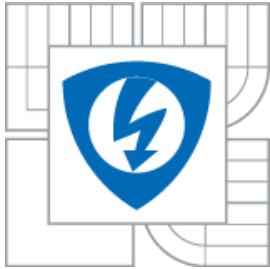




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

BEZDRÁTOVÉ KOMUNIKACE PRO DOMÁCÍ AUTOMATIZACI

WIRELESS COMMUNICATION IN HOME AUTOMATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

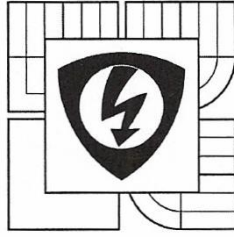
MICHAL PROKOP

VEDOUČÍ PRÁCE

SUPERVISOR

ING. ONDŘEJ HYNČICA

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor

Automatizační a měřicí technika

Student: Michal Prokop

Ročník: 3

ID: 154837

Akademický rok: 2014/15

NÁZEV TÉMATU:

Bezdrátové komunikace pro domácí automatizaci

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je seznámení se s technologií pro bezdrátové komunikace IQRF a vytvoření systému pro bezdrátový sběr dat.

- 1) Seznamte se s technologií IQRF a popište způsoby přenosu dat.
- 2) Vytvořte aplikace pro sběr dat v síti za využití různých možností technologie IQRF.
- 3) Porovnejte a vyhodnoťte přístupy přenosu dat z hlediska latence a spotřeby energie.
- 4) Vytvořte jednoduchou aplikaci pro demonstraci zvoleného přístupu.

DOPORUČENÁ LITERATURA:

STALLINGS W.: Wireless Communications & Networks. 2.vyd. Upper Saddle River (NJ):

Pearson Prentice Hall, 2005. ISBN 0-13-191835-4.

Termín zadání: 9. 2. 2015

Termín odevzdání: 25.5.2015

Vedoucí práce: Ing. Ondřej Hynčica

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

předseda oborové rady



UPOZORNĚNÍ:

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

Abstrakt

Tato bakalářská práce se zabývá vytvořením bezdrátové komunikační sítě technologií IQRF, která je určena pro domácí automatizaci. Pro její vytvoření se používá DPA custom handler, díky němuž lze síť vytvořit bez programování. Pro její realizaci je použit model ovládání žárovky a to pomocí koordinátora nebo tlačítkem na uzlu. Následně se pro tuto synchronní síť proměřuje její latence a spotřeba energie při různých možnostech synchronního provedení komunikace i asynchronním provedením.

Klíčová slova

IQRF, bezdrátová komunikace, koordinátor, uzel, bondování, discoverování, synchronní, asynchronní, DPA custom handler

Abstract

This bachelor thesis is about how to make wireless communication network by IQRF technology which is intended for a home automation. It's created by using DPA custom handler. This network can be made without programming thanks to DPA custom handler. This wireless network is realized by light model that is controlled by coordinator or by a button on a node. After that I measured latency and energy consumption on several various synchronous network and asynchronous network.

Keywords

IQRF, wireless communication, coordinator, node , bonding, discovering, synchronous, asynchronous, DPA custom handler

Bibliografická citace:

PROKOP, M. *Bezdrátové komunikace pro domácí automatizaci*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 44s. Vedoucí bakalářské práce byl Ing. Ondřej Hynčica.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Bezdrátové komunikace pro domácí automatizaci jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujícího autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **25. května 2015**

.....

podpis autora

Poděkování

Děkuji panu vedoucímu bakalářské práce Ing. Ondřeji Hynčicovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce a dále také za jeho vstřícnost a ochotu být kdykoliv nápomocen.

V Brně dne: **25. května 2015**

.....

podpis autora

OBSAH

Obsah	7
1 Úvod	8
1.1 Cíl práce	9
2 Technologie IQRf	10
2.1 Princip funkce IQRf sítě.....	11
2.2 Nastavení sítě	11
2.3 Navazování.....	12
2.4 Objevení uzlů	13
2.5 Doručování paketů	15
2.6 CATS Service Tools	16
3 Fast response commands (FRC).....	18
4 Asynchronní Komunikace	20
5 Použití DPA custom Handler	22
5.1 DpaEvent_Init	22
5.2 DpaEvent_Idle.....	22
5.3 DpaEvent_Interrupt.....	22
5.4 DpaEvent_ReceiveDpaRequest	23
5.5 DpaEvent_FrcValue	23
5.6 DpaEvent_BeforeSendingDpaResponse.....	23
5.7 DpaEvent_ReceiveDpaResponse	23
6 Sběr dat při různých možnostech IQRf	24
6.1 Průběh odbíraného proudu při funkci FRC	24
6.2 Průběh odbíraného proudu při Asynchronní komunikaci	28
7 Měření spotřeby energie	30
8 Měření latence	35
9 Realizace Aplikace	37
9.1 Synchronní sběr dat.....	37
9.1.1 Algoritmus řešení:	37
9.2 Asynchronní sběr dat.....	38
9.2.1 Algoritmus řešení	39
10 Vyhodnocení výsledků	40
11 Závěr	41

1 ÚVOD

Tato bakalářská práce se zabývá bezdrátovou komunikací. Konkrétně se zmiňuje o konfiguraci a naprogramování komunikačních modulů IQRF.

Prvním cílem dané bakalářské práce je obeznámit se, jak vytvořit síť s IQRF moduly při využití DPA custom handleru, který umožňuje vytvořit síť jednoduše bez programování, následně zjistit, která metoda získávání dat je vhodnější pro použití v praktických úlohách pro domácí automatizaci. Vhodnost se bude posuzovat na základě změřené latence a především na spotřebě energie. K tomu poslouží model rozsvícení světla.

Prvním přenosem dat, jímž se tato bakalářská práce zabývá, je synchronní sběr dat. Při něm je komunikace řízena koordinátorem a probíhá vždy v pravidelných cyklech na výzvu koordinátora. Koncové uzly jsou stále připraveny k přijetí dotazu. Výhodu tohoto provedení lze očekávat v rychlejším přenosu dat, ale za cenu větší spotřeby energie, protože uzly jsou stále v aktivním režimu. Příkladem může být měření teploty v pokoji, kdy se centrální uzel táže každých deset minut koncových uzlů, jaká je v místnosti teplota.

S cílem dosáhnout nižší spotřeby energie byla synchronní komunikace provedena v nízko příkonovém režimu a dále pak ve spánkovém režimu, kdy spánek trval určitou dobu po odeslání dat koordinátorovi.

V hlavní síti IQRF lze realizovat pouze synchronní přenos, proto při druhém řešení se musí použít vedlejší podřízená síť, která je řízena asynchronně. Komunikace je v tomto případě vyvolána koncovým uzlem, který na základě nějaké události pošle svá data hlavnímu uzlu. Zde už nemusí být koncové uzly neustále v aktivním režimu, neboť je stačí aktivovat pouze v době, kdy budou vysílat a pak přejdou opět do režimu spánku, čímž se předpokládá nižší spotřeba energie než u synchronní komunikace a rychlejší přenesení dat z uzlu do koordinátora. Komplikovanější je ale přenos dat z koordinátora do asynchronního uzlu, protože příkaz musí nejdříve přijít do uzlu v synchronní síti, který je koordinátorem pro asynchronní podsíť. Jelikož je asynchronní uzel většinu času v režimu spánku, není tudíž na příjmu, musí se proto asynchronní uzel dotazovat, zda koordinátor má pro něj data. Příkladem budiž žárovka, která se rozsvítí po obdržení paketu na základě sepnutí tlačítka u koncového uzlu v sub-síti.

1.1 Cíl práce

Hlavním cílem práce je dosáhnout nízké spotřeby energie, aby baterie, které napájí bezdrátové moduly, vydržely co nejdéle a nemusely se často měnit. Energií se budu snažit ušetřit v době, kdy moduly budou nečinné a zbytečně jsou na příjmu. Dalším cílem je popsat, jak vytvořit IQRF síť jednoduše bez programování za použití pluginů a DPA custom handleru. Tyto získané poznatky mají posloužit studentům, kteří by se rozhodli použít IQRF technologii v automatizaci.

2 TECHNOLOGIE IQRF

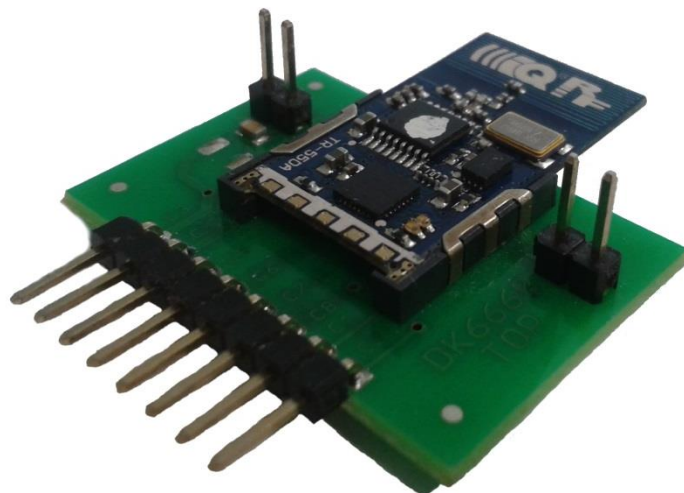
IQRF je nízko příkonová bezdrátová komunikace určená zejména pro domácí automatizaci, ale nachází uplatnění i v průmyslových aplikacích. Komunikace používá volná bez licenční frekvenční pásma 433 MHz, 868 MHz a pro americký kontinent 916 MHz. Na každém frekvenčním pásmu je pak určitý počet kanálů, které se mohou využít ke komunikaci. Firma pro komunikaci vytvořila svůj vlastní komunikační protokol IQMESH Open Protocol. Obdobu pro použití v bezdrátové komunikaci lze spatřit například v ZigBee. [5],[7]

Základem sítě jsou uzly, každá síť má pouze jeden hlavní uzel zvaný koordinátor. V jednotlivé síti můžeme použít až 239 uzlů, každý z nich můžeme nastavit jako směrovač nebo může sloužit jako sub koordinátor, který tvoří sub-síť. Sub-sítě umožňují asynchronní sběr dat, zatímco hlavní síť sbírá data synchronně. [1]

Při vytváření sítě se dá použít maximálně 240 spojů (hopů) mezi uzly. Rychlost jednoho hopu je pak 30-50 ms. Rychlost závisí na tom, kolik dat z maximálního počtu 56 B na paket se pošle v jednom paketu. Používaný DPA protokol automaticky zvolí délku paketu. Technologie IQRF funguje v 10 ms slotech, to znamená, že rychlost hopu může být 30, 40 nebo 50 ms. Přenosová rychlost sítě je 19,2 kBaud¹. [1]

Dosah mezi jednotlivými uzly sítě v interiéru výrobce udává v řádech desítek metrů. Závisí na vlivu prostředí. Při venkovním prostředí je pak rozsah dosahu mezi nimi až na stovky metrů. [1]

Komunikační moduly se dají napájet dvěma tužkovými bateriemi. Napájecí napětí se může pohybovat mezi 1,8 V, až 3,4 V. Modul se umísťuje na SIM konektor.



Obrázek 1 IQRF TR modul

¹ Baud [Bd] je modulační jednotka rychlosti, vyjadřuje počet změn stavu přenosového média za 1 sekundu.

2.1 Obdobné bezdrátové technologie

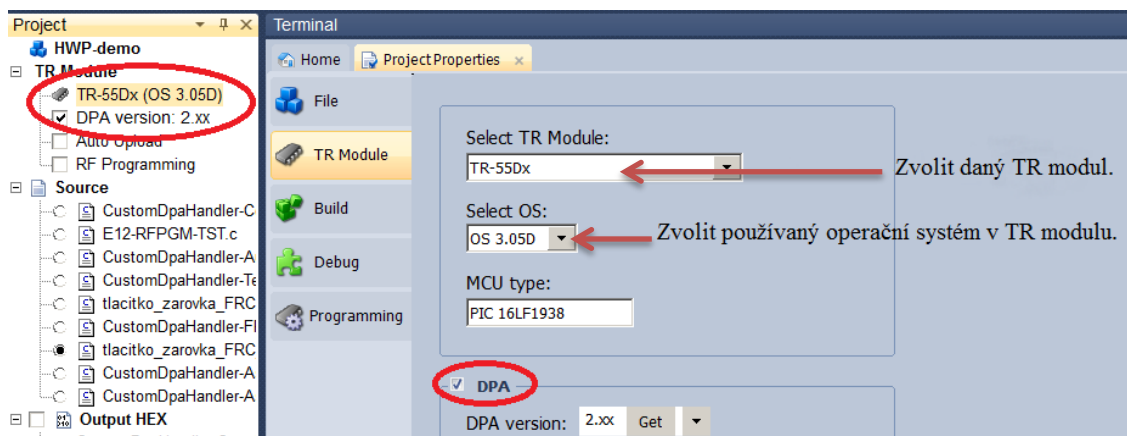
Pro obdobné bezdrátové komunikace, lze využít například Bluetooth Low Energy nebo zařízení založené na standardu IEEE 802.15.4 jako jsou například ZigBee, WirelessHart. Výhoda IQRF technologie vůči zde výše zmíněným komunikacím, je především v tom, že vlastní komunikace je zajištěna vlastním operačním systémem a použitím pluginů. Takže síť lze vytvořit bez složitého programování, což ulehčuje a zrychluje práci.

2.2 Princip funkce IQRF sítě

V síti se jeden hlavní koordinátor dotazuje uzlů, které mu zpětně odpovídají. Aby síť fungovala, musejí být uzly ke koordinátorovi přivázány, dále pak se jejich spojení s koordinátorem ověřuje discoverováním, čímž dostávají virtuální routovací čísla. Samotná komunikace, jako je například, že má vysílat či přijímat, zajišťuje operační systém, který je již nahrán do modulů. Funkci modulu jako je koordinátor či uzel zajišťují připravené hardwarové profily. Do modulů lze nahrát vlastní program, který je založen na rutinách, které nastávají pro určité události.

2.3 Nastavení sítě

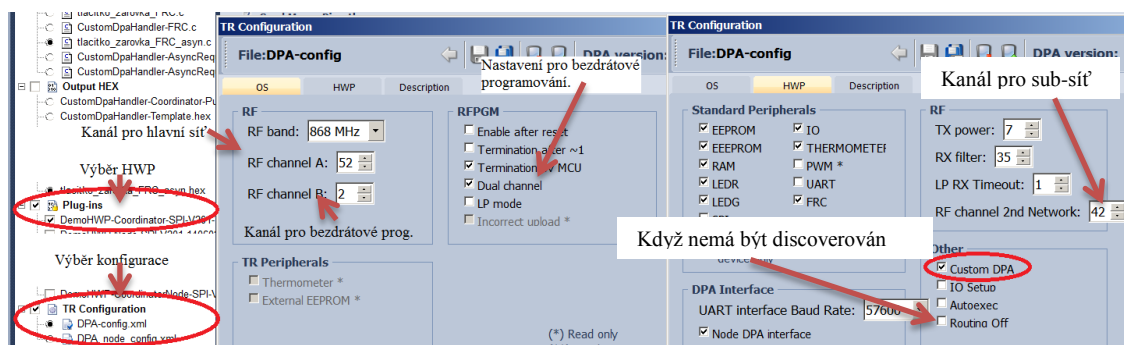
Síť se vytváří pomocí vývojového prostředí IQRF IDE4. Po vytvoření nového projektu se nejprve musí nastavit vlastnosti projektu. Komunikace s modulem a počítačem je zajištěna pomocí vývojové soupravy DK-EVAL04A nebo CK-USB-04A, která je spojena s počítačem přes USB.



Obrázek 2 nastavení s jakým modulem se bude pracovat

Do každého TR² modulu se musí nahrát Hardwarový profil a konfigurace. Vhodné nastavení konfigurace je zobrazeno na Obrázek 3. K dispozici jsou tři základní hardwarové profily:

- Coordinator koordinátor pouze jeden v síti
- Node uzly
- CoordinatorNode uzel v hlavní síti a jako koordinátor v sub-síti [2]



Obrázek 3 nastavení hardwarové konfigurace

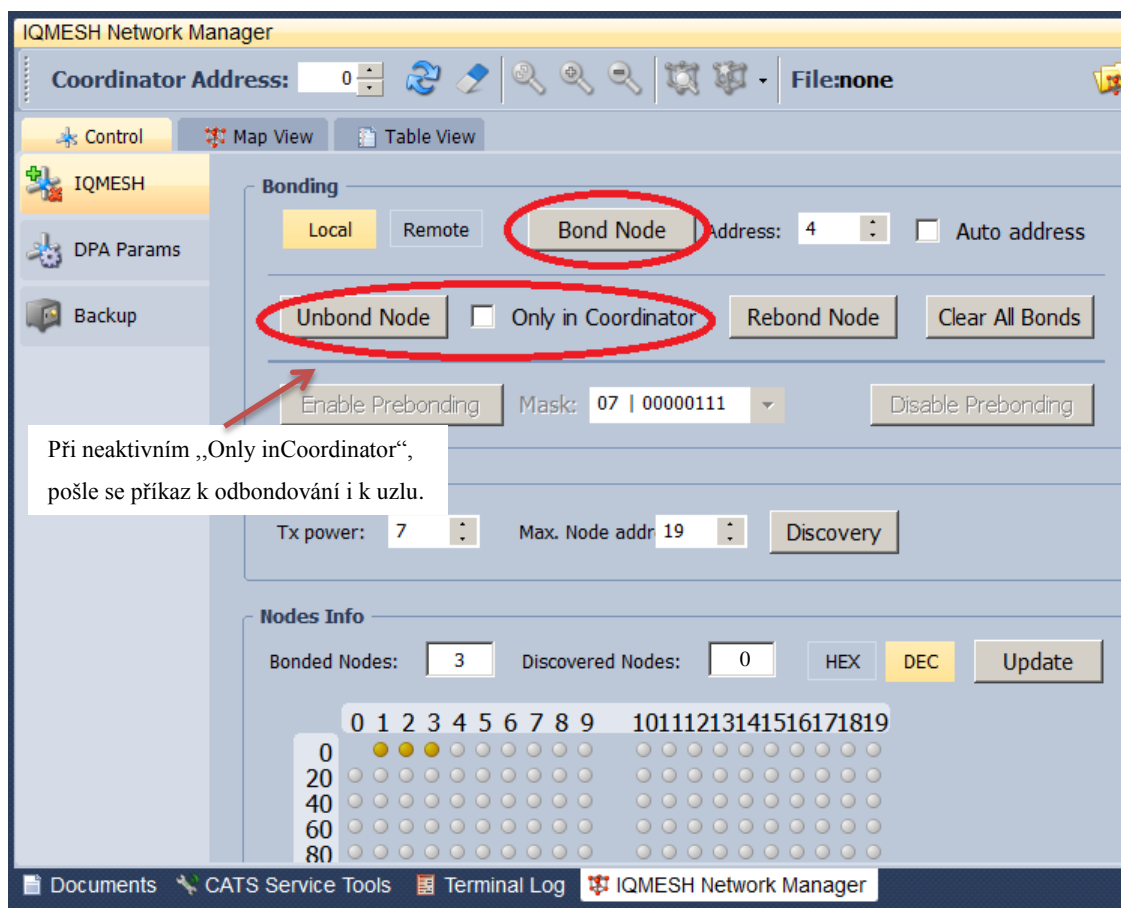
2.4 Navazování

Navazování a následné objevení uzlů se provádí v IQMESH Network Manager. Ke koordinátorovi se musí přivázat všechny uzly v síti. Ke správnému přivazování k síti musí být nejdříve uzel správně odvázan z předchozí sítě, pokud modul byl již použit dříve v jiné síti. Nejvíce praktické se zdá odvázat uzly ze staré sítě pomocí CATS service tool, jelikož při něm se k tomu nepotřebuje koordinátor, ke kterému jsou připojeny viz kapitola č. 2.7. [2]

Přivázaný uzel blikne pouze jednou červeně. Uzel, který je připraven k navazování bliká červeně po dobu deseti sekund, pak přejde do režimu spánku. Po kliknutí na tlačítko „Bond³ node“ se musí propojit pin –SS a pin VCC (pro modul TR 55D se jedná o piny C5 a C3). [3]

² TR anglická zkratka pro transceiver český vysílač

³ Anglické slovo bond český vázat.



Obrázek 4 svazování uzlů ve vývojovém prostředí

„Remote Bonding“ je určeno pro uzly, které nejsou v přímém dosahu koordinátora. Zvolená maska musí být vždy větší než počet uzlů k navazování. [1]

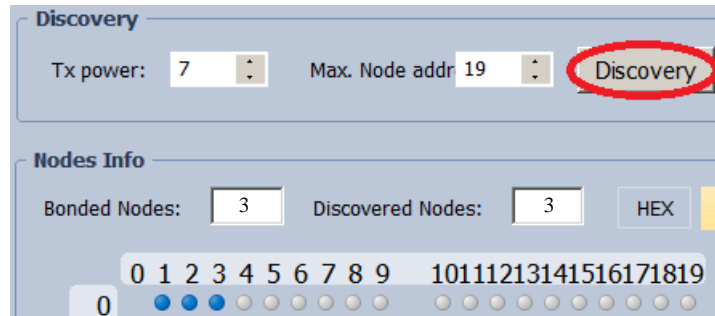
2.5 Objevení uzlů

Po navázání uzlů je potřeba zjistit, který uzel komunikuje s koordinátorem. Koordinátor zjišťuje, které uzly jsou v síti přímo kolem něj, a přiděluje jim virtuální směrovací čísla. Čísluje se od jedné a nejnižší číslo dostane nejbližší uzel s nejnižším adresovým číslem, na Obrázek 6 jsou to žluté číslice. Pokud některé uzly v síti mají nastaveno routing⁴ off, tak jako by při discovery⁵ procesu nebyly. [4],[5]

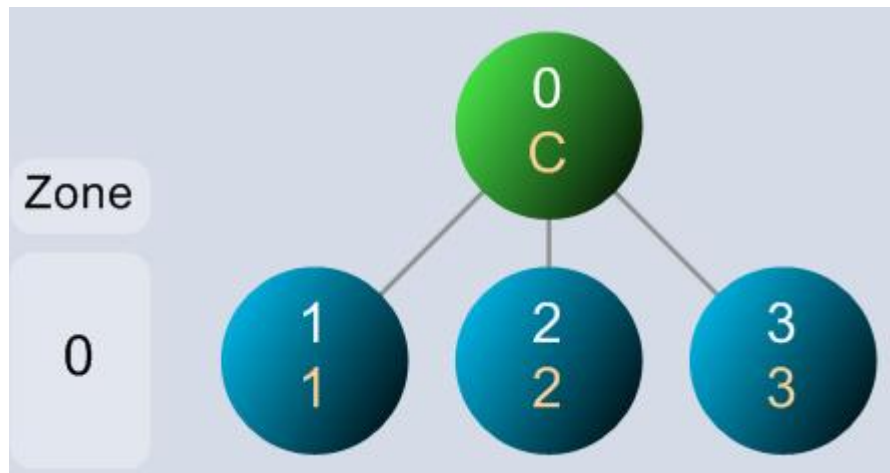
Celková síť je dělena do zón. Zóna nula obsahuje uzly, k nimž se koordinátor dostane bez dalšího směrovacího skoku. Následující zóny vznikají tak, že uzly z předchozí zóny se dostanou k dalším uzlům bez použití směrovacího skoku. [4]

⁴ Anglické slovo route česky směrovat

⁵ Anglického slovo discovery česky objevování



Obrázek 5 uzly nalezené pomocí discovery ve vývojovém prostředí

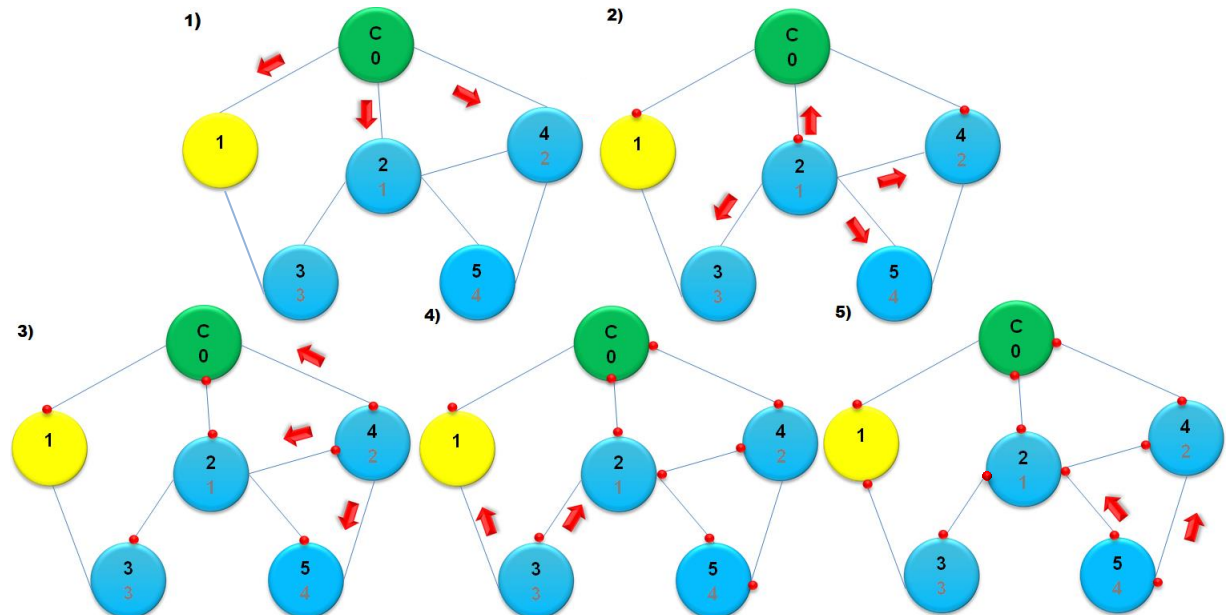


Obrázek 6 síť typologie hvězda

Samotný proces v IQMESH Network Manageru je velice jednoduchý. Kliknutím na tlačítko „Discovery“ se ke koordinátoru připojí všechny přivázané uzly. Dále IQMESH Network Manager automaticky navrhne typologii sítě. [2]

2.6 Doručování paketů

Komunikace je zahájena vždy koordinátorem. V prvním kroku koordinátor pošle data všem okolním uzlům. V dalším kroku modul s nejnižším číslem virtuálního směrovače pošle informaci okolním uzlům. Tento krok se opakuje, dokud všechny uzly neobdrží danou zprávu. Uzel číslo jedna neposílá data, protože nemá povolené směrování. [5]



Obrázek 7 zobrazení přenosu paketu v síti [1]

Tento princip posílání dat se nazývá Directed Flooding⁶. Technologie neumožňuje poslat data z více uzlů najednou, protože by ve vzduchu došlo k interferenci signálů a data by se ztratila. Výhodou daného principu šíření dat je to, že do uzlu se data dostanou vícekrát a díky tomu je nízká pravděpodobnost, že dojde ke ztrátě poslaných dat a navíc síť bude často fungovat, i když některé spoje mezi uzly fungovat nebudou. Za nevýhodu se dá považovat, když už paket obdržely všechny uzly, avšak zpráva je dále rozesílána, protože posílání paketu probíhá až do té doby, než ho odešle směrovací uzel s nejvyšším směrovacím číslem. [1]

Doba rozeslání paketu po celé síti se dá určit výpočtem z rovnice č. 1:

$$t = t_{\text{hop}} \cdot n_{\text{hopů}} \quad (1)$$

kde:

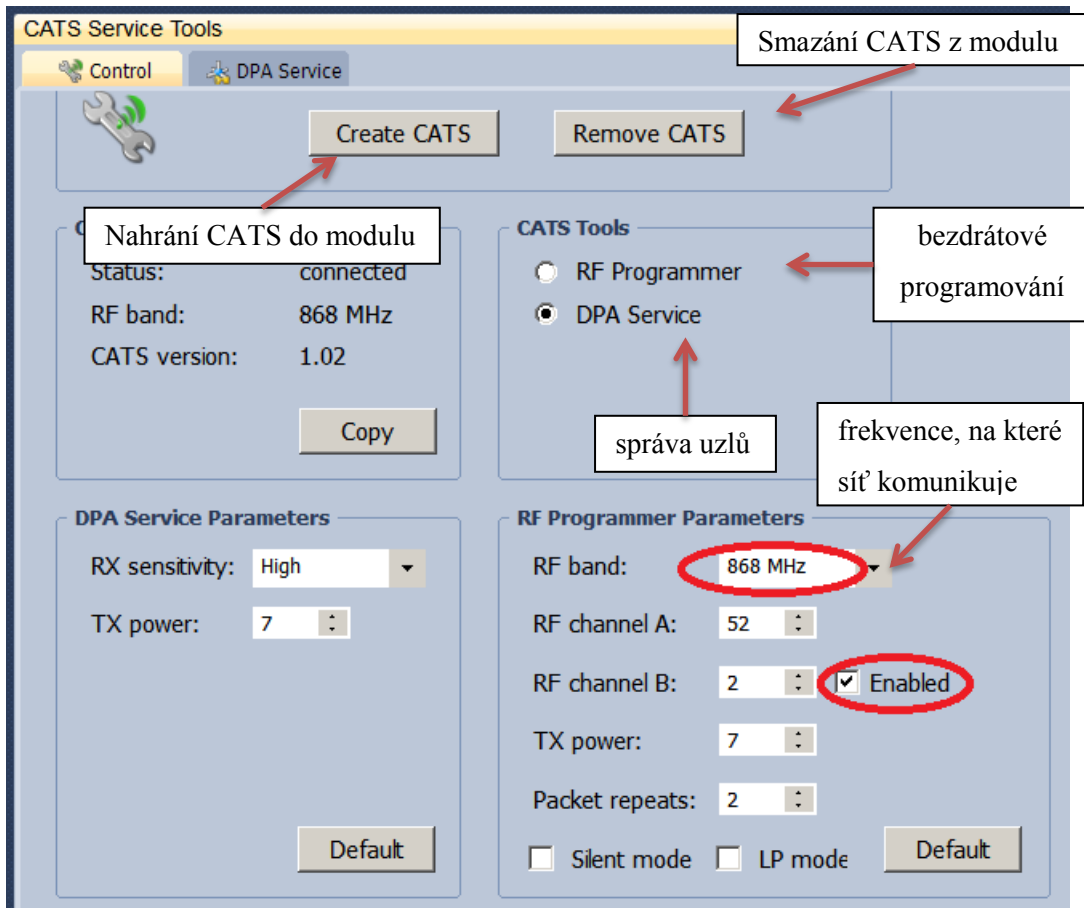
- t_{hop} je čas na vykonání jednoho hopu a může nabývat hodnot: 30, 40 nebo 50 ms.
- $n_{\text{hopů}}$ je počet hopů⁷ v síti, maximálně 240. [1]

⁶ Directed Flooding česky přímé zaplávání, což je algoritmus při kterém poslaný paket projede přes všechny uzly.

⁷ Jeden hop odpovídá jednomu směrovači.

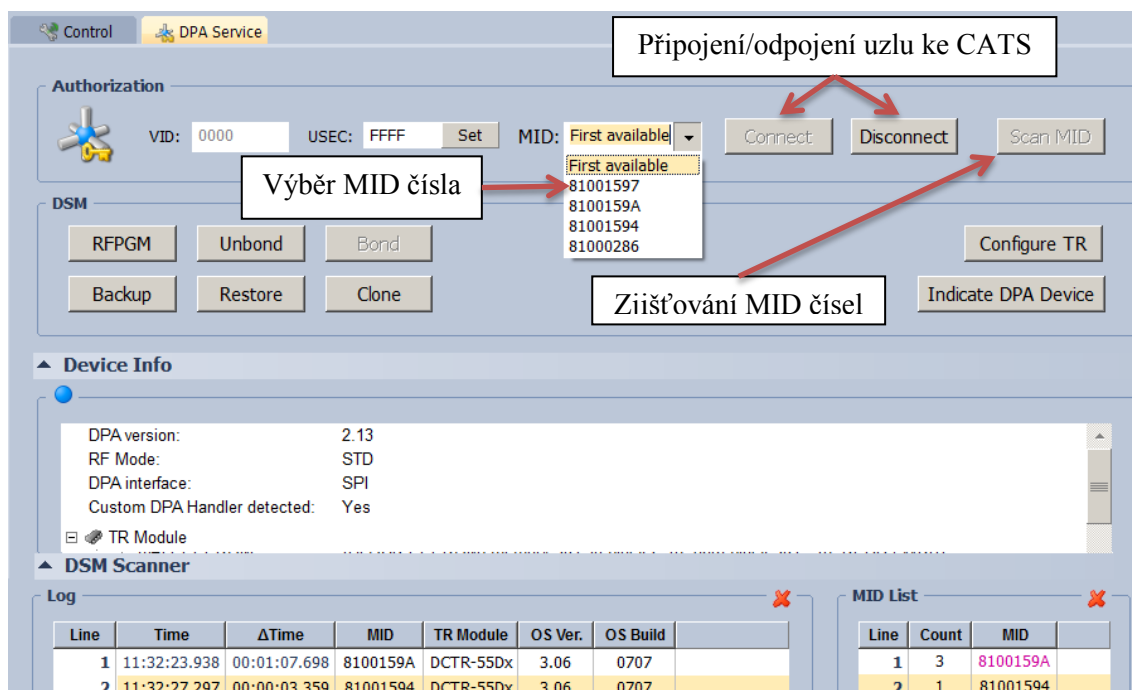
2.7 CATS Service Tools

Tento nástroj se používá pro bezdrátové programování, to znamená, že se nemusí fyzicky připojit modul k počítači, aby se do něj nahrál program, ale lze jej poslat po síti pomocí B kanálu. Další funkcí je správa uzlů pomocí DPA service. Komunikaci mezi počítačem a uzly zajišťuje modul, který je připojen do počítače a je do něj nahrán CATS. Tento modul ovšem ztrácí předchozí funkci uzlu nebo koordinátora, pokud ji měl.



Obrázek 8 vytvoření CATS

V DPA service lze pracovat vždy s jedním uzlem. Uzel lze odvázat od sítě nebo přivázat a lze do něj nahrát konfiguraci anebo zjistit informaci o jeho rozhraní.



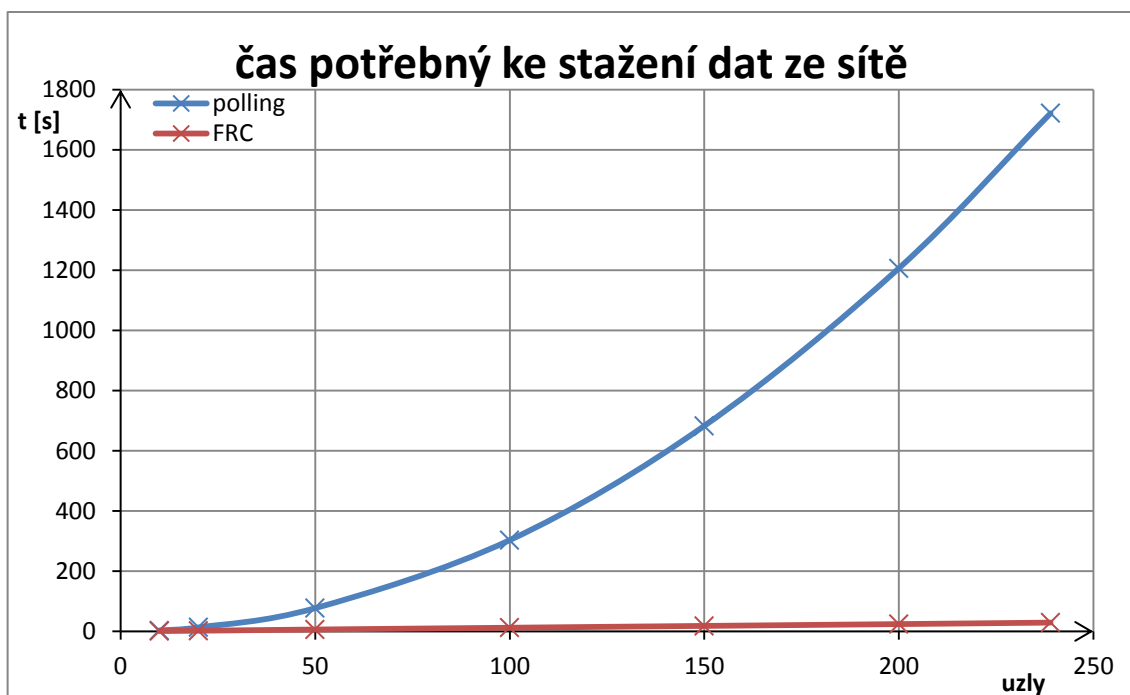
Obrázek 9 práce s uzlem v DPA Service

Uzel komunikuje s CATS zařízením na základě MID čísla, které je identické pro daný modul. CATS zjišťuje MID čísla od uzlů tak, že se uzel resetuje, například vysunutím a zasunutím do SIM konektoru. Připojit uzel se dá buď jako první, který je přístupný, což znamená ten, který se první resetoval, nebo dle zjištěných MID čísel při skenování.

3 FAST RESPONSE COMMANDS (FRC)

Při klasické komunikaci v síti pomocí pollingu se koordinátor postupně ptá každého uzlu zvlášť, což vede k exponenciálnímu růstu času pro šíření otázky v síti. Protože v síti s větším počtem uzlů je komunikace pomocí pollingu velice pomalá a prakticky nepoužitelná, používá se funkce FRC. Fast response commands se hodí používat tam, kde se klade pro celou síť stejná otázka a odpověď má malou velikost dat, například ano/ne. Graf 1 zobrazuje teoretickou dobu stahování odpovědi zpět koordinátorovi. [1] Výpočet doby poslání zpětného FRC paketu lze vypočítat dle rovnice č. 2.

$$t = 0,02 \cdot \text{svázané uzly} + 0,09 \cdot \text{objevené uzly} + 0,11 \text{ [s]} \quad (2) [4]$$



Graf 1 demonstrace jak je pomalý polling v síti s více uzly [1]

Žádost vždy může poslat jen koordinátor, který ji pošle do všech uzlů. K tomu slouží funkce `sendFRC(unsigned short)` nebo přes IQRF IDE: `_PNUM=0x0D` a číslo příkazu napsat do DATA. Otázka může mít velikost 2 až 30 bajtů. [6]

Vývojáři již předdefinovali pět FRC příkazů:

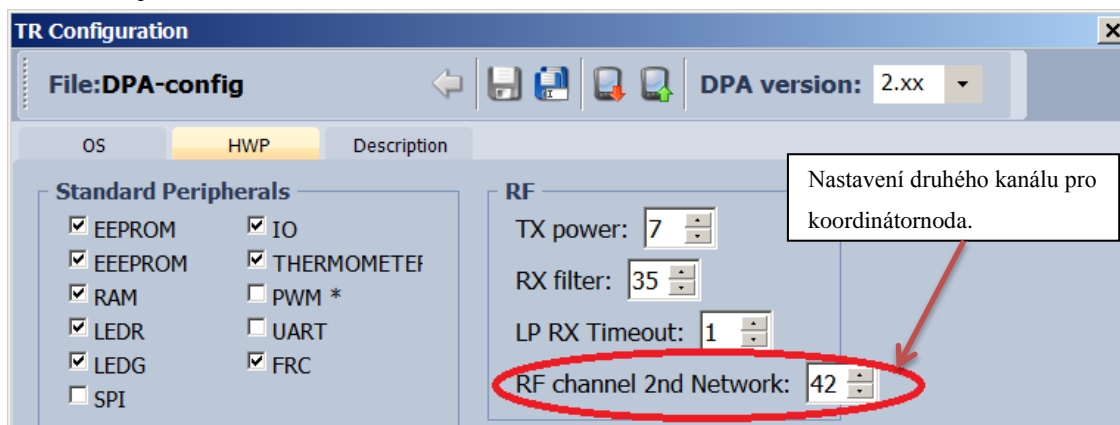
- 0x00 Prebonding
- 0x01 UART SPI data
- 0x02 Acknowledged broadcast- bits
- 0x80 Zjištění teploty
- 0x81 Acknowledged broadcast- bytes [6]

Paket s odpovědí postupuje zpět od nejnižšího uzlu (s nejvyšším směrovacím číslem) ke koordinátorovi a každý uzel do něj připisuje svou odpověď. Odpověď může být dvoubitová, a to maximálně od 239 uzlů nebo jednobajtová, ale jen od 62 uzlů. Dvoubitová odpověď se realizuje pomocí příkazu `responseFRCvalue.1`, příkazem `responseFRCvalue` se odešle jednobajtová odpověď. [1]

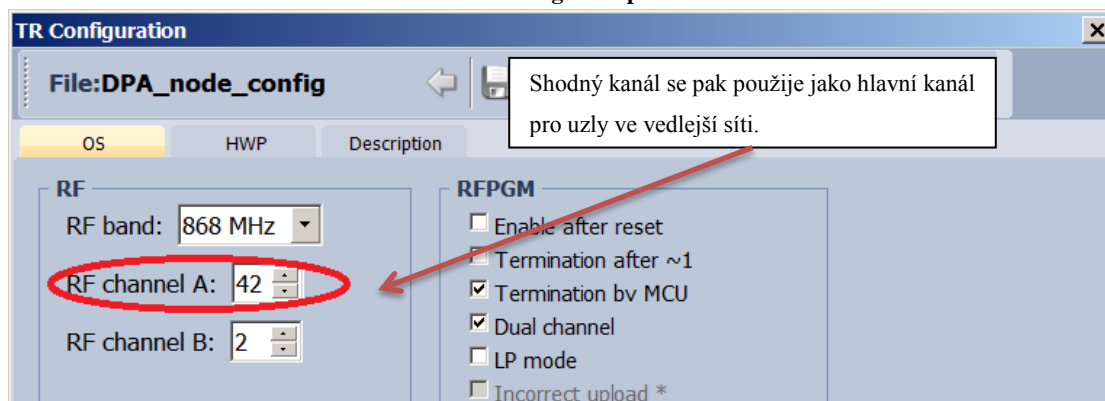
4 ASYNCHRONNÍ KOMUNIKACE

Asynchronní komunikace lze realizovat pouze v sub-sítích, kdy komunikaci vyvolává uzel sub-sítě, nikoliv koordinátor, jak je tomu při synchronní komunikaci. Asynchronní komunikace je vhodné použít tam, kde událost nenastává v periodickém čase, ale probíhá nepravidelně. Touto implementací se předpokládá úspora energie, neboť uzel je aktivní jen, když posílá zprávu a poté přejde opět do režimu spánku. Při synchronní komunikaci musí být uzel neustále v aktivním režimu, jelikož očekává žádost od koordinátora, na kterou mu pošle odpověď. Nevýhodou lze očekávat v tom, že rychlost komunikace od koordinátora do asynchronního uzlu bude o něco pomalejší než při synchronní realizaci, protože data musí nejdříve projít přes uzel v hlavní síti, který slouží jako koordinátor ve vedlejší podsíti. [1]

Sub síť lze vytvořit použitím druhého komunikačního kanálu RF channel 2nd Network u uzlu s hardwarovým profilem CoordinatorNode⁸. Neplést s RF channel B, který je pod záložkou OS, ten slouží k bezdrátovému programování. Číslo druhého kanálu, který je nastaven u koordinátornoda, se pak použije jako kanál A u uzlů, jenž tvoří vedlejší síť. [1]



Obrázek 10 TR konfigurace pro koordinátornoda

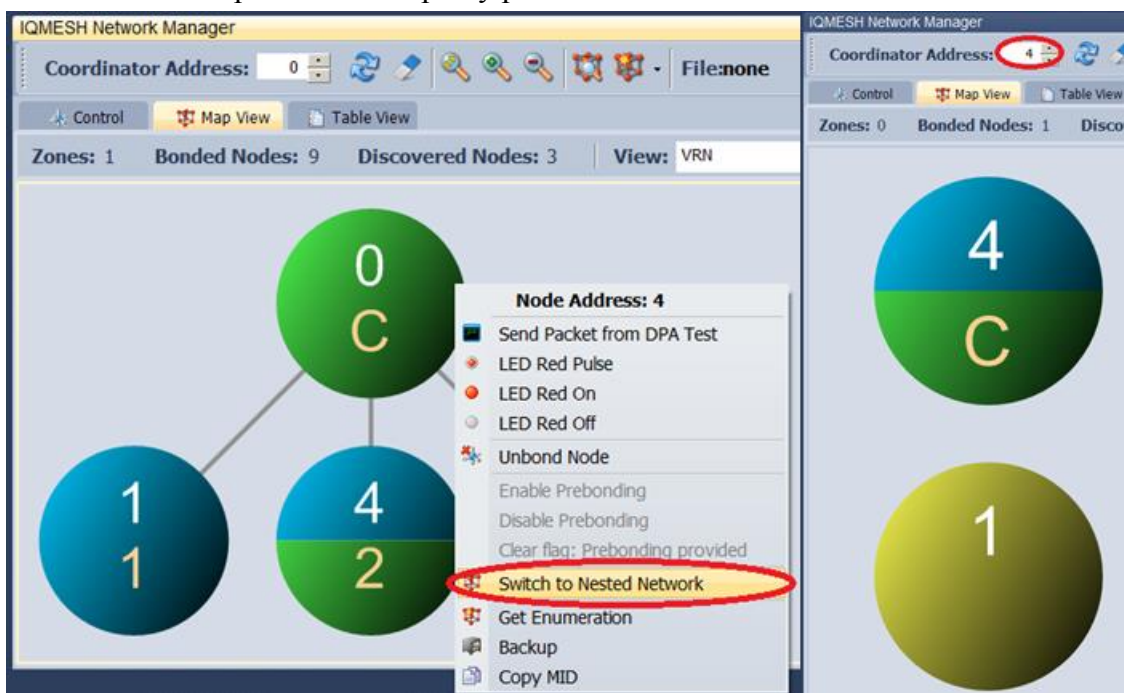


Obrázek 11 TR konfigurace pro uzel sub-sítě

⁸ CoordinatorNode je uzel, který se chová v hlavní síti jako klasický uzel, ale v sub-síti má vlastnosti koordinátora.

Například hlavní síť bude používat kanál A 52, dále tyto uzly mají nastaven RF channel 2nd Network 42. Pak uzly v podsíti používají jako svůj hlavní kanál A 42 a RF channel 2nd Network se zde nepoužívá, proto může být pak libovolný.

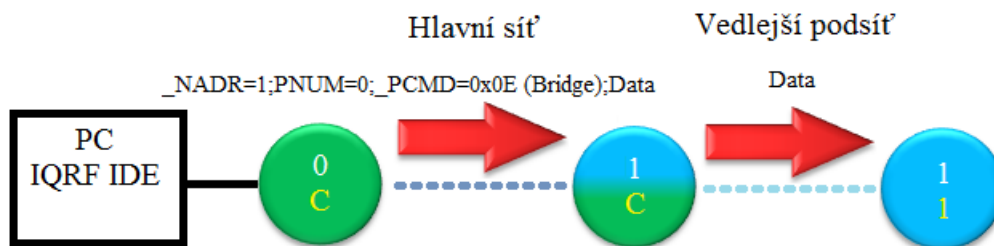
Pro přivázání uzlu ke koordinátoru se nejprve ve vývojovém prostředí musí přepnout do podsítě. K tomu stačí napsat v poli „Coordinator Address“ příslušné číslo uzlu, který má hardwarový profil CoordinatorNode. Pokud by to nefungovalo, je tu druhá možnost pro uzel, který má být koordinátor pro podsít, zjistíme jeho parametry pomocí „Get Enumeration“, tím se jeho původně modrá koule přebarví na zeleno modrou, jež značí, že se jedná o koordinátoruzel. Nyní se lze kliknutím na „Switch to Nested Network“ přesunout do správy podsítě.



Obrázek 12 přepnutí do podsítě

Dále jen stačí v „control“, při bondovat a udělat discovering, jak je popsáno v kapitole 2.4 a v kapitole 2.5.

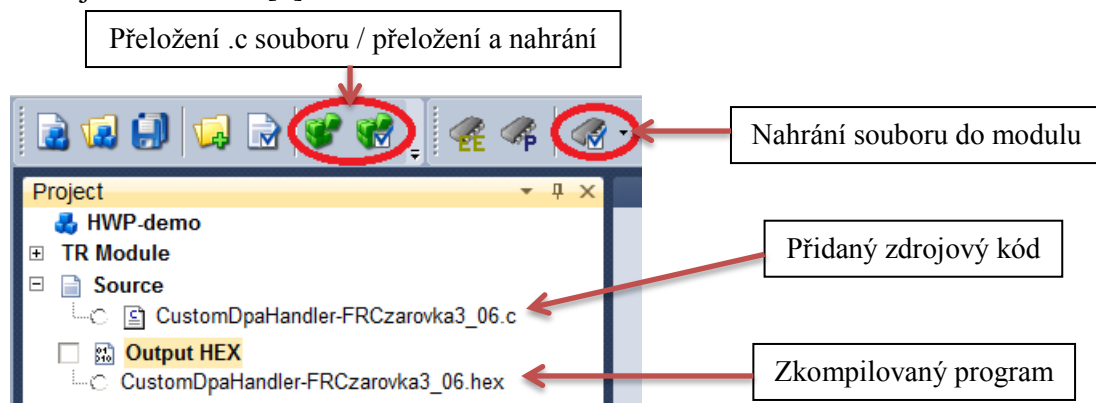
S uzlem z podsítě může hlavní koordinátor komunikovat pomocí DPA příkazu zvaný Bridge. Pokud se má příkaz poslat do vnořené sítě v podsíti, tak stačí pouze vepsat vnořený Bridge příkaz do dat Bridge, který je poslán do nadřazené podsítě. Jedná se pak ovšem o synchronní komunikaci.[6]



Obrázek 13 posílání příkazu do podsítě

5 POUŽITÍ DPA CUSTOM HANDLER

Vlastní programování se provádí v jazyce C. Zdrojové kódy se překládají ve vývojovém prostředí IQRF IDE, které vytvoří zkompileovaný soubor s příponou *.hex, který se pak nahraje do modulu. [1]



Obrázek 14 nahrání programu do modulu

Pokud se při vytvoření projektu povolilo DPA a povolilo se jeho užití v modulech, tak lze mikrokontrolér programovat na základě událostí, které na modulu mohou nastat. Zřejmě nejvíce používané rutiny, které jsou použity i v této bakalářské práci jsou popsány v následujících podkapitolách. Některé z nich ovšem lze použít jen pro General hardwarový profil, jenž lze nahrát pouze do DCTR modulů. [1]

K programování lze použít předpřipravené šablony, které obsahují již předpřipravené rutiny pro události.

5.1 DpaEvent_Init

Tato událost je volána pouze jednou, a to při spuštění modulu. Obvykle se zde inicializují proměnné nebo nastavují časovače, či nastavují vstupy a výstupy na počáteční podmínky. [6]

5.2 DpaEvent_Idle

Jedná se o hlavní událost, která je vykonávána cyklicky, pokud zrovna nenastane jiná událost. Až se jiná událost ukončí, začne se tato část programu znovu vykonávat. Tato událost se využívá pro zjišťování stavů na vstupech k obvolávání uzlů koordinátorem nebo k volání funkce spánku. [6]

5.3 DpaEvent_Interrupt

Událost přerušení lze například použít pro časovače.

5.4 DpaEvent_ReceiveDpaRequest

Událost se spustí v případě, když přišla k uzlu nebo ke koordinátorovi DPA žádost. Po vykonání se vrací odpověď `true`. Tato rutina se využívá například pro vyhodnocení příkazu poslaného k uzlu klasickým pollingem. [6]

5.5 DpaEvent_FrcValue

Událost slouží pro přijetí FRC příkazu uzly a aktivuje se v případě, pokud přišel FRC příkaz. FRC příkazy jsou rozlišovány přepínačem, jenž má vstupní parametr proměnnou `MFRW1`, která obsahuje číslo obdrženého FRC příkazu. V jednotlivých větveních se pak vykonávají instrukce a posílá odpověď koordinátorovi na daný FRC příkaz. V těle FRC události mohou být napsány instrukce i mimo větvení FRC příkazů, ty se pak budou vykonávat vždy při volání FRC události. Tato rutina se používá pouze u uzlů. [6]

5.6 DpaEvent_BeforeSendingDpaResponse

Tato událost se volá v případě, že je připravena odpověď, která má být vrácena do zařízení. Rutinu lze využít například pro vyhodnocení obdržených FRC odpovědích, co byly poslány zpět koordinátorovi. [6]

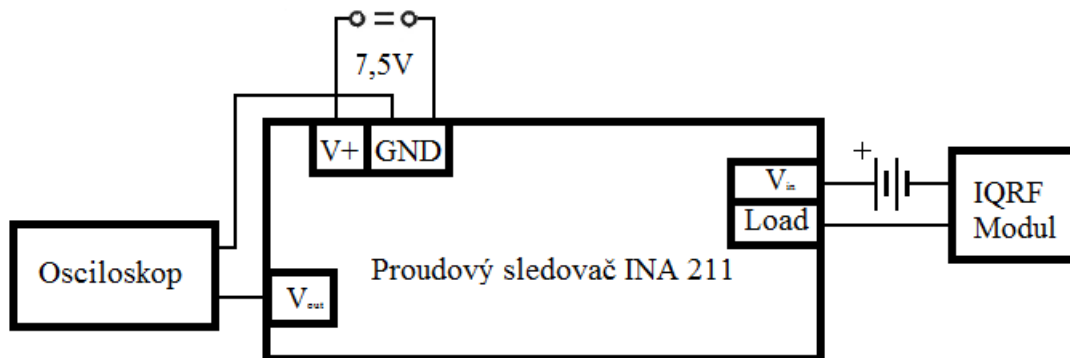
5.7 DpaEvent_ReceiveDpaResponse

Událost se využívá pro vyhodnocení DPA odpovědí od uzlů. Lze ji používat jen u koordinátora, popřípadě u koordinátornoda. Rutinu nelze využívat pro vyhodnocování FRC odpovědí, protože původní příkaz byl nejprve poslán koordinátorovi, aby poslal FRC příkaz uzlům. [6]

6 SBĚR DAT PŘI RŮZNÝCH MOŽNOSTECH IQRF

V bakalářské práci jsou porovnávány celkem tři synchronní možnosti komunikace za použití FRC funkce a jedno asynchronní provedení. Při prvním provedení synchronní komunikace je uzel neustále na příjmu, druhou možností je použití nízko příkonového módu a při třetí poslední variantě se ještě přidala doba spánku po zaslání odpovědi koordinátorovi. U těchto všech zmíněných variant provedení komunikace se měřil proud, který uzly odebíraly z napájecího zdroje. K měření proudu v časové závislosti se použila sestava osciloskop a proudový sledovač, pomocí něhož se měnil proud na napětí. Bez časové závislosti se pak měřil proud digitálním multimetrem Agilent 34401a. Z výsledných průběhů lze pak vidět, zda daná varianta bude či nebude energeticky úsporná, a jak dlouho trvá vlastní vysílání uzlu. Přesná spotřeba je pak měřena v kapitole č. 7.

Průběh proudu odebíraný uzlem byl měřen proudovým sledovačem INA211, který má zesílení 500V/V. Výstupem sledovače bylo pak napětí, které bylo v čase zobrazováno osciloskopem. Napájecí napětí mezi svorkami V+ a GND sledovače bylo zvoleno na 7,5V, výrobce udává, že hodnota může být nastavena mezi 2,7 V až 26 V. Sledovaný proud z baterie, která napájí modul, vstupoval do svorky V_{in} a vystupoval na svorce Load. Výstupní napětí se pak měřilo na svorce V_{out}. Toto popsané zapojení je určeno pro měření s bočním odporem R=0,1 Ω. [8]



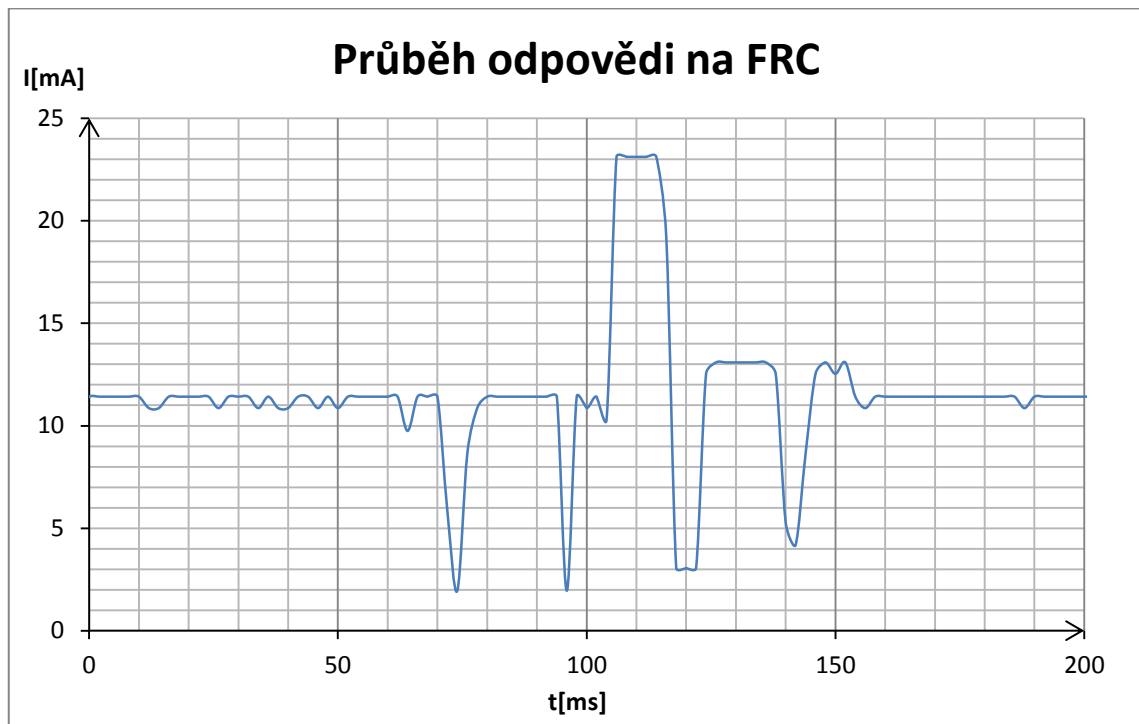
Obrázek 15 blokové schéma zapojení sledovače INA 211

Odebíraný proud se dále měřil digitálním multimetrem Agilent 34401a v jednotlivých stavech komunikačního uzlu. Tyto hodnoty pak posloužily, aby se dal průběh napětí zobrazovaný na osciloskopu převést na odpovídající průběh proudu.

6.1 Průběh odbíraného proudu při funkci FRC

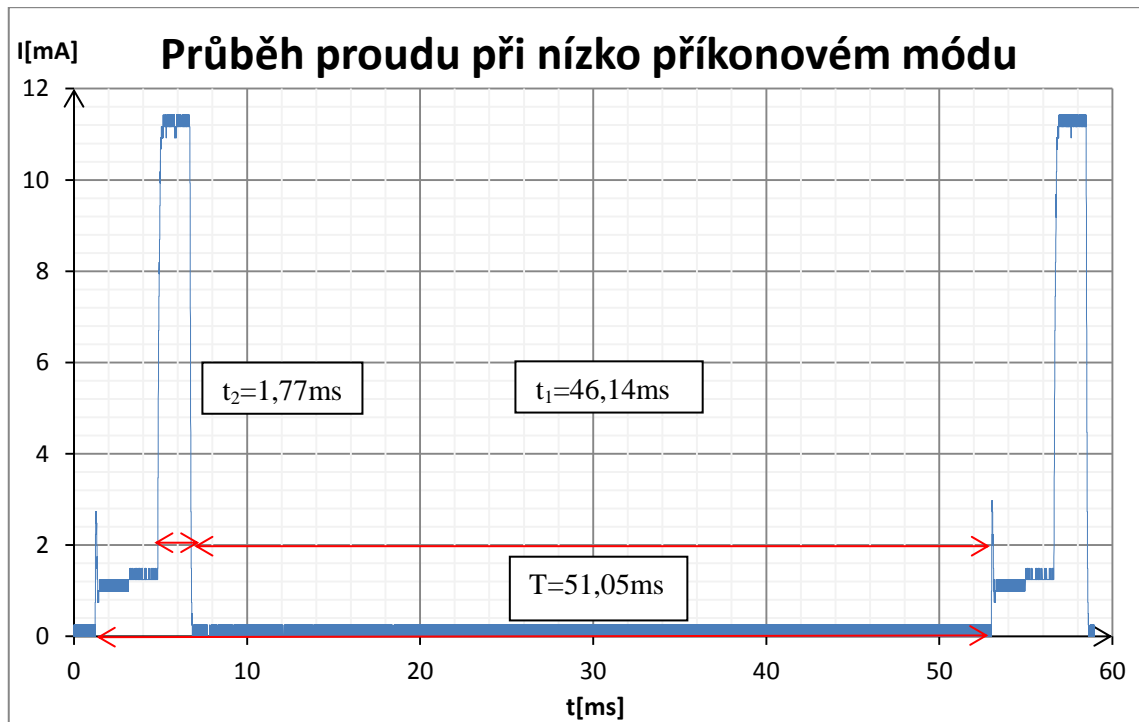
Komunikace mezi koordinátorem a koncovými uzly byla provedena ve třech možných synchronních provedeních, a to vždy za použití funkce FRC.

Při prvním provedení komunikace byly moduly stále na příjmu. V neustálém aktivním režimu modul odebíral z napájecí baterie proud 11,42 mA. Při odpovědi na dotaz koordinátora se proud zvýšil na 23,12 mA po dobu 14 ms.



Graf 2 průběh odpovědi na FRC ve standardním režimu

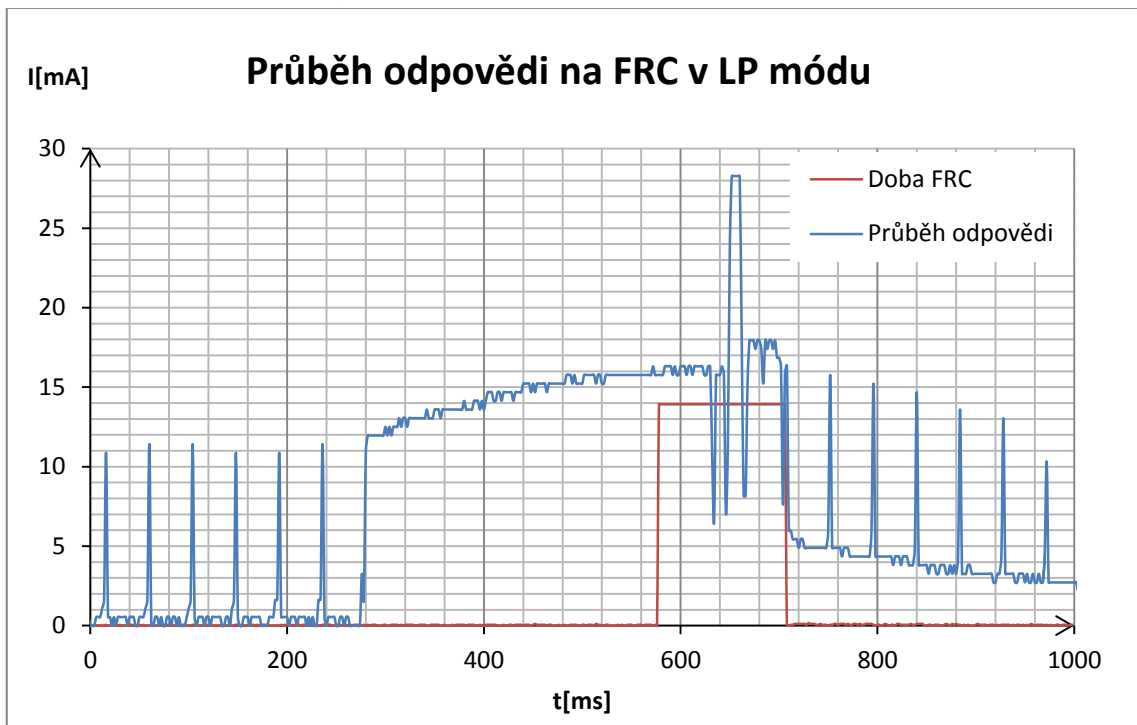
Druhou možností bylo nastavení nízko příkonového módu v DPA konfiguraci pro uzly. Díky tomu se u uzlů periodicky mění doba spánku a kdy je modul na příjmu.



Graf 3 příjem v LP módu

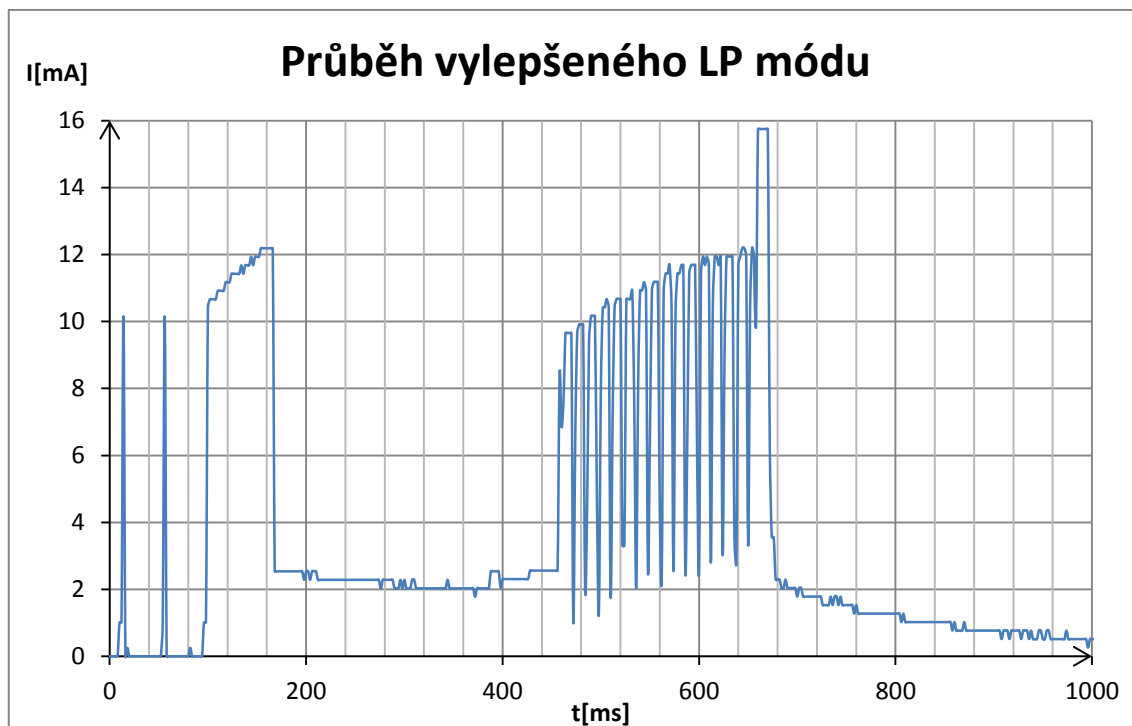
Uzel je na příjmu po dobu 1,77 ms, při němž odebírá proud 11,42 mA. Znovu na příjmu bude až za 51,05 ms. Během této doby komunikační modul přejde do spánkového režimu a odebírá dle údajů od výrobce 380 nA [3]. V nízko příkonovém módu je tudíž v aktivním režimu pouze 3,47% z celkového času provozu, a doba kdy modul šetří energii činí 90,38%. Střední hodnota proudu činila 0,480 mA. Přibližný celkový efektivní proud při nízko příkonovém provedení je dán tímto výpočtem:

$$I_{ef} = \sqrt{\frac{1,24^2 \cdot 3,14 + 11,42^2 \cdot 1,77 + (0,38^{-6})^2 \cdot 46,14}{51,05}} = 2,15 \text{ mA}$$



Graf 4 průběh odpovědi na FRC v nízko příkonovém módu

V Graf 4 zobrazující průběh odpovědi na FRC žádost v nízko příkonovém módu. Červený puls zobrazuje dobu jak dlouho v `case DpaEvent_FrcValue` probíhal příkaz `responseFRCvalue.1`, což je vlastní vysílání odpovědi koordinátorovi Tato doba činí 128 ms, avšak celková doba v aktivním režimu je 432 ms. To vede ke zbytečně vyšší spotřebě energie jak u nízko příkonového módu, tak i u třetího provedení se spánkovým režimem, než se očekávalo. Na tento nedostatek byla touto bakalářskou prací firma upozorněna, následně firma poskytla k proměření zkušební verzi s opravenou spotřebou energie při zpracování FRC příkazu.



Graf 5 průběh odpovědi na FRC ve vylepšeném nízko příkonovém módu

Třetí varianta byla provedena, tak že po odeslání zprávy se uzel uspal na dobu, než se bude očekávat další žádost od koordinátora. U této varianty lze očekávat nejnižší spotřebu ze zde uvedených tří variant, neboť koncový uzel je aktivní pouze po dobu příjmu žádosti a po dobu nezbytně nutnou k vykonání příkazu a odeslání k odpovědi. Čas určený ke spánku se nastavil na konci `case DpaEvent_FrcValue` do celočíselné proměnné `sleepTime` typu `integer`. V `case DpaEvent_Idle` se pak kontrolovalo, zda má modul přejít do spánku, pokud ano, je volána funkce `gosleep(sleepTime)`, která uspí modul na dobu 2,097 s vynásobenou hodnotou proměnné `sleepTime`.

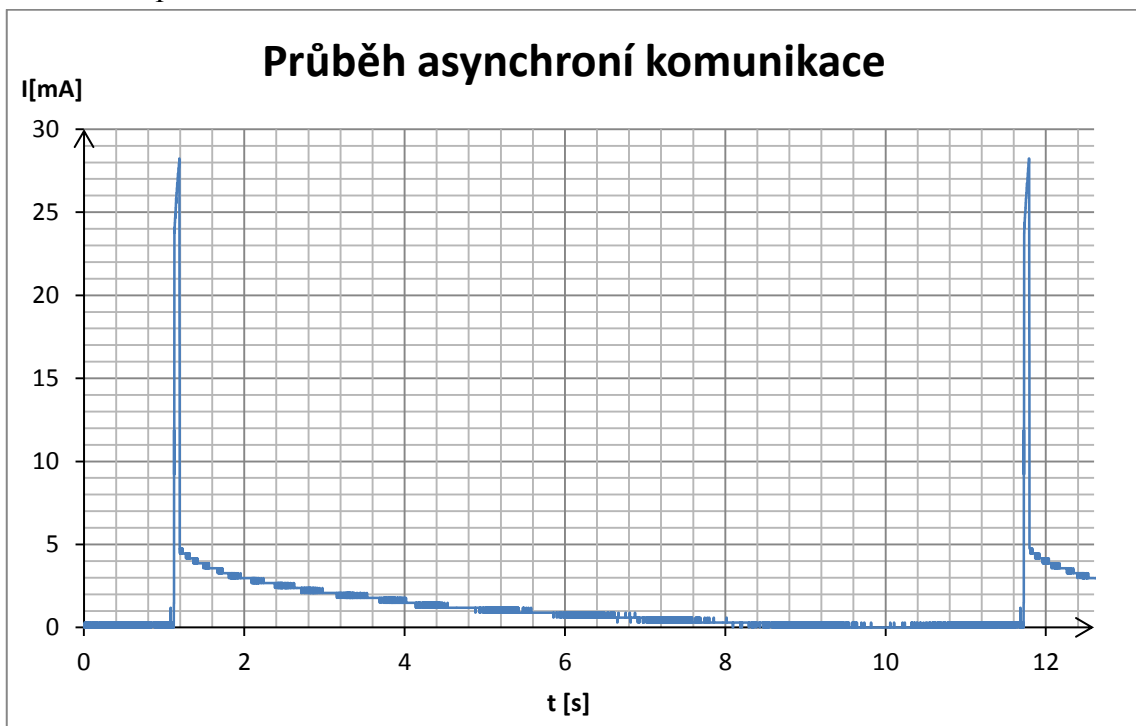
Přesnost doby `sleepTime` byla měřena při nastavení na velikost 150, takže by uzel měl teoreticky být v režimu spánku po dobu 314,550 s. Jelikož vnitřní hodiny modulu, tvoří frekvenční krystal, který by mohl různě kmitat při různých teplotách, byla tato nastavená doba spánku měřena při třech teplotách, při 18 °C, 24 °C a 33 °C. K měření teploty vzduchu byl použit modul TR52A, jenž má v sobě zabudovaný teploměr a díky předdefinovanému FRC příkazu na zjištění teploty se získávala data. Naměřené výsledky a odchylku od teoretické doby spánku zobrazuje Tabulka 1

t[°C]	Čas spánku [s]	Δ Čas spánku [s]	δ Čas spánku [%]
18	314,688	0,138	0,034
24	314,658	0,108	0,034
33	314,656	0,106	0,044

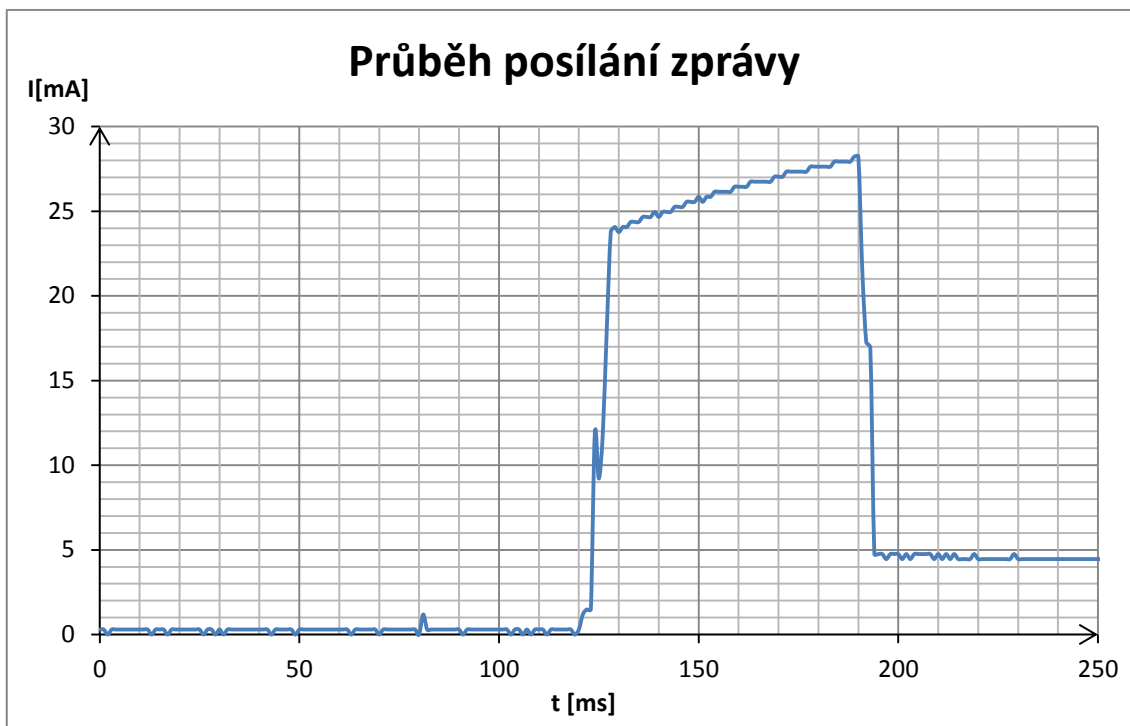
Tabulka 1 změřená doba spánku

6.2 Průběh odbíraného proudu při Asynchronní komunikaci

Jelikož při použití funkce FRC je uzel v aktivním režimu po dobu 350 ms než začne vysílat odpověď koordinátorovi, a tudíž se zcela nevyužije efektivita šetření energie pomocí spánku nebo nízkého příkonového módu. Byl proměřen průběh proudu i při možnosti posílání dat o stavu na koncovém uzlu koordinátorovi pomocí asynchronní komunikace. Při tomto provedení byly uzly ve spánkovém režimu, a jakmile uplynula nastavená doba spánku, uzly se probudily a vyslaly signál koordinátorovi a přešly opět do režimu spánku.



Graf 6 průběh proudu při nastavení doby spánku na 10s



Graf 7 průběh posílání zprávy koordinátorovi

Doba kdy uzel vysílá se tak zkrátila na pouhých 71 ms, a tím se dá předpokládat i výrazné snížení spotřeby energie v porovnání použitím FRC funkce.

7 MĚŘENÍ SPOTŘEBY ENERGIE

Spotřeba energie byla porovnáována mezi třemi možnostmi provedení, jak odpovídat na žádosti od koordinátora, které budou přicházet v pravidelných časových intervalech.

- FRC příkazy (FRC)
- FRC příkazy s použitím nízko příkonového módu (LP)
- FRC příkazy s následným uspaním, než se bude očekávat další dotaz (Sleep)
- Asynchronní komunikace s následným uspaním do dalšího vysílání (Asyn.)

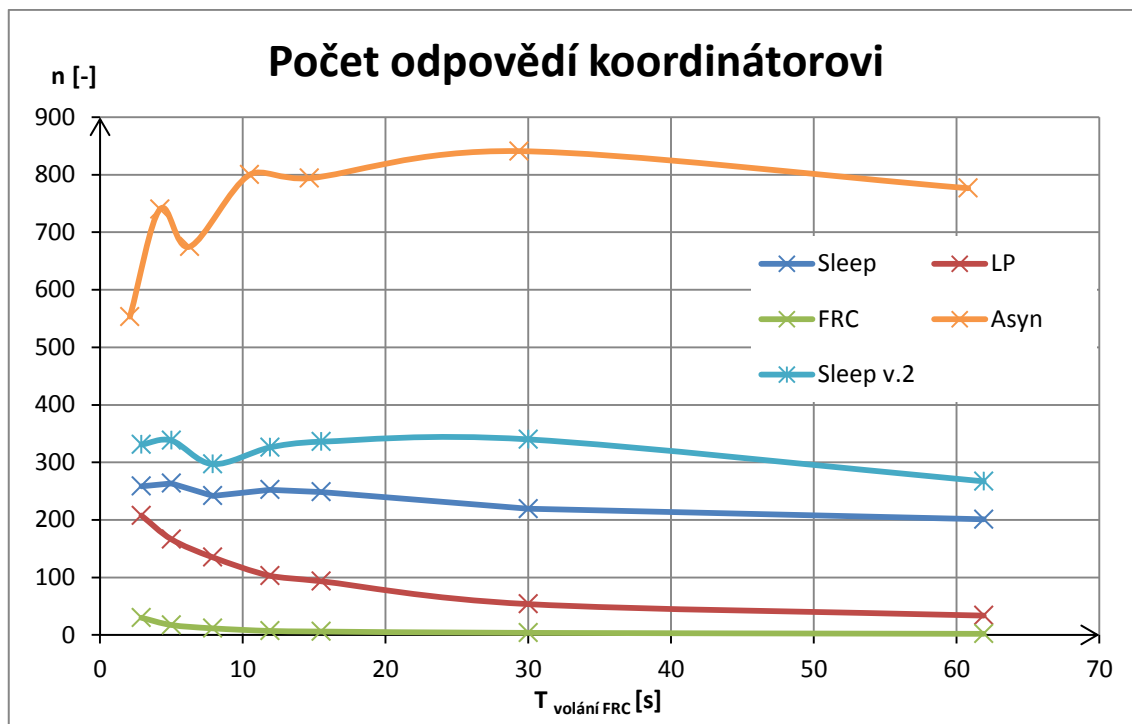
Uzly byly napájeny vysoce kapacitními kondenzátory o kapacitě $C=0,47$ F. Kondenzátory byly nabíjeny na maximální provozní napětí pro moduly tj. 3,4 V. Dále byla měřena doba, dokdy uzly odpovídaly ještě koordinátorovi na jeho dotazy. Taktéž se počítalo, kolikrát uzel za danou dobu odpověděl.

Měření bylo prováděno při několika různých periodách opakování dotazů od koordinátora, viz Tabulka 2. U asynchronní komunikace se zpráva koordinátorovi posílala periodicky dle celočíselných násobků doby pro spaní funkce `gosleep(int)`.

Každý uzel byl proměřen třikrát a výsledek následně zprůměrován, aby se předešlo vlivu odlišně nabitých kondenzátorů.

$T_{FRC}[s]$	2,900	5,000	7,900	11,900	15,500	30,000	61,900
FRC	30	17	12	7	6	4	2
LP	208	166	135	103	93	54	34
Sleep	259	264	242	253	249	219	201
Sleep v. 2	331	338	297	326	336	340	267
$T_{asyn}[s]$	2,097	4,1942	6,291	10,485	14,679	29,358	60,813
Asyn.	553	740	675	800	794	841	777

Tabulka 2 závislost počtu odpovědí na periodě dotazů



Graf 8 závislost počtu odpovědí na periodě dotazů

Nabitý kondenzátor měl napětí od 3,21 V, až po 3,31 V. Měřením bylo zjištěno, že modul pracuje do napájecího napětí 1,84 V. Odebraný příkon komunikačního uzlu byl pak stanoven jako podíl rozdílu energie na kondenzátoru mezi počátkem měření a koncem funkčnosti uzlu, viz rovnice č. 5 a změřenou dobu funkčnosti uzlu, viz Tabulka 4. Vypočítaný příkon zobrazuje Tabulka 3.

$$E_1 = \frac{1}{2} C U_1^2 \quad (3)$$

$$E_2 = \frac{1}{2} C U_2^2 \quad (4)$$

$$\Delta E = E_1 - E_2 \quad (5)$$

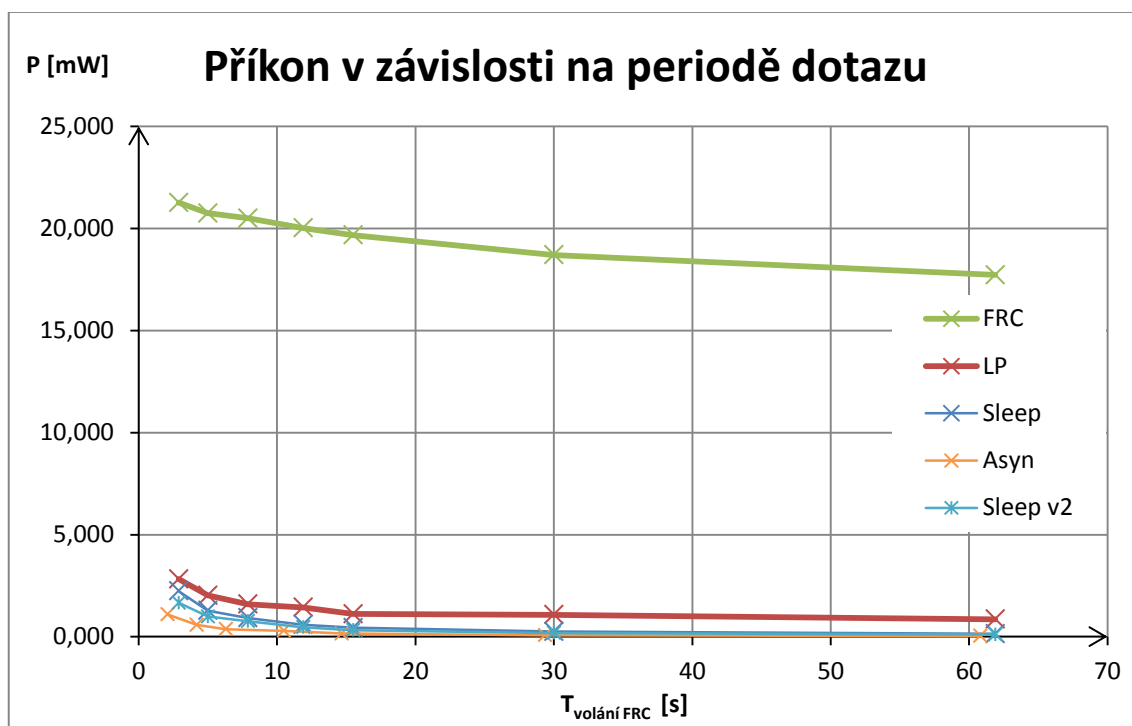
$$P = \frac{\Delta E}{t} \quad (6)$$

Kde:

- P – příkon uzlu [W]
- t – doba fungování uzlu [s]
- E₁ – počáteční energie kondenzátoru [J]
- E₂ – energie kondenzátoru v době, kdy uzel fungoval
- U₁ – počáteční napětí na kondenzátoru [V]
- U₂ – napětí na kondenzátoru, při němž uzel přestal fungovat

	T [s]	2,900	5,000	7,900	11,900	15,500	30,000	61,900
P[mW]	FRC	21,273	20,755	20,504	20,022	19,675	18,702	17,728
	LP	2,827	2,026	1,597	1,439	1,115	1,074	0,856
	Sleep	2,235	1,283	0,915	0,590	0,442	0,253	0,141
	Sleep v. 2	1,667	0,991	0,765	0,462	0,325	0,183	0,108
	T_{asyn}[s]	2,097	4,1942	6,291	10,485	14,679	29,358	60,813
	Asyn.	1,088	0,585	0,371	0,294	0,156	0,074	0,037

Tabulka 3 příkon uzlu v závislosti na periodě dotazů



Graf 9 závislost příkonu uzlu na periodě dotazů

	T [s]	2,900	5,000	7,900	11,900	15,50	30,000	61,900
t[min]	FRC	1,333	1,367	1,383	1,417	1,442	1,517	1,600
	LP	10,033	14,000	17,767	19,708	25,433	26,406	33,150
	Sleep	12,692	22,108	30,983	48,067	64,192	111,994	200,478
	Sleep v. 2	16,867	29,650	39,105	64,712	88,150	150,067	274,304
	T_{asyn}[s]	2,097	4,1942	6,291	10,485	14,679	29,358	60,813
	Asyn.	27,406	51,300	79,967	101,500	190,517	410,358	787,700

Tabulka 4 fungování uzlu při napájení super kondenzátorem

Z vypočítaných hodnot příkonu v Tabulka 3 se následně vypočítalo, kolik watt hodin spotřebuje uzel při ročním provozu, viz Tabulka 5. Z tohoto údaje se určilo, kolik let bude fungovat uzel, když bude napájen klasicky dvěma tužkovými NiMH akumulátory s kapacitou 2050 mAh. Pro lepší přehlednost Tabulka 6 zobrazuje i počet dnů pro jednotlivá provedení komunikace.

$$E_{IQRF} = \frac{P \cdot t_{rok}}{3600} \quad (7)$$

$$E_{baterie} = 2 \cdot Q \cdot U \quad (8)$$

$$n_{roků} = \frac{E_{baterie}}{E_{IQRF}} \quad (9)$$

Kde:

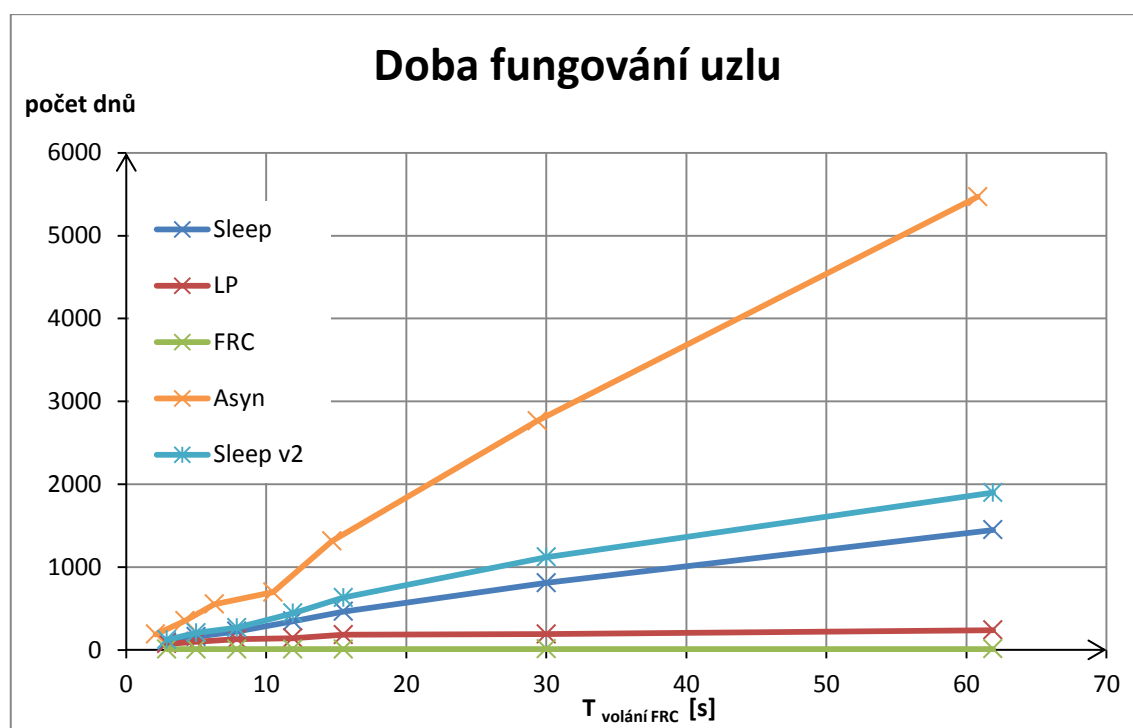
- E_{IQRF} – spotřebovaná energie uzlu za jeden rok [Wh]
- $E_{baterie}$ – energie dvou akumulátorů [Wh]
- P – příkon uzlu [W]
- t_{rok} – doba jednoho roku [s]
- Q – náboj akumulátoru [mAh]
- U – jmenovité napětí akumulátoru [V]

	T [s]	2,900	5,000	7,900	11,900	15,500	30,000	61,900
E[Wh]	FRC	186,355	181,810	179,619	175,393	172,351	163,828	155,296
	LP	24,765	17,748	13,985	12,608	9,770	9,410	7,495
	Sleep	19,578	11,239	8,020	5,169	3,871	2,219	1,239
	Sleep v.2	14,599	8,684	6,700	4,049	2,844	1,602	0,946
	T_{asyn}[s]	2,097	4,1942	6,291	10,485	14,679	29,358	60,813
	Asyn.	9,533	5,122	3,248	2,574	1,367	0,650	0,328

Tabulka 5 roční spotřeba uzlů

	T [s]	2,900	5,000	7,900	11,900	15,500	30,000	61,900
Počet let	FRC	0,0264	0,0271	0,0274	0,0281	0,0285	0,0300	0,0317
	LP	0,1987	0,2772	0,3518	0,3902	0,5036	0,5229	0,6564
	Sleep	0,2513	0,4278	0,6135	0,9518	1,2711	2,2176	3,9696
	Sleep v. 2	0,3370	0,5666	0,7343	1,2151	1,7299	3,0705	5,2033
	T_{asyn}[s]	2,097	4,1942	6,291	10,485	14,679	29,358	60,813
	Asyn.	0,5161	0,9605	1,5147	1,9114	3,5982	7,5733	14,9859
Počet dnů	FRC	9,64	9,88	10,00	10,24	10,42	10,96	11,56
	LP	72,51	101,18	128,41	142,44	183,82	190,84	239,59
	Sleep	91,73	159,78	223,93	347,39	463,94	809,42	1448,92
	Sleep v. 2	123,01	206,80	268,01	443,52	631,40	1120,72	1899,21
	T_{asyn}[s]	2,097	4,1942	6,291	10,485	14,679	29,358	60,813
	Asyn.	188,38	350,58	552,87	697,67	1313,35	2764,26	5469,85

Tabulka 6 jak dlouho bude fungovat uzel při na pájení dvěma AA bateriemi



Graf 10 kolik dní bude uzel fungovat

8 MĚŘENÍ LATENCE

Měření latence se provádělo mezi koordinátorem a k němu připojeným uzlům, které byly vždy připojeny v další zóně. Zón bylo na začátku celkem deset, tudíž i deset hopů, postupně byly uzly odebírány ze sítě, a tím pádem se snižoval i počet hopů.

Měření pro FRC příkazy se provádělo tak, že po odeslání z počítače DPA příkazu na FRC funkci koordinátorovi se ihned v handleru `case DpaEvent_ReceiveDpaRequest` sepnul koordinátorův výstupní pin C2, jehož náběžnou hranu zaznamenal osciloskop. Až packet nesoucí FRC žádost dorazil k uzlu v poslední zóně, vracel se packet s odpovědí zpátky koordinátorovi přes všechny uzly v síti. Jakmile koordinátor obdržel odpověď v handleru `case DpaEvent_BeforeSendingDpaResponse`, přešel výstupní pin C2 do stavu `false`. Doba latence se pak určila jako polovina z měřeného času mezi náběžnou a sestupnou hranou, jelikož se předpokládá, že paket vyslaný koordinátorem půjde stejnou rychlostí zpět. Teoretické doby latence se určily výpočtem z rovnice č. 2 pro výpočet doby poslání paketu zpět koordinátorovi ze strany 18 v kapitole 3.

Pro měření latence pro polling se výstupní pin C2 na koordinátorovi nastavil do stavu `true` poté, co se zpráva odeslal prvním uzlu. Dále se osciloskopem měřila náběžná hrana při rozsvícení žárovky na posledním uzlu, který byl obvolán.

Počet hopů	Doba latence FRC τ [ms]	Teoretická doba latence FRC τ [ms]	δ doba latence FRC[%]	Doba latence pollingu τ [ms]
1	533	220	58,7	288
2	547	330	39,7	756
3	567	440	22,4	1384
4	642	550	14,3	2174
5	718	660	8,1	3112
6	798	770	3,5	4208
7	908	880	3,1	5480
8	1012	990	2,2	6912
9	1119	1100	1,7	8480
10	1223	1210	1,1	10224

Tabulka 7 závislost latence na počtu hopů

Z Tabulka 7 je zřejmé, že polling nelze prakticky použít pro komunikaci s více uzly. Při jednom hopu polling trvá kratší dobu zejména proto, že při FRC komunikaci koordinátor nejprve musel vytvořit packet, do kterého budou uzly zapisovat svou odpověď. Jelikož i při pollingu byl uzel v aktivním režimu, než odvyšlal odpověď

koordinátorovi podobně jak při FRC, nemělo smysl se touto variantou komunikace nikterak dále zabývat a nebyla proměřena její spotřeba energie.

Skutečná doba latence je vzata od doby poslání příkazu z vývojového prostředí IQRF IDE až po rozsvícení žárovky. Teoretická doba latence zahrnuje pouze vypočítanou dobu pro přenos paketu v síti mezi uzly. Odchylka od teoretické doby latence u nižšího počtu hopů je způsobena tím, že do teoretického výpočtu není zahrnuta doba vytvoření paketu, který koordinátor nejdříve musí vytvořit. S narůstajícím počtem skoků je tato doba zanedbatelná. Rovnice z kapitoly 3, od které se teoretická doba latence odvozovala, ve skutečnosti nepočítá dobu poslání paketu uzlům, ale zpět. Avšak předpokládá se, že rychlost paketu bude oběma směry shodná.

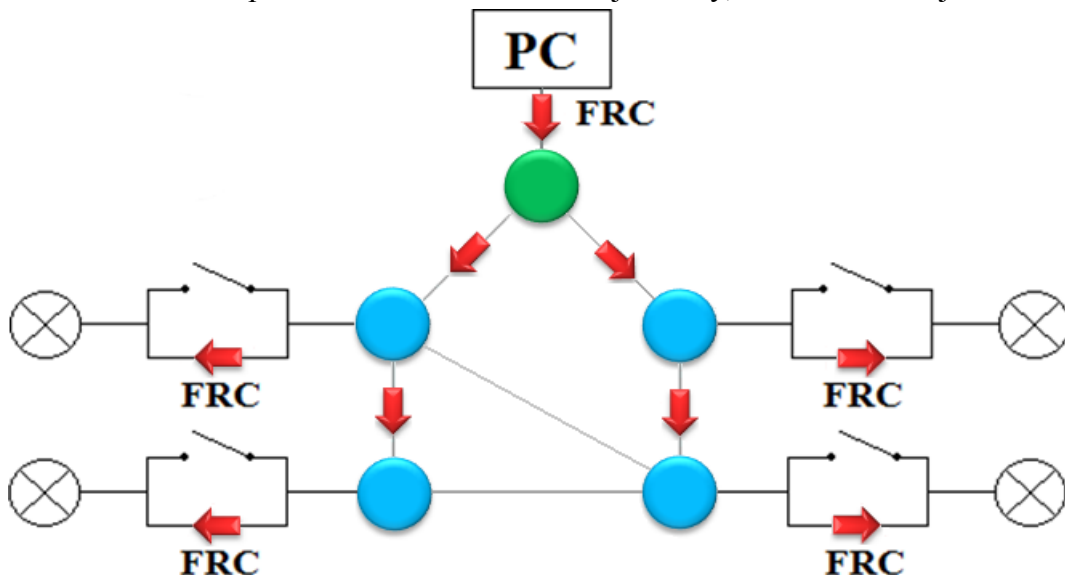
Při asynchronní komunikaci, byl uzel přivázán ke koordinátorovi nebo ke koordinátorovi, tudíž vždy to byl jeden hop. Doba latence zde byla 253 ms.

9 REALIZACE APLIKACE

Pro demonstraci sběru dat v síti byl zvolen model ovládání osvětlení pomocí tlačítka nebo z počítače přes koordinátora. Tato úloha byla řešena dvojitým způsobem komunikace, nejdříve synchronně a poté asynchronně.

9.1 Synchronní sběr dat

Přes koordinátora (Obrázek 16 zelený bod) lze posílat na uzly (Obrázek 16 modré body) dotazy, zda je žárovka či tlačítko sepnuto, dále příkazy na zapnutí či vypnutí žárovky. Žárovka může být ovládána i tlačítkem, které je přímo na uzlu. Komunikace mezi koordinátorem a uzly je provedena pomocí funkce Fast response command. Toto řešení se hodí pro systémy s centrálním řízením, jako je například pouliční osvětlení, které se celé rozsvítí na povel od centrální řídicí jednotky, v modelu to je koordinátor.



Obrázek 16 realizace ovládání žárovky při synchronním sběru dat

9.1.1 Algoritmus řešení:

Pro rozhodnutí zda žárovku rožnout nebo zhasnout slouží, `case DpaEvent_Idle`, v němž se nastavuje výstupní pin C2, ke kterému je připojena žárovka, do stavu `true` nebo `false`. Stav `true` může nastat jen v případě sepnutého tlačítka (vstupní pin C1), nebo když bude proměnná `zarovka = true`. Proměnná `zarovka` je ovládána FRC příkazy.

FRC příkazy jsou řešeny v `case DpaEvent_FrcValue`, kde se dále kód větví pro jednotlivé FRC příkazy:

- `case 0x08` Dotaz zda tlačítko je sepnuté

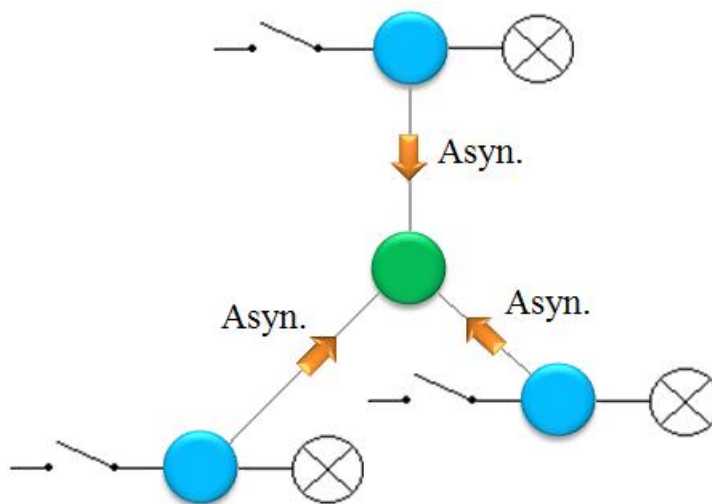
Odpověď koordinátorovi `responseFRCvalue.1=1` přijde jen v případě, když pin tlačítka C1 bude sepnutý (ve stavu `true`).

- `case 0x18` Dotaz zda svítí žárovka
Odpověď koordinátorovi `responseFRCvalue.1=1` přijde jen v případě, když pin C2 bude sepnutý (ve stavu `true`).
- `case 0x28` Příkaz na zapnutí žárovky
Proměnná `zarovka` se nastaví do stavu `true`.
- `case 0x38` Příkaz na vypnutí žárovky
Proměnná `zarovka` se nastaví do stavu `false`.

Ovládání FRC příkazů je prováděno pomocí vytvořených maker ve vývojovém prostředí IQRF IDE.

9.2 Asynchronní sběr dat

Při tomto způsobu sběru dat je komunikace zahájena od koncového uzlu ke koordinátorovi. Žárovka, která je připojena k uzlu, na němž je taktéž připojeno tlačítko, díky němuž lze žárovku rozsvěcovat nebo zhasínat. Uzel posílá koordinátorovi informaci o žárovce, v jakém stavu se žárovka nachází pouze tehdy, dojde-li ke změně stavu žárovky. Pokud uzel nevysílá, je v režimu spánku, a tudíž není ani na příjmu. Takovéto řešení v domácí automatizaci lze například aplikovat na monitorování automaticky se rozsvěčujících světel nebo bezpečnostní hlásiče.



Obrázek 17 realizace ovládání žárovky při asynchronním sběru dat

Pokud je potřeba ovládat více prvků než jeden koordinátor, lze k němu připojit v hlavní síti uzly s hardwarovým profilem `CoordinatorNode`, a ty jsou pak ovládány z uzlů ze svých podsítí. To lze aplikovat například na schodišťové osvětlení, kde žárovka je na koordinátornodu a tlačítka jsou na uzlech v podsíti.

9.2.1 Algoritmus řešení

Všechny události jsou řešeny v `case DpaEvent_Idle`, princip ovládní žárovky je zcela stejný jako při synchronní komunikaci (viz kapitola 9.1.1), akorát již nelze žárovku zapínat a vypínat pomocí FRC příkazů. Pokud dojde ke změně stavu žárovky, to znamená, že předtím svítila, a tudíž bylo tlačítko sepnuté a nyní je tlačítko rozepnuté a žárovka nesvítila. Z uzlu se pošle zpráva ke koordinátorovi dle příslušného stavu, ve kterém se žárovka právě nachází, a to pomocí bitového pole `_DpaMessage.Request.PData[]`. Stav, ve kterém se žárovka nachází, monitoruje proměnná `stav_zarovky`, jenž je typu `enum`. Dále proměnná `zmena_stavu` indikuje, zda došlo právě ke změně stavu. Tato proměnná je ve stavu `true`, například pokud bude sepnuté tlačítko (`PORTA.0=true`) a zároveň `stav_zarovky=NESVITI`. Na konci bloku `case DpaEvent_Idle` se nastavuje čas spánku pro funkci `goSleep(int)`. Čím je doba spánku delší, tak se snižuje doba odezvy žárovky na změnu tlačítka, neboť je modul ve spánku a nepracuje, ale zase se více ušetří energie.

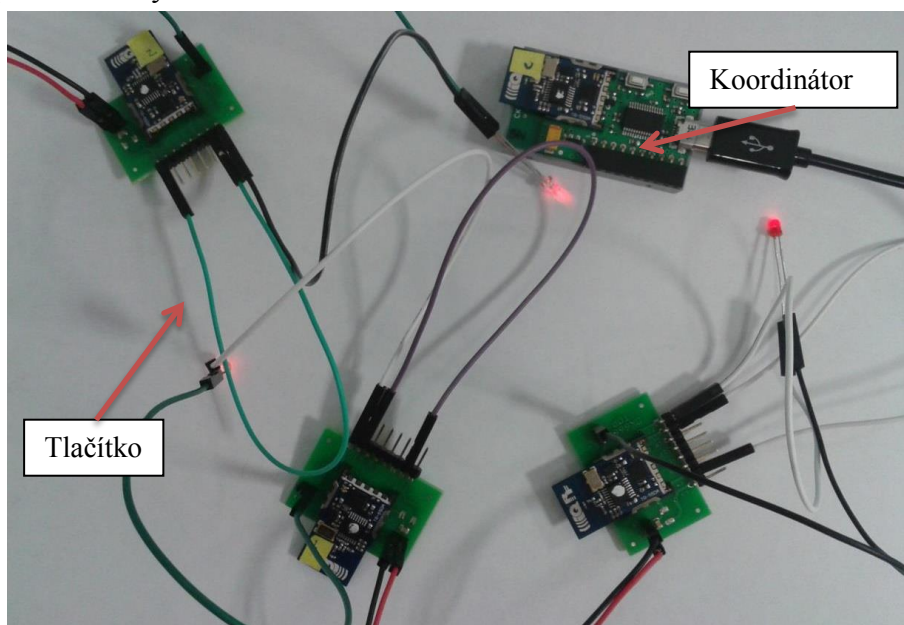
10 VYHODNOCENÍ VÝSLEDKŮ

V porovnání tří způsobů realizací synchronní komunikace vychází při porovnání spotřeby energie nejvýhodněji použití spánku s kombinací nízko příkonového módu. Při periodě opakování dotazu co 2,9 s uzel odebíral 2,235 mW, při takové spotřebě by vydržel na dvě AA baterie 91,73 dnů. Při snižování frekvence dotazů až do posílání žádosti co 61,9 s se příkon snížil na 0,141 mW, to by teoreticky dvojice AA baterie vystačily 3,97 let. Naopak standardní FRC komunikace měla téměř při periodě 2,9 s desetkrát a při periodě 61,9 s dokonce 125 krát větší spotřebu, a tudíž i uzly vydržely pouze zlomek času než při použití spánku. Jelikož tato doba činí pouze 9,64 - 11,56 dnů, není standardní FRC vhodné používat, místo něj lze použít nízko příkonový mód, kdy taktéž uzel nemůže zaspat žádost od koordinátora a zároveň se šetří energie. U LP módu je příkon jen zhruba 1,3 krát až 6 krát větší než při použití spánkového režimu v závislosti na periodě dotazů.

Při asynchronním řešení se dosáhlo ještě lepších výsledků. Příkon při době opakování dotazu každých 2,097 s činil 1,088 mW a životnost pak 188,38 dní. Při periodě dotazů 60,813 s příkon činil pouze 0,050 mW a životnost teoreticky až 14,99 let.

Dodatečným proměřením spotřeby energie při vylepšeném průběhu odpovědi FRC byla prokázána lepší životnost, a to zhruba 1,4 krát než předtím. Avšak stále úsporněji vychází asynchronní řešení.

Doba latence při použití FRC funkce činila pro jeden routovací skok 533 ms, dále pak se tato doba s navyšujícími skoky zvyšovala v řádech desítek milisekund. Při porovnání s dobou latence při pollingu, kde se sice při obvolání jednoho uzlu latence rovnala jen 288 ms, ale narůstala exponenciálně až na 10,224 s, je použití FRC jediným možným použitím u synchronních sítí.



Obrázek 18 síť ovládaných světel

11 ZÁVĚR

V bakalářské práci jsem se nejdříve zabýval, jak vytvořit síť pomocí vývojového prostředí IQRF IDE 4 a využitím DPA custom handleru. Nahrál jsem do jednoho modulu hardwarový profil pro koordinátora a ostatním modulům hardwarový profil pro uzly. Spojení uzlů s koordinátorem jsem dělal ve dvou krocích. V prvním kroku jsem přivázal všechny uzly ke koordinátorovi, posléze díky discoveringu, který přiřadil uzlům virtuální routovací čísla, jsem ověřil, zda uzly komunikují.

Pro takto vytvořenou síť jsem použil aplikaci k ovládní žárovky. Ovládní žárovky jsem nejprve řešil synchronně, kdy koordinátor posílal FRC příkazy k zapnutí nebo vypnutí žárovky, dále jím šlo zjišťovat, zda právně žárovka svítí nebo je zhasnutá. Žárovku bylo možné taktéž ovládat tlačítkem, které bylo na uzlu. Druhou možností ovládní žárovky bylo pomocí asynchronní komunikace.

Synchronní možnost komunikace byla dále rozšířena na tři varianty s cílem ušetřit energii, aby modulům déle vydržely baterie, jimiž jsou napájeny. Při použití holého FRC příkazu se vypočítaná životnost pohybuje v řádech několika dnů, s použitím nízko příkonového módu se životnost prodloužila na několik měsíců, a když se uzel bude i uspávat, lze dosáhnout životnosti i několik let. Doba životnosti se taktéž proměřovala v závislosti na periodě odpovídání koordinátorovi.

Asynchronní komunikace byla provedena tak, že se žárovka ovládala pouze tlačítkem a uzel poslal zprávu koordinátorovi v případě, že došlo na daném uzlu ke změně stavu žárovky, jinak byl uzel ve spánkovém režimu. Měření spotřeby bylo provedeno tak, aby korespondovalo s periodou dotazů od koordinátora při synchronním provedení, tudíž vždy po uplynutí nastavené doby spánku uzel poslal zprávu. Toto řešení bylo nejvíce úsporné z hlediska odebírané energie z baterie. Životnost při zasílání zprávy každou jednu minutu je teoreticky téměř 15 let.

Nicméně do výpočtu nebylo zahrnuto samovybíjení akumulátorů, jenž se pohybuje okolo 15% z kapacity akumulátoru za jeden rok, což se nejvíce projeví při spánkovém režimu. [9] Další vliv na spotřebu mělo to, že během měření nebyl na výstupy přiváděn logický signál, čímž by se zvýšila celková spotřeba.

Na závěr jsem provedl měření latence, při níž se signál od koordinátora musel dostat ke koncovému uzlu přes několik hopů. U FRC příkazu doba latence při jednom skoku činila 533 ms, při deseti již 1233 ms. V porovnání s pollingem, kde doba jednoho skoku byla 288 ms a při deseti až 10224 ms, jde jasně vidět, že pro síť s více uzly lze použít pouze FRC funkci.

Použitá literatura

- [1] SOS WEBINÁŘ , Jak udělat z vašeho produktu bezdrátové zařízení Jednoduše!
[video] *Youtube*[online]. 2014 [cit. 2014-10-18] Dostupné z
<https://www.youtube.com/watch?v=2LJb9IhXJY4#t=6335>
- [2] DPA QUICK START GUIDE V2.01. *IQRF DPA Quick Start Guide* [online]. 2014,
s. 8 [cit. 2014-12-03]. DOI: Quick_Start_Guide_DPA-200_140513. Dostupné z:
<http://www.iqrf.com/weben/downloads.php?id=352>
- [3] TR-55D datasheet. *TR-55D* [online]. 2014, s. 11 [cit. 2014-12-08]. DOI:
Datasheet_TR-55D_140430. Dostupné z:
<http://www.iqrf.com/weben/downloads.php?id=239>
- [4] IQRF OS V3.05D USER'S GUIDE. *IQRF OS: User's guide* [online]. 2014, s. 54
[cit. 2014-12-01]. DOI: User_Guide_IQRF-OS-304D_140430. Dostupné z:
<http://www.iqrf.com/weben/downloads.php?id=375>
- [5] IQRF TECHNICAL GUIDE. *Technical Guide* [online]. 2014, s. 20 [cit. 2014-12-01]. DOI: IQRFTG_rev.121031. Dostupné z:
<http://www.iqrf.com/weben/downloads.php?id=79>
- [6] DPA FRAMEWORK TECHNICAL GUIDE V2.01. *IQRF DPA Framework*
[online]. 2014, s. 58 [cit. 2014-12-08]. DOI: Tech_Guide_DPA-Framework-201_140604. Dostupné z: <http://www.iqrf.com/weben/downloads.php?id=353>
- [7] KUČHTA Radek, VRBA Radimír a Vladimír ŠULC. *IQMESH implementation in IQRF wireless communication platform*. Advances in Mesh Networks, 2009. MESH 2009. Second International Conference [online]. 2009, s. 4 [cit. 2014-12-16]. ISBN 978-0-7695-3667-5. DOI: 10.1109/MESH.2009.18. Dostupné z:
<http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/xpl/articleDetails.jsp?arnumber=5222983>
- [8] INA210-214EVM USER'S GUIDE. *Quick Start Setup and Use* [online]. 2008, s. 17
[cit. 2015-03-21]. DOI: SBOU065. Dostupné z:
<http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=sbou065&fileType=pdf>
- [9] Akumulátory NiMh s nízkým samovybíjením LSD NiMh. [online]. [cit. 2015-04-22]. Dostupné z: <http://www.podzemi.net/news/akumulatory-nimh-s-nizkym-vybijenim/>

Seznam obrázků

Obrázek 1 IQRF TR modul	10
Obrázek 2 nastavení s jakým modulem se bude pracovat	11
Obrázek 3 nastavení hardwarové konfigurace	12
Obrázek 4 svazování uzlů ve vývojovém prostředí	13
Obrázek 5 uzly nalezené pomocí discovery ve vývojovém prostředí	14
Obrázek 6 síť typologie hvězda	14
Obrázek 7 zobrazení přenosu paketu v síti [1]	15
Obrázek 8 vytvoření CATS	16
Obrázek 9 práce s uzlem v DPA Service	17
Obrázek 10 TR konfigurace pro koordinátornoda	20
Obrázek 11 TR konfigurace pro uzel sub-sítě	20
Obrázek 12 přepnutí do podsítě	21
Obrázek 13 posílání příkazu do podsítě.....	21
Obrázek 14 nahrání programu do modulu	22
Obrázek 15 blokové schéma zapojení sledovače INA 211	24
Obrázek 16 realizace ovládní žárovky při synchronním sběru dat	37
Obrázek 17 realizace ovládní žárovky při asynchronním sběru dat	38
Obrázek 18 síť ovládaných světel.....	40

Seznam tabulek

Tabulka 1 změřená doba spánku.....	27
Tabulka 2 závislost počtu odpovědí na periodě dotazů	30
Tabulka 3 příkon uzlu v závislosti na periodě dotazů	32
Tabulka 4 změřená doba fungování uzlu	32
Tabulka 5 roční spotřeba uzlů.....	33
Tabulka 6 jak dlouho bude fungovat uzel.....	34
Tabulka 7 závislost latence na počtu hopů	35

Seznam grafů

Graf 1 demonstrace jak je pomalý polling v síti s více uzly [1]	18
Graf 2 průběh odpovědi na FRC ve standardním režimu	25
Graf 3 příjem v LP módu	25
Graf 4 průběh odpovědi na FRC v nízko příkonovém módu	26
Graf 5 průběh odpovědi na FRC ve vylepšeném nízko příkonovém módu.....	27
Graf 6 průběh proudu při nastavení doby spánku na 10s	28
Graf 7 průběh posílání zprávy koordinátorovi.....	29
Graf 8 závislost počtu odpovědí na periodě dotazů	31
Graf 9 závislost příkonu uzlu na periodě dotazů	32
Graf 10 kolik dní bude uzel fungovat	34

Seznam příloh

CD s elektronickou verzí bakalářské práce, s použitými pluginy a se zdrojovými kódy

Příloha naměřených hodnot

Příloha naměřených hodnot

Tabulka počtu odpovědí

T [s]	Měření pro FRC			Měření pro LP			Měření pro Sleep			Měření pro Sleep v. 2		
2,9	32	30	28	207	209	207	259	258	257	330	332	330
5,0	19	16	17	167	174	158	263	264	264	341	332	341
7,9	11	12	12	138	132	135	242	237	247	297	291	303
11,9	7	7	7	102	104	103	256	249	254	323	330	326
15,5	6	6	6	99	90	91	243	254	250	332	336	339
30,0	4	3	4	54	52	55	220	218	220	340	338	341
61,9	2	2	2	35	32	34	182	220	201	249	267	285

Tabulka doby fungování FRC při napájení vysoko kapacitním kondenzátorem

T [s]	Měření pro FRC t [s]			Měření pro LP t [s]			Měření pro Sleep t [s]			Měření pro Sleep v. 2 t [s]		
2,9	80	78	92	601	604	601	763	760	757	1010	1016	1010
5,0	77	87	82	840	870	810	1324	1329	1326	1779	1747	1811
7,9	83	80	86	1094	1038	1066	1864	1880	1833	2346	2299	2393
11,9	85	85	84	1158	1207	1182	2840	2848	2884	3882	3846	3918
15,5	94	79	87	1528	1535	1515	3938	3765	3851	5289	5226	5352
30,0	91	90	92	1594	1534	1625	6614	6803	6742	10200	10140	10230
61,9	96	94	98	2074	1845	2039	11269	13610	11207	15460	16578	17697

Tabulka doby fungování asynchronního provedení při napájení vysoko kapacitním kondenzátorem

T [s]	2,097	4,194	6,291	10,485	14,679	29,358	60,813
Asyn. t [s]	1619	3129	4785	6100	11259	24383	48060
	1593	3110	4726	6080	11360	24860	47251
	1721	2995	4883	6090	11674	24621	46444

Počet asynchronních odpovědí se průměr rovnou určil z počtu přijatých zpráv v terminálu vývojového prostředí IQRF IDE.