spotřeby energie jsou pak k dispozici v tabulce 6.5. Poznamenejme, že jde stále o teoretické hodnoty, protože hodnoty tab. 6.2 a tab. 6.3 nezohledňují saturaci přenosového kanálu, opakovaný přenos zpráv při interferenci či zarušení kanálu. Nereálnost podtrhuje požadavek až 512-ti bezdrátových uzlů a to ve vzájemném radiovém dosahu (tab. 6.2).

S využitím výpočetních možností platformy EE-WSN, lze snadno dokázat její energetickou efektivitu při počítání všech bodů, resp. sekcí, FFT na jediném uzlu (viz tab. 6.1). V kontrastu s distribuovanými algoritmy výpočtu FFT vykazuje přímé zpracování pomocí knihovny CMSIS-DSP na platformě EE-WSN v průměru **6236 krát** nižší spotřebu energie pro případ přímé komunikace a **64047 krát** pro případ multi-hop sítě s hloubkou h=3. Srovnání spotřeby energie pomocí grafu je v tomto případě zbytečné. Zdůrazněme, že jde o srovnání dvou rozdílných přístupů zpracování signálů na uzlech bezdrátové senzorové sítě a to na stejné platformě a za rovnocenných podmínek.

Uvedenými výsledky tak popíráme závěry [1, 20, 21, 25], které tvrdí, že distribuované výpočty poskytují flexibilitu, spolehlivost, ale především zvýšení energetické efektivity. Tímto také potvrzujeme hypotézu H_1 ve stanovených cílech práce.

	Přímé zpracování	UPAD FFT (jeden hop)	UPAD FFT (multi-hop, h=3)
\mathbf{FFT}	Spotřeba energie $[\mu J]$	Spotřeba energie $[\mu J]$	Spotřeba energie $[\mu J]$
N = 64	2,96	$15889,\!35$	55612,72
N = 128	6,10	$37075,\!15$	$224216,\!36$
N = 256	12,81	$84743,\!19$	$900396,\!41$
N=512	27,68	$190672,\!18$	$3608647{,}58$

Tab. 6.5: Spotřeba energie pro výpočet FFT na uzlu EE-WSN.

* MCLK = 16 MHz, 3,0V

6.4 Autonomní systém příjmu dat

Pro posouzení energetické efektivnosti navrženého systému autonomního příjmu dat z kap. 4.4.1 bylo využito modifikované protokolové sady LWM, tedy EE-LWM (viz kap. 5.3.1), doplněné o implementaci zmíněného systému z kap. 5.3.1. Tato kapitola tak experimentálně ověřuje navržený systém pro splnění cíle b, této práce.

Tento scénář demonstruje na platformě EE-WSN periodicky přijímání zpráv o celkové velikosti 127 Byte. V prvním případě jde o konvenční způsob popsaný v kap. 4.4.2. a druhý způsob zohledňuje implementaci autonomního systému. V obou případech je měřena mikrokontrolérem spotřebovaná energie pomocí technologie Advanced Energy Monitor a programu energyAware Profiler. Energie spotřebovaná RF transceiverem je v obou případech shodná. Procesor byl taktován na 48 MHz a napájecí napětí bylo 3,3 V.

Průběh spotřeby energie pro konvenční příjem dat je zachycen na obr. 6.5, přičemž průměrná spotřeba energie dosahuje 349,49 µJ. Průběh spotřeby energie s využitím autonomního systému přenosu dat je na obr. 6.6 a průměrná spotřeba energie odpovídá 41,82 µJ. Navržený systém tedy vykazuje ve srovnání s běžným způsobem příjmu **8,36 krát** nižší spotřebu energie a dokáže tak **ušetřit přes 88** % energie. V důsledku se jedná o výrazné zvýšení energetické efektivity a tedy o nalezení energeticky efektivního přístupu pro přenos dat ve WSN, čímž došlo ke splnění cíle *b*, této práce.



Obr. 6.5: Průběh spotřeby energie pro konvenční příjem dat.



Obr. 6.6: Průběh spotřeby energie pro autonomní příjem dat.

7 EXPERIMENTÁLNÍ OVĚŘENÍ SYSTÉMU EE-RTOS

Za účelem demonstrace a ověření vlastností implementovaného operačního systému EE-RTOS byla vytvořena aplikace, jenž by měla představovat zvýšené zatížení. Foto tranzistorem je s frekvencí 1 kHz snímána intenzita okolního osvětlení a každých 0,5 sekund je nad získanými vzorky vypočítána 512-ti bodová FFT, společně s algoritmem pro odhad frekvence s největší energií ve spektru (mimo DC složky) pomocí Sinc interpolace. Tato informace je pak odesílána sousednímu uzlu. Samozřejmě v praxi by bylo vhodné výsledky agregovat do větších celků, nicméně v tomto případě jde o modelový příklad, a proto i velikost odesílané zprávy je vyplněna do 127 Byte. Jelikož aplikace využívá autonomního systému příjmu dat, RF transceiver je periodicky přepínán do režimu příjmu.

Aby bylo možné dosažené výsledky srovnat, byla tato aplikace připravena i v modifikované verzi LWM, tedy EE-LWM. Lze předpokládat, že aplikace bez operačního systému reálného času (EE-LWM) bude vykazovat nižší výpočetní náročnost, neboť je oproštěna od režie RTOS. Ta se projeví např. s každým přerušením pro získání nových dat z foto tranzistoru, tedy s frekvencí 1kHz. Nutno ale dodat, že aplikace (pod EE-LWM) je striktně jednoúčelová a nedisponuje řadou výhod RTOS systémů (viz kap. 5).

Srovnání průběhu spotřeby energie mikrokontrolerem je možné z obr. 7.1 a obr. 7.2. V grafech jsou patrny delší okamžiky zpracování dat pomocí FFT a periodické špičky, které jsou způsobeny systémem autonomního příjmu. Šum připomínající oblast mezi zpracováním dat je spotřeba energie obsluhy přerušení pro snímání a ukládání vzorků z foto tranzistoru. U obou implementacích pracoval procesor s rychlostí taktu 24 MHz a napájecím napětím 3,3 V, přičemž měřená data jsou získána pomocí technologie Advanced Energy Monitor a programu energyAware Profiler.



Obr. 7.1: Průběh spotřeby energie pro FFT aplikaci EE-RTOS.



Obr. 7.2: Průběh spotřeby energie pro FFT aplikaci EE-LWM.

Integrací získaná spotřeba energie pro jeden běh programu (vzorkování dat a výpočet FFT) 2346 J pro EE-LWM a 4122 J pro EE-RTOS potvrzuje zvýšenou režii běhu RTOS. Nicméně s ohledem na extrémní charakter aplikace a nabízené vlastnosti jde o výborný výsledek. Je nutné upozornit, že ani v jednom případě nešlo o optimalizovaný program a šlo o záměrné testování limitních podmínek. Důležitým závěrem je poznatek, že přestože taktovací frekvence procesoru dosahovala jen poloviny jeho maxima, obě implementace potvrdily bezproblémový chod. Rychlost zpracování a odeslání dat tedy není u obou implementace limitující a dokáží v mezičase využívat režimů spánku.

8 DISKUSE

Jelikož se tato práce zabývá zpracováním dat na uzlech bezdrátové senzorové sítě, bylo nutné pro ověření hypotéz a výsledků zajistit hardwarové i softwarové prostředky. Nejprve byla analyzována dostupná softwarová řešení v podobě operačních systémů Contiki OS, TinyOS a ZigBee Pro, z nichž z pohledu energetické efektivity nejlépe vyhovovala komunikační sada ZigBee Pro [109]. Výsledky reálných měření přinesly několik nových poznatků o chování tohoto systému v praxi. Široký potenciál systému LWM byl identifikován ihned po vydání jeho první beta verze, kdy započalo jeho testování. To vyústilo ve srovnávací analýzu těchto systémů, tedy konkrétně implementace ZigBee Pro – BitCLoud a LWM (viz kap.3).

Již na počátku analýzy vyšel z pohledu režijní zátěže jako optimální řešení právě LWM. Nízká datová režie se nutně projevila ve vysoké datové propustnosti, kde ve všech testovaných scénářích LWM poráží BitCloud. Velmi odlehčená implementace mesh komunikace se podepsala na rychlé odezvě systému i na 8-bitových MCU, nízkém zpoždění vlivem směrování a nízkých výpočetních a paměťových nárocích. V rámci testování samo opravné schopnosti sítě sice LWM vykazuje kratší čas zotavení, nicméně datová režie pro nalezení alternativní cesty je pro všechny případy vyšší. V souvislosti s hledáním cesty k cíli pomocí broadcast zpráv lze identifikovat několik slabých míst implementace LWM, které lze optimalizovat např. pomocí časovačů (viz [41]). Nicméně pro další vývoj představuje LWM ideální cestu.

Jak již bylo řečeno, nutnost vývoje vlastní hardwarové platformy vyvstala z faktu, že jak na komerčním trhu, tak v současné literatuře není dostupná platforma, splňující požadavky kap. 4.1. Bylo přistoupeno ke kontroverznímu řešení v podobě využití procesoru ARM Cortex-M4F. Ten představuje nejvýkonnější procesor ARM Cortex procesorů a kombinuje vlastnosti MCU a výkonných DSP. Disponuje celou řadou hardwarových modulů, včetně DMA, MPU a FPU jednotky, které ve finále umožňují vykonávat např. náročné algoritmy signálového zpracování i nad daty s plovoucí řádovou čárkou. Experimentální ověření výkonu z kap. 6.1 tak není ničím překvapivým, neboť pro tyto náročné operace byl procesor navržen. Druhou stranou mince jsou však energetické nároky, které v případě jediné známé platformy s ARM Cortex-M4F [74] představují nejhorší energetické vlastnosti režimu spánku ze všech uvedených platforem. Stejně tak je tomu i s jinými implementacemi ARM Cortex-M4, např. STM32F4 [142], kde je příkon v režimu spánku až 490 krát vyšší, než u navržené platformy EE-WSN. Tím se dostáváme k hardwarové architektuře EE-WSN, která byla navržena pro těsnou součinnost se softwarem a tvoří tak neoddělitelný celek. Řeč je především o vhodném pamětovém uspořádání vzhledem k využívaným periferiím a využití několikavrstvé maticové sběrnice AHB. Tato sběrnicová struktura pak společně se systémem PRS umožňuje mj. realizaci navrženého autonomního systému příjmu dat (viz. kap. 4.4.1). Důležitost vhodného návrhu hardwarové architektury podtrhují např. výsledky [50], z kterých je patrné, že při využití stejných komponent je možné dosáhnout i patnáctinásobně rozdílných rychlostí zpracování dat.

Navržená platforma EE-WSN je unikátní především v poskytovaném výpočetním výkonu a možnosti využití autonomních periferií. Přesto však dosahuje velmi nízké spotřeby i v režimu spánku, konkrétně 24 krát méně než platforma Iris.

Návrh systému autonomního příjmu dat patří ke stěžejním částem celé práce, na němž stojí splnění cíle *b* a v širším smyslu i *d*. Přestože práce hovoří o systému příjmu dat, obdobný mechanizmus je využíván i k odeslání dat. Nicméně vzhledem k tomu, že odesílání dat většinou iniciuje samotný operační systém, není potřeba využívat autonomní systém v plném rozsahu, ale pouze k přesunu dat do vyrovnávací paměti RF transceiveru pomocí DMA. Reálná měření na platformě EE-WSN ukázala pro příjem stejné zprávy 8,36 krát nižší spotřebu energie pomocí autonomního příjmu oproti běžnému způsobu. Tímto způsobem lze tedy ušetřit až 88% el. energie. Dodejme, že realizace tohoto systému je díky jedinečné architektuře možná pouze na platformě EE-WSN.

Jako další podporu přímého zpracování dat na senzorických uzlech byl prezentován systém EE-RTOS a modifikovaná verze LWM, EE-LWM. První ze jmenovaných dovoluje plně využívat výhod RTOS, z kterých mohou těžit jak aplikace, v podobě přesné časové synchronizace, tak samotní aplikační inženýři, kteří se nyní nemusí zabývat nižšími vrstvami systému a mohou se plně soustředit na vývoj specifické aplikace. Ta navíc může být díky systému EE-RTOS jednou z několika simultánně běžících aplikací na senzorovém uzlu, ovšem za cenu mírně zvýšené režie. Systém EE-LWM pak představuje velmi lehkou implementaci mesh protokolu s podporou nízko energetických vlastností platformy EE-WSN. Nejvýznamnějším rozdílem obou implementací je především nedeterminičnost časových událostí v případě EE-LWM, což může mít v určitých aplikacích (např. ve zdravotnictví) fatální následky. V případě vzorkování může docházet k aliasingu a funkce synchronních MAC protokolů je značně omezena. Problémy způsobuje proměnná aktivní doba i asynchronním MAC protokolům (např. Contiki-MAC s phase-lock optimalizací). Oba uvedené systémy pak plně podporují úsporné režimy platformy EE-WSN a mesh komunikaci v rámci bezdrátové sítě.

S odkazem k dosaženým výsledkům kapitoly 6.3 lze potvrdit i hypotézu H_1 a zároveň tak popřít závěry prací [1, 20, 21, 25], které tvrdí, že distribuované výpočty poskytují flexibilitu, spolehlivost, ale především zvýšení energetické efektivity.

9 ZÁVĚR

Tato práce se zabývá problematikou bezdrátových senzorových sítí se zaměřením na aplikace vyžadující vysoký výpočetní výkon. Pro uspokojení specifických potřeb těchto aplikací byla představena hardwarová platforma EE-WSN, představující možnou odpověď při hledání současně neslučitelných požadavků na výpočetní výkon a nízkou spotřebu energie. Společně s navrženým systémem autonomního příjmu dokáže významným způsobem zvýšit energetickou efektivitu celého systému. Pomocí série měření byl potvrzen dostatečný výkon platformy, přičemž jeho energetická náročnost je při výpočtu 256-ti bodové FFT až 130 krát nižší, ve srovnání s platformou Iris. Stejně tak v režimu spánku dosahuje platforma až 24 krát nižší spotřeby. Autonomní systém příjmu dat pak při příjmu paketu vykazuje až 8,36 krát nižší spotřebu energie oproti běžnému řešení na téže platformě. V neposlední řadě byla představena dvojice kompletních softwarových řešení. Operační systém reálného času EE-RTOS a modifikovaná verze LWM, tedy EE-LWM. Oba uvedené systémy pak plně podporují úsporné režimy platformy EE-WSN a mesh komunikaci v rámci bezdrátové sítě.

Měřením byly potvrzeny dosažené cíle, včetně hypotézy H_1 ze str. 28 a bylo tedy dokázáno, že uzly bezdrátové senzorové sítě mohou být dostatečně výkonné pro přímé výpočty algoritmů signálového zpracovaní, jako např. FFT či aplikace FIR a IIR filtrů, a zároveň vykazují nízkou spotřebu v aktivním i úsporném režimu. Ve srovnání s prezentovaným řešením pak představují distribuované výpočty zcela energeticky neefektivní řešení.

Nutno dodat, že navržený WSN systém profituje z těsné součinnosti hardwaru a softwaru/ firmwaru, bez níž lze jen těžko pokračovat v dalším zvyšování energetické efektivity bezdrátových senzorových sítí. Cenným zjištěním je tedy poznatek, že další zvýšení energetické efektivity WSN systémů vede přes hlubokou znalost dílčích oblastí a holistický přístup při návrhu softwarové a hardwarové architektury.

Výsledky tohoto výzkumu nachází přímé uplatnění v průmyslu i akademické

sféře, kde lze hardwarovou platformu společně s operačním systémem bez dalších úprav či znalosti fungování WSN využít. To řeší běžný problém, kdy senzorové sítě využívají experti jiných oblastí výzkumu, jako biologové, geologové či stavební inženýři. Systém autonomního přenosu dat pak nabízí cestu pro další vývoj snižování energetické náročnosti bezdrátových systémů. Uvedené návrhy byly publikovány na konferencích i časopisech s impakt faktorem, viz str.119.

LITERATURA

- Abdelhak, S.; Ghosh, S.; Tessier, J.; Bayoumi, M.; ; aj.: Resource-aware Distributed Split Radix FFT on Wireless Sensor Networks. In *High Performance Embedded Computing, Massachussetts*, 2009.
- [2] Akhavan, M. R.; Watteyne, T.; Aghvami, A.: Enhancing the performance of RPL using a Receiver-Based MAC protocol in lossy WSNs. In *Telecommunications (ICT)*, 2011 18th International Conference on, 2011, s. 191–194.
- [3] Alippi, C.; Camplani, R.; Roveri, M.; Viscardi, G.: Netbrick: A high-performance, low-power hardware platform for wireless and hybrid sensor networks. In *Mobile Adhoc and Sensor* Systems (MASS), 2012 IEEE 9th International Conference on, 2012, s. 111–117.
- [4] ARM Holdings. ARM7 Processor Family. [Online]. 2014.
 URL http://www.arm.com/products/processors/classic/arm7/index.php
- [5] ARM Holdings. big.LITTLE Processing. [Online]. 2014. URL http://www.arm.com/products/processors/technologies/ biglittleprocessing.php
- [6] ARM Holdings. Classic Processors. [Online]. 2014. URL http://www.arm.com/products/processors/classic/index.php
- [7] ARM Holdings. Cortex-M4 Processor. [Online]. 2014.
 URL http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php
- [8] ARM Holdings. Cortex-M4 Processor Technical Reference Manual. [Online]. 2014. URL http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0439d/ index.html
- [9] Atmel Corporation. ATmega128. [Online]. 2012.URL http://www.atmel.com/devices/atmega128.aspx
- [10] Atmel Corporation. AT86RF233. [Online]. 2014. URL http://www.atmel.com/Images/Atmel-8351-MCU_Wireless-AT86RF233_ Datasheet.pdf
- [11] Atmel Corporation, Atmel AVR2130: Lightweight mesh developer guide, Application note.
 [Online]. 2014.
 URL http://www.atmel.com/tools/LIGHTWEIGHT_MESH.aspx?tab=documents

- [12] Bachir, A.; Dohler, M.; Watteyne, T.; Leung, K.: MAC Essentials for Wireless Sensor Networks. *Communications Surveys Tutorials, IEEE*, ročník 12, č. 2, 2010: s. 222–248, ISSN 1553-877X.
- [13] Bertrand, A.: Signal Processing Algorithms for Wireless Acoustic Sensor Networks. Dizertační práce, KU Leuven, University of Leuven, Belgium, 2011.
- [14] Bhatti, S.; Carlson, J.; Dai, H.; Deng, J.; Rose, J.; aj.: MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. *Mob. Netw. Appl.*, ročník 10, č. 4, 2005: s. 563–579, ISSN 1383-469X.
- [15] Bischoff, R.; Meyer, J.; Feltrin, G.: Wireless Sensor Network Platforms. John Wiley & Sons, Ltd, 2009, ISBN 9780470061626.
- [16] Borkar, A.; Ranjan, P.: Optical wireless sensor network design for a conducting chamber. In Local Computer Networks (LCN), 2011 IEEE 36th Conference on, 2011, ISSN 0742-1303, s. 990–993.
- [17] Boyinbode, O.; Le, H.; Mbogho, A.; Takizawa, M.; Poliah, R.: A Survey on Clustering Algorithms for Wireless Sensor Networks. In Network-Based Information Systems (NBiS), 2010 13th International Conference on, 2010, ISSN 2157-0418, s. 358–364.
- [18] Buettner, M.; Yee, G. V.; Anderson, E.; Han, R.: X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks. In *Proceedings of the 4th International Conference* on Embedded Networked Sensor Systems, SenSys '06, ACM, 2006, ISBN 1-59593-343-3, s. 307–320.
- [19] Camillò, A.; Nati, M.; Petrioli, C.; Rossi, M.; Zorzi, M.: IRIS: Integrated data gathering and interest dissemination system for wireless sensor networks. *Elsevier Ad Hoc Networks*, ročník 11, č. 2, 2013: s. 654 – 671, ISSN 1570-8705, special Issue on Cross-layer design in ad hoc and sensor networks.
- [20] Canli, T.: Power Efficient Computations and Communication Primitives in Wireless Sensor Networks. University of Illinois at Chicago, 2010, 202 s.
- [21] Canli, T.; Gupta, A.; Khokhar, A.: Power Efficient Algorithms for Computing Fast Fourier Transform over Wireless Sensor Networks. In *Computer Systems and Applications, 2006. IEEE International Conference on.*, 2006, s. 549–556.
- [22] Cao, Q.; Abdelzaher, T.; Stankovic, J.; He, T.: The LiteOS Operating System: Towards Unix-Like Abstractions for Wireless Sensor Networks. In Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on, 2008, s. 233–244.
- [23] ChaN. Fixed-point FFT routines for megaAVRs. [Online]. 2005. URL http://elm-chan.org/docs/avrlib/fftavr.zip
- [24] Chen, F.; Wen, F.; Jia, H.: Algorithm of Data Compression Based on Multiple Principal Component Analysis over the WSN. In Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on, 2010, s. 1–4.

- [25] Chiasserini, C.: On the concept of distributed digital signal processing in wireless sensor networks. In *MILCOM. Proceedings*, ročník 1, 2002, s. 260–264 vol.1.
- [26] Clarke, P.: Android, FreeRTOS top EE Times 2013 embedded survey. *EE Times*, 2013. URL http://www.eetimes.com/document.asp?doc_id=1119278
- [27] Coates, M.: Distributed Particle Filters for Sensor Networks. In Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, IPSN '04, 2004, ISBN 1-58113-846-6, s. 99–107.
- [28] van Dam, T.; Langendoen, K.: An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03, ACM, 2003, ISBN 1-58113-707-9, s. 171–180.
- [29] Dannies, A.; Palafox-Albarrán, J.; Lang, W.; Jedermann, R.: Smart dynamic software components enabling decision support in Machine-to-machine networks. *International Journal* of Computer Science Issues (IJCSI), ročník 10, č. 1, 2013: str. 540, ISSN 1694-0814.
- [30] Delicato, F.; Pires, P.; Zomaya, A.: Middleware Platforms: State of the Art, New Issues, and Future Trends. In *The Art of Wireless Sensor Networks*, editace H. M. Ammari, Signals and Communication Technology, Springer Berlin Heidelberg, 2014, ISBN 978-3-642-40008-7, s. 645–674.
- [31] Du, S.; Saha, A.; Johnson, D.: RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, ISSN 0743-166X, s. 1478–1486.
- [32] Dutta, P.; Dawson-Haggerty, S.; Chen, Y.; Liang, C.-J. M.; Terzis, A.: Design and Evaluation of a Versatile and Efficient Receiver-initiated Link Layer for Low-power Wireless. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, ACM, 2010, ISBN 978-1-4503-0344-6, s. 1–14.
- [33] Dziengel, N.; Wittenburg, G.; Adler, S.; Kasmi, Z.; Ziegert, M.; aj.: Event Detection in Wireless Sensor Networks. Taylor & Francis, 2012, ISBN 9781439892817.
- [34] El-Hoiydi, A.; Decotignie, J.-D.: WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks. In Algorithmic Aspects of Wireless Sensor Networks, Lecture Notes in Computer Science, ročník 3121, Springer Berlin Heidelberg, 2004, ISBN 978-3-540-22476-1, s. 18–31.
- [35] Ellaithy, R. A.: A Comprehensive Survey on Sensors and Sensors Networks. Dizertační práce, 2012.
- [36] Červenka, V.; Komosný, D.; Kathiravelu, G.: IETF 6LoWPAN and Sensor Networking. In *ICT2011*, Vysoké učení technické v Brně, 2011, ISBN 978-80-214-4231-3, s. 69–73.
- [37] Červenka, V.; Komosný, D.; Malina, L.; Mráz, L.: Energy Efficient Public Key Cryptography in Wireless Sensor Networks. In *Innovations and Advances in Computer, Information, Systems Sciences, and Engineering*, USA, Berkeley: Springer New York, 2012, ISBN 978-1-4614-3534-1, s. 497–509.

- [38] Červenka, V.; Mráz, L.; Šimek, M.: Reusing AES Coprocessor in Public Key Cryptography. In Poster and Demo Proceedings of 9th EuropeanConference on Wireless Sensor Network, 2012.
- [39] Červenka, V.; Mráz, L.; Šimek, M.: Integration and Visualization of Adaptive Wireless Sensor Network. In Poster and Demo Proceedings of 10th European Conference on Wireless Sensor Network (EWSN 2013), 2013.
- [40] Červenka, V.; Mráz, L.; Komosný, D.: Top-Down Architecture for Environmental Monitoring Using Wireless Sensor Network. In Proceedings of 35th International Conference on Telecommunications and Signal Processing (TSP 2012), 2012, ISBN 978-1-4673-1118-2, s. 91-95.
- [41] Červenka, V.; Mráz, L.; Komosný, D.: Comprehensive Performance Analysis of Lightweight Mesh and its comparison with ZigBee Pro Technology. Wireless Personal Communication, 2014, doi: 10.1007/s11277-014-1832-x, v tisku.
- [42] Fafoutis, X.; Vuckovic, D.; Di Mauro, A.; Dragoni, N.; Madsen, J.: Energy-Harvesting Wireless Sensor Networks. 2012, s. 84–85.
- [43] Farooq, M. O.; Kunz, T.: Operating Systems for Wireless Sensor Networks: A Survey. Sensors, ročník 11, č. 6, 2011: s. 5900–5930.
- [44] Fernandez-Lopez, H.; Correia, J.; Simoes, R.; Afonso, J.: Experimental Evaluation of IEEE 802.15.4/ZigBee for Multi-patient ECG Monitoring. In *Electronic Healthcare, Lecture Notes* of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, ročník 69, Springer Berlin Heidelberg, 2012, ISBN 978-3-642-23634-1, s. 184–191.
- [45] Gajjar, S.; Choksi, N.; Sarkar, M.; Dasgupta, K.: Comparative analysis of wireless sensor network motes. In Signal Processing and Integrated Networks (SPIN), 2014 International Conference on, 2014, s. 426–431.
- [46] Gao, W.; Zhao, H.; Song, C.; Xu, J.: A New Distributed Particle Filtering for WSN Target Tracking. In 2009 International Conference on Signal Processing Systems, 2009, s. 334–337.
- [47] Gorlatova, M.; Sharma, T.; Shrestha, D.; Xu, E.; Chen, J.; aj.: Prototyping Energy Harvesting Active Networked Tags (EnHANTs) with MICA2 Motes. In Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on, 2010, s. 1–3.
- [48] Grübl, A.: VLSI Implementation of a Spiking Neural Network. Dizertační práce, Ruprecht-Karls-Universität Heidelberg, 2007.
- [49] Hammoodi, I. S.; Stewart, B.; Kocian, A.; McMeekin, S.: A Comprehensive Performance Study of OPNET Modeler for ZigBee Wireless Sensor Networks. In Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International Conference on, 2009, s. 357–362.
- [50] Healy, M.; Newe, T.; Lewis, E.: Efficiently securing data on a wireless sensor network. Journal of Physics: Conference Series, ročník 76, č. 1, 2007.

- [51] Heidemann, J.; Stojanovic, M.; Zorzi, M.: Underwater sensor networks: applications, advances and challenges. *Physical and Engineering Sciences*, ročník 370, 2012: s. 158–175.
- [52] Hill, J.; Culler, D.: A wireless embedded sensor architecture for system-level optimization. Technická zpráva, 2001.
- [53] Huang, L.; Ashouei, M.; Yazicioglu, R. F.; Penders, J.; Vullers, R. J. M.; aj.: Ultra-Low Power Sensor Design for Wireless Body Area Networks - Challenges, Potential Solutions, and Applications. *JDCTA*, ročník 3, č. 3, 2009: s. 136–148.
- [54] Huang, P.; Wang, C.; Xiao, L.; Chen, H.: RC-MAC: A receiver-centric medium access control protocol for wireless sensor networks. In *Quality of Service (IWQoS), 2010 18th International* Workshop on, 2010, ISSN 1548-615X, s. 1–9.
- [55] Huang, P.; Xiao, L.; Soltani, S.; Mutka, M.; Xi, N.: The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey. *Communications Surveys Tutorials*, *IEEE*, ročník 15, č. 1, 2013: s. 101–120, ISSN 1553-877X.
- [56] Hyncica, O.; Kucera, P.; Honzik, P.; Fiedler, P.: Performance evaluation of symmetric cryptography in embedded systems. In *Intelligent Data Acquisition and Advanced Computing* Systems (IDAACS), 2011 IEEE 6th International Conference on, ročník 1, 2011, s. 277–282.
- [57] IEEE 802.15 WPAN Low Rate Alternative PHY. Task Group 4a (TG4a). [Online]. 2012. URL http://www.ieee802.org/15/pub/TG4a.html
- [58] IEEE 802.15 WPAN. Task Group 4b (TG4b). [Online]. 2012. URL http://www.ieee802.org/15/pub/TG4b.html
- [59] IEEE 802.15 WPAN. Task Group 4c (TG4c). [Online]. 2012. URL http://www.ieee802.org/15/pub/TG4c.html
- [60] IEEE 802.15 WPAN. Task Group 4d (TG4d). [Online]. 2012. URL http://www.ieee802.org/15/pub/TG4d.html
- [61] IEEE 802.15 WPAN. Task Group 4e (TG4e). [Online]. 2012.URL http://www.ieee802.org/15/pub/TG4e.html
- [62] IEEE 802.15 WPAN. Task Group 10 (TG10) Layer 2 Routing. [Online]. 2014. URL http://www.ieee802.org/15/pub/TG10.htm
- [63] Šimek, M.; Mráz, L.; Červenka, V.; Pechanec, V.: Adaptive Wireless Sensor Network for Urban Crisis Management. In Poster and Demo Proceedings of 9th EuropeanConference on Wireless Sensor Network, 2012.
- [64] Šimek, M.; Mráz, L.; Červenka, V.; Botta, M.; Julina, V.: Wireless Light-weight Snow Sensor. In Poster and Demo Proceedings of 10th European Conference on Wireless Sensor Network (EWSN 2013), 2013.
- [65] Intel. Embedded Microcontrollers. [Online]. 2012. URL www.intel.com/design/embcontrol/index.htm

- [66] Intel-National Taiwan University NSC. WSN M3-SuperNode. [Online]. 2014. URL http://ccc.ntu.edu.tw
- [67] Internet Engineering Task Force. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks RFC 6550. [Online]. 2012.
 URL https://datatracker.ietf.org/doc/rfc6550/
- [68] Internet Engineering Task Force. The Constrained Application Protocol. RFC7252. [Online].
 2012.
 URL https://datatracker.ietf.org/doc/rfc6550/
- [69] IPv6 over Low power WPAN. [Online]. 2012. URL http://tools.ietf.org/wg/6lowpan
- [70] Ismail, N.-S.; Hussein, L.; Ariffin, S. H. S.: Analyzing the Performance of Acoustic Channel in Underwater Wireless Sensor Network (UWSN). In *Mathematical/Analytical Modelling* and Computer Simulation (AMS), 2010 Fourth Asia International Conference on, 2010, s. 550–555.
- [71] Jennic Ltd.: Calculating 802.15.4 Data Rates, Application Note JN-AN-1035. Technická zpráva, 2006.
 URL http://www.jennic.com/support/application_notes/jn-an-1035_calculating_data_rates_in_an_ieee_802154-based_network
- [72] Jiang, H.; Hallstrom, J.: Fast, Accurate Event Classification on Resource-Lean Embedded Sensors. In Wireless Sensor Networks, Lecture Notes in Computer Science, ročník 6567, Springer Berlin Heidelberg, 2011, ISBN 978-3-642-19185-5, s. 65–80.
- [73] Jiang, H.; Jin, S.; Wang, C.: Prediction or Not? An Energy-Efficient Framework for Clustering-Based Data Collection in Wireless Sensor Networks. *Parallel and Distributed Sys*tems, IEEE Transactions on, ročník 22, č. 6, 2011: s. 1064–1071, ISSN 1045-9219.
- [74] Jiang, J.; Claudel, C.: A wireless computational platform for distributed computing based traffic monitoring involving mixed Eulerian-Lagrangian sensing. In *Industrial Embedded Sys*tems (SIES), 2013 8th IEEE International Symposium on, 2013, s. 232–239.
- [75] Jiang, S.; Wang, W.; Sun, D.; Li, Z.: Design of energy self-sufficient wireless sensor network node for orchard information acquisition. *Transactions of the Chinese Society of Agricultural Engineering*, ročník 28, č. 9, 2012: s. 153–158.
- [76] Jin-Shyan Lee, Y.-M. W.; Jin-Shyan Lee, Y.-M. W.: Experimental Evaluation of ZigBee-Based Wireless Networks in Indoor Environments. *Journal of Engineering*, 2013.
- [77] Johnson, M.; Healy, M.; van de Ven, P.; Hayes, M.; Nelson, J.; aj.: A comparative review of wireless sensor network mote technologies. In *Sensors, 2009 IEEE*, 2009, ISSN 1930-0395, s. 1439–1442.
- [78] Ju, H.; Cui, L.: EasiPC: a packet compression mechanism for embedded WSN. In Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on, 2005, ISSN 1533-2306, s. 394–399.

- [79] Jurdak, R.; Klues, K.; Kusy, B.; Richter, C.; Langendoen, K.; aj.: Opal: A Multiradio Platform for High Throughput Wireless Sensor Networks. *Embedded Systems Letters, IEEE*, ročník 3, č. 4, 2011: s. 121–124, ISSN 1943-0663.
- [80] Karapistoli, E.; Stratogiannis, D.; Tsiropoulos, G.; Pavlidou, F.: MAC protocols for ultrawideband ad hoc and sensor networking: A survey. In Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on, 2012, ISSN 2157-0221, s. 834–841.
- [81] Kargl, A.; Pyka, S.; Seuschek, H.: Fast Arithmetic on ATmega128 for Elliptic Curve Cryptography. 2008.
- [82] Khanafer, M.; Guennoun, M.; Mouftah, H.: A Survey of Beacon-Enabled IEEE 802.15.4 MAC Protocols in Wireless Sensor Networks. *Communications Surveys Tutorials, IEEE*, ročník 16, č. 2, 2014: s. 856–876, ISSN 1553-877X.
- [83] Khiani, S.; Dethe, C.; Thakare, V.: Designing a Secure and Reliable Node Disjoint Multipath Routing Algorithm and a Survey on Existing Techniques. In *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on*, 2014, s. 496–500.
- [84] Ko, J.; Klues, K.; Richter, C.; Hofer, W.; Kusy, B.; aj.: Low Power or High Performance? A Tradeoff Whose Time Has Come (and Nearly Gone). In *Proceedings of the 9th European Conference on Wireless Sensor Networks*, EWSN'12, Springer-Verlag, 2012, ISBN 978-3-642-28168-6, s. 98–114.
- [85] Ko, J.; Wang, Q.; Schmid, T.; Hofer, W.; Dutta, P.; aj.: Egs: A Cortex M3-Based Mote Platform. In Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on, 2010, s. 1–3.
- [86] Korkua, S.; Jain, H.; Lee, W.-J.; Kwan, C.: Wireless health monitoring system for vibration detection of induction motors. In *Industrial and Commercial Power Systems Technical Conference (I CPS)*, 2010 IEEE, 2010, s. 1–6.
- [87] Kuorilehto, M.; Kohvakka, M.; Suhonen, J.; Hamalainen, P.; Hannikainen, M.; aj.: Ultra-Low Energy Wireless Sensor Networks in Practice: Theory, Realization and Deployment. Wiley Publishing, 2008, ISBN 0470057866, 9780470057865.
- [88] Lang, Jean-Philippe. Nano-RK. [Online]. 2012. URL http://www.nanork.org/projects/nanork/wiki
- [89] Langendoen, K.; Meier, A.: Analyzing MAC Protocols for Low Data-rate Applications. ACM Trans. Sen. Netw., ročník 7, č. 1, Srpen 2010: s. 10:1–10:34, ISSN 1550-4859.
- [90] Lee, J. S.: Performance Evaluation of IEEE 802.15.4 for Low-rate Wireless Personal Area Networks. *IEEE Trans. on Consum. Electron.*, ročník 52, č. 3, 2006: s. 742–749, ISSN 0098-3063.
- [91] Levis, P.; Madden, S.; Polastre, J.; Szewczyk, R.; Whitehouse, K.; aj.: TinyOS: An Operating System for Sensor Networks. In *Ambient Intelligence*, editace W. Weber; J. Rabaey; E. Aarts, Springer Berlin Heidelberg, 2005, ISBN 978-3-540-23867-6, s. 115–148.

- [92] Li, J.; Zhu, X.; Tang, N.; Sui, J.: Study on ZigBee network architecture and routing algorithm. In Signal Processing Systems (ICSPS), 2010 2nd International Conference on, ročník 2, 2010, s. V2–389–V2–393.
- [93] Lingyun, Y.; Xingchao, W.: Study on performance evaluation method based on measurement for wireless sensor network. In *Communications Technology and Applications, 2009. ICCTA* '09. IEEE International Conference on, 2009, s. 201–206.
- [94] Liu, S.; Fan, K.-W.; Sinha, P.: CMAC: An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks. In Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on, 2007, s. 11–20.
- [95] Lonare, S.; Wahane, G.: A survey on energy efficient routing protocols in wireless sensor network. In Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on, 2013, s. 1–5.
- [96] Low, K.-S.; Talampas, M. C. R.: Wireless Sensor Networks for Intelligent Transportation Applications. CRC Press, 2013, ISBN 978-1-4665-0051-8, str. 2009.
- [97] Lu, B.; Gungor, V.: Online and Remote Motor Energy Monitoring and Fault Diagnostics Using Wireless Sensor Networks. *Industrial Electronics, IEEE Transactions on*, ročník 56, č. 11, 2009: s. 4651–4659, ISSN 0278-0046.
- [98] Lu, G.; Krishnamachari, B.; Raghavendra, C.: An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In *Parallel and Distributed Processing* Symposium, 2004. Proceedings. 18th International, 2004.
- [99] Magno, M.; Benini, L.; Spagnol, C.; Popovici, E.: Wearable low power dry surface wireless sensor node for healthcare monitoring application. In Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on, 2013, ISSN 2160-4886, s. 189–195.
- [100] Magno, M.; Porcarelli, D.; Brunelli, D.; Benini, L.: Endless Smart Power for WSNs: Combining Multiple Harvesting and Fuel Cells. In Adjunct Proceedings of the 9th European Conference on Wireless Sensor Networks (EWSN'12), 2012.
- [101] Memsic. LPR2400 LOTUS Processor Radio Processor Board. [Online]. 2014. URL http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0705-01_A_ LOTUS.pdf
- [102] Memsic. Stagrate SPB400. [Online]. 2014. URL http://www.memsic.com/wireless-sensor-networks/
- [103] Memsic. XM2110CA 2.4GHz IRIS OEM Edition Module. [Online]. 2014. URL http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf
- [104] Microchip Technology Inc. PIC18. [Online]. 2012.URL www.microchip.com/PIC18

- [105] MiWi Development Environmen. [Online]. 2012. URL http://www.microchip.com/miwi
- [106] Müller, J.; Červenka, V.; Komosný, D.: Taking Advantage of DLNA Technology in Education. In New Information and Multimedia Technologies (NIMT), VUT Brno, 2011, ISBN 978-80-214-4241-2, s. 1–4.
- [107] Mráz, L.; Červenka, V.: Complete System Design Based on Wireless Sensor Network for Disaster Prevention. In Proceedings of 35th International Conference on Telecommunications and Signal Processing (TSP 2012), ročník 10a, Przeglad Elektrotechniczny, 2012, ISSN 0033-2097, s. 273–277.
- [108] Mráz, L.; Červenka, V.; Šimek, M.: Packet Analyser for IEEE 802.15.4 Networks. In Poster and Demo Proceedings of 9th European Conference on Wireless Sensor Network (EWSN 2012), 2012.
- [109] Mráz, L.; Červenka, V.; Šimek, M.; Komosný, D.: Comprehensive Performance Analysis of ZigBee Technology Based on Real Measurements. Wireless Personal Communication, 2013: s. 2783–2803, ISSN 0929-6212.
- [110] Mráz, L.; Komosný, D.; Červenka, V.; Morávek, P.; Šimek, M.: Open-Packet Analyser Platform for Wireless Sensor Networks Based on IEEE 802.15.4. In 34th International Conference on Telecommunications and Signal Processing (TSP 2011), 2011, ISBN 978-1-4577-1409-2, s. 145 – 149.
- [111] Nachman, L.; Huang, J.; Shahabdeen, J.; Adler, R.; Kling, R.: IMOTE2: Serious Computation at the Edge. In Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International, 2008, s. 1118–1123.
- [112] National Institute of Standards and Technology. NIST Special Publication 800-57. [Online]. 2012. URL http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_ general.pdf
- [113] Nefzi, B.; Song, Y.-Q.: Performance Analysis and improvement of ZigBee routing protocol. In 7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems - FeT'2007, IFAC, 2007.
- [114] NXP Semiconductors. LPC4300. [Online]. 2014. URL http://www.nxp.com/products/microcontrollers/cortex_m4/lpc4300
- [115] Oppermann, F.; Boano, C.; Römer, K.: A Decade of Wireless Sensing Applications: Survey and Taxonomy. In *The Art of Wireless Sensor Networks*, Springer Berlin Heidelberg, 2014, ISBN 978-3-642-40008-7, s. 11–50.
- [116] Oracle Labs. Sun Small Programmable Object Technology (Sun SPOT) Owner's Manual Release 2.0. [Online]. 2007. URL http://www.sunspotworld.com/docs/Orange/SunSPOT-OwnersManual.pdf
- [117] Oracle Labs. Sun SPOT World. [Online]. 2014. URL http://www.sunspotworld.com

- [118] Padmavati; Aseri, T.: Comparison of routing protocols in wireless sensor network using mobile sink- A survey. In Engineering and Computational Sciences (RAECS), 2014 Recent Advances in, 2014, s. 1–4.
- [119] Pandatron. Spojení Cortex-M3 obvodu a RTX real-time OS. [Online]. 2009. URL http://pandatron.cz/?780&spojeni_cortex-m3_obvodu_a_rtx_real-time_os
- [120] Panitantum, N.: Ultra-Low-Energy Transmitters for Battery-Free Wireless Sensor Networks. Dizertační práce, 2011.
- [121] Pasha, M. A.; Derrien, S.; Sentieys, O.: A novel approach for ultra low-power WSN node generation. In Signals and Systems Conference (ISSC 2010), IET Irish, June 2010, s. 204– 209.
- [122] Patil, H. K.; Szygenda, S. A.: kapitola Implementation of Elliptic Curve Cryptography in Wireless Sensor Networks. 2013, s. 173–200.
- [123] Perkins, C.; Royer, E.: Ad-hoc on-demand distance vector routing. In Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on, 1999, s. 90–100.
- [124] Pesovic, U.; Jovanovic, Z.; Randjic, S.; Markovic, D.: Benchmarking performance and energy efficiency of microprocessors for wireless sensor network applications. In *MIPRO*, 2012 Proceedings of the 35th International Convention, 2012, s. 743–747.
- [125] Petrova, M.; Riihijarvi, J.; Mahonen, P.; Labella, S.: Performance study of IEEE 802.15.4 using measurements and simulations. In Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE, ročník 1, 2006, ISSN 1525-3511, s. 487–492.
- [126] Philipp, S.; Schemmel, J.; Meier, K.: A QoS network architecture to interconnect largescale VLSI neural networks. In *Neural Networks*, 2009. IJCNN 2009. International Joint Conference on, 2009, ISSN 1098-7576, s. 2525–2532.
- [127] Piguet, C.: Ultra-Low-Power Processor Design. In High-Performance Energy-Efficient Microprocessor Design, Series on Integrated Circuits and Systems, Springer US, 2006, ISBN 978-0-387-28594-8, s. 1–30.
- [128] Pister, K. S. J.; Doherty, L.: TSMP: Time Synchronized Mesh Protocol. In In Proceedings of the IASTED International Symposium on Distributed Sensor Networks (DSN08, 2008.
- [129] Polastre, J.; Hill, J.; Culler, D.: Versatile Low Power Media Access for Wireless Sensor Networks. In Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems, SenSys '04, 2004, ISBN 1-58113-879-2, s. 95–107.
- [130] Real Time Engineers Ltd. FreeRTOS. [Online]. 2014. URL http://www.freertos.org/
- [131] Reinhardt, A.; Hollick, M.; Steinmetz, R.: Stream-oriented Lossless Packet Compression in Wireless Sensor Networks. In Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on, 2009, s. 1–9.

- [132] Renner, C.; Meier, F.; Turau, V.: Energy Assessment in Praxis. In Adjunct Proceedings of the 9th European Conference on Wireless Sensor Networks (EWSN'12), 2012.
- [133] Riliskis, L.: Methodologies and Practical Tools for Realistic Large Scale Simulations of Wireless Sensor Networks. Dizertační práce, 2014.
- [134] Rukpakavong, W.; Phillips, I.; Guan, L.; Oikonomou, G.: RPL router discovery for supporting energy-efficient transmission in single-hop 6LoWPAN. In *Communications (ICC)*, 2012 IEEE International Conference on, 2012, ISSN 1550-3607, s. 5721–5725.
- [135] Saadon, S.; Sidek, O.: Ambient vibration-based MEMS piezoelectric energy harvester for green energy source. In Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on, 2011, s. 1–6.
- [136] Savvides, A.; Srivastava, M.: A distributed computation platform for wireless embedded sensing. In Computer Design: VLSI in Computers and Processors, 2002. Proceedings. 2002 IEEE International Conference on, 2002, ISSN 1063-6404, s. 220–225.
- [137] Serpen, G.; Li, J.; Liu, L.; Gao, Z.: WSN-ANN: Parallel and distributed neurocomputing with wireless sensor networks. In *Neural Networks (IJCNN)*, The 2013 International Joint Conference on, 2013, ISSN 2161-4393, s. 1–8.
- [138] Sharkawy, B.; Khattab, A.; Elsayed, K.: Fault-tolerant RPL through context awareness. In Internet of Things (WF-IoT), 2014 IEEE World Forum on, 2014, s. 437–441.
- [139] Silicon Labs. EFM32WG Reference Manual. [Online]. 2014. URL http://www.silabs.com/products/mcu/lowpower/Pages/efm32wg-wonder-gecko. aspx
- [140] Silicon Labs. Simplicity Studio: FreeRTOS Tickless. [Online]. 2014. URL http://www.silabs.com/products/mcu/pages/simplicity-studio.aspx
- [141] Smékal, Z.: Systémy a signály: 1D a 2D diskrétní a číslicové zpracování. Sdělovací technika, Praha, 2013, ISBN 80-86645-22-0.
- [142] STMicroelectronics. STM32F427xx. [Online]. 2014. URL http://www.st.com/st-web-ui/static/active/en/resource/technical/ document/datasheet/DM00071990.pdf
- [143] Strazdins, G.; Elsts, A.; Nesenbergs, K.; Selavo, L.: Wireless Sensor Network Operating System Design Rules Based on Real-World Deployment Survey. *Journal of Sensor and Actuator Networks*, ročník 2, č. 3, 2013: s. 509–556, ISSN 2224-2708.
- [144] Suarez, P.; Renmarker, C.-G.; Dunkels, A.; Voigt, T.: Increasing ZigBee Network Lifetime with X-MAC. In *Proceedins of REALWSN 2008*, 2008.
- [145] Sun, Y.; Du, S.; Gurewitz, O.; Johnson, D. B.: DW-MAC: A Low Latency, Energy Efficient Demand-wakeup MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 9th* ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '08, ACM, 2008, ISBN 978-1-60558-073-9, s. 53-62.

- [146] Sun, Y.; Gurewitz, O.; Johnson, D. B.: RI-MAC: A Receiver-initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. In *Proceedings* of the 6th ACM Conference on Embedded Network Sensor Systems, SenSys '08, ACM, 2008, ISBN 978-1-59593-990-6, s. 1–14.
- [147] Suriyachai, P.; Roedig, U.; Scott, A.: A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks. *Communications Surveys Tutorials, IEEE*, ročník 14, č. 2, 2012: s. 240–264, ISSN 1553-877X.
- [148] Tada, Y.; Manju; Singh, V.: A survey on Energy Efficient Routing Protocols in Wireless Sensor Network. In International journal for research in applied science and engineering technology (IJRASET), 2014, ISSN 2321-9653.
- [149] Tang, L.; Sun, Y.; Gurewitz, O.; Johnson, D.: PW-MAC: An energy-efficient predictivewakeup MAC protocol for wireless sensor networks. In *INFOCOM*, 2011 Proceedings IEEE, 2011, ISSN 0743-166X, s. 1305–1313.
- [150] Texas Instrument. MSP430 Ultra-Low Power 16-Bit Microcontrollers. [Online]. 2012. URL www.ti.com/MSP430
- [151] Texas Instruments: Using a DC-DC Converter to Reduce Power (Current) Consumption In CC430 Systems. Technická zpráva, 2006.
 URL http://www.ti.com.cn/cn/lit/an/slaa500/slaa500.pdf
- [152] The Contiki OS. [Online]. 2014. URL http://www.contiki-os.org
- [153] Trevor, M.: The Designer's Guide to the Cortex-M Processor Family: A Tutorial Approach. Elsevier Science, 2013, ISBN 9780080982991.
- [154] University of California, Berkeley. Tech. project: Smart dust. [Online]. 2012. URL http://robotics.eecs.berkeley.edu/~pister/SmartDust/
- [155] University of California, Berkeley. A WSN look at the ARM ecosystem. [Online]. 2010. URL http://wsn.eecs.berkeley.edu/?p=128
- [156] Vasseur, J.-P.; Dunkels, A.: Interconnecting Smart Objects with IP: The Next Internet. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010.
- [157] Virtenio. Preon32. [Online]. 2014. URL http://www.virtenio.com/en/products/radio-module.html
- [158] Vodel, M.; Hardt, W.: Energy-efficient communication in distributed, embedded systems. In Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt), 2013 11th International Symposium on, 2013, s. 641–647.
- [159] Warneke, B.; Last, M.; Liebowitz, B.; Pister, K.: Smart Dust: communicating with a cubicmillimeter computer. Computer, ročník 34, č. 1, 2001: s. 44–51, ISSN 0018-9162.
- [160] Weiye, T.; Jieqiong, Z.: The performance study of ZigBee mesh network. In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, 2011, s. 407– 409.

- [161] Yan-li, Z.; Xiao-ping, F.; Shao-qiang, L.; Zhe-yuan, X.: Improved LZW algorithm of lossless data compression for WSN. In *Computer Science and Information Technology (ICCSIT)*, 2010 3rd IEEE International Conference on, ročník 4, 2010, s. 523–527.
- [162] Yao, Y.; Wan, L.; Cao, Q.: System Architecture and Operating Systems. In *The Art of Wireless Sensor Networks*, Springer Berlin Heidelberg, 2014, ISBN 978-3-642-40008-7, s. 697–738.
- [163] Ye, W.; Heidemann, J.; Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, ročník 3, 2002, ISSN 0743-166X, s. 1567– 1576 vol.3.
- [164] Yerva, L.; Bansal, A.; Campbell, B.; Dutta, P.; Schmid, T.: An IEEE 802.15.4-compatible, Battery-free, Energy-harvesting Sensor Node. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, ACM, 2011, s. 389–390.
- [165] Yu, Y.; Krishnamachari, B.; Kumar, V.: Information Processing and Routing in Wireless Sensor Networks. World Scientific, 2006, ISBN 9789812701466.
- [166] Zheng, J.; Wang, P.; Li, C.: Distributed Data Aggregation Using Slepian-Wolf Coding in Cluster-Based Wireless Sensor Networks. *Vehicular Technology, IEEE Transactions on*, ročník 59, č. 5, 2010: s. 2564–2574, ISSN 0018-9545.
- [167] ZigBee Aliance. [Online]. 2012. URL http://www.zigbee.org

SEZNAM ZKRATEK

AAP	Authentication Access Port
ACK	Acknowledgement
ADC	Analog to Digital Converter
ADCNRM	ADC Noise Reduction Mode
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AHB-AP	Advanced High-performance Bus Access Port
AMBA	Advanced Microcontroller Bus Architecture
AODV	Ad hoc On-Demand Distance Vector
APB	Advanced Peripheral Bus
API	Application Programming Interface
AppACK	Application Acknowledgement
ASIC	Application Specific Integrated Circuits
BPSK	Binary-Phase Shift Keying
CFFT	Complex Fast Fourier Transform
CLBD	Conventional Load Balanced Distributed
CMSIS	Cortex Microcontroller Software Interface Standard
COAP	Constrained Application Protocol
CPU	Central Processing Unit
CRC	Cyclic redundancy check
CSMA	Carrier Sense Multiple Access
DAC	Digital to Analog Converter
DDSP	Distributed Digital Signal Processing
DFT	Discrete Fourier Transform
DLNA	Digital Living Network Alliance
DLW	Debug Lock Word
DMA	Direct Memory Access
DMIPS	Dhrystone Million Instructions Per Second

DPF	Distributed Particle Filtering
DSP	Digital Signal Processor
EE-LWM	Energy Efficient Lightweight Mesh
EE-RTOS	Energy Efficient Real-Time Operating Systém
EE-WSN	Energy Efficient Wireless Sensor Node
EH	Energy Harvesting
FCS	Frame Check Sequence
FFT	Fast Fourier Transform
FIFO	First In First Out
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
GPIO	General-purpose input output
HAL	Hardware Abstraction Layer
HFRCO	High Frequency RC Oscillator
HFXO	High Frequency Crystal Oscillator
HW	Hardware
I^2C	Inter-Integrated Circuit Interface
IEEE	Institute of Electrical and Electronics Engineers
IIR	Infinite Impulse Response
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRQ	Interrupt Request
IRS	Interrupt Service Routine
ISM	Industrial Scientific Medicine
LFRCO	Low Frequency RC Oscillator
LFXO	Low Frequency Crystal Oscillator
LPL	Low Power Listening
LPP	Low Power Polling/ Probing
LWM	Lightweight Mesh
MAC	Medium Access Control
MAC_{math}	Multiplier-Accumulator
MCU	Micro Controller Unit
MFR	MAC Footer
MHR	MAC Header
MPU	Memory Protection Unit
MSDU	MAC Service Data Unit
NIST	National Institute of Standards and Technology
NVIC	Nested Vectored Interrupt Controller

OS	Operating System
OTAU	Over The Air Upgrade
PHR	Physical Layer Header
PPDU	Physical Protocol Data Unit
PRS	Peripheral Reflex System
RAM	Random-Access Memory
RF	Radio Frequency
ROM	Read-Only Memory
RSSI	Received signal strength indication
RTC	Real-Time Clock
RTOS	Real-Time Operating System
SHR	Synchronization header
SIMD	Single Instruction Multiple Data
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SW	Software
SW-DP	Serial Wire Debug Port
TSMP	Time Synchronized Mesh Protocol
UART	Universal Asynchronous Receiver Transmitter
UPAD	Unbalanced Power-Aware Distributed
USART	Universal Synchronous/ Asynchronous Receiver and Transmitter
VBA	Visual Basic for Applications
WBAN	Wireless Body Area Network
WBSN	Wireless Body Sensor Network
WDT	Watchdog Timer
WIC	Wakeup Interrupt Controller
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

SEZNAM PŘÍLOH

A	Přehled dostupných hardwarových platforem	108
В	Elektrotechnické schéma EE-WSN B.1 Seznam součástek	110 . 114
\mathbf{C}	Ukázka analytického frameworku	115

A PŘEHLED DOSTUPNÝCH HARDWAROVÝCH PLATFOREM
Platforma	Architektura [b]	Mikrokontrolér	Rádiový modul	Rozměry [cm]	Max. takt MCU [MHz]
Btnode v.3	×	ATmega128L	ZV4002+CC1000	5,8 imes3,3	7,383
INGA	×	ATmega 1284P	AT86RF231	4,0 imes 5,0 imes 0,7	16
IRIS	×	XM2110CB (ATmega1281V)	AT86RF230	5,8 imes3,2 imes0,7	7,383
Mica2	×	${ m ATmega128L}$	CC1000	5,8 imes3,2 imes0,7	7,383
MicaZ	×	${ m ATmega128L}$	CC2420	5,8 imes3,2 imes0,7	7,383
TinyNode 584	×	MSP430	XE1205	3,0 imes 4,0	8
Waspmote	×	${ m ATmega1281}$	XBee -802.15.4	$7,35\times5,1\times1,3$	8
G-Node G301	16	$\mathrm{MSP430F2418}$	CC1101	6,0 imes 1,9 imes 0,6	16
Telos	16	MSP430F149	CC2420	6,6 imes 3,2 imes 0,7	8
TelosB	16	MSP430F1611	CC2420	6.5 imes3.1 imes0.6	8
TmoteSky	16	MSP430F1611	CC2420	6,6 imes 3,2 imes 0,7	8
Z1	16	MSP430F2617	CC2420	5,68 imes3,45	16
Imote2	32	Intel PXA271 Xscale	CC2420	4,8 imes3,6 imes0,9	13-416
Lotus	32	LPC1758	AT86RF231	7,6 imes 3,4 imes 0,7	10-100
SunSpot	32	ARMSAM9G20	CC2420	4,1 imes 7,0	400
Stargate	32	Intel PXA255 Xscale	(Mica2)	9,53 x 6,33 x 0,7	400
Preon32	32	STM32 N/A	AT86RF231	$2,75 \ge 1,9 \ge 0,3$	72

Tab. A.1: Přehled dostupných hardwarových platforem.

B ELEKTROTECHNICKÉ SCHÉMA EE-WSN







File: A.sch		
Sheet: /A/		
Title: Platfor	ma EE-WSN	
Size: A4	Date: 19 aug 2014	Rev:
KiCad E.D.A.	eeschema (2013-07-07 BZR 4022)-stable	ld; 2/3
•	4	5





File: B.sch			
Sheet: /B/			
Title: Platfor	ma EE-WSN		
Size: A4	Date: 19 aug 2014		Rev:
KiCad E.D.A.	eeschema (2013-07-07 BZR	4022)-stable	ld: 3/3
	4	5	

B.1 Seznam součástek

B1	24501	BM15A0015
C1	$100 \ \mathrm{nF}$	X5R, 10%, 16V
C2	$10 \ \mu F$	X5R, 10%, 16V
C3, C7, C11	$10 \ \mu F$	X5R, 15%
C4, C8, C9, C10	$100 \ \mathrm{nF}$	X5R, 15%
C5, C6	$10 \ \mathrm{nF}$	X5R, 15%
CB1, CB3	$100~\mathrm{nF}$	X7R, 10%, 16V
CB2, CB4	$1 \ \mu F$	X5R, 10%, 16V
CF1, CF2	22 pF	C0G, 5%, 50V
CP1	$10 \ \mu F$	X5R, 20%
CP2, CP3	$2,2 \ \mu F$	X5R, 20%
CX1, CX2	4 pF	C0G, 5%, 50V
CX3, CX4	12 pF	C0G, 5%, 50V
FB1	BLI	M21B102S
L1	BLM1	5BB221SN1
LP1	LQM2	1PN2R2NGC
R1	$10~\mathrm{k}\Omega$	1%
R2	$1 \ \Omega$	1%
R3	8,2 k Ω	1%
U1	EFM3:	2WG332F256
U2	АТ	86RF233
U3	TS	5A3159A
U4	Т	PS62732
X1	CX2016	DB48CWPLA2
X2	$16 \mathrm{~MHz}$	SX-4025

C UKÁZKA ANALYTICKÉHO FRAMEWORKU

1 No.W 🔻 No. 🔻 Ti	ne 🔻 MCU_Time 🔻 Source	▼ Destination	▼ Protocol ▼ 12.	seq 🔻 LwM.seq	🔻 LwM.ack.seg 🔻	Info	Comment	ength 🔻 Diff 👻
21443 21745 38785	144,65075 2536,359136 0×0001	0×0000	LwMesh	241 2	40	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00141
21444 21746 38786	144,651185 2536,360129		IEEE 802.15.4	241		Ack		5 0,00099
21445 21747 38787	144,653509 2536,361892 0x0001	0×000	LwMesh	242 2	41	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00176
21446 21748 38788	144,653948 2536,362885		IEEE 802.15.4	242		Ack		5 0,00099
21447 21749 38789	144,657233 2536,365615 0x0001	0×0000	LwMesh	243 24	42	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00273
21448 21750 38790	144,65767 2536,366608		IEEE 802.15.4	243		Ack		5 0,00099
21449 21751 38791	144,659948 2536,368336 0x0001	0×0000	LwMesh	244 2	43	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00173
21450 21752 38792	144,660397 2536,369329		IEEE 802.15.4	244		Ack		5 0,00099
21451 21753 38793	144,66362 2536,372008 0×0001	0×0000	LwMesh	245 2	44	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00268
21452 21754 38794	144,664058 2536,373002		IEEE 802.15.4	245		Ack	2006	5 0,00099
21453 21781 38795	161,41367 2553,122113 0x0001	Broadcast	LwMesh	1	1	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 16,7491
21454 21782 38796	161,415584 2553,123942 0×0000	0×0001	LwMesh	1	1 1	LwMesh ACK		21 0,00183
21455 21783 38797	161,415994 2553,124999		IEEE 802.15.4	1		Ack		5 0,00106
21456 21784 38798	161,417507 2553,125968 0x0002	Broadcast	LwMesh	1	1	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00097
21457 21785 38799	161,95629 2553,664739 0×0002	Broadcast	LwMesh	2	1	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0002		19 0,53877
21458 21786 38800	161,95818 2553,666565 0x0000	0×0002	LwMesh	2	2 1	LwMesh ACK		21 0,00183
21459 21787 38801	161,95863 2553,667623		IEEE 802.15.4	2		Ack		5 0,00106
21460 21788 38802	161,960441 2553,668898 0×0001	Broadcast	LwMesh	2	1	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0002		19 0,00128
21461 21789 38803	163,041237 2554,749687 0×0001	0×000	LwMesh	ę	2	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 1,08079
21462 21790 38804	163,041681 2554,75068		IEEE 802.15.4	m		Ack		5 0,00099
21463 21791 38805	163,042775 2554,751238 0×0002	0×0000	LwMesh	m	2	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0002		19 0,00056
21464 21792 38806	163,04321 2554,752232		IEEE 802.15.4	n		Ack		5 0,00099
21465 21793 38807	163,044622 2554,75309 0x0001	0×0000	LwMesh	4	m	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00086
21466 21794 38808	163,04507 2554,754084		IEEE 802.15.4	ব		Ack		5 0,00099
21467 21795 38809	163,046579 2554,755048 0x0001	0×0000	LwMesh	4	07	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00096
21468 21796 38810	163,047027 2554,756042		IEEE 802.15.4	4		Ack		5 0,00099
21469 21797 38811	163,048182 2554,756647 0x0002	0×0000	LwMesh	4	0	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0002		19 0,00061
21470 21798 38812	163,048621 2554,757639		IEEE 802.15.4	ব		Ack		5 0,00099
21471 21799 38813	163,050027 2554,758498 0x0001	0×000	LwMesh	IJ	4	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00086
21472 21800 38814	163,050478 2554,759492		IEEE 802.15.4	'n		Ack		5 0,00099
21473 21801 38815	163,051866 2554,760333 0×0002	0×0000	LwMesh	2	4	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0002		19 0,00084
21474 21802 38816	163,052303 2554,761327		IEEE 802.15.4	Ω		Ack		5 0,00099
21475 21803 38817	163,053447 2554,761909 0x0001	0×0000	LwMesh	9	ſ	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00058
21476 21804 38818	163,053881 2554,762902		IEEE 802.15.4	9		Ack		5 0,00099
21477 21805 38819	163,054921 2554,763392 0x0002	0×0000	LwMesh	9	S	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0002		19 0,00049
21478 21806 38820	163,055371 2554,764386		IEEE 802.15.4	9		Ack		5 0,00099
21479 21807 38821	163,058429 2554,766894 0x0001	0×0000	LwMesh	7	9	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00251
21480 21808 38822	163,058866 2554,767888		IEEE 802.15.4	7		Ack		5 0,00099
21481 21809 38823	163,061781 2554,770253 0×0001	0×0000	LwMesh	œ	2	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00237
21482 21810 38824	163,062232 2554,771246		IEEE 802.15.4	00		Ack		5 0,00099
21483 21811 38825	163,063886 2554,772355 0x0001	0×0000	LwMesh	00	2	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00111
21484 21812 38826	163,064332 2554,773348		IEEE 802.15.4	00		Ack		5 0,00099
21485 21813 38827	163,065488 2554,773953 0x0002	0×0000	LwMesh	9	ſ	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0002		19 0,0006
21486 21814 38828	163,065925 2554,774946		IEEE 802.15.4	9		Ack		5 0,00099
21487 21815 38829	163,068383 2554,776853 0x0001	0×0000	LwMesh	б	00	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00191
21488 21816 38830	163,068834 2554,777846		IEEE 802.15.4	σ		Ack		5 0,00099
21489 21817 38831	163,071122 2554,779587 0×0001	0×0000	LwMesh	10	σ	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00174
21490 21818 38832	163,071558 2554,780579		IEEE 802.15.4	10		Ack		5 0,00099
21491 21819 38833	163,074864 2554,783328 0x0001	0×0000	LwMesh	11	10	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001		19 0,00275
21492 21820 38834	163,075307 2554,784322		IEEE 802.15.4	11		Ack		5 0,00099
21493 21821 38835	163,077138 2554,785607 0x0002	0×0000	LwMesh	2	9	Lightweight Mesh, Nwk Dst: 0x0000, Nwk Src: 0x0002		19 0,00128
HART FRANK LINE AND							100	

Obr. C.1: Analytický framework. Test propustnosti dat na aplikační vrstvě.

Purpurová barva značí sestavení spojení, červená konec relace, zelená a tyrkysová potvrzené pakety a žlutá duplicitně potvrzené pakety.

1 No.W 👻	No. 🔻 Time	🔻 MCU_Time 🔻 Source	▼ Destination	▼ Protocol ▼ 12.5	eq 🔻 LwM.seq	 LwM.ack.seq 	r Info	▼ Comment	🔻 Length 🔻 Diff 🔻
3552 3569	5E+05 29,636	619 82001,60256 0×0001	0×0002	LwMesh	61	61	Encrypted data (77 bytes)		99 0,00723
3553 3570	5E+05 29,636.	775 82001,60612		IEEE 802.15.4	61		Ack		5 0,00355
3554 3571	5E+05 29,643	931 82001,60978 0x0002	0×0003	LwMesh	84	61	Encrypted data (77 bytes)		99 0,00366
3555 3572	5E+05 29,644	031 82001,61334		IEEE 802.15.4	84		Ack		5 0,00355
3556 3573	5E+05 29,650	709 82001,61659 0x0003	0x0004	LwMesh	72	61	Encrypted data (77 bytes)		99 0,00325
3557 3574	5E+05 29,650	804 82001,62014		IEEE 802.15.4	72		Ack		5 0,00355
3558 3575	5E+05 29,669.	936 82001,63588 0×0001	0×0002	LwMesh	62	62	Encrypted data (77 bytes)		99 0,01574
3559 3576	5E+05 29,670	092 82001,63943		IEEE 802.15.4	62		Ack		5 0,00355
3560 3577	5E+05 29,676	514 82001,64244 0×0002	0×0003	LwMesh	85	62	Encrypted data (77 bytes)		99 0,00301
3561 3578	5E+05 29,677.	185 82001,646		IEEE 802.15.4	85		Ack		5 0,00355
3562 3579	5E+05 29,682	339 82001,64826 0x0003	0x0004	LwMesh	73	62	Encrypted data (77 bytes)		99 0,00227
3563 3580	5E+05 29,682.	476 82001,65182		IEEE 802.15.4	73		Ack		5 0,00355
3564 3581	5E+05 29,702.	164 82001,6681 0×0001	0×0002	LwMesh	63	63	Encrypted data (77 bytes)		99 0,01628
3565 3582	5E+05 29,702:	307 82001,67165		IEEE 802.15.4	63		Ack		5 0,00355
3566 3583	5E+05 29,707(805 82001,67375 0×0002	0×0003	LwMesh	86	63	Encrypted data (77 bytes)		99 0,0021
3567 3584	5E+05 29,707:	959 82001,6773		IEEE 802.15.4	86		Ack		5 0,00355
3568 3585	5E+05 29,71	464 82001,68052 0x0003	0x0004	LwMesh	74	63	Encrypted data (77 bytes)		99 0,00322
3569 3586	5E+05 29,714.	733 82001,68407		IEEE 802.15.4	74		Ack		5 0,00355
3570 3587	5E+05 29,734	403 82001,70034 0x0001	0×0002	LwMesh	64	64	Encrypted data (77 bytes)		99 0,01627
3571 3588	5E+05 29,734;	554 82001,7039		IEEE 802.15.4	64		Ack		5 0,00355
3572 3589	5E+05 29,74(076 82001,7067 0×0002	0×0003	LwMesh	87	64	Encrypted data (77 bytes)		99 0,0028
3573 3590	5E+05 29,740:	909 82001,71025		IEEE 802.15.4	87		Ack		5 0,00355
3574 3591	5E+05 29,746	576 82001,71252 0x0003	0×0004	LwMesh	75	64	Encrypted data (77 bytes)		99 0,00227
3575 3592	5E+05 29,746.	731 82001,71607		IEEE 802.15.4	75		Ack		5 0,00355
3576 3593	5E+05 29,766	084 82001,73193 0×0001	0×0002	LwMesh	65	65	Encrypted data (77 bytes)		99 0,01585
3577 3594	5E+05 29,766.	177 82001,73548		IEEE 802.15.4	65		Ack		5 0,00355
3578 3595	5E+05 29,77.	351 82001,73945 0x0002	0×0003	LwMesh	88	65	Encrypted data (77 bytes)		99 0,00397
3579 3596	5E+05 29,773(665 82001,74301		IEEE 802.15.4	88		Ack		5 0,00355
3580 3597	5E+05 29,780	271 82001,74622 0x0003	0×0004	LwMesh	76	65	Encrypted data (77 bytes)		99 0,00321
3581 3598	5E+05 29,780	426 82001,74977		IEEE 802.15.4	76		Ack		5 0,00355
3582 3599	5E+05 29,799	499 82001,76544 0x0001	0×0002	LwMesh	66	66	Encrypted data (77 bytes)		99 0,01567
3583 3600	5E+05 29,799.	659 82001,76899		IEEE 802.15.4	99		Ack		5 0,00355
3584 3601	5E+05 29,807.	116 82001,77306 0×0002	0×0003	LwMesh	89	66	Encrypted data (77 bytes)		99 0,00407
3585 3602	5E+05 29,807.	286 82001,77662	_	IEEE 802.15.4	89	_	Ack		5 0,00355
3586 3603	5E+05 29,813	263 82001,7792 0x0003	0x0004	LwMesh	27	66	Encrypted data (77 bytes)		99 0,00259
3587 3604	5E+05 29,813	412 82001,78276		IEEE 802.15.4	77	_	Ack		5 0,00355
3588 3605	5E+05 29,831	401 82001,79734 0×0001	0×0002	LwMesh	67	67	Encrypted data (77 bytes)		99 0,01458
3589 3606	5E+05 29,83.	156 82001,80089		IEEE 802.15.4	67		Ack		5 0,00355
3590 3607	5E+05 29,838	1995 82001,80488 0x0002	0×0003	LwMesh	06	67	Encrypted data (77 bytes)		99 0,00398
3591 3608	5E+U5 29,839	092 82001,80843	_	IEEE 802.15.4	8	_	Ack		5 U,UU355
3592 3609	5E+05 29,846	103 82001,81195 0×0003	0×0004	LwMesh	78	67	Encrypted data (77 bytes)		99 0,00352
3593 3610	5E+05 29,846.	198 82001,8155		IEEE 802.15.4	78		Ack		5 0,00355
3594 3611	5E+05 29,865.	193 82001,83084 0x0001	0x0002	LwMesh	68	68	Encrypted data (77 bytes)		99 0,01533
3595 3612	5E+05 29,865.	194 82001,83439		IEEE 802.15.4	68	_	Ack		5 0,00355
3596 3613	5E+05 29,872.	176 82001,83812 0x0002	0×0003	LwMesh	91	68	Encrypted data (77 bytes)		99 0,00373
3597 3614	5E+05 29,872.	331 82001,84167		IEEE 802.15.4	91		Ack		5 0,00355
3598 3615	5E+05 29,878	956 82001,8449 0×0003	0x0004	LwMesh	79	68	Encrypted data (77 bytes)		99 0,00323
3599 3616	5E+05 29,879.	118 82001,84846		IEEE 802.15.4	79		Ack		5 0,00355
3600 3617	5E+05 29,897	185 82001,86307 0x0001	0×0002	LwMesh	69	69	Encrypted data (77 bytes)		99 0,01461
3601 3618	5E+05 29,897.	277 82001,86662	_	IEEE 802.15.4	69	_	Ack		5 0,00355
3602 3619	5E+05 29,904	011 82001,86996 0×0002	0×0003	LwMesh	92	69	Encrypted data (77 bytes)		99 0,00334
IA A P PI LIS	t1 / Listz / List3	(M)							

Obr. C.2: Analytický framework. Test zpoždění směrováním.

Tyrkysová barva značí začátek relace a purpurová konec relace.

1 No.W	V ▼ No.	▼ Time	▼ MCU_Time ▼ Source	 Destination 	Protocol 🔻 12.	seq 🔻 LwM	seq 💌 LwM.ack.seq 🔻	↓	Comment 🗾 🗸 Length 🗸 🛛	Diff 🗸
690 3.	911 2E	+06 725,92	20974 13706,45834 0x0001	0×0010	LwMesh	120	120	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01617
691 3	912 2E	+06 725,92	21444 13706,45933		IEEE 802.15.4	120		Ack	U	0,00099
692 3	913 2E	+06 725,92	25189 13706,46256 0x0010	0×0000	LwMesh	121	120	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00323
3693 3.	914 2E	+06 725,92	25674 13706,46356		IEEE 802.15.4	121		Ack	Ω.	0,00099
3694 3	915 2E	+06 725,94	42078 13706,47939 0x0001	0×0010	LwMesh	121	121	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01584
3695 33	916 2E	+06 725,94	43851 13706,48121 0×0001	0×0010	LwMesh	121	121	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00182
3696 33	917 2E	+06 725,94	45671 13706,48303 0x0001	0×0010	LwMesh	121	121	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00182
3697 3:	918 2E	+06 725,94	47794 13706,48517 0x0001	0×0010	LwMesh	121	121	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00213
3698 3:	919 2E	+06 725,96	53415 13706,50078 0x0001	0×0010	LwMesh	122	122	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01562
3699 3:	920 2E	+06 725,96	57153 13706,50452 0x0001	0×0010	LwMesh	122	122	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00373
3700 3:	921 2E	+06 725,97	70558 13706,50793 0x0001	0×0010	LwMesh	122	122	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00341
3701 33	922 2E	+06 725,97	73011 13706,51038 0×0001	0×0010	LwMesh	122	122	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00245
3702 3.	923 2E	+06 725,98	95745 13706,52311 0x0001	0×0010	LwMesh	123	123	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01273
3703 3.	924 2E	+06 725,96	38518 13706,52589 0×0001	0×0010	LwMesh	123	123	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00277
3704 3:	925 2E	+06 725,95	92249 13706,52962 0×0001	0×0010	LwMesh	123	123	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00373
3705 33	926 2E	+06 725,95	95341 13706,53271 0×0001	0×0010	LwMesh	123	123	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00309
3706 3	927 2E	+06 726,00	06202 13706,54357 0x0001	Broadcast	LwMesh	124	124	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01086
3707 3:	928 2E	+06 726,00	38196 13706,54557 0x0011	Broadcast	LwMesh	-	124	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00199
3708 31	929 2E	+06 726,00	19954 13706,54725 0x0000	0×0010	LwMesh	2	2 12	t LwMesh ACK	21	0,00168
3709 31	930 2E	+06 726,01	12149 13706,54945 0x0000	0×0010	LwMesh	2	2 12	t LwMesh ACK	21	0,0022
3710 31	931 2E	+06 726,01	14662 13706,55196 0x0000	0×0010	LwMesh	2	2 12	1 LwMesh ACK	21	0,00252
3711 3:	932 2E	+06 726,01	17819 13706.55512 0x0000	0×0010	LwMesh	2	2	t LwMesh ACK	21	0.00316
3712 34	933 2F.	+0.6 726.02	28553 13706 56593 0×0001	Broadcast	IwMesh	125	125	Lightweight Mesh Nwk Dst Ox0000 Nwk Src. 0x0001	6	0.01081
3713 3	934 2E	+06 726.03	30792 13706,56816 0x0011	Broadcast	LwMesh	2	125	Lightweight Mesh, Nwk Dst 0x0000, Nwk Src: 0x0001	19	0.00224
3714 39	935 2F	+06 726.03	32545 13706 56984 0x0000	0×0010	twMesh		3	5 IwMesh ACK	21	0 00168
3715 3	936 2E	+06 726,03	34425 13706,57173 0x0000	0×0010	LwMesh	0	3 12	5 LwMesh ACK	21	0,00188
3716 3	937 2E	+06 726.0	13631 13706.57361 0×0000	0×0010	LwMesh	m	3 12	5 LwMesh ACK	21	0.00188
3717 3:	938 2E	+06 726,03	38188 13706,57549 0×0000	0×0010	LwMesh	0	3	5 LwMesh ACK	21	0,00188
3718 3:	939 2E	+06 726,05	51512 13706,58889 0x0001	Broadcast	LwMesh	126	126	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01339
3719 3:	940 2E	+06 726,05	54714 13706,59208 0x0011	Broadcast	LwMesh	m	126	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,0032
3720 3:	941 2E	+06 726,05	56438 13706,59373 0×0000	0×0010	LwMesh	4	4 12(5 LwMesh ACK	21	0,00165
3721 3:	942 2E	+06 726,05	58317 13706,59562 0×0000	0×0010	LwMesh	4	4 12(5 LwMesh ACK	21	0,00188
3722 3:	943 2E	+06 726,06	50513 13706,59781 0x0000	0×0010	LwMesh	4	4 12(5 LwMesh ACK	21	0,0022
3723 3:	944 2E	+06 726,06	53031 13706,60033 0x0000	0×0010	LwMesh	4	4 12	5 LwMesh ACK	21	0,00252
3724 3:	945 2E	+06 726,07	72895 13706,61027 0x0001	Broadcast	LwMesh	127	127	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00994
3725 3.	946 2E	+06 726,07	75156 13706,61253 0x0011	Broadcast	LwMesh	4	127	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00226
3726 3.	947 2E	+06 726,07	78529 13706,61583 0x0000	0×0011	LwMesh	ы	5 12	7 LwMesh ACK	21	0,0033
3727 3.	948 2E	+06 726,07	78996 13706,61688		IEEE 802.15.4	u		Ack	D	0,00106
3728 3.	949 2E	+06 726,08	82533 13706,61922 0×0011	0×0001	LwMesh	ы	5 12	7 LwMesh ACK	21	0,00233
3729 3	950 2E	+06 726,06	82534 13706,62027	_	IEEE 802.15.4	n		Ack	Q	0,00106
3730 3	951 2E	+06 726,05	93933 13706,6313 0×0001	0×0011	LwMesh	128	128	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01102
3731 3.	952 2E	+06 726,05	94535 13706,63229		IEEE 802.15.4	128		Ack	ы	0,00099
3732 3.	953 2E	+06 726,05	96234 13706,6336 0×0011	0×0000	LwMesh	9	128	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00131
3733 3.	954 2E	+06 726,05	97542 13706,6346		IEEE 802.15.4	9		Ack	U	0,00099
3734 3.	955 2E	+06 726,11	14545 13706,65174 0x0001	0×0011	LwMesh	129	129	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01715
3735 3.	956 2E	+06 726,11	14847 13706,65273		IEEE 802.15.4	129		Ack	U	0,00099
3736 3.	957 2E	+06 726,11	17536 13706,65437 0×0011	0×0000	LwMesh	7	129	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,00163
3737 3.	958 2E	+06 726,11	17537 13706,65536		IEEE 802.15.4	7		Ack	U	0,00099
3738 3.	959 2E	+06 726,13	36545 13706,67375 0x0001	0×0011	LwMesh	130	130	Lightweight Mesh, Nwk_Dst: 0x0000, Nwk_Src: 0x0001	19	0,01839
3739 3.	960 2E	+06 726,15	36866 13706,67475		IEEE 802.15.4	130		Ack	α	0,00099
3740 3.	961 2E	+06 726.1	14054 13706,67734 0x0011	0×0000	LwMesh	œ	130	Lightweight Mesh, Nwk Dst. 0x0000, Nwk Src: 0x0001	19	0,00259
14 A B B	LISTI	LISTZ / LIST3								

Obr. C.3: Analytický framework. Test samo opravné schopnosti sítě.

Purpurová barva značí první nedoručený paket, žlutá paket s žádostí o novou cestu, tyrkysová paket využívající nové cesty.

PUBLIKACE AUTORA

Publikace v časopisech s impaktním faktorem

- ([41]) Červenka, V.; Mráz, L.; Komosný, D.: Comprehensive Performance Analysis of Lightweight Mesh and its comparison with ZigBee Pro Technology. Wireless Personal Communication, 2014, doi: 10.1007/s11277-014-1832-x, v tisku. (IF: 0,979)
- ([109]) Mráz, L.; Červenka, V.; Šimek, M.; Komosný, D.: Comprehensive Performance Analysis of ZigBee Technology Based on Real Measurements. Wireless Personal Communication, 2013: s. 2783–2803, ISSN 0929-6212. (IF: 0,979)
- ([107]) Mráz, L.; Červenka, V.: Complete System Design Based on Wireless Sensor Network for Disaster Prevention. In *Proceedings of 35th International Conference on Telecommunications and Signal Processing (TSP 2012)*, ročník 10a, Przeglad Elektrotechniczny, 2012, ISSN 0033-2097, s. 273–277. (IF: 0,244)

Ostatní publikace

- ([39]) Červenka, V.; Mráz, L.; Šimek, M.: Integration and Visualization of Adaptive Wireless Sensor Network. In Poster and Demo Proceedings of 10th European Conference on Wireless Sensor Network (EWSN 2013), 2013.
- ([64]) Šimek, M.; Mráz, L.; Červenka, V.; Botta, M.; Julina, V.: Wireless Light-weight Snow Sensor. In Poster and Demo Proceedings of 10th European Conference on Wireless Sensor Network (EWSN 2013), 2013.
- ([37]) Červenka, V.; Komosný, D.; Malina, L.; Mráz, L.: Energy Efficient Public Key Cryptography in Wireless Sensor Networks. In *Innovations and Advances in Computer, Information, Systems Sciences, and Engineering*, USA, Berkeley: Springer New York, 2012, ISBN 978-1-4614-3534-1, s. 497–509.
- ([40]) Červenka, V.; Mráz, L.; Komosný, D.: Top-Down Architecture for Environmental Monitoring Using Wireless Sensor Network. In *Proceedings of 35th International Conference* on *Telecommunications and Signal Processing (TSP 2012)*, 2012, ISBN 978-1-4673-1118-2, s. 91–95.
- ([108]) Mráz, L.; Červenka, V.; Šimek, M.: Packet Analyser for IEEE 802.15.4 Networks. In Poster and Demo Proceedings of 9th European Conference on Wireless Sensor Network (EWSN 2012), 2012.
- ([63]) Šimek, M.; Mráz, L.; Červenka, V.; Pechanec, V.: Adaptive Wireless Sensor Network for Urban Crisis Management. In Poster and Demo Proceedings of 9th EuropeanConference

on Wireless Sensor Network, 2012.

- ([38]) Červenka, V.; Mráz, L.; Šimek, M.: Reusing AES Coprocessor in Public Key Cryptography. In Poster and Demo Proceedings of 9th EuropeanConference on Wireless Sensor Network, 2012.
- ([36]) Červenka, V.; Komosný, D.; Kathiravelu, G.: IETF 6LoWPAN and Sensor Networking. In *ICT2011*, Vysoké učení technické v Brně, 2011, ISBN 978-80-214-4231-3, s. 69–73.
- ([106]) Müller, J.; Červenka, V.; Komosný, D.: Taking Advantage of DLNA Technology in Education. In New Information and Multimedia Technologies (NIMT), VUT Brno, 2011, ISBN 978-80-214-4241-2, s. 1–4.
- ([110]) Mráz, L.; Komosný, D.; Červenka, V.; Morávek, P.; Šimek, M.: Open-Packet Analyser Platform for Wireless Sensor Networks Based on IEEE 802.15.4. In 34th International Conference on Telecommunications and Signal Processing (TSP 2011), 2011, ISBN 978-1-4577-1409-2, s. 145 149.

Produkty

- 1. Červenka, V.; Velký, P.; Mráz, L.; Komosný, D. WSN Server Control Centre. 2012.
- Mráz, L.; Komosný, D.; Červenka, V.: Mininode, Platforma pro bezdrátové senzorické sítě založené na standardech IEEE 802.15.4/ Zigbee. 2010.

Curriculum Vitæ Vladimír Červenka

Osobní informace

Datum narození:	29. dubna, 1986
Místo narození:	Brno
Adresa:	Berlínská 2747, 390 05 Tábor
E-mail:	cervenka.v@email.cz

Vzdělání

2010-2014	Vysoké učení technické v Brně, Fakulta elektrotechniky a ko-
	munikačních technologií, Technická 3058/10, 616 00 Brno,
	titul: Ph.D.
2008-2010	Vysoké učení technické v Brně, Fakulta elektrotechniky a ko-
	munikačních technologií, Technická 3058/10, 616 00 Brno,
	titul: Ing.
2005 - 2008	Vysoké učení technické v Brně, Fakulta elektrotechniky a ko-
	munikačních technologií, Technická 3058/10, 616 00 Brno,
	titul: Bc.
2001 - 2005	Střední škola spojů a informatiky, Bydlinského 2474, 390 11
	Tábor

Dodatečné informace ke vzdělání

2009-2010	Tampere Uni	versity of	Technology,	${ m Korkeakoulunkatu}$	10,
	FI-33720 Tam	pere, Fins	ko		

Ocenění

2010 Magisterský titul s vyznamenáním

Předchozí zaměstnání

2011–2014 *vědecký pracovník*: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, Technická 12, 616 00 Brno

2011–2012 *embedded developer*: Herman elektronika, Brněnská 993, 664 42 Modřice

Participace na projektech

2014 - 2015	OPVK CZ.1.07/3.2.04/06.0005, Moderní síťové technologie
2012 - 2015	OPVK CZ.1.07/2.2.00/28.0096, Příprava specialistů pro te-
	lekomunikace a informatiku - magisterské studium telekomu-
	nikační a informační technika
2012-2013	HS-Honeywell, Výzkum přenosu dat v oblasti UWB a tech-
	nologie IEEE 802.15.4a
2012-2013	FRVŠ1365/2012, Začlenění problematiky bezpečnosti počíta-
	čových sítí a technologie Intrusion Prevention System do vý-
	uky předmětu Cisco akademie 2
2011 - 2012	FRVŠ2693/2011, Začlenění problematiky bezdrátových sítí a
	technologie Digital Living Network Alliance do výuky před-
	mětu Cisco akademie 2
2011 - 2014	OPVK CZ.1.07/2.3.00/20.0094, Podpora zapojení vědecko-
	výzkumných týmů do mezinárodní spolupráce v oblasti zpra-
	cování obrazových a zvukových signálů
2011 - 2012	OPVK CZ.1.07/1.3.10/03.0007, Vzdělávání pedagogů pro
	lepší využití ICT při výuce na základních a středních školách
2011 - 2013	OPVK CZ.1.05/2.1.00/03.0072, Centrum senzorických, in-
	formačních a komunikačních systémů (SIX)
2011 - 2013	MPO/FR-TI2/571, Adaptibilní bezdrátové senzorové sítě s
	vizualizací dat pro krizové řízení
2011 - 2012	OPVK CZ.1.07/2.3.00/09.0222, Vzdělávací centrum pro zvý-
	šení zájmu mládeže o výzkumné profese v oblasti informač-
	ních a komunikačních technologií

Vyžádané přednášky

- Wireless Sensor Networks in Biomedicine, University Institute for Technological Development and Innovation in Communication, University of Las Palmas, Španělsko.
- Advances in Analysis of Wireless Sensor Networks, Faculty of Sciences and Technology, University of Coimbra, Portugalsko.

Vyžádané recenze

- Elsevier, Computer Communications, ISSN: 0140-3664.
- Springer, Wireless Personal Communication, ISSN: 0929-6212.
- Hindawi Publishing Corporation, Journal of Applied Mathematics, ISSN: 1110-757X.
- International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems, ISSN: 1805-5443.
- International Work Conference on Bioinspired Intelligence (IWOBI 2014), ISBN: 978-1-4799-6174-0.
- 6th International Conference on Teleinformatics (ICT 2011).