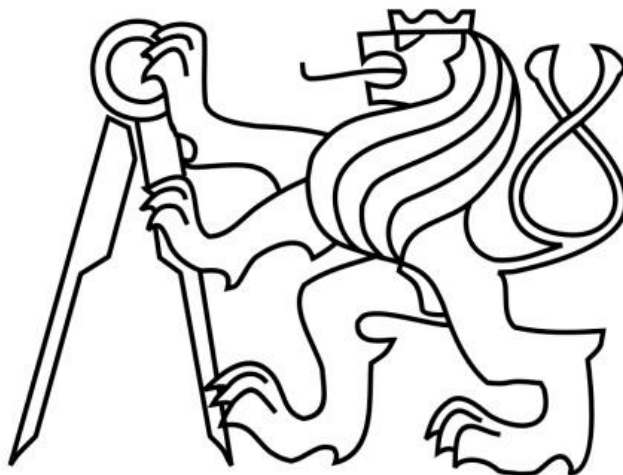


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

Bezdrátový senzorový systém

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

.....

Podpis autora práce

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jakub Štoud**

Studijní program: Otevřená informatika (magisterský)
Obor: Počítačové inženýrství

Název tématu: **Bezdrátový senzorový systém**

Pokyny pro vypracování:

1. S pomocí platformy Texas Instruments MSP430 realizujte systém pro bezdrátový přenos informace ze senzorů s vestavěným zpracováním signálů.
2. Systém bude tvořen autonomními jednotkami (uzly) provádějícími snímání signálů ze senzorů (teplota, vlhkost a další), jejich zpracování a přenos do sousedních jednotek a do centrálního uzlu připojeného po rozhraní USB k počítači. V jednotlivých uzlech implementujte několik metod pro hodnocení informačního obsahu snímané veličiny (dle [1]) a jednoduchý inferenční senzor.
3. Maximalizujte otevřenost systému a kvalitu dokumentace pro budoucí použití při vývoji metod distribuovaného zpracování signálů ze senzorů.

Seznam odborné literatury:

- [1] O. Kreibich, J. Neuzil, R. Šmíd, Quality-based Multiple Sensor Fusion in an Industrial Wireless Sensor Network for MCM, to appear in IEEE Transactions on Industrial Electronics, 2013.
[2] Dokumentace MSP430, firemní literatura Texas Instruments, on-line.

Vedoucí: Doc.Ing. Radislav Šmíd, Ph.D.

Platnost zadání: do konce letního období 2014/2015

Poděkování

Děkuji především svému vedoucímu práce doc. Ing. Radislavu Šmídovi za jeho ochotu a trpělivost mi vždy poradit a nasměrovat mě správným směrem ve vypracování diplomové práce. Také děkuji všem profesorům, přednášejícím a cvičícím, kteří mě vyučovali po celou dobu studia.

Anotace

Cílem této diplomové práce je navrhnout a vytvořit bezdrátovou senzorovou síť s přenosem výstupů do nadřazeného zařízení. Bezdrátová senzorová síť bude složena z vývojových kitů od firmy Texas Instruments. Z naměřených hodnot bude v uzlech sítě zpracován ukazatel kvality výsledků, který bude spolu s výsledky měření odesílán do centrálního prvku sítě. Ten bude tato data dále předávat do počítače, pro libovolnou interpretaci uživateli.

Annotation

The Goal of this diploma thesis is to design and create a wireless sensor network capable of reporting measured values to a superior device. The wireless sensor network consists of development kits from Texas Instruments. Quality indicator, which will be transmitted to a central point of the network along with the results of measurements, will be computed in the nodes of the network from measured values. This central point will hand the data over to a computer for user's interpretation.

Obsah

1	Úvod.....	1
2	State-of-the-art	2
2.1	Aplikace bezdrátových sensorových sítí.....	2
2.2	Typická architektura uzlu v síti.....	3
2.3	Topologie sítí.....	4
2.4	Open-source a proprietární platformy	4
2.5	Srovnání používaných platforem	5
2.5.1	Intel XScale	6
2.5.2	Crossbow	7
2.5.3	TI MSP430 a rádio CC2500RF	8
2.5.4	SmartMesh	10
2.5.5	BTnode	11
2.6	Druhy programování systémů	11
2.6.1	Jazyk C a knihovny pro konkrétní hardware.....	11
2.6.2	Událostmi řízené programování	12
2.6.3	S operačním systémem.....	12
3	Vstupní zařízení a periferie	14
3.1	Základní aplikace s tlačítkem.....	14
3.2	Knihovna pro práci se sériovým rozhraním	15
3.3	Senzor SHT11	17
3.3.1	Komunikace se senzorem.....	18
4	Bezdrátová komunikace	24
4.1	Protokolový zásobník SimpliciTI.....	24
4.1.1	Popis protokolového zásobníku SimpliciTI.....	24
4.1.2	Jak postupovat při implementaci.....	25
4.2	Přístupový bod.....	26

4.3	Koncový uzel	27
4.4	Rozšiřovač dosahu	28
5	Metody pro vyhodnocování výsledků měření snímané veličiny	28
5.1	Q1 – fyzikální limity.....	31
5.2	Q2 – úroveň napětí baterie.....	32
5.3	Q3 – síla signálu RF	33
5.4	Q4 – lineární regrese.....	33
5.5	Q5 – maximální rozdíl mezi vzorky	35
5.6	Q6 – směrodatná odchylka	36
5.7	Knihovna pro výpočet parametrů Q	37
6	Praktické ověření bezdrátové senzorové sítě.....	38
6.1	Změna teploty a vlhkosti ve vnějším prostředí.....	38
6.2	Změna teploty a vlhkosti ve vnitřním prostředí.....	40
7	Závěr.....	42
	Seznam obrázků.....	43
	Seznam tabulek.....	43
	Obsah příloženého CD.....	44
	Reference	45

1 Úvod

V této práci se jedná o realizaci systému pro bezdrátový přenos informací ze senzorů s vestavěným zpracováním signálu. Aby toto bylo možné, bude navržena bezdrátová senzorová síť tvořená centrálním bodem a koncovými uzly, které se budou k tomuto centrálnímu prvku připojovat. V koncových bodech bude realizována metoda pro hodnocení informačního obsahu snímaných veličin ze senzorů. Toto spočtené hodnocení naměřených dat bude spolu s daty samotnými odesílání přístupovému bodu, který bude připojen pomocí rozhraní k počítači, kde budou data zpřístupněna pro interpretaci uživateli.

Navrhovaná implementace tohoto zadání sestává z vývojového nástroje od firmy Texas Instruments eT430-RF2500 založeném na procesoru MSP430 stejného výrobce a bezdrátovém rádiu CC2500RF. Tato platforma bude využita jak pro implementaci centrálního prvku sítě, tak i pro vytvoření koncových uzlů v síti. Komunikace v bezdrátové senzorové síti bude probíhat v pásmu 2,4 GHz. Bude použit výrobcem dodávaný protokolový zásobník SimpliciTI. Měřenými veličinami bude teplota a relativní vlhkost vzduchu. K měření bude použit senzor SHT11, který měří obě tyto fyzikální veličiny.

Tato metoda hodnocení informačního obsahu naměřených veličin je tvořena spočtením dílčích ukazatelů kvality závislých na aktuálním stavu měřícího uzlu, nebo na předchozích naměřených hodnotách. K jejímu ověření bude nutné implementovat několik dílčích částí.

- Bezdrátovou senzorovou síť s přístupovým bodem a koncovými uzly
- Komunikační protokol s vybraným senzorem
- Knihovnu pro výpočet a sjednocení dílčích ukazatelů kvality naměřených dat.

2 State-of-the-art

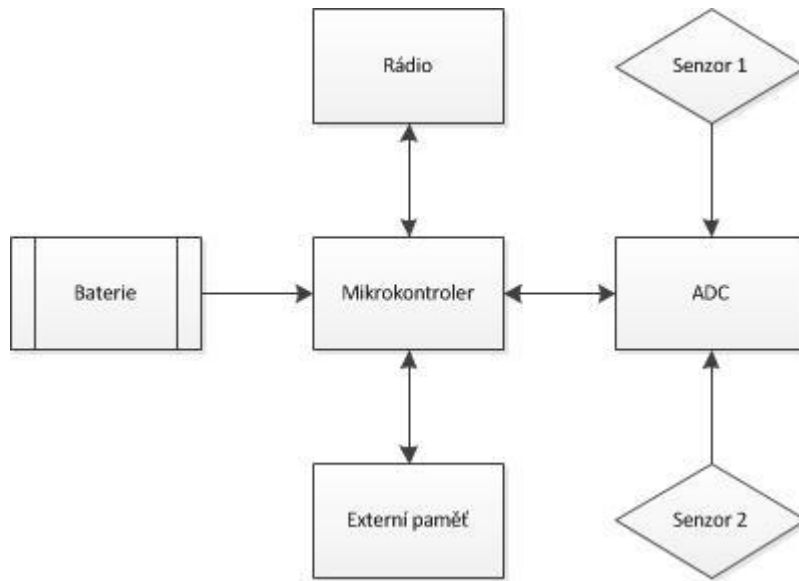
V této části teoreticky proberu aktuální stav problematiky. V jakých oblastech se bezdrátové senzorové sítě využívají, jakými způsoby je možné tyto sítě programovat a jaké jsou různé vývojové platformy, na kterých se síť dá vytvořit. Rozeberu jejich výhody a nevýhody.

Bezdrátové senzorové sítě (WSN Wireless sensor network) jsou bezdrátové sítě, skládající se z prostorově distribuovaných autonomních zařízení, která používají senzory k monitorování okolního prostředí a fyzikálních veličin v něm. Takovýto senzorový systém obsahuje výchozí bránu, která zprostředkovává drátové připojení do okolního světa a další uzly. V praxi se využívá velké množství bezdrátových komunikačních standardů. Výběr konkrétního standardu vždy závisí na specifikaci a požadavcích aplikace. Používají se standardy Bluetooth, Wi-Fi, nebo ZigBee.

2.1 Aplikace bezdrátových senzorových sítí

Tyto sítě jsou tvořeny a využívány v mnoha aplikacích, jako například vzdálený monitoring oblasti, péče o zdraví, správu inteligentních budov, nebo monitorování průmyslových strojů. Díky těmto sítím je možné vzdáleně sledovat zdravotní stav pacientů bez přílišného narušení jejich životního stylu, nebo soukromí. Senzorové sítě jsou hojně využívány ve správě veřejných prostředků a služeb. Například v rozvodné vodovodní síti, elektrické síti nebo v pouličním osvětlení poskytují tyto sítě velice levnou metodu, jak vzdáleně kontrolovat stav a funkčnost těchto systémů. Nahrazují drátové systémy díky snižování nákladů na fyzická média. V nových kancelářských budovách je možné bez zbytečných nákladů nastavovat klimatizace, nebo třeba automaticky stahovat žaluzie podle intenzity slunečního osvětlení. [3]

2.2 Typická architektura uzlu v síti

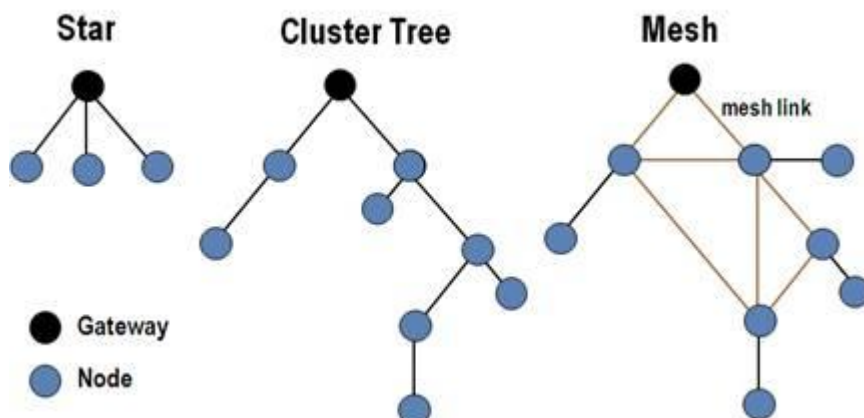


Obrázek 1 - architektura koncového uzlu sítě

Uzel v bezdrátové senzorové síti se typicky skládá z několika komponentů. Je to baterie, anténa, mikrořadič, senzory a rozhraní pro komunikaci se senzory. Vždy je důležité systém dobře nadimenzovat a vybrat si, co je důležité. Zvyšování rychlosti datového toku a častější komunikace s ostatními uzly zvětšuje spotřebu energie a tím zkracuje dobu, po jakou systém dokáže fungovat na baterii. V praxi jsou běžným požadavkem tři roky provozu na baterii. Ovšem není důležité pouze, aby baterie měla co nejdélší životnost. V některých aplikacích je potřeba vzít v potaz i velikost a hmotnost baterie. Velice často je v bezdrátových senzorových sítích využíván standard ZigBee. Má velmi malou spotřebu energie. Vzhledem k tomu, že správa a úspora napětí integrovaných systémů a kvalita baterií se stále vyvíjejí, Wi-Fi je také jedna z možností i přes větší odběr energie.

Z důvodu prodloužení životnosti systému na baterii se většinou uzel periodicky probouzí, odesílá data a poté opět uspává do nízko napěťového režimu. Kvůli maximalizaci úspory energie je také nutné vybrat pro uzel správný procesor a jeho architekturu. Je důležité, aby probouzení a uspávání procesoru bylo co nejučinnější a následná spotřeba ve spánku co nejmenší. V závislosti na dané aplikaci je potřeba vyvážit spotřebu energie vybraného procesoru a jeho výpočetní sílu.

2.3 Topologie sítí



Obrázek 2 - topologie bezdrátových senzorových sítí

Uzly bezdrátové senzorové sítě jsou klasicky uspořádány do jedné ze tří síťových struktur. V klasické hvězdicové topologii je každý uzel spojen přímo s výstupní branou systému. V topologii cluster tree jsou uzly vždy připojeny k uzlu výše ve stromové struktuře. Data jsou poté předávána od uzlů na nejnižší úrovni, přes uzly vyšší úrovně a to až k výstupní bráně systému. Třetí strukturou je tzv. Mesh, který obsahuje uzly připojené k více uzlům různých úrovní a předává data skrze nejspolehlivější cestu k výstupní bráně. Těmto uzlům připojeným k několika dalším se často říká směrovače. [4]

2.4 Open-source a proprietární platformy

Open-source software, je počítačový systém, který má jak dostupný a otevřený kód pro všechny uživatele, ale je také legálně dostupný pro využití. Konkrétní podmínky vždy záleží na tom, pod jakou licenci software spadá. V posledních letech se open-source systémy stávají zdrojem inovací a přebírají tak štafetu od proprietárních systémů. Nejdiskutovanějšími rozdíly mezi těmito dvěma přístupy jsou bezpečnost, vyšší kvalita, nižší náklady a žádné proprietární uzamčení otevřeného softwaru. Toto jsou jeho často zmiňované devíza. Vyšší bezpečnost a nižší chybovost otevřeného kódu vychází z toho, že každý uživatel může procházet, nebo upravovat kód. Je tedy pravděpodobnější, že chybu v kódu neodhalí jedinec s úmysly ji zneužít, ale někdo z komunity a chybu napraví. Studie zdrojového kódu jádra operačního systému Linux

zjistila 0,17 chyb na jeden tisíc řádek kódu, zatímco u proprietárního softwaru výsledky dosahují zhruba 25 chyb na stejný počet řádků.

Vývojář se zde tedy musí rozhodnout, jaké vlastnosti od platformy očekává a na základě těchto požadavků nějakou zvolit. Často zmiňovanou nevýhodou otevřeného softwaru je nižší, nebo nedostatečná úroveň technické a obecné podpory, právě díky tomu, že je systém zdarma. V této situaci je tedy nutné obrátit se na komunitu vyvíjející konkrétní otevřený software. [5]

2.5 Srovnání používaných platforem

Existuje velké množství systémů, na kterých je možné bezdrátovou senzorovou síť založit. Liší se v mnoha faktorech, jako jsou výpočetní výkon, dosah, nebo třeba přenosová rychlost bezdrátového spojení. To je samozřejmě určeno také tím, jaký standard komunikace platforma používá. Nejčastějšími jsou standardy 802.15.4, Bluetooth, nebo ZigBee. Ještě před začátkem návrhu si tedy vývojáři musí určit, jak bude síť výkonově náročná, jak dlouho by měla vydržet na baterii a jaké budou potřeba přenosové rychlosti. Vždy jde hlavně o to co nejlépe vyvážit tyto parametry a vybrat systém, který tomu odpovídá. Čím větší je požadován výkon, tím větší bude většinou i spotřeba energie baterií a stejně tak to bude s přenosovými rychlostmi.

Název	8/16 bitů	Výrobní proces [nm]	Paměť [KB]	Vdd [V]	Clk [Mhz]	Spotřeba [pJ/ins]
Atmel ATmega 128L	8	350	4	3	7.3	3200
TI MSP430	16	N/A	10	3	8	750
S. Hanson, JSSC '08	8	130	0.3125	0.35	0.354	3.5
B. Zhai, VLSI'06	8	130	0.25	0.36	0.833	2.6
Phoenix, VLSI'08	8	180	0.41	0.5	0.106	2.8
J. Kwong,	16	65	128	0.5	0.434	27.3

ISSCC'08						
Charm, VLSI'06	N/A	130	68	1, 0.3	9	96
SNAP, ASPLOS'04	16	180	8	1.8, 0.6	200, 23	218, 24
SmartDust, ISSCC'04	8	250	3.125	1	0.5	12

Tabulka 1 - přehled používaných systémů

[6]

Při výběru platformy je také důležité rozhodnout se, zda použít osmi, nebo šestnácti bitový systém. Z tabulky vyplývá, že ve většině případů má osmi bitová architektura nižší spotřebu energie, ale také menší výpočetní výkon, což může být pro komplexnější úlohy na škodu. Šestnácti bitová architektura tedy zvýší spotřebu, ale také nebude muset pracovat tak dlouho, vzhledem k urychlení výpočtů. Většinou se tedy v „hloupějších“ uzlech senzorových sítí používá spíše osmi bitové systémy a v uzlech, které naměřená data ze senzorů ještě dále zpracovávají naopak šestnácti bitové řadiče.

2.5.1 Intel XScale

XScale je architektura mikrořadičů původně navržená firmou Intel. Tato architektura využívá instrukční sadu architektury ARM ve verzi 5. Tuto architekturu obsahuje několik rodin procesorů s tím, že některé byly odkoupeny firmou Marvell Technology Group a dále vyvíjeny. Některé novější procesory od této společnosti se stejnou značkou již měly jinou architekturu, například ARM Cortex. Další byly navrženy jako SoC (System on a Chip). Jedná se o integrovaný obvod, který obsahuje veškeré komponenty počítače, nebo jiného zařízení na jediném čipu. Všechny procesory s touto architekturou jsou 32-bitové a vyrobené technologií s tranzistory širokými 130 nm, nebo 190 nm. Mají instrukční i datovou cache velkou 32 KB.

Vzhledem k těmto parametrům jsou řadiče s touto architekturou trochu méně vhodné pro použití v bezdrátových senzorových sítích. Uzly jsou ve většině případů napájené z baterie, což se neshoduje s vyšší spotřebou, kterou mají tyto procesory. Také ale poskytují řádově vyšší výpočetní výkon. Hodinový takt se pohybuje ve stovkách MHz. V některých aplikacích s vyššími výpočetními nároky se toho dá využít, ale většinou je použita architektura s nižším výkonem, zato ale s o hodně nižší spotřebou.

Kromě bezdrátových senzorových sítí jsou procesory z této rodiny k nalezení v běžně používaných zařízeních, jako jsou například starší telefony značky BlackBerry, kapesní počítače DELL, čtečka knih Kindle, nebo ve stolním počítači Lyonix s operačním systémem založeném na Linuxu. Jsou také používána v multimediálních zařízeních, jako jsou video přehrávače, nebo domácí mediální centra. [7], [8]

2.5.2 Crossbow

Vývojové desky crossbow s radičem a radiem, nejčastěji označované (crossbow motes) umožňují několika sensorům rozmístěným napříč relativně velkou oblastí bezdrátově přenášet jejich data k centrálnímu prvku, většinou připojenému k počítači. V uzlech běží operační systém TinyOS, který zajišťuje napájení, bezdrátový přenos a síťování průhledně pro uživatele. Síť podporuje tzv. multihopping, takže když je uzel mimo dosah centrálního prvku, může data předat přes ostatní články sítě. Uzly po spuštění si samy utvoří síťovou strukturu tak, aby byla co nejefektivnější.

Desky prodávané firmou Crossbow byly původně vyvinuté na kalifornské univerzitě v Berkley. Mozkem uzlů je mikroprocesor ATmega 128L a jsou programovatelné jazykem nesC, který je odnoží jazyka C s přidáním konceptu rozdělení celých aplikací na dílčí komponenty, které jsou následně spojovány dohromady podobně jako například v jazyce Verilog. Tyto komponenty jsou tvořeny událostmi a rozhraními, jako jsou například časovače, nebo rozhraní s diodou LED. [9]

2.5.3 TI MSP430 a rádio CC2500RF

MSP430



Obrázek 3 - vývojový kit ez430-RF2500

Vývojová platforma eZ430-RF2500 je založená na komunikačním rozhraní USB. Poskytuje veškerý potřebný hardware a software pro práci a vývoj s mikroprocesorem MSP430F2274 a bezdrátovým vysílačem CC2500 pracujícím v pásmu 2.4 GHz, které jsou součástí tohoto kitu. K programování a ladění ez430-RF2500 je možné použít IAR Embedded Workbench IDE nebo Code Composer Studio, což je IDE založené na prostředí eclipse. Oba tyto vývojové prostředky jsou ke stažení pro nekomerční účely zdarma, s tím, že je omezena maximální velikost nahraného kódu. Součástí vývojové desky je také odpojitelná část pro nahrávání a ladění kódu. Ten je možné krokovat, nebo přidávat hardwarové breakpointy bez zvýšení nároků na samotný hardware. Tato část zároveň přidává virtuální COM port po připojení k PC, takže je možné s procesorem komunikovat s pomocí rozhraní UART. Součástí balení je i modul pro dvě AAA baterie, do kterého se zapojí pouze část s procesorem a vysílačem pro použití ve finální aplikaci. Na této části je vyvedeno celkem 21 dostupných pinů pro další použití, dvě LED diody kvůli zpětné vazbě s uživatelem a jedno tlačítko, připojené na pin procesoru s možností nastavení přerušování. Procesor MSP430F2274 může běžet na frekvenci až 16 MHz a to s pěti nízko napěťovými módy s minimálním odběrem 700 nA v klidovém režimu. Obsahuje několik čítačů a USCI modul podporující komunikační protokoly UART, LIN, SPI, I2C a IrDA. [10]

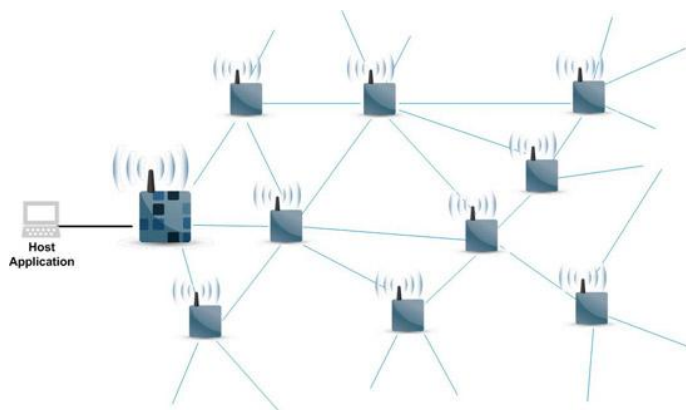
CC2500RF

CC2500 je nízko nákladový vysílač a přijímač pracující v pásmu 2,4 GHz. Je navrhnutý pro nízko napěťové aplikace. Je určen pro vysílání v pásmu ISM (Industrial, scientific and medical) 2400 – 2483.5 MHz a v pásmu SRD (Short range device) 900 MHz. Pracovní napětí je od 1.8 V do 3.3 V s tím, že musí být samozřejmě stejné na všech napájecích pinech. Provozní teplota je od -40 °C do 85 °C. RF vysílač je integrován společně s nastavitelným modemem. Ten podporuje různé druhy modulačních formátů a má nastavitelnou datovou rychlost a to až 500 kBaud. Vysílač CC2500 poskytuje velkou hardwarovou podporu pro funkce, jako jsou packet handling, data buffering, burst transmissions, clear channel assessment, link quality indication a wake-on-radio. Hlavní pracovní parametry a 64 bajtové vysílací a přijímací fronty mohou být řízeny, skrze rozhraní SPI. V typickém zapojení je CC2500 použit společně s mikrořadičem a několika pasivními prvky. [11]

Baud

Jedná se o jednotku modulační rychlosti. Jinak se také označuje, jako symbolová, nebo znaková rychlost. Tato veličina udává počet změn stavu přenosového média za jednu sekundu. Pro kódy, jako je RS232 je 1 baud roven 1 bitu. U některých druhů komunikace ale tento poměr neplatí, protože jedna změna stavu média může znamenat větší množství dat, než jeden bit.

2.5.4 SmartMesh



Obrázek 4 - topologie sítě SmartMesh

SmartMesh je kompletní řešení pro tvorbu bezdrátových sítí od firmy Linear technology. Celý systém je autonomní a síťová struktura se vybuduje sama. Klasicky se skládá z jednoho uzlu, tzv. manažera sítě a většího množství uzlů, které mohou plnit libovolný účel, například sbírat data, nebo pouze přeposílat dalším. Manažer za běhu sleduje výkon sítě a případně vyvolává automatické úpravy, či případné opravy při chybách, nebo výpadech sítě. Klasicky je také spojen s počítačem, kam předává informace ze sítě. Každý článek sítě funguje za běhu jako směrovač, takže kromě svého hlavního účelu i přeposílá a řídí síťovou komunikaci. Díky tomu jsou v síti redundantní cesty, tedy je odolnější a je i možné za běhu sítě přidávat další články. Uzly se chlubí velice nízkou spotřebou.

Síť je založena na protokolovém zásobníku TSMP (Time synchronized mesh protocol). Tento zásobník obsahuje několik vrstev ISO modelu, takže podporuje síťové zabezpečení, kódování zpráv, kontrolu chybovosti zpráv a ověřování připojených zařízení. TSMP obsahuje také TSCH (Time slotted channel hopping). Ten rozděluje čas na časové intervaly a ty následně mapuje na jednotlivé kanály. Každý uzel tedy přesně ví, kdy má vysílat, přijímat, nebo spát. Díky tomu nevznikají v síti kolize a přenos datových paketů je synchronizovaný. Tímto postupem se komunikace lehce zpomalí, ale zmenší se spotřeba energie uzlů díky tomu, že není třeba před vysíláním žádným způsobem kontrolovat médium a před při příjmu je uzel probuzen pouze na nezbytně nutnou dobu. Bezdrátová komunikace probíhá v pásmu 2,4 GHz (2400 – 2483.5 MHz).

[12]

2.5.5 BTnode



Obrázek 5 - vývojový kit BTnode

BTnode je autonomní platforma pro výpočty a bezdrátovou komunikaci pomocí standardu bluetooth. Vývojový kit obsahuje mikrořadič, bluetooth rádio, ISP, UART, SPI, I2C a další běžné periferie. Používá se primárně pro bezdrátové senzorové sítě, ale také pro testování sítí. Byla vyvinuta univerzitě ETH v Curychu laboratoří počítačového inženýrství a počítačových sítí. Kromě bluetooth radia obsahuje ještě další nízkonapěťové rádio. Ty mohou pracovat souběžně, nebo být nezávisle na sobě odpojována od napájení, což dramaticky snižuje celkovou spotřebu zařízení v klidovém stavu. Srdcem vývojového kitu je mikrořadič Atmel ATmega 128L s hodinovým taktem 8 MHz. K dispozici je vývojářům 244 KB paměti RAM a 128 KB paměť flash. Zařízení je programováno jazykem C a je kompatibilní s operačním systémem TinyOS. [13]

2.6 Druhy programování systémů

V této části rozeberu možné přístupy k programování a návrhu bezdrátových senzorových sítí. Zmíním výhody a nevýhody jednotlivých přístupů.

2.6.1 Jazyk C a knihovny pro konkrétní hardware

Tento přístup je určen především pro menší a jednodušší aplikace a sítě. Nesporná výhoda je ušetření paměti zařízení. Tu bude zabírat pouze kód aplikace a případně knihovny. Nevýhodou je, že vývojáři přichází o možnosti operačního systému, které by se u větších aplikací mohli hodit. Je nutné vše důkladně otestovat, zatímco při

použití nějakého, ať už open-source, nebo proprietárního systému, bude kód otestován komunitou, nebo přímo výrobce, který za ním stojí.

2.6.2 Událostmi řízené programování

Událostmi řízené programování, neboli Event-driven programming je model programování ve kterém je běh programu namísto posloupnosti akcí řízen událostmi bez předem určeného pořadí. Na počítači jsou to například uživatelské akce, jako stisknutí klávesy, nebo klik myši. Může to ale také být připravený výstup ze senzoru, nebo přijatá bezdrátová zpráva jako v některých modelech bezdrátových senzorových sítí.

Program napsaný s touto koncepcí se většinou skládá z hlavní smyčky, kde se čeká na tyto události. Ty spustí callback funkci, která je dále zpracuje. Ve vestavěných systémech může být stejné funkcionality dosaženo pomocí hardwarového přerušeni, namísto neustálého čekání v hlavní smyčce. Tím je možné dramaticky snížit spotřebu zařízení. [14]

2.6.3 S operačním systémem

Na rozdíl od klasického programování nabízí ve vestavěných systémech operační systém řadu výhod za cenu využití paměti. U bezdrátových senzorových sítí se v podstatě jedná o knihovnu provázanou s aplikací. S operačním systémem se také běžně pojí programovací jazyk. Klasicky to je jazyk C, nebo nějaký jeho dialekt. Například pro TinyOS je to nesC. Nejběžněji využívaným operačním systémem ve WSN je TinyOS, dále potom Contiki, SOS, nebo RETOS.

TinyOS

TinyOS je open-source operační systém, podléhající licenci BSD, navržený hlavně pro nízkonapěťová bezdrátová zařízení, která jsou používána v bezdrátových senzorových sítích, PAN sítích, nebo v inteligentních budovách. Zprostředkovává vývojářům zjednodušení v práci s nižším hardwarem, přičemž je komunitou stále vyvíjen a dobře testován. Vzhledem k jeho malé velikosti je obzvláště vhodný pro zařízení s mikrořadiči s omezenou pamětí. Výhodou operačního systému je, že již existuje mnoho kódu, ze kterého mohou vývojáři vycházet, na mnoha různých rodinách řadičů. Tyto knihovny jsou uživateli používány a mnohokrát testovány, takže je

zaručena jejich funkčnost. Podporuje nejen single-hop komunikaci, ale také multi-hop. Pro některá bezdrátová rádia obsahuje i procesy pro zajištění bezpečnosti a spolehlivosti komunikace. Velice silná je podpora bezdrátového rádia standardu ZigBee CC2420. Slabinou TinyOS je jeho model programování. Pro začátečníky s tímto systémem může být obtížnější si na něj zvyknout a tím pádem téměř nemožné vytvořit komplexnější aplikace. Vzhledem k tomu, že je určen pro čipy s velice malou pamětí RAM je většina API funkcí neblokující. Například při použití funkce send se program vrátí téměř okamžitě a až po delší době je systémem vyvolána callback funkce sendDone. Je tedy nutné se tomuto modelu přizpůsobit a veškeré rutiny napsané vývojáři musí být taktéž neblokující, aby se nepromeškaly tyto události. V opačném případě by mohlo dojít k vypadávání paketů, nebo podobným problémům. Aktuální verze operačního systému obsahuje knihovnu pro práci s vlákny, což může v těchto problémových případech pomoci. Také to vede k možnosti používat jazyk C, namísto pro TinyOS klasického nesC. [15]

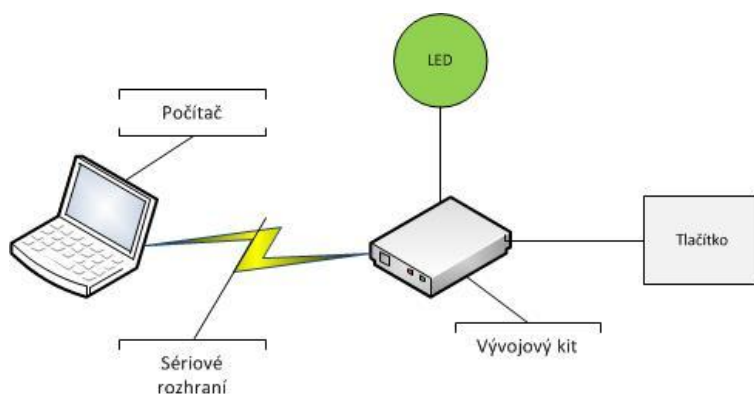
Contiki OS

Jedná se o open-source operační systém, který je k volnému využití jak v komerční, tak i nekomerční sféře. Je určen především pro nízko napěťová zařízení s možností připojení na internet. Podporuje internetové standardy IPv4 i IPv6, ale i standardy pro bezdrátovou komunikaci, jako jsou 6lowpan, RPL, nebo CoAP. Na rozdíl od TinyOS se programuje standardním jazykem C a podporuje velké množství vývojových platforem od různých výrobců. Například TI MSP430, nebo Atmel ATmega 128.

3 Vstupní zařízení a periferie

V této části je postupně rozebráno, s jakými periferiemi a zařízeními se pracovalo při návrhu této konkrétní aplikace. Jak se postupovalo od základních dílčích celků až po větší a složitější systémy. Je zde také řečeno, jak je navržena knihovna pro komunikaci se senzorem SHT11 pomocí jeho komunikačního protokolu.

3.1 Základní aplikace s tlačítkem



Obrázek 6 - topologie základní testovací aplikace

Pro implementaci zdrojových kódů zařízení jsou pro vybranou platformu TI ez430-RF2500 jsou k dispozici dvě různá vývojová prostředí. Code composer studio (CCS) a IAR embedded workbench. Vzhledem k tomu, že Code composer studio je vývojové prostředí založené na vývojovém prostředí Eclipse, které je mi blízké. Zvolil jsem pro návrh tento software. Vývojové prostředí již obsahuje knihovny pro pohodlnou práci s mikrořadiči. V našem případě msp430f2274. Ta obsahuje makra pro přístup k registrům procesoru nastavujícím jednotlivé periferie, například časovače, převodníky, nebo sériové rozhraní.

Pro otestování základních periferií a prací s nimi jsem navrhl jednoduchý senzorový systém. Namísto senzoru zde postačí tlačítko, jehož stisky snímá mikroprocesor. Na základě stisků rozsvěcí LED diodu a odesílá po sériové lince počítadlo stisků tlačítka. Nejprve je nutné nastavit použité porty, tedy jeden vstupní pro tlačítko a jeden výstupní pro diodu LED. Zde se velice hodí zmíněná knihovna k tomuto konkrétnímu procesoru. Obě tyto periferie jsou připojeny k portu 1, takže je nutné nastavit registr P1DIR. Pro tlačítko je určitě vhodné nastavit i pull-up rezistor v registru P1REN. Tlačítko bylo nejdříve snímáno neustálou kontrolou změny hodnoty a následně

obslužnou rutinou přerušení. Jakmile je detekována změna stavu spínače, tak se přepne rozsvícení/zhasnutí LED diody a inkrementuje se počítadlo stisknutí. Toto počítadlo je následně odesláno po sériovém rozhraní do připojeného počítače. Tímto jednoduchým demem jsem si vyzkoušel různá nastavení portů procesoru, nastavení přerušení a používání sériové linky UART.

3.2 Knihovna pro práci se sériovým rozhraním

Vzhledem k tomu, že ve výsledné aplikaci se počítá se odesíláním dat z centrálního prvku sítě do počítače pomocí sériového rozhraní, je určitě vhodné si pro tento účel napsat pohodlnou knihovnu.

Inicializace

Nejprve je nutné UART inicializovat, takže jsem vytvořil příslušnou funkci `uart_init()`. V této funkci provádíme všechny kroky nutné k rozchození komunikace. Piny určené pro sériovou komunikaci jsou pro tento procesor na portu 3. Pro vysílání je to čtvrtý pin a pro příjem pátý. Aby plnili svou sekundární funkci, tedy sériovou linku je třeba nastavit registr `P3SEL`. Dále se nastaví přenosová rychlost v registru `UCA0BR0` a počet stop bitů.

Odeslání znaku

Vzhledem k tomu, že odesílání nebude prováděno moc často, tak je funkce napsána, jako blokující. Nejprve zkontroluje, zda je odesílací zásobník připraven, pomocí stavových registrů `IFG2` a `UCATXIFG` a pokud ano, tak vloží znak, předaný jako parametr do odesílacího zásobníku `UCA0TXBUF`.

Odeslání řetězce

Této funkci je předáno pole znaků. Ona stejně, jako při odeslání jediného znaku kontroluje registry `IFG2` a `UCATXIFG`, které udávají, zda je periferie volná pro odesílání a následně nahrává znaky z pole po jednom do odesílacího zásobníku. Funkce počítá s konvencí, že na konci každého řetězce je takzvaný `NULLCHAR`, neboli proměnná s datovým typem `CHAR` a hodnotou 0. Jakmile tedy odesílací cyklus narazí na tento znak v poli, zastaví odesílání a funkce končí. Zde je nutné, aby uživatel

knihovny počítal s touto konvencí, a každý odesílaný řetězec zakončil tímto znakem. V opačném případě by došlo k výjimce, protože by cyklus sahal mimo zadané pole.

Odeslání celého čísla

Tato funkce využívá knihovní funkci `sprintf()`, která do zadaného pole znaků vloží znaky příslušného čísla, předaného parametrem, jako text. Například tedy pro číslo 12 vloží na začátek pole znak '1' a vedle něj znak '2'. Poté text odešle s pomocí funkce pro odeslání řetězce.

Odeslání reálného čísla

Vzhledem k tomu, že měřené hodnoty ze senzoru teploty a vlhkosti SHT11 nemusí být celá čísla a stejně tak vypočtený ukazatel kvality je necelé číslo mezi 0 a 1, je nutné, aby knihovna obsahovala i funkci pro odeslání datového typu float.

V této funkci je opět použito pomocné pole znaků, do kterého je postupně plníme. Nejprve se do tohoto pole naplní celočíselná část zadaného čísla. To se jednoduše provede přetypováním čísla na datový typ LONG. Zde přichází na řadu opět funkce `sprintf()`. Vzhledem k tomu, že tato funkce dodržuje konvenci vkládání nulového znaku na konec řetězce, tak nalezneme pozici tohoto znaku v poli a nahradíme ho tečkou. Zde celý proces opakujeme ještě jednou s tím rozdílem, že od zadaného čísla odečteme jeho celočíselnou část a vynásobíme ho požadovanou přesností. Počtem desetinných míst. Toto číslo přidáme do pole funkcí `sprintf()` za vloženou tečku tak, že funkci namísto ukazatele na první prvek v poli přidáme ukazatel inkrementovaný o původní pozici nulového znaku zvětšenou o jedna. Nyní se řetězec odešle dříve napsanou funkcí pro odeslání řetězce.

2	3	.	6	2	1
---	---	---	---	---	---

Tabulka 2 - příklad výsledného pole pro číslo 23,621

3.3 Senzor SHT11

SHT11 je senzor teploty a vlhkosti. Je to jedno z rodiny čidel vyráběných společností Sensiron. Obsahuje, jak snímací prvky, tak i část pro zpracování signálu a komunikaci na velmi malé ploše. Vyčtené hodnoty jsou spolehlivé a dlouhodobě stabilní. Systém má velice rychlou odezvu a je odolný proti vnějšímu rušení. Kalibrační koeficienty čidla změřené výrobcem jsou uloženy v paměti zařízení. Následně jsou interně použity při vyčítání hodnot ze senzorů. Výstup senzoru je plně digitální. Pro měření teploty používá band-gap senzor a pro vlhkost kapacitní prvek. Pracovní rozsah teplotního čidla je od $-40\text{ }^{\circ}\text{C}$ do $124\text{ }^{\circ}\text{C}$ s přesností měření $0.4\text{ }^{\circ}\text{C}$. Měření relativní vlhkosti vzduchu pracuje na rozsahu od 0% do 100% s přesností 0.3% . Oba tyto senzory jsou připojeny na čtrnácti bitový AD převodník a sériové rozhraní pro komunikaci. To se skládá z dvou vodičů. Bohužel ho není možné adresovat pomocí protokolu rozhraní I2C i přes podobnost připojení, ale je možné čidlo na toto rozhraní připojit bez toho, aby narušovalo komunikaci ostatních zařízení. Procesor poté může přepínat mezi komunikačním protokolem I2C a vlastním protokolem SHT11. Pracovní napětí senzoru je 3.3V . Zabírá plochu zhruba $8 \times 5 \times 2.5\text{ mm}$ a váží 100 mg . Je osazen na epoxidové desce v pouzdře SIL 4 pin. V případě potřeby je číslo vybaveno vlastním systémem vyhřívání a systémem pro kontrolu nízkého napájecího napětí. Vyhřívání je možné programově upravovat. Teplotu a relativní vlhkost vzduchu je možné vyčítat s rozlišením $8/12/14$ bitů a měření poté trvá $11/55/210\text{ ms}$. [16], [17]

Band-gap senzory

Band-gap je velice často užívaná technologie měření teploty. Její výhodami jsou hlavně velice nízká cena a snadná integrace přímo do křemíku integrovaných obvodů. Principem této technologie měření teploty je, že napětí v propustném směru na diodě, která může být přechod báze – emitor na bipolárním tranzistoru, je závislé na teplotě. Pomocí rovnice je tedy teplotu možné vypočítat. Proces se zjednodušuje tím, že se měří dvě napětí, na dvou přechodech báze – emitor při stejné teplotě, ale rozdílných prouděch. Tím je možné výpočet ještě zjednodušit a eliminovat některé parametry. [18]

Kapacitní senzory vlhkosti

Měření vlhkosti vzduchu pomocí kapacitoru je založeno na principu kondenzátoru s izolantem z polymeru, který má hydroskopické vlastnosti. Je postaven

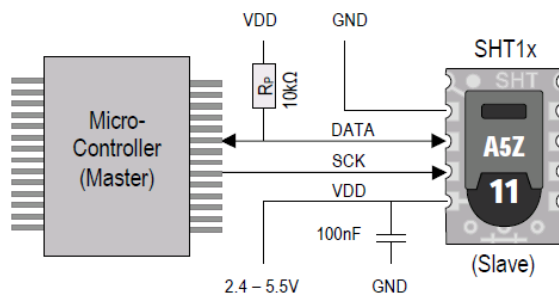
tak, že umožňuje vodním páram difundovat do polymeru. Tím, jak dielektrikum nasakuje vodu, mění své izolační vlastnosti a mění se i kapacita kondenzátoru. Jejím měřením je poté možné měřit vlhkost vzduchu.

Relativní vlhkost vzduchu

Udává poměr aktuální vlhkosti vzduchu k absolutní vlhkosti, která by při dané teplotě byla v plně nasyceném vzduchu. Vyjadřuje stupeň nasycení vzduchu vodní parou. Obvyklá pokojová hodnota relativní vlhkosti vzduchu se pohybuje okolo 40%.

ZDROJ Český hydrometeorologický ústav

3.3.1 Komunikace se senzorem



Obrázek 7 - schéma připojení senzoru SHT11

V této části práce jsou postupně vysvětleny všechny nutné základy pro implementaci komunikačního protokolu se senzorem SHT11. Po jejich objasnění je dále popsána implementace tohoto protokolu v testovací aplikaci.

Napájecí piny (Vdd, Gnd)

Senzor SHT11 je možné napájet napětím v rozmezí 2.4 V – 5.5 V s tím, že doporučené napájecí napětí od výrobce je 3.3 V. Vzhledem k tomu, že procesor MSP430F2274 také pracuje na tomto napětí, rozhodli jsme se čidlo napájet přímo z vyvedených napájecích pinů na vývojovém kitu. K pinu Vdd musí být připojen blokovací kondenzátor.

Pin SCK (Seriál clock input)

Pin SCK je používán k synchronizaci komunikace mezi mikroprocesorem a čidlem SHT11. Rozhraní uvnitř senzoru se skládá z plně statické logiky, takže čidlo

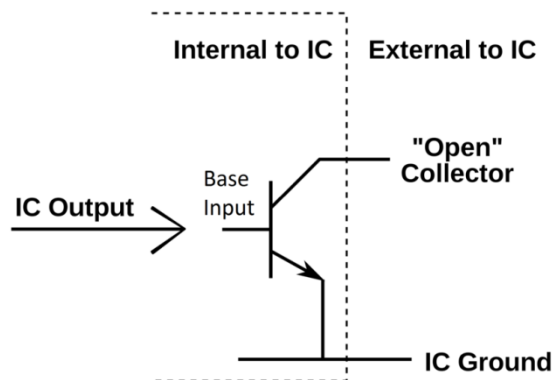
nemá žádnou minimální frekvenci SCK. Je tedy možné posílání, nebo příjem dat z čidla klidně přerušit a pokračovat až po provedení jiných časově kritických akcí.

Pin DATA (serial data)

Pin DATA je tří stavový pin používaný k přenosu dat jak do senzoru, tak i z něj. Při posílání dat do senzoru je hodnota na vodiči je platná při vzestupné hraně na vodiči SCK a musí zůstat stabilní, dokud je SCK na úrovni log. 1. Po spádové hraně je možné data změnit. Pro hodinové tiky jsem zvolil délku 5 us. Při vyčítání ze senzoru jsou data na vodiči platná od spádové hrany hodin a to až do další spádové hrany. Poté již senzor posílá další bit.

Aby se zabránilo ovlivnění vyčítaných hodnot procesorem, je nutné, aby hodnotu na tomto vodiči nastavoval pouze do log. 0. Pro vyjádření log. 1 je využíván pull-up rezistor s hodnotou 10 k Ω . Pin procesoru je tedy nutné nastavit do open drain režimu. Výrobce doporučuje používat externí pull-up rezistor, ale je možné použít i vnitřní odpor v procesoru, pokud je dostupný.

Pin zapojený jako open drain (open collector)



Obrázek 8 - schéma zapojení open drain

Jedná se o běžně využívaný druh výstupu u integrovaných obvodů. Místo, aby byl na výstup přiveden přímo signál se specifickou hodnotou, je signál přiveden na bázi vnitřního NPN tranzistoru. Kolektor tohoto tranzistoru je připojen na vnitřní zemnicí vodič. Pokud je na výstupu integrovaného obvodu použit MOSFET, tak se tomuto druhu zapojení říká také open drain a plní totožnou funkci. Signál přivedený do tranzistoru tedy řídí výstupní hodnotu pinu. Jeví se tedy buďto jako otevřený obvod,

nebo jako připojení k zemi. Typicky je s tímto nastavením pinu spojen externí pull-up rezistor. Tento princip se využívá v mnoha aplikacích.

Komunikace se senzorem SHT11

Po připojení senzoru na zvolenou hodnotu napětí je nutné držet klidový stav komunikace po 11 ms. Po této době přejde senzor do režimu spánku a očekává příkazy. Není možné cokoli posílat dříve.

Zahajovací sekvence

Před vysláním příkazu do senzoru je nutné nejdříve poslat tzv. zahajovací sekvenci. Ta se skládá ze dvou pulzů na vodiči SCK. Při prvním procesor stáhne napětí na datovém vodiči na nulu a ve druhém vodič opět uvolní. Díky pull-up rezistoru se napětí na vodiči vrátí na log. 1. Mezi oběma pulzy je potřeba vyčkat, aby se dodržela maximální frekvence hodinového vodiče. Použil jsem čekací smyčku po 5 us.

Příkazy senzoru

Jakmile je dokončena sekvence pro zahájení spojení, je možné senzoru poslat jeden z několika příkazů. Jedná se o pěti bitové číslo, které je předcházené ještě třemi bity adresy. V případě tohoto senzoru jsou to však pouze tři nuly. Těchto osm bitů je tedy jeden po druhém odesláno do čidla. Každý bit je na datovém vodiči vystaven společně s pulzem na SCK, které trvá 5 us. Po odeslání všech osmi bitů a po spádové hraně osmého hodinového tiku vynutí čidlo na datovém vodiči nulu, která trvá až do spádové hrany dalšího hodinového pulzu. Touto logickou nulou potvrzuje SHT11 přijetí příkazu. Jedná se o takzvaný ACK (acknowledgement) bit.

Příkaz	Kód
Rezervováno	0000x
Měření teploty	00011
Měření relativní vlhkosti	00101
Čtení STATUS registru	00111
Zápis do STATUS registru	00110
Rezervováno	0101x – 1110x
Soft reset	11110

Tabulka 3 - seznam příkazů senzoru SHT11

Soft reset čidla vyčistí status registr do výchozích hodnot udaných výrobcem a resetuje komunikační rozhraní. Po provedení tohoto příkazu je stejně jako po připojení napájení nutné vyčkat 11 ms, než přejde senzor do spánkového režimu. Poté je možné posílat další příkazy.

Měření teploty a relativní vlhkosti vzduchu

K zahájení měření teploty, nebo vlhkosti vzduchu je nutné do senzoru nejdříve odeslat příslušný příkaz podle tabulky příkazů výše. SHT11 podporuje tři režimy měření teploty a relativní vlhkosti vzduchu. Ty se liší různým rozlišením. Na výběr je měření s rozlišením 8/12/14 bitů. Doba měření se pro jednotlivé režimy liší. Podle informací od výrobce by měření mělo trvat 20/80/320 ms pro jednotlivé režimy. Po proběhnutí měření senzor nastaví logickou nulu na datový vodič. Jakmile se tak stane, tak je možné začít pomocí řízení hodinového signálu vyčítat jednotlivé bity naměřené teploty, nebo relativní vlhkosti. Od výrobce je jako výchozí nastaveno rozlišení 12 bitů pro měření relativní vlhkosti vzduchu a 14 bitů pro měření teploty. Pokud je zvolen režim 12, nebo 14 bitů, senzor odešle dva celé bajty. Data v těchto bajtech jsou zarovnána doprava a nejvíce významný bit je vysílán jako první. To znamená, že 2, nebo 4 první bity nemají žádný význam pro rozlišení 14 respektive 12 bitů. Je-li zvolen režim rozlišení měření pouze na 8 bitů, tak se první bajt výsledku nepoužívá. Po jednom, nebo dvou bajtech vyčtených hodnot ještě může následovat jeden bajt CRC kontrolního součtu. Je nutné, aby procesor po každém přijatém bajtu na datový vodič vystavil logickou nulu, jako potvrzení (ACK) přijetí. Po přijetí posledního ACK bitu od procesoru se senzor sám přepne do spánkového režimu, dokud mu není předán další příkaz. Výrobce doporučuje, aby byl senzor využíván maximálně 10% času. Toto vychází zhruba na jedno měření s rozlišením 12 bitů za vteřinu.

Sekvence pro reset spojení

Pokud je z nějakého důvodu přerušena, nebo ztracena komunikace se senzorem, tak je po opětovném navázání nutné použít resetující sekvenci, která sériové rozhraní čidla vyčistí. Tato sekvence nemá za následek nic jiného. Není tedy potřeba znovu nastavovat status registr. Skládá se z devíti po sobě jdoucích hodinových pulzů. Vodič DATA se po celou dobu sekvence nechá volný a tedy v úrovni logické 1. Před zasláním příkazu po resetu spojení je opět nutné poslat do čidla sekvenci pro zahájení spojení (Transmission start).

Status registr

Senzor SHT11 podporuje některé pokročilé funkce, jako je upozornění na nízké napětí a tedy předpokládaný konec životnosti baterie, nebo použití ohřívače, který dokáže teplotu senzoru zvýšit asi o 5 – 10 °C a to na odběru asi 8 mA. Toto a také třeba rozlišení měření teploty, nebo relativní vlhkosti vzduchu se nastavuje ve status registru. Pro zápis, respektive čtení z registru slouží příslušný příkaz čidla. Po odeslání příkazu zápisu senzor jako vždy stáhne DATA do hodnoty logické 0 a poté očekává bajt pro zápis do stavového registru. Nejvíce významný bit je odeslán opět jako první. Po přijetí celé zprávy následuje ACK. Při vyčítání aktuálního nastavení status registru ze zařízení je opět možné použít po přijetí jednoho bajtu registru 8 bitů kontrolního součtu CRC.

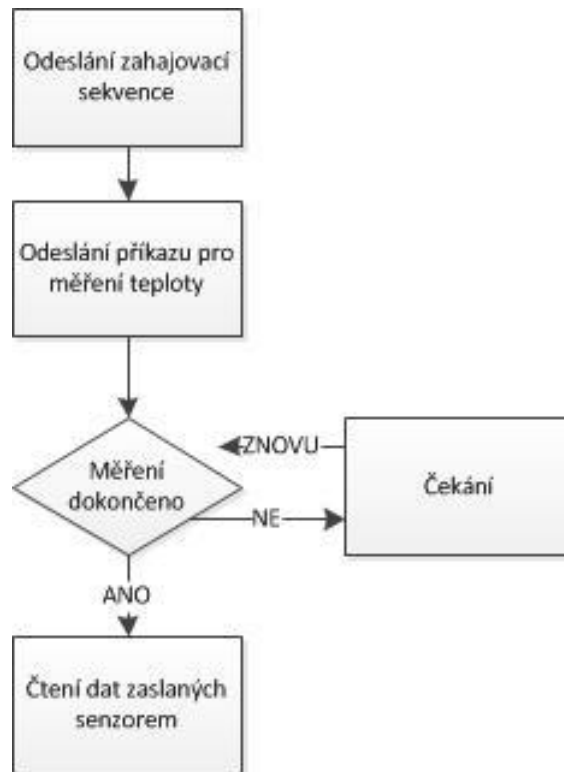
Číslo Bitu	Typ	Popis	Výchozí hodnota
7		Rezervováno	0
6	R	Detekce nízkého napájecího napětí 0 - VDD > 2,47 V 1 – VDD < 2,47 V	X – aktualizováno po každém měření
5		Rezervováno	0
4		Rezervováno	0
3		Rezervováno	0
2	R/W	Ohřívač	0 – vypnuto
1	R/W	OTP	0
0	R/W	1 – 8b měření vlhkosti a 12b měření teploty 0 – 12b měření vlhkosti a 14b měření teploty	0

Tabulka 4 - popis jednotlivých bitů status registru senzoru SHT11

Jak je vidět v tabulce, tak některé položky jsou nastavitelné uživatelem, ale kontrola měření nízkého napájecího napětí je zapnuta stále a je už pouze na konkrétní aplikaci, zda tuto hodnotu bude kontrolovat.

Aplikace pro měření veličin

Pro komunikaci se senzorem SHT11 byla vytvořena knihovna obsahující potřebné funkce a definice. Například to jsou funkce pro odeslání příkazů senzoru, pro příjem bajtu, reset spojení, nebo pro odeslání sekvence zahajující komunikaci.



Tabulka 5 - blokové schéma komunikace se senzorem SHT11

Pro ilustraci komunikačního protokolu je na obrázku vidět postup pro změření teploty senzorem SHT11. Nejprve je nutné odeslat zahajovací sekvenci tak, jak je popsáno v příslušné kapitole. V tuto chvíli je čidlo připraveno pro příjem příkazu, takže je možné odeslat příslušných osm bitů. Vzhledem k tomu, že koncový uzel v naší bezdrátové senzorové síti je většinu času v režimu spánku a pouze jednou za příslušnou dobu změří teplotu, je možné ho na chvíli zablokovat čekáním. Měření trvá řádově desítky milisekund v závislosti na druhu měřené veličiny a zvoleném rozlišení. Čekáme tedy, dokud senzor neoznámí dokončení měření stažením vodiče DATA do hodnoty logické nuly. V tuto chvíli je možné začít vyčítat naměřená data pomocí funkce pro čtení bajtu. Čidlo odesílá nejdříve více významný bajt a následně méně významný. Tyto hodnoty tedy vložíme vedle sebe do proměnné s datovým typem, do kterého se vejdou. Vyčtené hodnoty registrů senzoru ještě neodpovídají skutečným hodnotám a je nutné je

přepočíst na skutečné hodnoty pomocí vztahů uvedených v dokumentaci senzoru SHT11.

4 Bezdrátová komunikace

4.1 Protokolový zásobník SimpliCI

V této části je teoreticky popsán protokolový zásobník SimpliCI, je zmíněno k čemu slouží a jaké jsou jeho obecné vlastnosti. Poté kapitola obsahuje postup, jak pomocí tohoto nástroje komunikovat mezi uzly sítě, ať už pouze jako rovný s rovným, nebo jako přístupový bod a koncové uzly. Je zde řečeno, jaké kroky je nutné před začátkem implementace aplikační vrstvy udělat a jak poté dále krok po kroku pokračoval.

4.1.1 Popis protokolového zásobníku SimpliCI

SimpliCI je síťový protokol určený hlavně pro nízkonapěťové RF sítě obsahující až sto uzlů v síti. Je vytvořen pro snadnou implementaci ve vestavěných systémech s minimálními výpočetními nároky na mikroprocesor. Je primárně určen pro rodinu procesorů společnosti Texas Instruments MSP430 a pro několik různých RF vysílačů.

V typickém případě malé nízkonapěťové sítě využívají zařízení pracující s bateriovým napájením. Aby byla prakticky využitelná, je vyžadována velká životnost baterie. Proto se využívají spíše nižší přenosové rychlosti, a co nejvíce se snižuje doba aktivní práce uzlu oproti době, po kterou je v režimu spánku a tedy výrazně šetří energii baterie. Dále se snažíme co nejvíce minimalizovat počet přímých spojení mezi jednotlivými uzly. SimpliCI má minimální nároky na výkon mikroprocesoru, což vede k nižší spotřebě uzlů a tedy delší životnosti.

I přes nízké nároky na výpočetní sílu, podporuje simpliCI síťovou topologii peer to peer, možnost, aby přístupový bod ukládal a přeposílal komunikaci a uzly, které pouze prodlužují komunikační dosah a to až přes čtyři úrovně. V budoucích verzích

výrobce plánuje přidávat více a více možností protokolu, jako například rutiny pro kódování zpráv.

Protokol je možné využít v mnoha nízko napěťových aplikacích. Typickými příklady jsou bezpečnostní alarmy a zabezpečující zařízení pro detekci kouře, prolomení skla, nebo světelné senzory. Dále se také využívá v inteligentních domácnostech, nebo snímání a zaznamenávání stavu okolního prostředí.

SimpliciTI je dostupný ke stažení jako zdrojový kód k volnému využití s volnou licencí. Je možné, aby si vývojáři protokol upravili podle svých specifických potřeb konkrétní aplikace.

4.1.2 Jak postupovat při implementaci

Protokolový zásobník SimpliciTI je složen ze tří vrstev. První je Fyzická/Linková, druhá síťová a třetí je aplikační. O obsluhu obou spodních vrstev se zásobník postará sám za uživatele. Vývojář se tedy při implementaci stará pouze o aplikační vrstvu SimpliciTI. Zde je navrhována síťová komunikace, připojování uzlů, nebo odesílání datových paketů.

Po vytvoření nového projektu vývojového prostředí Code composer studio je nutné do projektu přidat zdrojové kódy a hlavičkové soubory protokolového zásobníku. Ty jsou rozděleny do několika adresářů, podle toho na jaké vrstvě pracují. Jsou k dispozici k volnému stažení na stránkách výrobce, nebo jsou i dodávány na CD spolu s vývojovým kitem ez430-RF2500. Kromě těchto souborů je do projektů také nutné přidat příkazové soubory. Tyto soubory se automaticky spustí po startu zařízení, do kterého byly nahrány. Nastavují komunikaci rádia CC2500RF. Do projektu se přidají ve vlastnostech projektu, v záložce Build/Command Files.

Prvním krokem při implementaci aplikace využívající tento protokolový zásobník je funkce BSP_INIT. Ta inicializuje všechny periferie využívané zásobníkem specifické pro konkrétní vývojovou desku. Například tedy tlačítka, LED diody a podobně. V tuto chvíli může vývojář inicializovat další periferie, které bude potřebovat. V našem případě sériové rozhraní UART. Další akce jsou závislé na tom, jakou funkci bude konkrétní zařízení plnit.

4.2 Přístupový bod

Jedná se o centrální prvek bezdrátové senzorové sítě. Plní úlohu uzlu, který se spojuje se všemi koncovými body a přijímá od nich jejich naměřená data ze senzorů a příslušné ukazatele kvality těchto dat. Tento uzel je připojen k uživatelskému počítači pomocí sériového rozhraní, po kterém předává uživateli výsledky měření ze všech uzlů. Zároveň také z těchto výsledků počítá vážený průměr na základě ukazatelů kvality měřených dat. Z takto spočtené teploty a relativní vlhkosti vzduchu potom pomocí vzorce počítá rosný bod.

Implementace je zahájena inicializační funkcí periférií protokolového zásobníku BSP_INIT. Po této funkci a dalších uživatelských inicializacích je zavolána funkce SMPL_Init, která inicializuje bezdrátové rádio a komunikační protokol SimpliciTI. Funkci se také parametrem předá funkce, která bude obsluhovat příchozí pakety ze sítě.

Této funkci se říká callback funkce a je vyvolána protokolovým zásobníkem v přerušení, takže by měla být co nejmenší. Z tohoto důvodu ve funkci pouze pomocí parametru Link_ID zjistíme, zda se jedná o paket uzlu již zapojeného v síti, nebo od uzlu, který ještě v síti není. Hodnota parametru Link_ID je rovna nule v případě, že je příchozí paket od uzlu mimo síť. Na základě tohoto zjištění inkrementujeme jeden ze dvou semaforů, které jsou dále obsluhovány v hlavní smyčce programu mimo přerušení.

Semafor nového uzlu

Jestliže jsme v obslužné rutině přerušení inkrementovali tento semafor, znamená to, že se k centrálnímu uzlu sítě snaží připojit dosud nepřipojený koncový uzel. Zkontrolujeme tedy, zda není ještě překročen limit maximálního počtu spojení a pokud ne, zavoláme funkci SMPL_LinkListen, která s koncovým uzlem naváže spojení a přiřadí mu Link_ID.

Semafor nového paketu

V případě, že v callback funkci vyvolané z přerušení jsme zjistili, že přišel nový paket od zařízení již připojeného v síti, je třeba tento paket přečíst a obsloužit. Přijatá data získáme použitím funkce SMPL_Receive s tím, že jako parametr funkce předáme postupně všechny Link_ID, které byly přiděleny koncovým uzlům. Tím zajistíme, že obsloužíme všechny přijaté pakety. Po přečtení zprávy jsme obdrželi naměřené hodnoty

fyzikálních veličin teploty a relativní vlhkosti vzduchu, a jejich příslušené ukazatele kvality. Tyto hodnoty v tuto chvíli předáme po sériovém rozhraní vyššímu zařízení v systému, tedy uživatelskému počítači, ke kterému je centrální prvek připojen.

4.3 Koncový uzel

Tyto prvky sítě tvoří základ celého systému, bez kterého by nemohl fungovat. Ke každému koncovému bodu je připojen senzor SHT11 pro měření. Po spuštění je opět zavolána funkce `BSP_INIT` inicializující periferie potřebné protokolovým zásobníkem. Následně je zavolána funkce `SMPL_Init`, která inicializuje rádio a `SimpliciTI`. Na rozdíl od centrálního prvku sítě ji ale není předána žádná callback funkce, protože ke koncovému uzlu v síti se nebude žádný jiný uzel připojovat. Po úspěšné inicializaci je zavolána funkce `SMPL_Link`, která v nekonečné smyčce čeká na spojení s přístupovým bodem. Zároveň také vrátí přidělené `Link_ID`. Jakmile jsou provedeny tyto nutné postupy pro bezdrátovou komunikaci, tak je nastaven časovač na zvolený interval, ve kterém bude uzel měřit fyzikální veličiny.

V okamžiku, kdy časovač vyvolá přerušení, tak nastaví semafor pro měření. V hlavní smyčce programu je poté tento semafor kontrolován, a jakmile je nastaven, tak začne uzel komunikovat se senzorem SHT11 a vyžádá si od něj měření teploty a relativní vlhkosti vzduchu. V okamžiku, kdy jsou veličiny změřeny a procesor vyčetl ze senzoru data, tak je zavolána funkce z knihovny pro ukazatele kvality a je spočten hlavní ukazatel kvality. Jak naměřené hodnoty, tak příslušné ukazatele naplníme do pole znaků a pomocí funkce `SMPL_Send` je společně s jejich velikostí a přiděleným `Link_ID` odešleme do centrálního prvku sítě pro další zpracování.

Vzhledem k tomu, že na rozdíl od přístupového bodu jsou koncové uzly napájeny z baterie, tak je velice důležité, aby měli co nejnižší spotřebu. V bezdrátových senzorových sítích by měly koncové uzly mít co nejdelší životnost. Z tohoto důvodu je mimo provádění výše zmíněných rutin uzel uspán, čímž se minimalizuje spotřeba energie a dramaticky prodlužuje výdrž na baterie.

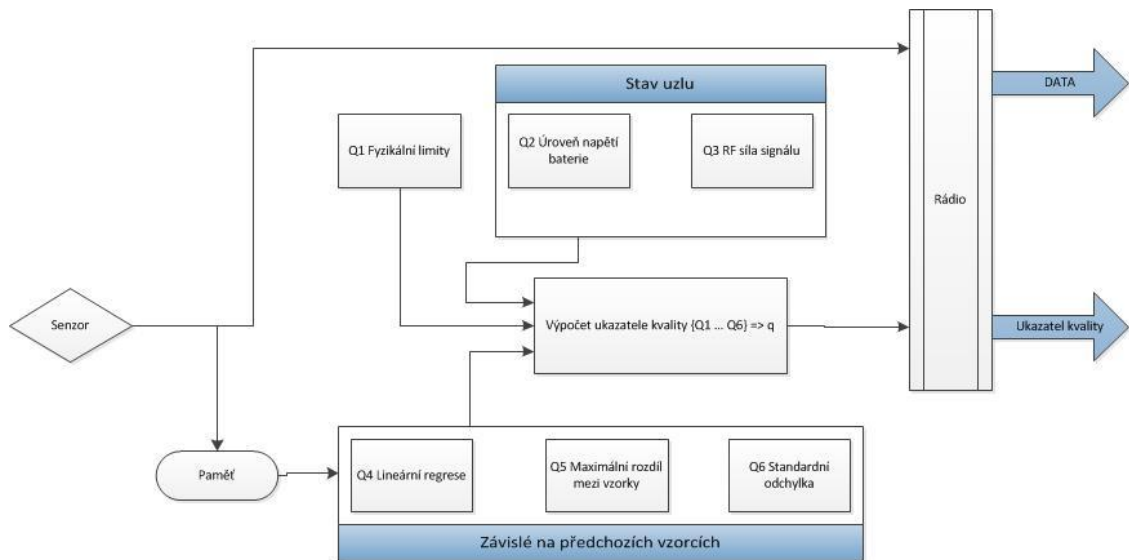
4.4 Rozšiřovač dosahu

Tento prvek bezdrátové senzorové sítě slouží k opakování přijatých paketů a tím prodloužení dosahu sítě. Vzhledem k jeho funkci musí být stále zapnutý a nepřechází do režimu spánku. Tím se snižuje jeho životnost, protože je stejně jako koncový uzel napájen z baterie. V síti založené na protokolovém zásobníku SimpliCI je možné až čtyřikrát „skočit“ přes nějaký uzel. To znamená, že paket z koncového zařízení může nejdřív projít přes čtyři tyto opakovatele, než dorazí do centrálního uzlu v síti.

5 Metody pro vyhodnocování výsledků měření snímané veličiny

V moderních bezdrátových senzorových sítích využívaných například pro sledování stavu prostředí v průmyslových zařízeních, nebo automatické nastavování klimatizace, žaluzií a dalších systémů v inteligentních budovách je nutné, aby hodnoty měřených veličin ze senzorů byly co nejpřesnější. V průběhu činnosti jsou ale koncové uzly vystaveny velkému množství vlivů, které mohou tato měření ovlivnit. Může to být třeba klesající napětí baterie, zhoršení síly signálu, nebo další vnější vlivy způsobené prostředím, kterému je uzel sítě vystaven.

Bezdrátové senzorové sítě jsou centralizovaným systémem a centrální prvek sítě potřebuje vědět, do jaké míry jsou přijatá data od dílčích koncových uzlů relevantní a správná. Z tohoto důvodu je možné kromě naměřených dat dané veličiny přenášet ještě ukazatel kvality dat, který koncové uzly spočtou po vyčtení dat z čidel. Tento ukazatel je reálné číslo mezi nulou a jedničkou a vyjadřuje do jaké míry je možné se na výsledek měření spolehnout. Tento postup zvýší spolehlivost všech naměřených dat, ale za cenu zvýšení přenosové náročnosti systému. Tento ukazatel kvality je závislý na více dílčích faktorech.

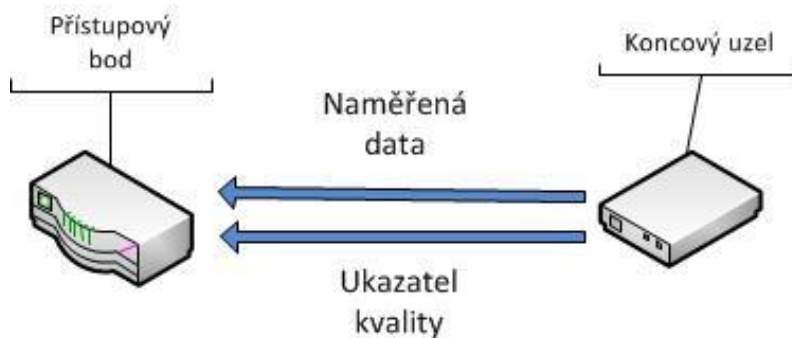


Obrázek 9 - topologie koncového uzlu sítě z pohledu ukazatelů kvality

Na obrázku je vidět, že ukazatel kvality Q je pomocí vztahu uvedeného níže spočten z ukazatelů různých druhů. Některé vyjadřují aktuální stav uzlu, tedy napětí zdroje a sílu přenášeného signálu a jiné pomocí různých matematických metod porovnávají aktuální vyčtená data ze senzoru s předchozími naměřenými hodnotami ze senzorů uložených v paměti. Pohybují se v intervalu mezi nulou a jedničkou, přičemž hodnota jedna znamená, že data jsou v pořádku a nula, že na naměřenou hodnotu se není možné spolehnout.

$$Q = Q1 * \sqrt{\text{mean}(Q2 \dots Q6) * \min(Q2 \dots Q6)}$$

Důležitou částí vztahu je dílčí ukazatel kvality $Q1$, který vyjadřuje fyzikální limity prostředí. Je nutné předem určit krajní body, mezi kterými by se naměřené hodnoty měly pohybovat. Například pokud senzor díky chybě měření určí teplotu jako 300 °C v kanceláři, tak tento ukazatel nabývá hodnoty nula a tím pádem i výsledný odesílaný ukazatel kvality měření do centrálního prvku bude nulový.



Obrázek 10 - spojení přístupového bodu a koncového uzlu

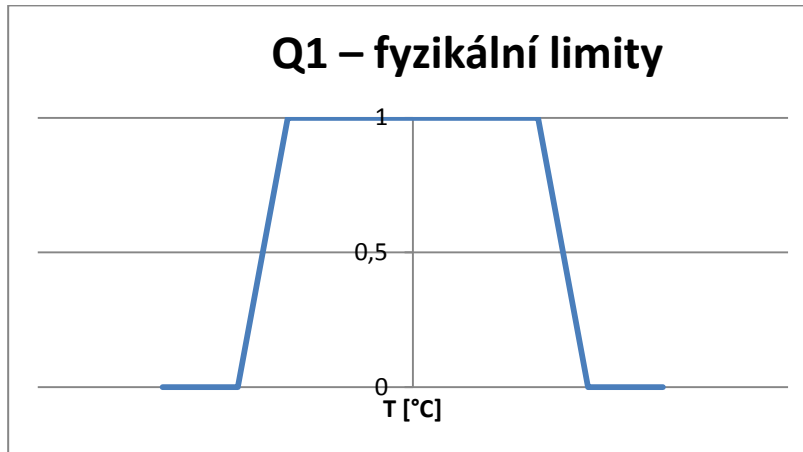
Oproti běžné bezdrátové senzorové síti tedy nejsou do centrálního uzlu sítě přenášena pouze naměřená data ze senzorů, ale i tento vypočtený ukazatel kvality. Dále už záleží na uživatelské aplikaci, jak tyto výsledky využije. Je možné například spočítat průměrnou hodnotu dané veličiny pomocí váženého průměru, kde se za váhu každé hodnoty dosadí její ukazatel kvality.

$$\bar{x} = \frac{\sum_{i=1}^n w_i * x_i}{\sum_{i=1}^n w_i}$$

V naší aplikaci jsou měřenými veličinami teplota a relativní vlhkost vzduchu. Po získání spolehlivých hodnot pomocí váženého průměru je poté spočten rosný bod podle následujícího vztahu, kde T_d je teplota rosného bodu ve stupních celsia, T je složená teplota ze všech koncových bodů a RH je takto sloučená relativní vlhkost vzduchu.

$$T_d = T - \frac{100 - RH}{5}$$

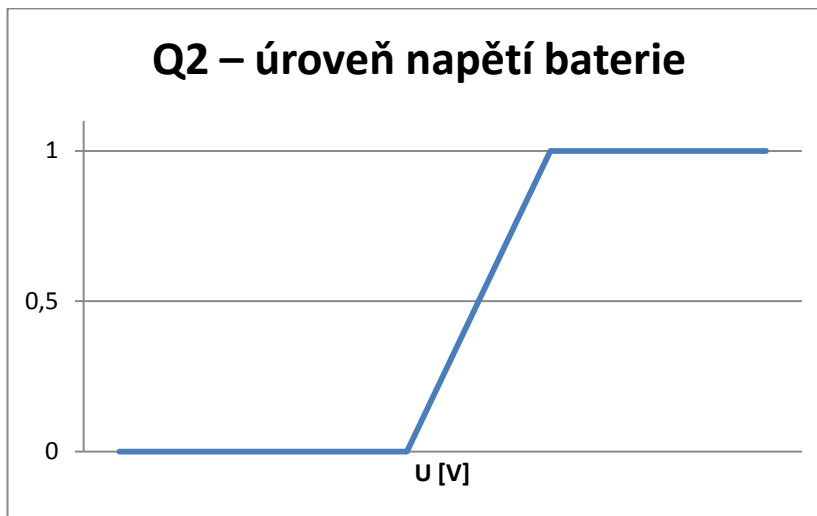
5.1 Q1 – fyzikální limity



Obrázek 11 - graf pro ukazatel kvality Q1

Tento ukazatel kvality porovnává aktuálně naměřená data senzorem s předem danými fyzikálními limity. Jak je vidět v grafu výše, tak ukazatel Q1 je roven 0 jak pro příliš nízké teploty, tak pro příliš vysoké. Při jeho výpočtu je toto tedy nutné vzít v potaz. Tento ukazatel tedy na rozdíl od ostatních nemá pouze dvě meze, mezi kterými je naškálován na číslo mezi jedničkou a nulou, ale má tyto meze čtyři. Levou dvojici pro nižší teploty a pravou pro vyšší. Tento ukazatel vychází ze znalosti prostředí, ve kterém bude zařízení použito. Pro účely testování byly tyto meze například nastaveny vlevo na -20 °C a -10 °C a vpravo na 60 °C a 80 °C. Naopak pokud je výsledek měření někde mezi vyšší levou mezí a nižší pravou, tak je hodnota tohoto ukazatele nastavena na 1. Pro relativní vlhkost vzduchu je to trochu jiný případ. Jsou dvě cesty, jak zde nastavit meze tohoto kvalifikátoru. Buď počítat s tím, že relativní vlhkost vzduchu je v pořádku od 0% do 100% a tedy tento ukazatel bude roven 0 pouze v případě, že by ze senzoru bylo vyčteno číslo mimo tyto meze, což je prakticky chyba měření. Druhou možností je nastavit meze trochu přísněji. Při testování tohoto parametru byly použity hodnoty mezí vlevo 0% a 5% a vpravo 95% a 100%. V těchto dvou intervalech tedy byl ukazatel škálován mezi nulu a jedničku.

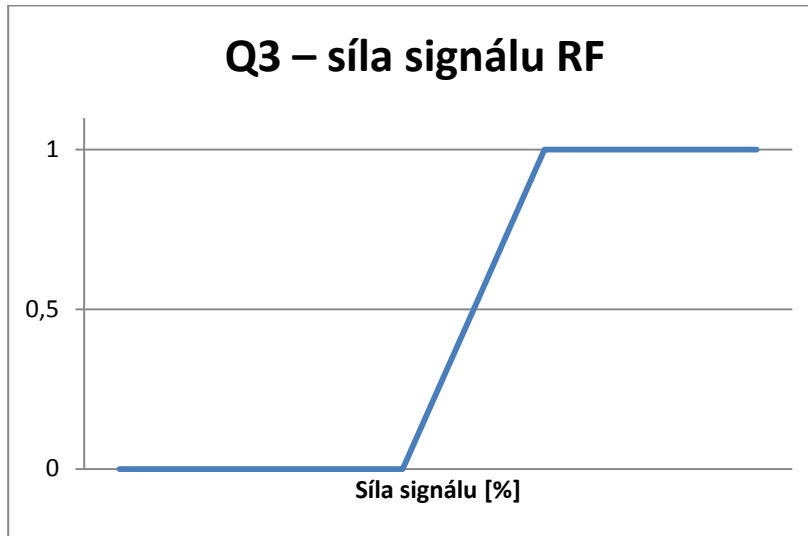
5.2 Q2 – úroveň napětí baterie



Obrázek 12 - graf pro ukazatel kvality Q2

Úroveň napětí baterie je první z ukazatelů kvality, který je závislý na aktuálním stavu uzlu v síti. Úroveň napětí baterie uzlu má velký vliv na vysílaný signál, takže je třeba ji vzít v potaz při počítání celkového ukazatele kvality. Na rozdíl od ukazatele Q1, který porovnává naměřená data s předem danými limity, má ale tento ukazatel pouze 2 meze. Od nuly do určitého napětí je tento ukazatel roven nule, vzhledem k tomu, že uzel nemůže s nízkým napětím fungovat správně. Naopak pokud je napětí dosti vysoké, tak je roven jedničce. V předem daném intervalu je opět škálován mezi nulou a jedničkou. Při testování byly použity hodnoty 2,6 V a 2,9 V.

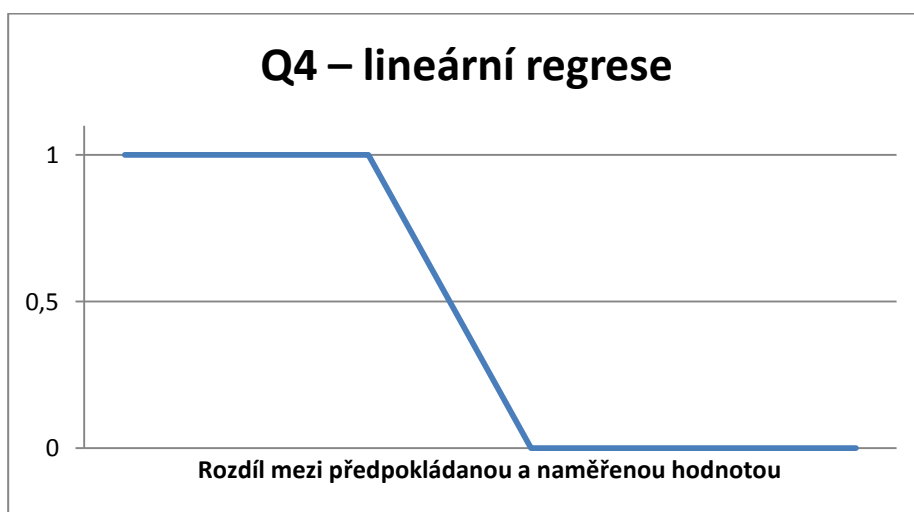
5.3 Q3 – síla signálu RF



Obrázek 13 - graf pro ukazatel kvality Q3

Síla signálu vzhledem k centrálnímu prvku je druhým ukazatelem, který je závislý na aktuálním stavu uzlu. Stejně jako napětí baterie uzlu i toto má samozřejmě vliv na vysílaný signál. Když je signál příliš slabý, tak je ukazatel nastaven na nulu, pokud je naopak dostatečně silný, tak je nastaven na jedničku. Mezi dvěma mezními hodnotami je škálován mezi nulu a jedničku. Při testování sítě byly použity hodnoty 50% a 70%.

5.4 Q4 – lineární regrese



Obrázek 14 - graf pro ukazatel kvality Q4

Lineární regrese je matematická metoda, pomocí které se prokládá množinou bodů přímka. Díky této přímce potom můžeme spočítat předpokládanou následující hodnotu. Vychází z metody nejmenších čtverců. Předpokládáme, že hodnoty na ose x jsou přesné, zatímco hodnoty na ose y jsou proměnné a ovlivněné nějakou chybovostí. V našem případě je na ose x pouze pořadí vzorků, zatímco na ose y jsou naměřené hodnoty teploty, respektive relativní vlhkosti vzduchu.

Díky tomu, že bereme v potaz vždy fixní počet předchozích bodů, je možné vyjádřit konstantní matici, kterou při každém měřicím cyklu vynásobíme čtyři předchozí vzorky a získáme parametry proložené přímky.

$$\begin{bmatrix} 1 & 0,5 & 0 & -0,5 \\ -0,3 & -0,1 & 0,1 & 0,3 \end{bmatrix}$$

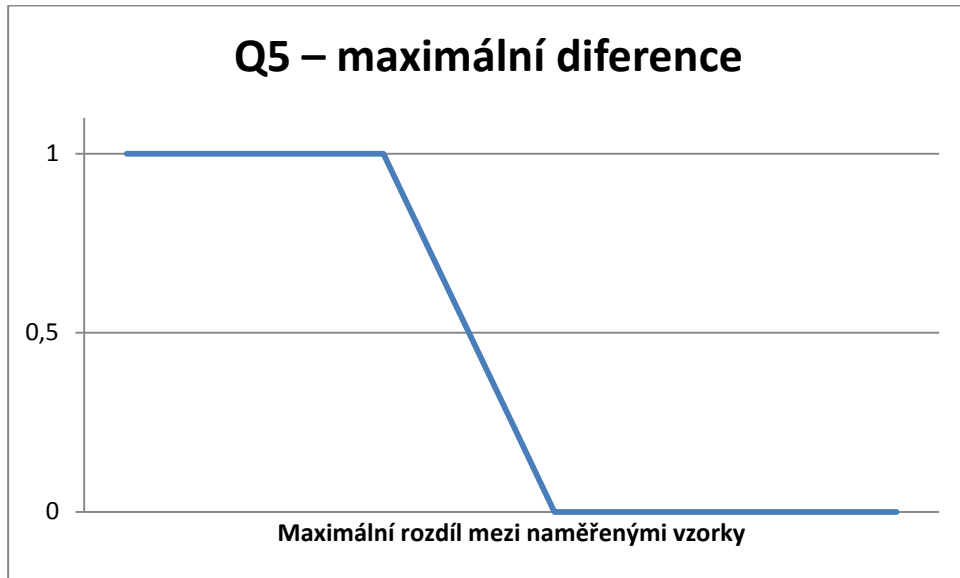
Touto maticí vynásobíme sloupcový vektor čtyř předchozích naměřených hodnot. Výsledkem této operace je sloupcový vektor o dvou prvcích. První z těchto prvků je potom parametr přímky A a druhý parametr B v rovnici přímky.

$$y = A + Bx$$

Do výsledné rovnice následně dosadíme další pořadové číslo prvku. V našem případě tedy číslo pět. Spočtením této rovnice dostaneme další předpokládanou naměřenou hodnotu.

Tento ukazatel je prvním z kategorie ukazatelů závislých na předchozích naměřených hodnotách ze senzorů. Pracuje se čtyřmi posledními vzorky teploty a relativní vlhkosti vzduchu. Z těchto uložených vzorků vyjádří rovnici lineárního trendu a následně spočte další předpokládanou naměřenou hodnotu. Tuto hodnotu odečte od skutečně naměřených dat. Na základě tohoto rozdílu potom vyhodnocuje ukazatel kvality Q4. Pokud je rozdíl malý, tak je vše v pořádku a ukazatel nabývá hodnoty jedna. Jakmile je příliš velký, tak nabývá hodnoty nula. Pro teplotu byly použity meze pro škálování 2 °C a 4 °C a pro relativní vlhkost vzduchu meze 25% a 40%.

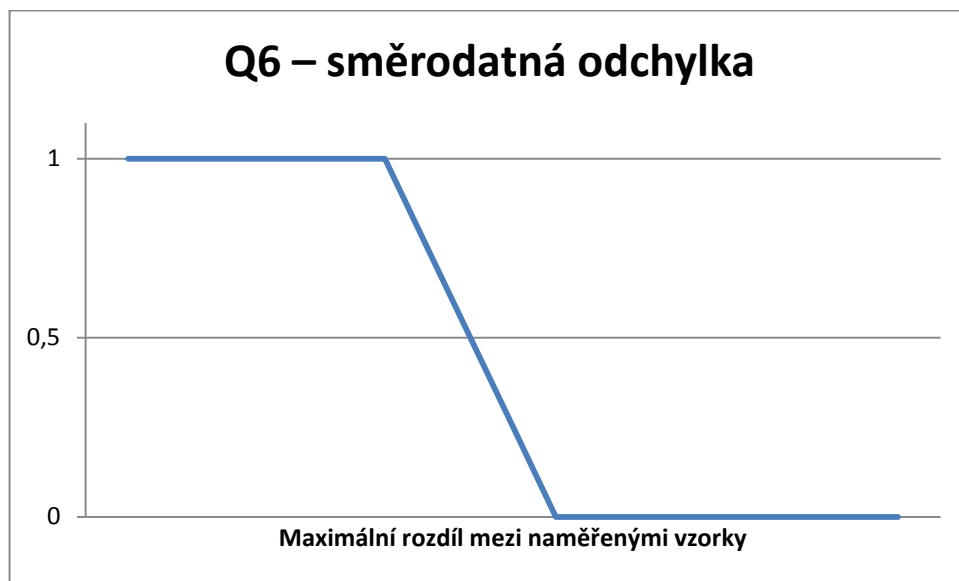
5.5 Q5 – maximální rozdíl mezi vzorky



Obrázek 15 - graf pro ukazatel kvality Q5

Tento ukazatel kvality měřených dat je další z kategorie ukazatelů závislých na předchozích naměřených hodnotách. Bere v potaz sedm předchozích vyčtených výsledků ze senzoru jak u měření teploty, tak u relativní vlhkosti vzduchu. Jak je vidět na obrázku, tak pokud je maximální diference malá, ukazatel nabývá hodnoty jedna. Jakmile je příliš velký rozdíl mezi nějakými vzorky, signalizuje to možnou chybu měření, nebo hardwaru, takže ukazatel klesá na nulu. Pokud je diference v určitých mezích, tak je opět škálován mezi jedničku a nulu. Při testování byly hodnoty mezi nastaveny na 10 °C a 15 °C pro teplotu a 30% a 60% pro měření relativní vlhkosti vzduchu. Tato funkce je implementována tak, že projde zvolený počet uložených hodnot, přičemž si zaznamená nalezené maximum a minimum. Ty potom odečte a dál vyhodnocuje, jak je popsáno výše v textu.

5.6 Q6 – směrodatná odchylka



Obrázek 16 - graf pro ukazatel kvality Q6

Směrodatná odchylka je v teorii pravděpodobnosti často používanou mírou statistické disperse. Vypovídá o tom, do jaké míry se od sebe navzájem liší typické případy v souboru zadaných hodnot. Je-li směrodatná odchylka malá, jsou si prvky v souboru většinou podobné. Velká směrodatná odchylka signalizuje velké rozdíly mezi zaznamenanými hodnotami.

Aritmetický průměr:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

Výběrová směrodatná odchylka:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Tento ukazatel je poslední ze skupiny ukazatelů závislých na předchozích naměřených hodnotách. Z osmi posledních hodnot je vždy spočtena výběrová směrodatná odchylka, která je dále posuzována ve smyslu grafu výše. Pokud je malá, tak to indikuje, že jsou si vzorky podobné a vše by mělo být v pořádku, jakmile se začne zvyšovat, je škálována od určitých mezí a ukazatel se pohybuje od jedničky k nule a

dále už je vždy nulový. Pro teplotu byly meze nastaveny po pozorování výsledků empiricky na 15 a 20 a pro relativní vlhkost vzduchu na 50 a 70.

5.7 Knihovna pro výpočet parametrů Q

Pro pohodlný výpočet těchto ukazatelů kvality měřených veličin byla vytvořena knihovna obsahující všechny potřebné funkce a konstanty. Hlavičkový soubor knihovny obsahuje definice mezí pro škálování všech dílčích ukazatelů, takže je možné je pohodlně měnit. Obsahuje také konstantní matici pro výpočet lineární regrese pro ukazatel Q4.

Quality_structure

Hlavní částí této knihovny je struktura `quality_structure`. Tato struktura obsahuje dva rotační zásobníky, pro ukládání naměřených hodnot teploty i vlhkosti a `FLOAT` proměnné pro ukládání všech dílčích a i hlavního ukazatele kvality

Compute_qualifiers

Tato funkce nemá žádnou návratovou hodnotu, ale jako parametr má ukazatel na proměnnou s datovým typem `Quality_structure`. Z této proměnné získá aktuální i předešlé vyčtené hodnoty ze senzorů a spočte všechny matematické operace, jako je lineární regrese, nebo směrodatná odchylka. S výsledky těchto operací poté zavolá funkce pro výpočet jednotlivých ukazatelů kvality `Compute_QN`, kde N je pořadové číslo příslušného ukazatele. Dále už jen spočte celkový ukazatel kvality Q a uloží ho pomocí ukazatele zpět do struktury zadané parametrem.

Compute_QN

Tyto funkce slouží k rozhodnutí, zda jsou výsledky matematických operací mimo zvolené meze, nebo uvnitř. Pokud jsou mimo, tak vrací nulu, respektive jedničku v závislosti na konkrétním ukazateli. Jestliže je hodnota zadaná parametrem funkce mezi zvolenými mezemi, tak ji naškáluje na číslo mezi nulou a jedničkou pomocí následujícího vztahu.

$$\text{Výsledek} = \frac{\text{Horní mez} - \text{zadaná hodnota}}{\text{Horní mez} - \text{dolní mez}}$$

Tento vztah odpovídá vztahu například pro rozdíl mezi předpokládanou hodnotou získanou lineární regresi a skutečnou naměřenou hodnotou veličiny. Pro jiné ukazatele kvality se mírně liší.

Použití knihovny

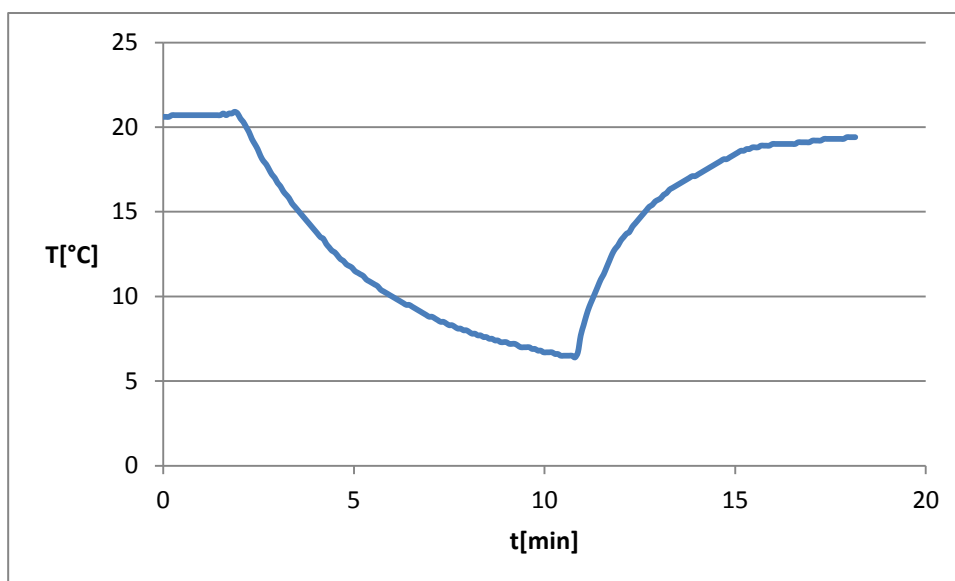
Předpokládané využití této knihovny spočívá ve vytvoření proměnné typu `Quality_structure`, do které se poté ve zvolených měřicích intervalech ukládají vyčtená data z čidel a další potřebné hodnoty, jako je stav napětí baterie. Následně je zavolána funkce `Compute_qualifiers`, které je předán ukazatel na tuto strukturu. Funkce do struktury doplní spočtené jak dílčí, tak i hlavní ukazatel kvality. Dále už je na uživatelské aplikaci, jak tyto hodnoty využije. Je možné je například s naměřenými daty odeslat do centrálního prvku, jako je to v testovací aplikaci.

6 Praktické ověření bezdrátové senzorové sítě

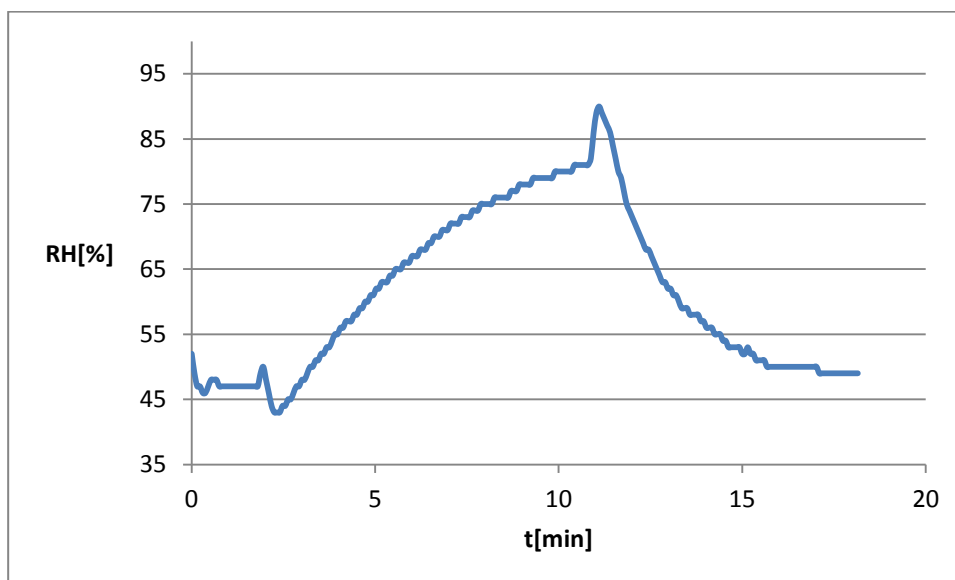
V této části bude několika způsoby otestováno měření teploty a relativní vlhkosti vzduchu. Koncový uzel bezdrátové senzorové sítě bude odesílat naměřená data a ta budou zaznamenávána v počítači, kterému je předá centrální prvek bezdrátové senzorové sítě. Měření budou úmyslně probíhat za zvláštních podmínek, které budou vždy zmíněny u jednotlivých pokusů, aby byla na výsledných grafech vidět závislost a chování měřených veličin.

6.1 Změna teploty a vlhkosti ve vnějším prostředí

Při tomto pokusu byl koncový uzel ve večerních hodinách a sychravém počasí položen za okno na parapet. Z důvodu ochrany elektroniky byl v igelitovém pytlíku. Tato skutečnost mohla ovlivnit měření relativní vlhkosti vzduchu. Zhruba v půlce časového úseku byl měřicí prvek přemístěn opět do pokojového prostředí.



Obrázek 17 - závislost teploty na čase při vystavení vnějšimu prostředí

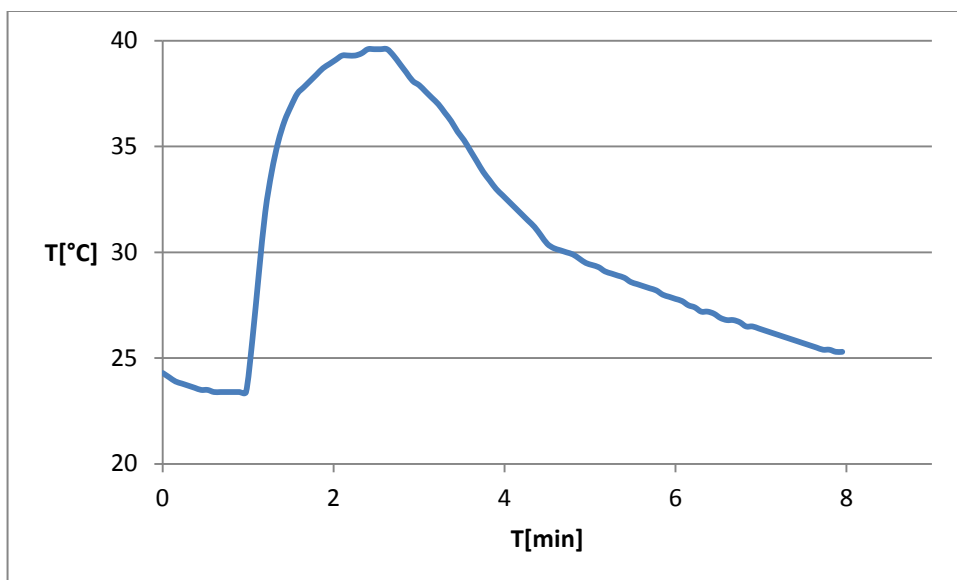


Obrázek 18 - závislost vlhkosti na čase při vystavení vnějšimu prostředí

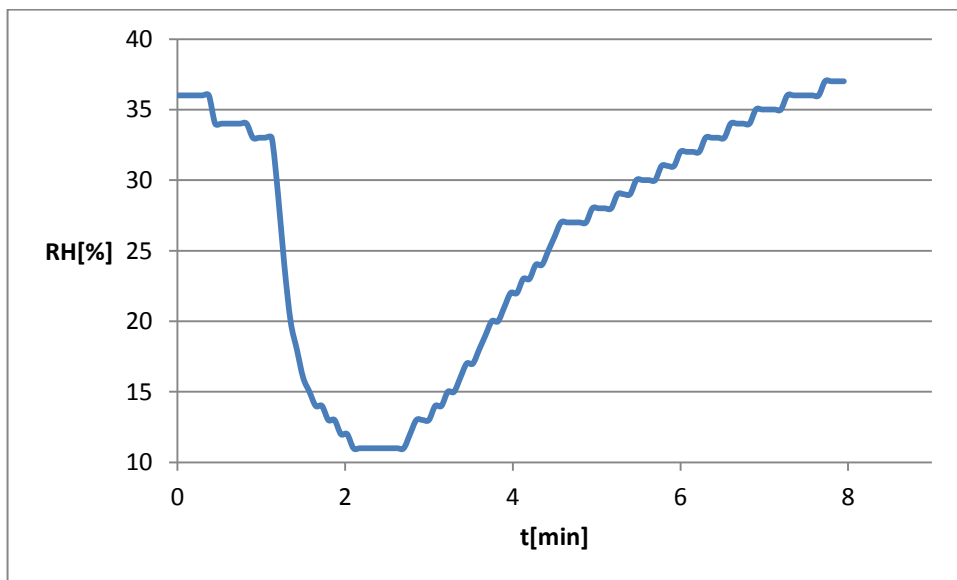
Na prvním grafu je vidět, jak naměřená teplota nejdříve klesá rychleji, ale postupně konverguje ke skutečné hodnotě. Po přemístění měřícího uzlu je vidět stejné chování. Nejdříve teplota vzrůstá rychleji a postupně se tempo růstu zpomaluje. Při porovnání obou grafů je vidět, jak v igelitovém pytlíku stoupala vlhkost. Zvláště po přemístění ochlazeného zařízení do teplejšího prostředí vyskočila naměřená hodnota vlhkosti skokově nahoru.

6.2 Změna teploty a vlhkosti ve vnitřním prostředí

V tomto experimentu bylo měřicí zařízení ponecháno chvíli v klidu a následně na něj byl zaměřen horký vzduch z fénu nastaveného na nejnižší úroveň zhruba ze vzdálenosti 15 cm. V okamžiku, kdy růst teploty a klesání relativní vlhkosti vzduchu začal zpomalovat byl fén vypnut a měřicí prvek postupně samovolně chladnul.



Obrázek 19 - závislost teploty na čase při fénování senzoru



Obrázek 20 - závislost relativní vlhkosti vzduchu při fénování senzoru

Z grafů je patrné, jak rapidně teplota stoupá pod horkým vzduchem z fénu. Na rozdíl od prvního pokusu, kde nebyl pokles teploty ve venkovním prostředí tak strmý. Při porovnání grafů je také zřejmé, že ohřátý vzduch z fénu je velice suchý, takže klesá

relativní vlhkost vzduchu. Samovolné chladnutí senzoru je opět pozvolné a není zdaleka tak příkré, jako při jeho zahřívání fénem.

7 Závěr

Cílem této práce bylo implementovat metodu pro hodnocení informačního obsahu snímaných veličin pomocí senzorů.

Byla implementována bezdrátová senzorová síť na platformě ez430-RF2500 tvořená přístupovým bodem připojeným pomocí rozhraní UART k počítači a několika koncovými uzly, které provádějí toto vestavěné zpracování měřeného signálu a následně jej předávají skrz přístupový bod sítě uživateli pro interpretaci.

Měřené veličiny, teplota a relativní vlhkost vzduchu jsou měřeny pomocí čidla SHT11 připojeného ke koncovým uzlům bezdrátové senzorové sítě. Byl implementován komunikační protokol pro ovládání a vyčítání naměřených hodnot z tohoto senzoru.

Tato metoda hodnocení obsahu dané veličiny je tvořena několika ukazateli kvality naměřených dat. Z toho důvodu byla implementována knihovna pro spočtení těchto ukazatelů a jejich následné sjednocení do hlavního ukazatele kvality měření dané veličiny.

Několika měřeními v různých prostředích byla ověřena funkčnost bezdrátové senzorové sítě měřící teplotu a relativní vlhkost vzduchu v koncových uzlech sítě.

Seznam obrázků

Obrázek 1 - architektura koncového uzlu sítě.....	3
Obrázek 2 - topologie bezdrátových senzorových sítí	4
Obrázek 3 - vývojový kit ez430-RF2500.....	8
Obrázek 4 - topologie sítě SmartMesh.....	10
Obrázek 5 - vývojový kit BTnode.....	11
Obrázek 6 - topologie základní testovací aplikace.....	14
Obrázek 7 - schéma připojení senzoru SHT11	18
Obrázek 8 - schéma zapojení open drain	19
Obrázek 9 - topologie koncového uzlu sítě z pohledu ukazatelů kvality.....	29
Obrázek 10 - spojení přístupového bodu a koncového uzlu	30
Obrázek 11 - graf pro ukazatel kvality Q1	31
Obrázek 12 - graf pro ukazatel kvality Q2	32
Obrázek 13 - graf pro ukazatel kvality Q3	33
Obrázek 14 - graf pro ukazatel kvality Q4	33
Obrázek 15 - graf pro ukazatel kvality Q5	35
Obrázek 16 - graf pro ukazatel kvality Q6	36
Obrázek 17 - závislost teploty na čase při vystavení vnějšímu prostředí	39
Obrázek 18 - závislost vlhkosti na čase při vystavení vnějšímu prostředí.....	39
Obrázek 19 - závislost teploty na čase při fénování senzoru	40
Obrázek 20 - závislost relativní vlhkosti vzduchu při fénování senzoru	40

Seznam tabulek

Tabulka 1 - přehled používaných systémů.....	6
Tabulka 2 - příklad výsledného pole pro číslo 23,621	16
Tabulka 3 - seznam příkazů senzoru SHT11	20
Tabulka 4 - popis jednotlivých bitů status registru senzoru SHT11	22
Tabulka 5 - blokové schéma komunikace se senzorem SHT11.....	23

Obsah přiloženého CD

1. Zdrojový kód centrálního prvku sítě
2. Zdrojový kód koncového bodu
3. Implementované knihovny pro UART, ukazatele kvality a komunikace se senzorem
4. Text diplomové práce ve formátu pdf

Reference

- [1] O. Kreibich, J. Neužil, R. Šmíd: Quality-based Multiple Sensor Fusion in an Industrial Wireless Sensor Network for MCM, to appear in IEEE Transactions on Industrial Electronics, 2013
- [2] K. Holger, A. Willing: Protocols and Architectures for Wireless Sensor Networks, Wiley, 2005
- [3] Wireless Sensor Network [online]. [cit. 2014-04-19]. Dostupné z WWW:
<http://en.wikipedia.org/wiki/Sensor_network>
- [4] What is a Wireless Sensor Network [online]. [cit. 2014-04-20]. Dostupné z WWW:
<<http://www.ni.com/white-paper/7142/en/>>
- [5] Otevřený software [online]. [cit. 2014-11-20]. Dostupné z WWW:
<http://cs.wikipedia.org/wiki/Otev%C5%99en%C3%BD_software>
- [6] M. Hempstead: An Accelerator-Based Wireless Sensor Network Processor in 130nm CMOS, Harvard University, 2009
- [7] Intel I/O Processors [online]. [cit. 2014-11-29]. Dostupné z WWW:
<<http://www.intel.com/content/www/us/en/intelligent-systems/previous-generation/i-o-processors-intel-iop-series-reduces-communications-bottleneck.html/index.htm>>
- [8] XScale [online]. [cit. 2014-11-29]. Dostupné z WWW:
<<http://en.wikipedia.org/wiki/XScale>>
- [9] Guide to Crossbow Motes [online]. [cit. 2014-10-23]. Dostupné z WWW:
<<http://www.pages.drexel.edu/~kws23/tutorials/motes/motes.html>>
- [10] MSP430x2xx Family User's Guide, Texas Instruments, Revised 2013
- [11] CC2500 [online]. [cit. 2014-07-02]. Dostupné z WWW:
<<http://www.ti.com/lit/ds/symlink/cc2500.pdf>>
- [12] Dust Networks Technology [online]. [cit. 2014-10-03]. Dostupné z WWW:
<http://www.linear.com/designtools/wireless_mesh_networks.php>

[13] BTnodes – A Distributed Enviroment for Prototyping Ad Hoc Networks [online]. [cit. 2014-10-05]. Dostupné z WWW: <<http://www.btnode.ethz.ch/>>

[14] Event-driven programming [online]. [cit. 2014-12-03]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Event-driven_programming>

[15] What is TinyOS [online]. [cit. 2014-09-15]. Dostupné z WWW: <http://tinycos.stanford.edu/tinycos-wiki/index.php/FAQ#What_is_TinyOS.3F>

[16] Datasheet SHT1x Humidity and Temperature sensor, Sensirion, Revised 2010

[17] Vlhkost vzduchu a její měření [online]. [cit. 2014-04-13]. Dostupné z WWW: <<http://www.tzb-info.cz/3137-vlhkost-vzduchu-a-jeji-mereni>>

[18] Silicon bandgap temperature sensor [online]. [cit. 2014-04-13]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Silicon_bandgap_temperature_sensor>