

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Chytrá řešení na platformě IQRF
Bakalářská práce

Autor: Tomáš, Mrňák
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Mgr. Tomáš Kozel, Ph.D.

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Mrňák Tomáš	Dřiteč 187, Dřiteč	I1600583

TÉMA ČESKY:

Chytrá řešení na platformě IQRF

TÉMA ANGLICKY:

Smart solutions based on IQRF platform

VEDOUcí PRÁCE:

doc. Mgr. Tomáš Kozel, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Seznámení s platformou IQRF.org a vývoj vlastních chytrých řešení s její pomocí.

1. Úvod
2. Obecná problematika IoT
3. Přehled vybraných řešení pro IoT
4. Popis IQRF a možností využití
5. Demonstrace praktického použití IQRF
6. Výsledky
7. Závěr

SEZNAM DOPORUČENÉ LITERATURY:

Prozatím:
<http://iqrf.org>

Podpis studenta: Mrňák

Datum: 15.10.2018

Podpis vedoucího práce: Jana Kozel

Datum: 15.10.18

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 29.4.2019

.....
Tomáš Mrňák

Poděkování:

Děkuji vedoucímu bakalářské práce doc. Mgr. Tomáši Kozlovi Ph.D za metodické vedení práce a zapůjčení IQRF vývojářského balíčku.

Anotace

Tato práce je zaměřená na charakterizování struktury Internetu věcí a ukázkou praktického využití této technologie v domácnosti. Začátek práce pojednává o základní struktuře Internetu věcí a popisuje činnosti senzorů a aktuátorů na uzlu. Další část této práce se zabývá popisem požadavků na uzel v ekosystému Internetu věcí a popisuje praktická využití této technologie v několika odvětvích. V práci jsou popsány nejčastěji využívaná řešení pro aplikaci této technologie s bližším popisem technologie IQRF. Technologii IQRF je v této práci věnovaná samostatná kapitola, ve které jsou popsány principy a základní charakteristiky této technologie.

Tato práce obsahuje praktickou část, kterou tvoří dva odlišné pokusy o naprogramování vlastního řešení na platformě IQRF. V prvním pokusu byla vytvořena aplikace, která na základě naměřených dat z fotorezistoru, zhasíná nebo rozsvěcí led diody na uzlu. Obsahem druhého pokusu v praktické části této práce je rozsvícení led diod na základě dat z teploměrů. Podmínkou, pro rozsvícení led diody je porovnání hodnoty naměřené na teploměru s limitní hodnotou, která se nachází v externí databázi.

Klíčová slova

internet věcí, technologie, uzel, senzor, vysílač, DPA

Annotation

Title: Smart solutions based on IQRF platform

This thesis is focused on characterizing the structure of the Internet of things and demonstrating the practical use of this technology at home. The beginning of the thesis deals with the basic structure of Internet of things and describes the activities of sensors and actuators on the node. The next part of this thesis deals with the description of node requirements in the Internet of Things ecosystem and describes practical applications of this technology in several sectors. The most frequently used solutions for application of this technology are described in the thesis. In this thesis, IQRF technology is devoted to a separate chapter describing the principles and basic characteristics of this technology.

This thesis contains a practical part, which consists of two different attempts to program own solution on the IQRF platform. In the first experiment, an application was created which, based on the measured data from the photoresist, extinguishes or turns on the LEDs on the node. The second attempt in the practical part of this thesis is to turn on LEDs based on thermometer data. The condition for switching on the LED is to compare the value measured on the thermometer with the limit value located in the external database.

Keywords

internet of things, technology, node, sensor, transceiver, DPA

Obsah

1	Úvod	1
2	Cíl práce.....	3
3	Internet věcí	4
3.1	Senzory	5
3.1.1	Senzor RFID	6
3.2	Aktuátory	7
3.2.1	Ovládání IoT zařízení	7
3.3	Požadavky na IoT uzly	8
3.3.1	Fyzické omezení	8
3.3.2	Interakce s okolním světem	8
3.3.3	Schopnosti IoT uzlu.....	9
3.3.4	Uživatelská omezení.....	11
3.4	Obory, ve kterých se Internet věcí využívá	11
3.4.1	Zemědělství	11
3.4.2	Inteligentní doprava a automobilový průmysl.....	12
3.4.3	Průmyslové procesy.....	12
3.4.4	Inteligentní domy.....	13
3.4.5	Inteligentní Města	13
3.4.6	Zdravotní péče	14
3.5	IoT Protokoly pro komunikaci.....	14
4	Přehled vybraných řešení pro IoT	15
4.1	ZigBee.....	15
4.2	Z-Wave Alliance.....	16
4.3	LoRa Alliance	16
4.4	IQRF Alliance.....	17
5	Technologie IQRF	18
5.1	Komunikační vlastnosti IQRF vysílače	18

5.2	IQMESH Protokol	18
5.3	Základní informace o IQMESH.....	19
5.4	Aplikační přístupy pro IQRF vysílače	20
5.4.1	Programovací přístup.....	20
5.4.2	DPA – přímý periferní přístup.....	20
5.5	Architektura DPA	21
5.5.1	Struktura DPA příkazu	21
5.6	IQRF zařízení.....	22
5.6.1	IQRF vysílač.....	22
5.7	IQRF Vývojářské moduly.....	23
5.7.1	DK-EVAL-04A	23
5.7.2	DDC-SE-01	24
5.8	IQRF Cloud Server	25
6	Ukázka využití IQRF v domácnosti	25
6.1	Vytvoření IQRF IoT Sítě	26
6.1.1	Nahrání základních komponent do IQRF vysílačů.....	26
6.1.2	Zprovoznění IQMESH sítě.....	27
6.2	Rozsvícení diod na zařízeních, při nízké hladině světla	27
6.2.1	Založení základní struktury CustomDPAHandleru.....	27
6.2.2	Snímání dat ze sensorového modulu	28
6.2.3	Analýza dat na koordinátoru.....	30
6.2.4	Zprovoznění 1. ukázky	33
6.3	Ukázka inteligentní regulace teploty	34
6.3.1	Konfigurace Uzlů a Brány	34
6.3.2	Komunikace mezi IQRF Cloudem a bránou	35
6.3.3	Analýza dat ze sensorů.....	37
6.3.4	Vizuální reprezentace dat ze sensorů	39
7	Shrnutí výsledků.....	42

8	Závěry a doporučení	43
9	Seznam použité literatury	44
10	Seznam použitých zkratkách	46
11	Přílohy	47

Seznam obrázků

Obrázek 1: Schéma architektury Internetu věcí [2]	5
Obrázek 2: Minimální složení IoT uzlu [4].....	6
Obrázek 3: Průběh spotřeby energie v IoT uzlu [2]	9
Obrázek 4: Graf požadovaného rozlišení u senzorů [2]	10
Obrázek 5: Graf požadované propustnosti a spotřeby uzlu [2].....	10
Obrázek 6: diagram IQRF Bezdrátové sítě [15]	19
Obrázek 7: Schéma vysílače s podporou DPA [21].....	23
Obrázek 8: IQRF vysílač [23]	23
Obrázek 9: Schéma uzlového modulu [24]	24
Obrázek 10: Sensorový modul [25]	24
Obrázek 11: Ukázka implementace základní struktury [27].....	28
Obrázek 12: Zjednodušené schéma sensorového modulu [28].....	28
Obrázek 13: Ukázka zapnutí požadovaného portu na IQRF DDC.....	29
Obrázek 14: Kontrola přijatého DPA příkazu) [27].....	29
Obrázek 15: Zapnutí a nastavení analog-digitálního převodníku) [27]	30
Obrázek 16: Zapnutí a nastavení analog-digitálního převodníku) [27]	30
Obrázek 17: Vydefinování proměnných	31
Obrázek 18: Příznak switch a case [27]	31
Obrázek 19: Odeslání požadavku na uzel s fotorezistorem.....	33
Obrázek 20: Naplnění 16 bitové proměnné hodnotou z fotorezistoru.....	33
Obrázek 21: Struktura pole s DPA příkazy a parametr „requestParams“	37
Obrázek 22: Načítání minimální požadované teploty v aktuální hodinu	39
Obrázek 23: Vydefinování polí, sloužících jako vstupní parametr do grafů.....	39
Obrázek 24: Výstupní graf sensorických hodnot z fotorezistoru).....	41
Obrázek 25: Výstupní graf sensorických hodnot z teploměru).....	41

Seznam Tabulek

Tabulka 1: Struktura DPA příkazu [22]	21
Tabulka 2: Zařízení v IQMESH síti 1. Ukázka	34
Tabulka 3: Zařízení v IQMESH síti 2. Ukázka	35
Tabulka 4: První 3 záznamy v listu sensor_days.....	38

1 Úvod

Bezdrátová technologie internet věcí byla poprvé představena již v roce 1999. Tato technologie se však dočkala největšího rozšíření až po několika letech od svého představení. Novými optimalizacemi procesů došlo k vývoji levnějších polovodičových součástek, které následně umožnily její větší aplikační možnosti. [2]

Síťová zařízení pro aplikaci internetu věcí jsou rozdělena do tří vrstev. Zařízení, která se nacházejí na nejnižší vrstvě se nazývají uzly. Tato zařízení mají obvykle možnost vnímat změny ve fyzickém světě podobně jako člověk, který vnímá pěti smysly svět kolem sebe. Uzly mohou také pohybovat s objekty ve fyzickém světě, a tím našla tato technologie uplatnění ve firmách a podnicích, které hledají efektivnější a ekonomičtější řešení svých procesů. Technologie internetu věcí však nenachází uplatnění pouze ve firemním prostředí. Lze ji využít například pro zvýšení bezpečnosti domácností, nastavení domácí pohody nebo třeba k rekreačním účelům. Její rozšíření záleží na požadavku klienta.

Jelikož využití této technologie je pestré, dochází zde velmi často k vymezování požadavků na vlastnosti těchto uzlů. Uzel je obvykle napájen externí baterií. Snížením požadované přesnosti měření, lze snížit i cenu uzlu a zároveň snížit jeho energetickou spotřebu. Požadavky jsou kladeny i na maximální velikosti uzlu. Velikost uzlu přímo ovlivňuje maximální velikost baterie.

Pro aplikaci této technologie se využívají již hotová řešení. Tato řešení jsou vyvíjena s předem určenými požadavky využití u koncových zákazníků. Mezi hotovými řešeními lze nalézt technologie, které se specializují na přenos dat v řádech kilometrů nebo technologie, které se specializují na využití pouze v domácnosti.

Technologické řešení IQRF má své využití především v podnikové sféře. Tuto technologii lze využívat například v průmyslu 4.0 nebo při automatizaci budov. Schopnost senzorů ovšem dovolí jejich použití i v moderní domácnosti, neboť senzory, které jsou použity k vylepšení montážních hal, mohou zároveň chránit hodnoty nebo snižovat náklady v běžném životě.

Aplikace postavené na technologii Internetu věcí nemusí být vždy jednoduché a jsou velmi často rozšířeny i o internetový přístup. Ovládání této aplikace tak může být ovlivněno i daty z internetu (předpovědi počasí, kurz měn, online mapy, atd..) Základem

rozvoje a aplikace Internetu věci je průmyslový pokrok a snaha lidstva snižovat energetické potřeby procesů, snižovat emise škodlivých plynů a bránit oteplování planety. Tyto technologie jsou stavebními buňkami v průmyslu, podnikání, řízení dopravy nebo třeba jen jako malá buňka v řízení tepelné pohody domácnosti. Díky své nízké spotřebě elektrické energie ukazují novou cestu v tomto oboru.

2 Cíl práce

Cílem této bakalářské práce je seznámení zájemců z řad čtenářů o možnosti využití technologie Internet věcí.

Obsahovým cílem této práce je přiblížit funkci uzlu v technologii a vydefinovat požadavky, které jsou nejčastěji řešeny při vývoji těchto zařízení.

Smyslem této práce je přiblížit obory, ve kterých dnes tato technologie spolehlivě pracuje a následně popsat vybraná řešení, pro aplikaci v nových projektech.

Klíčovým cílem této práce je seznámení čtenáře s technologií IQRF a vysvětlit mu, jak jsou programové principy Internetu věcí využívány v praktických ukázkách.

V první ukázce je použit programový přístup DPA, k vytvoření funkčního mini systému. To znamená – naprogramovat systém tak, aby zapínal, a vypínal signalizační led diody podle úrovně světla dopadajícího na bezdrátový fotorezistor s jeho RFIDs přenosem dat.

Druhá ukázka má za cíl vytvořit program podobný inteligentnímu domácímu termostatu s využitím IQRF technologie. To znamená – naprogramovat bezdrátová čidla tak, že budou sledovat teplotní hodnotu, a z těchto výstupů bude následně vyhodnocena potřeba pro systém, zda „topit“ v sledovaném prostoru v domácnosti.

3 Internet věcí

Nové objevy z odvětví počítačové informatiky jsou obvykle nejdříve prakticky aplikovány a později definovány jako nové pojmy. Jinak tomu nebylo ani v případě Internetu věcí (IoT – *angl. Internet of things*). V listopadu roku 1993 několik nadšených počítačových expertů zkonstruovalo elektronický projekt, při kterém sledovali hladinu kávy v kávovaru. Nechtěli se namáhat chozením do místnosti s kávovarem, pokud nebyla káva připravená. Tito experti však ještě netušili, že vymysleli první reálnou aplikaci z oboru Internetu věcí. [1]

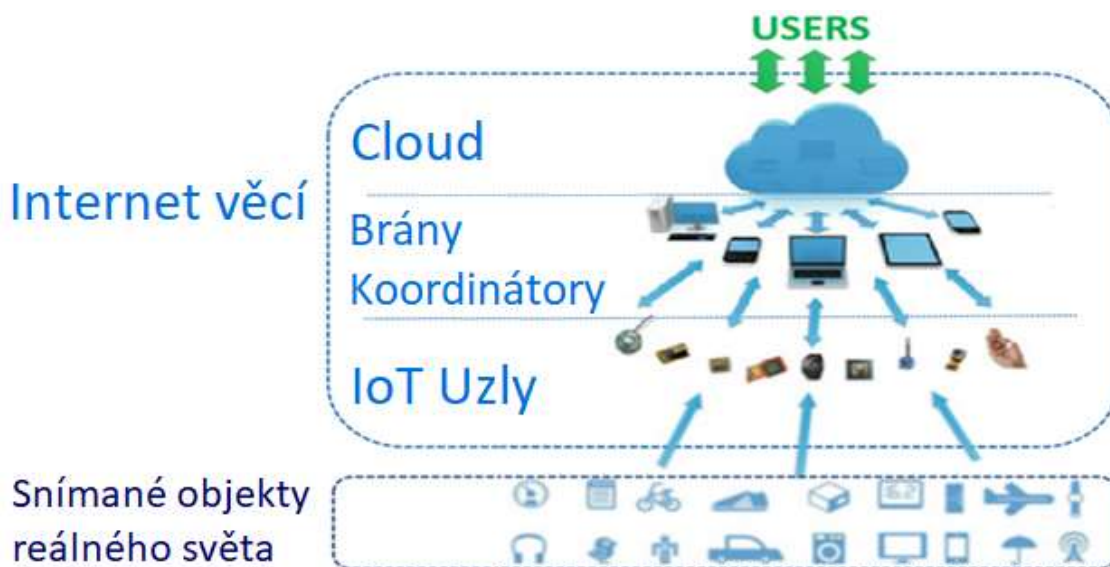
Hlavní osobností, která stála u vzniku tohoto dnes velmi často skloňovaného pojmu, se stal Kevin Ashton. Byl to on, kdo na prezentaci firmy Procter & Gamble v roce 1999, popsal tento neznámý pojem jako „sít' chytrých zařízení“, které spolu komunikují pomocí rádiové technologie (RFIDs). [2]

Dnes je Internet věcí chápán, jako jakési „propojení věcí“, s cílem zjišťovat a ohlašovat informace, které probíhají ve skutečném světě. Internet věcí, lze definovat několika různými způsoby a význam tohoto pojmu se v dnešní době stal tak širokým, že každé zařízení, které je připojeno k internetu (například automobil, mobilní telefony, průmyslové nástroje, meteostanice), lze považovat za chytré zařízení z oboru Internetu věcí. [2]

„Obecněji lze však Internet věcí definovat jako síťovou paradigmatu pro kybernetické fyzické systémy, které se vyznačují fyzickým objektem a kybernetickými prostředky, které mohou být vzdáleně zjišťovány a řízeny přes existující síťovou infrastrukturu, což umožňuje integraci mezi fyzickým světem a počítačem, a proto rozšiřuje internet do reálného světa.“ [3]

Pro jednoduché vysvětlení základních částí architektury Internetu věcí slouží *obrázek 1*. Na tomto obrázku se nachází ve spodní části objekty reálného světa. Různé vlastnosti a stavy těchto objektů jsou sledovány zařízeními, které se nachází na nejnižší (třetí) vrstvě Internetu věcí. Tyto zařízení se nazývají uzly (*angl. nodes*) a kromě vnímání, mohou i interagovat s objekty reálného světa. Pro správu uzlů a zajištění požadované rozšiřitelnosti, se na druhé vrstvě nachází koordinátoři. Jestliže je koordinátor rozšířen i o síťový přístup, nazývá se bránou. Ta narozdíl od koordinátoru umožňuje shromážděná data zasílat na server. Hlavní aktivitou je pro zařízení na druhé vrstvě shromažďování

data z uzlů. Na třetí vrstvě dochází ke zpracování dat z brán, do uživatelsky přívětivé podoby v závislosti na konkrétním řešení. Hardwarové požadavky na zařízení, která se nachází na třetí vrstvě, jsou velmi odlišné v závislosti na úrovni požadované dostupnosti. [2]

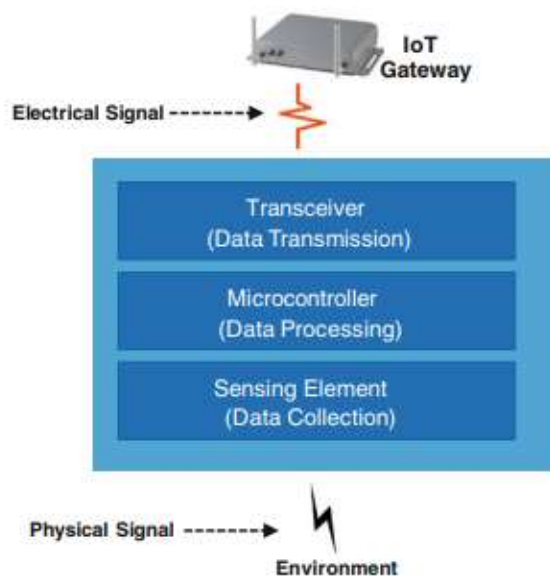


Obrázek 1: Schéma architektury Internetu věcí [2]

3.1 Senzory

Senzor je zařízení (obvykle elektronické), které je často označováno za analog-digitální převodník. Hlavním úkolem každého senzoru je snímat a detekovat různé fyzikální nebo chemické změny, které probíhají v reálném světě. Snímanou veličinou může být například zvuk, obraz, teplota, tlak a další. Vstup do senzoru je obvykle tvořen analogovým signálem, který musí být analog-digitálním převodníkem modulován do digitálních dat, aby bylo možné snímaná data ze senzoru analyzovat (zpracovat). [4]

Dnes se lze setkat s velmi jednoduchými senzory, které disponují pouze funkcemi pro snímání dat a přenášení dat. Nebo se lze setkat i se senzory, které obsahují i jistou míru inteligence, takovéto senzory umí například filtrovat duplicitu, nebo odesílat data až po splnění několika podmínek. V případě inteligentních senzorů, je vyžadováno, aby se IoT uzel skládal z minimálně tří částí (senzoru, mikrokontroleru a konektivity pro odesílání dat do IoT brány), jak je zobrazeno na obrázku 2. [4]



Obrázek 2: Minimální složení IoT uzlu [4]

3.1.1 Senzor RFID

Jedním z mnoha typů senzorů je i RFID senzor (identifikátor rádiových frekvencí). Tento senzor společně s anténou slouží k zachycování informací, které jsou ve formě rádiových vln. Používání RFID technologie je regulováno Mezinárodní telekomunikační unií, která spravuje kmitočtová pásma. Pro Evropu je stanoven povolený rozsah od 865 MHz do 869 MHz. [4]

Mechanismus RFID se typicky skládá ze dvou částí, značky (vysílače) a čtečky (přijímače). Značka obsahuje konkrétní sériové číslo, pro jeden konkrétní objekt a je obvykle spojena s anténou pro příjem a vysílání signálu. V oblasti IoT se využívají tři typy značek: [3]

1. Pasivní – Napájeny čtečkou a obsahují budící anténu. Tímto způsobem jsou označena například zboží v obchodech nebo cestovní pasy.
2. Aktivní – Obsahují vlastní zdroj energie.
3. Semi-Aktivní – Napájené externím zdrojem energie.

Aktivní a Semi-Aktivní značky jsou obvykle propojeny a napájeny IoT uzlem a mají typický dosah až 100 m, naopak pasivní značky mají maximální dosah až 12 metrů. [3]

Čtečka je zařízení, které obsahuje modul rádiového rozhraní (RFI), řídicí jednotku a anténu. Hlavní funkcí čtečky je vyslání signálu pro aktivování značky a navázání komunikace pro přenášení dat mezi značkou a RFID aplikací. [5]

Mezi hlavní výhody RFID patří možnost čtení až stovek značek, což výrazně šetří čas. A výhodou RFID je také fakt, že čtecí zařízení a značka nemusí být v přímém pohledu. [3]

Stejně jako jiné technologie, má RFID i řadu nevýhod. Tou nejvýznamnější z nich, je zablokování RFID rádiových vln, například hliníkovou folií. Další nevýhodou může být rušení mezi čtečkami a značkami při nesprávném nastavení celkového systému. Je potřeba tedy vždy dbát na rádiový rozsah každé RFID čtečky, neboť čtečka skenuje vždy všechny značky, ve svém rozsahu. Například: při aplikaci RFID v maloobchodu by mohlo snadno dojít k zaúčtování položky zákazníkovi, kterou má, ale ve skutečnosti jiný zákazník ve svém košíku. [4]

3.2 Aktuátory

Aktuátor je zařízení, které pro svou práci vyžaduje soubor úkonů a energii. Tyto zdroje jsou po dodání přeměněny na jistou činnost ve fyzickém světě. Typickým příkladem aktuátorů může být například pneumatický píst nebo motor. [4]

3.2.1 Ovládání IoT zařízení

Při návrhu řešení, jak ovládat a monitorovat IoT zařízení se lze setkat s principem lokálního řízení a principem globálního řízení. Aplikace lokálního řízení je oproti globálnímu jednoduchá a spočívá v instalaci ovládacího prvku do IoT uzlu (např. ovládání kotle a klimatizace domácím termostatem). [4]

Naopak aplikace globálního přístupu spočívá v ovládání jednotlivých zařízení globálně (např. ovládání kotle a klimatizace přes internet v závislosti na dalších podmínkách). K docílení této myšlenky je zapotřebí přemístit ovládací prvek například na cloud a vložit do všech zařízení levné senzory. Při aplikaci globálního přístupu je zapotřebí nainstalování více senzorů (vyšší náklady), ale nabízí možnost pro řízení ovládacího prvku v závislosti na datech z více zdrojů (např. venkovní senzory, teplotní senzory, předpověď počasí). [4]

3.3 Požadavky na IoT uzly

Dle Massimo Aliota, autora knihy *Enabling the Internet of Things*: „*Jsou charakteristické rysy IoT uzlů definovány pomocí požadavků z hlediska fyzických omezení, druhu interakce s vnějším světem, požadovanými schopnostmi na IoT uzel a uživatelskými požadavky na IoT uzel.*“ [2]

A jsou rozdělena do následujících subkategorií: [2]

Fyzické omezení

- Design, Velikost (*angl. Form factor*)
- Schopnost nabíjet energii

Interakce s okolím

- Životnost
- Schopnost provozu po celý den

Schopnosti uzlu

- Bezdrátové připojení
- Senzory

Uživatelské požadavky

- Cena
- Bezpečnost

3.3.1 Fyzické omezení

Aplikace IoT technologie by neměla nijak narušovat stávající rozmístění objektů a je tedy nezbytné, aby velikost takového IoT uzlu byla dostatečně malá. Obvyklá velikost IoT uzlu se pohybuje od několik milimetrů krychlových, až po stovky milimetrů krychlových. Společně s velikostí IoT uzlu jsou kladeny požadavky na způsob napájení těchto uzlů. Většina IoT uzlů jsou napájena baterií a jsou tedy nastaveny velmi vysoké nároky na co nejnižší spotřebu IoT uzlu, protože velikost každého uzlu, je ovlivněna velikostí baterie. [2]

Při řešení fyzických omezení je často kladen i požadavek, na vyhnutí se jakékoli budoucí údržbě (např. Výměna baterie). [2]

3.3.2 Interakce s okolním světem

IoT uzel je tedy obvykle nepájen baterií, životnost této baterie, je vhodné učit podle objektu nebo prostředí, do kterého je uzel vložen, protože výměna baterie, nemusí být vždy možná. Při aplikaci IoT technologie v životním prostředí nebo infrastruktuře, je

požadovaná životnost několik desetiletí. Životnost okolo jednoho desetiletí je vyžadována v prostředí jako je průmysl nebo doprava. A nejnižší nároky na životnost baterie v uzlu jsou v oblastech, maloobchodu, sledování provozu nebo zábavy. [2]

Z výše uvedených řádků, je tedy patrné, že možný rozsah životnosti baterie v IoT uzlu je velmi široký. Při zjišťování spotřeby IoT uzlu, je potřeba určit dobu, kdy je uzel zatížen neboli provádí úkony (např. snímání, zpracovávání) a dobu, kdy je uzel v režimu spánku. Uzel je obvykle uspaný a probudí se na dobu nezbytně nutnou k vykonání operace (může se například probudit na 1,5 sec. každou minutu, jako je znázorněno na *obrázku 3*) [2]



Obrázek 3: Průběh spotřeby energie v IoT uzlu [2]

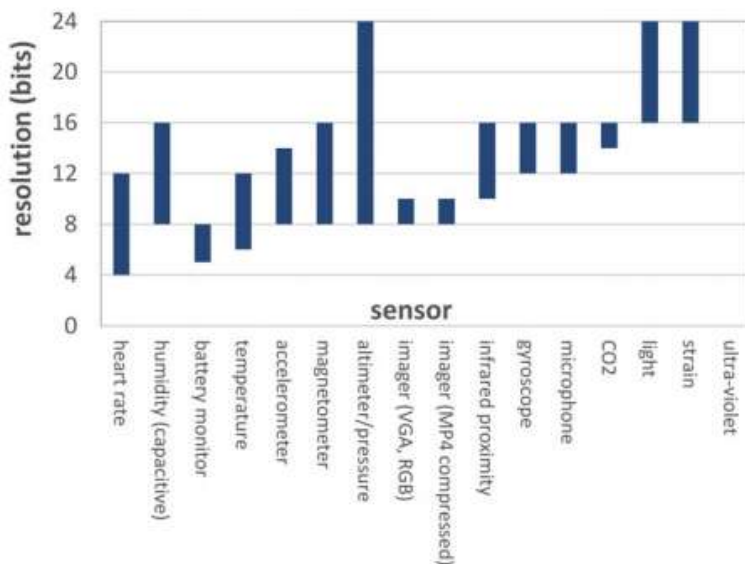
IoT uzel je vždy zapnuté zařízení a řadí se tak do podsystému ALWON. Tato architektura se vyznačuje periodickým cyklem probouzení (Twkup) a zpracováváním informací napříč aktivními úlohami. Jedním ze způsobů, jak snížit spotřebu energie, je kompromis, mezi počtem snímání za jednotku času a dobou spánku mezi snímáními.[2]

3.3.3 Schopnosti IoT uzlu

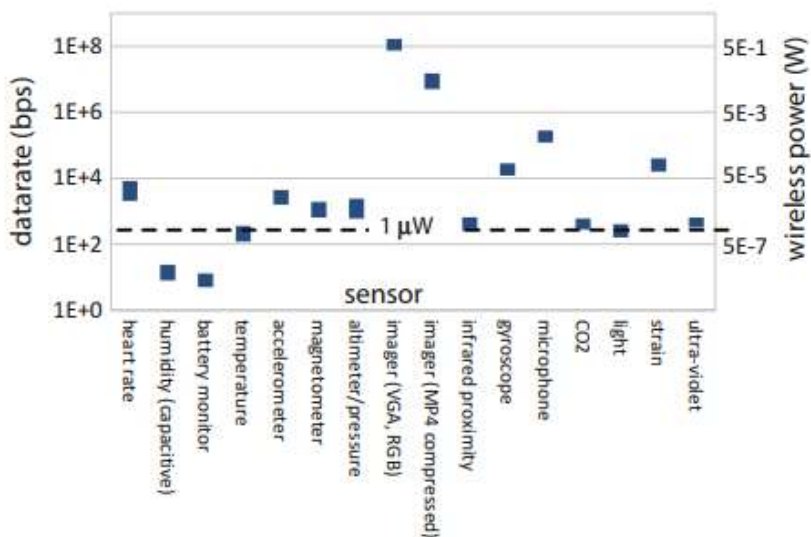
Snímání, výpočet matematických operací a poskytování bezdrátové komunikace jsou funkce, kterými musí disponovat každý IoT uzel. Existují aplikace, které nevyžadují, aby každý IoT uzel disponoval těmito vlastnostmi v plné šíři, a tak je například schopnost snímání u některých uzlů méně potřebná. Jestliže by bylo použití univerzálních uzlů, které jsou vhodné pro všeobecné použití zbytečné a nákladné, může být pro specifický účel navržena a přizpůsobena MEMS (MikroElektroMechanický Systém) konstrukce IoT uzlu, která může mít na rozdíl od univerzálních uzlů, snížené vlastnosti u jednotlivých funkcí. [2]

Na jednoduchém *obrázku 4* jsou vyobrazeny různé senzory a jejich doporučené rozlišení. Z obrázku je na první pohled patrné, že rozlišení 24 bitů je doporučováno jen pro několik senzorů. A pro většinu praktických aplikací je rozlišení mezi 8 bity a 12 bity zcela dostačující. [2]

Požadovaná datová propustnost pro jednotlivé aplikace, je znázorněna na obrázku 5. Z tohoto obrázku je patrné, že se zvyšující se datovou propustností se zvedá i spotřeba uzlu. Při výběru vhodného řešení je tedy potřeba zapotřebí i určit, jaké operace budou s daty prováděny (porovnávání, filtrování). Celkové rozložení nákladů, výpočetního výkonu a spotřeby tak záleží vždy na konkrétním využití. [2]



Obrázek 4: Graf požadovaného rozlišení u senzorů [2]



Obrázek 5: Graf požadované propustnosti a spotřeby uzlu [2]

3.3.4 Uživatelská omezení

Další požadavky, které klade uživatel na výrobce IoT zařízení, souvisí s cenou a bezpečností. [2]

Uživatel obvykle klade důraz, na dlouhodobou životnost IoT uzlu (např. desetiletí), což v dlouhodobém hledisku znamená, že trh s IoT je zcela odlišný od spotřebního trhu, který zpravidla spoléhá na periodické vlny nové poptávky stimulované příchozí generací produktu se zlepšenými vlastnostmi. Tento požadavek, je potřeba vyřešit prostřednictvím designu. Je vhodné, aby platforma umožňovala uživateli rekonfigurování. Tento přístup pomůže snížit náklady při rozšíření IoT uzlu o nové aplikace (senzory a funkce). [2]

Zajištění bezpečnosti, je velmi důležitý požadavek, neboť IoT se z důvodu svého velkého rozšíření, vybízí k častým útokům. Navíc tradiční řešení proti počítačovým útokům, jako je např. firewall nejsou použitelná na IoT uzlech, z důvodu dodržení stanoveného rozpočtu. [2]

3.4 Obory, ve kterých se Internet věcí využívá

V odvětví IoT technologie došlo v uplynulých několika letech k velkému rozmachu na všech základních IoT vrstvách. Díky tomu se rozšířili i různé potenciální aplikace pro IoT technologii.

3.4.1 Zemědělství

V zemědělství se dnes využívají IoT infrastruktury hned k několika různým aktivitám. Všechny aktivity, ale mají za hlavní cíl snížit náklady a zvýšit kvalitu vypěstovaných potravin a chovaného dobytka. [6]

Dle Organizace pro výživu a zemědělství je přibližně 70 % z celkové vyčerpané vody použito pro zemědělství. V zemědělství je dle této organizace zároveň vyplýváno až 60 % používané vody, nejčastěji z důvodu špatného zavlažovacího systému nebo výběru špatné potraviny pro danou lokalitu. [7]

Aktuátory a senzory tak mohou z efektivit využívání vody a snížit její plýtvání v tomto odvětví. Uzly, které obsahují senzory pro měření vlhkosti půdy obvykle měří i její teplotu nebo pH. Například na základě dat z těchto sensorů může ovocnář zefektivnit přihnojování a zalévání v ovocných sadech kapkovou závlahou. Výměra

sadu má tak své závlahové sekce, které jsou řízeny výstupními daty z těchto senzorů a podle potřeby přesně dávkuje vodu nebo hnojivo pro jednotlivé ovocné stromy. Tím je zaručená optimální spotřeba vody a skutečná potřeba vody pro dané vegetační období v ovocném sadu. Nedochozí tak k plošnému plýtvání vodou, elektřinou a lidskou prací. [6,7]

3.4.2 Inteligentní doprava a automobilový průmysl

Přítomnost IoT technologie, je nejčastěji využita k monitorování kritických součástí ve vozidlech. Tyto součástky mají senzory, pro zjišťování nárazu, vibrací nebo také opotřebení. Na základě dat z těchto senzorů, může být provedena preventivní údržba, která zajistí zvýšení životnosti vozidla. Tato možnost aplikace je velmi užitečná pro efektivní řízení vozového parku. [2]

Budoucnost IoT technologie ve vozidlech má usnadnit vývoj nových bezpečnostních systémů. Každé vozidlo by mělo obsahovat svůj IoT uzel, který bude komunikovat s ostatními vozidly ve svém rozsahu a vyměňovat si tak data na vzdálenost až 300 metrů. Vozidla by si tak v budoucnu mohla předávat informace, jako jsou změny počasí (sněžení) nebo informace o haváriích. Tyto informace jsou užitečné například pro systém na vyhnutí se kolizi. [1]

3.4.3 Průmyslové procesy

IoT technologii v průmyslovém prostředí lze aplikovat v různých částech v rámci celého výrobního procesu. Na straně skladu, lze sledovat skladové zásoby a dle snímaných dat přizpůsobovat plány na objednání dalšího materiálu. Při výrobě, lze sledovat polotovary v reálném čase. Díky čemuž může docházet k detekci událostí, které by potenciálně mohli zpomalit proces. Pomocí IoT lze také provádět diagnostiku výrobního stroje nebo jiného zařízení a provádět na základě snímaných dat plánovanou údržbu. Na straně logistiky, lze zjišťovat umístění produktu a tím se zefektivní dodávka a distribuce produktu. [2]

Aplikace IoT technologie v tomto prostředí snižuje náklady, zvyšuje výnos a umožňuje lépe kontrolovat výrobní proces. [2]

3.4.4 Inteligentní domy

V této poměrně rozsáhlé kategorii, se nejčastěji využívají zařízení, které spadají do kategorie spotřební elektroniky. Každé zařízení obvykle obsahuje několik senzorů a aktuátorů. Sensory na těchto zařízeních nejčastěji monitorují umístění, monitorují stav, detekují krizové anomálie (přehřívání) nebo sledují kvalitu ovzduší (teplotu, vlhkost, úniky plynu, koncentrace oxidu uhelnatého). Používání těchto zařízení a senzorů poté vede k zvýšení bezpečnosti, snížení nákladů a dosažení požadovaného pohodlí v domácnosti. [2]

Jestliže aplikace IoT technologie v oblasti inteligentních rodinných domů a bytů slouží ke zvýšení lidského pohodlí. Je obvykle požadováno, aby bylo například osvětlení, klimatizace nebo také vytápění, nastavované podle individuálních potřeb každého člena domácnosti. Monitorování kvality vzduchu a nastavování tepelné pohody v jednotlivých místnostech, může vést i ke snížení nákladů na topení. V blízké budoucnosti by měla chytrá domácnost spravovat a zajišťovat například včasné doplnění spotřebního materiálu (toaletní papír, sáčky do odpadkového koše). [2]

Pokud je požadováno využít IoT technologii pro zvýšení bezpečnosti v domácnosti, lze monitorováním pohybu osob v domácnosti předejít tragédiím. IoT umožňuje upozorňovat, na přítomnost dítěte v oblasti, kde je zapnuté nebezpečné zařízení (např. elektrické nebo pneumatické nástroje). Podobně lze vypínat i hračky, je-li v blízkosti nich batole, aby se zabránilo případnému nebezpečí. Existují ale i senzory a aktuátory, které reagují na krizové situace (požár nebo potopa) a snaží se v případě těchto událostí minimalizovat lidské a materiálové ztráty. [2]

3.4.5 Inteligentní Města

Město je pro vývojáře IoT technologie velmi zajímavé, neboť je zde velmi široký prostor pro různé oblasti aplikace IoT. Mezi městské aplikace IoT technologie patří například, inteligentní osvětlení, monitorování úniků vody nebo plynu, inteligentní parkování včetně dynamického spravování cen. Chytré město může v případě poruchy využít informačních tabulí, aby zabránilo nebezpečí, poskytnutím informací občanům v reálném čase. Chytré zavlažování zelených ploch a parků je další z možností, které chytré město spravuje. Dnes je velký zájem o Inteligentní turismus, ten slibuje, že

turistům umožní okamžité sledování přeplněnosti nebo hluku na často vyhledávaných místech turisty. Turisté si tak mohou dynamicky naplánovat své zájezdy. [2]

Aplikace IoT technologie ve městech tak najde uplatnění na různých místech a snaží se snížit náklady a zvýšit bezpečnost ve městě. Chytré město je současně i odolnější vůči dočasným poruchám a katastrofám. [2]

3.4.6 Zdravotní péče

IoT technologie díky možnostem vyrobit uzel o malé velikosti a s dlouhou životností, našla uplatnění i v oblasti zdravotní péče. Doktoři tak mohou sledovat například vitální funkce pacientů nebo lidské chování pacientů. Na základě dat, které tyto uzly snímají, pak doktoři mohou lépe pochopit zdravotní stav a mentální vývoj pacienta. [2]

Sledování zdravotního stavu pacientů a následná prediktivní diagnóza pacientů u kterých nemoc ještě nezačala, ale s nejvyšší pravděpodobností začne, se ověřila jako jedna z metod, pro zachování nezávislosti jednotlivce na nemocnici. Například v případě respiračních onemocnění je sledována hloubka a frekvence dýchání. Tyto data jsou poté použity k doporučení správného dýchacích tréninků. [1]

3.5 IoT Protokoly pro komunikaci

„IoT Protokol, který se užívá pro komunikaci, si lze představit jak rozšíření TCP/IP protokolu a skládá se z následujících částí:“ [4]

- Fyzické vrstvy
- Linkové vrstvy
- Síťové vrstvy
- Transportní vrstvy
- Aplikační / Protokolové vrstvy
- Vrstva aplikačních služeb

IoT uzly mohou být plně schopná zařízení nebo také velmi jednoduchá zařízení. Všechny tyto uzly mají obvykle předem vypočítanou a vymezenou spotřebu energie na zpracování operací a zajištění komunikace. Operace spojené s komunikací mají obvykle vyšší spotřebu energie než samotné zpracování dat ze senzorů a z toho vyplývá, že komunikační technologie musí být optimalizovány, aby vyhovovali i

zařizováním s nízkým výkonem. Nejvýznamnější dopad, na spotřebu energie má linková vrstva, která je zodpovědná za nuance během fyzického přenosu, rámcování, adresaci (MAC adresa) a re-transmisi. [4]

Komunikace nemusí být vždy stoprocentně úspěšná a znovu zaslání poškozeného rámce nebo nedoručeného rámce může být energeticky neefektivní. Z tohoto důvodu mají IoT aplikace různě nastavené požadavky na spolehlivost (ztrátovost paketů, latenci, dostupnost atd.). Za zmínění stojí kontrast mezi aplikací pro monitorování meteorologie a aplikací pro řízení tryskového motoru. Obě aplikace mohou využívat stejné senzory (snímání teploty a tlaku), rozdíl však bude v charakteristikách (požadavcích), které jsou kladeny na síťový provoz. [4]

Aplikace IoT technologie se stává stále rozmanitější a charakteristika IoT sítě se často liší od tradičních IT sítí. Tam dominují lokální sítě (LAN) a rozsáhlá síť (WAN). Rozmanitost IoT sítě je dána novým prostředím kde se využívá, například: ropná pole, námořní plošiny, doly, elektrické rozvody, vozidla. Zařízení v těchto sítích se mohou připojit k síti různými způsoby, ať už bezdrátově, drátově nebo kombinací těchto dvou technologií. [4]

Aby bylo možné tuto rozmanitost zachytit, bylo vyžadováno definovat nové protokoly na propojovacích vrstvách. [4]

4 Přehled vybraných řešení pro IoT

4.1 ZigBee

ZigBee je otevřený standard, který se zaměřuje primárně na monitorovací aplikace. Klíčovými vlastnostmi tohoto standardu jsou nízká přenosová rychlost (maximálně 250 Kb za sekundu), nízká spotřeba energie a bezdrátový přenos dat na vzdálenost až 100 metrů. Zařízení, které podporují standard ZigBee operují na bezdrátové frekvenci 868 MHz, 915 MHz nebo 2,4 GHz a jsou tak použitelné po celém světě. Při využití tohoto protokolu k monitorování, jsou senzorová zařízení většinu svého času v režimech spánku, a proto je velmi časté, že baterie v těchto zařízeních vydrží i několik let, než je potřeba je vyměnit. [8]

Od roku 2002, kdy vznikla ZigBee Alliance, je tento standard vyvíjen více než 170 společnostmi, kteří se stali jejími účastníky. Mimo tyto společnosti, existuje ještě

dalších 230 společností, kteří si ZigBee standard adoptovali a mají tak přístup ke standardům této technologie. [4]

Standard ZigBee má svou strukturu založenou na síťovém modelu OSI. Zároveň se však snaží o co nejvyšší interoperabilitu, a tak je prostřednictvím aplikační vrstvy sjednocena síťová vrstva a servisní vrstva tohoto protokolu. Fyzická vrstva a MAC vrstva jsou založeny na standardu IEEE 802.15.4. [9]

Pomocí takto rozdělených vrstev je ZigBee protokol snadno upravitelný na libovolnou aplikaci. Společnosti, které tento protokol využívají tak mohou pokrýt širší spektrum trhu s monitorovacími zařízeními a rychle reagovat na nové požadavky trhu. [4]

4.2 Z-Wave Alliance

Z-Wave je bezdrátový protokol využívající vlastní nízkoenergetické rádiové vlny k domácí automatizaci a automatizaci středně velkých podniků. Tento protokol byl původně vyvinut dánskou společností Zensys a dnes je vyvíjen průmyslovým konsorciem Z-Wave Alliance čítající více než 300 společností.[4]

Typickými příklady využití tohoto protokolu jsou automatické vytápění, automatické klimatizování nebo automatické ovládání oken, dveří a garáží. Protokol Z-Wave definuje komunikační vrstvu, síťovou vrstvu a aplikační vrstvu v OSI modelu. Z-Wave narozdíl od ZigBee není otevřený protokol a však velká část aplikační vrstvy je odhalena ve zdrojovém kódu OpenZwave. Fyzická vrstva a MAC vrstva tohoto protokolu jsou ratifikovány Mezinárodní telekomunikační unií (ITU G.9959). Síťová vrstva tohoto protokolu je veřejnosti nedostupná a při implementaci OpenZwave je vždy vyžadován Z-Wave přijímač. [10]

4.3 LoRa Alliance

Protokol LoRaWAN poskytuje bezdrátové spojení na dlouhé vzdálenosti. Narozdíl od protokolů ZigBee a Z-Wave, které mají maximální komunikační dosah 100 metrů, umožňuje protokol LoRaWAN komunikovat až na vzdálenost více než 100 km v příznivém počasí. V typickém venkovském prostředí je udávaná vzdálenost přibližně poloviční a v hustém městském prostředí je minimální udávaná vzdálenost 2 km.

LoRaWAN protokol má narozdíl od těchto dvou technologií velmi pomalou přenosovou rychlost, která se udává mezi 0,3 kb/s až 50 kb/s. [4]

Protokol LoRaWAN je vyvíjen LoRa Aliancí, která definuje a formuluje také technologii LoRa. LoRa reprezentuje fyzickou vrstvu OSI modelu a stará se o modulaci elektromagnetických vln pro komunikaci na velké vzdálenosti. [4]

Hlavními oblastmi, kde lze využít tento protokol jsou odvětví, jako například zemědělství, chytrá města nebo oblast inteligentních budov. Mezi velké nevýhody této technologie patří vysoká cena a nízká dostupnost modulů.[11]

4.4 IQRF Alliance

IQRF Alliance se zabývá vývojem vlastních vysílačů, které komunikují bezdrátově se základnou, popřípadě mezi sebou pomocí rádiových vln (RFID). Pro komunikaci využívají vysílače privátní IQMESH protokol, který byl vyvinut společností Microrisc (Zakladatelem IQRF technologie). Dnes je IQRF vyvíjeno s podporou členů IQRF Alliance. [12,13]

Mezi hlavní přednosti této technologie patří malá spotřeba energie, nízká datová náročnost a dosah až několik desítek nebo stovek metrů, přičemž ve speciálních případech, lze dosáhnout i několika kilometrové přenosové vzdálenosti. Naopak nevýhodou této technologie je nízká přenosová rychlost, přibližně 19 kb/s. [12]

IQRF technologie je primárně navržena, aby splnila všechny požadavky firemních zákazníků. Její využití lze rozdělit do čtyř primárních bloků: [13]

- První kategorii tvoří chytrá města
- Druhou kategorií jsou chytré budovy (montážní haly): Využití se najde především v budovách, které jsou starší a rekonstrukce by vyšla několika násobně draž než aplikace řešení pomocí IQRF Technologie. Do této kategorie se zahrnují i řešení pro chytré domácnosti (rodinné nebo bytové domy), kde jsou kladeny stejné nároky na automatizaci, jen aplikace probíhá v menším měřítku.
- Třetím blokem, jak technologii IQRF využít, je v Průmyslu 4.0, příkladem je monitorování správného fungování robotů nebo jiných automatizovaných strojů, jako jsou například bezpilotní vysokozdvizné vozíky.
- Ve čtvrté skupině se nachází ostatní zařízení, která nebyla více uvedena, jsou to například snímače spotřeby vody, elektřiny a plynu.

5 Technologie IQRF

5.1 Komunikační vlastnosti IQRF vysílače

IQRF vysílač je zařízení, které využívá ke komunikaci s koordinátorem nebo s jiným vysílačem bezplatné rádiové frekvence. Udávaná spotřeba vysílače, který vysílá informace je 15 μA , a jestliže je v režimu spánku, tak spotřebovává méně než 1 μA . Vysílače mají tedy velmi nízkou spotřebu, ale nebude-li zařízení mít externí napájení, je tedy potřeba před instalací vypočítat, jak často bude zařízení v režimu běžící a zařídit tak případné nabíjení. Maximální velikost jednoho poslaného paketu v síti je 64 B. Každý vysílač umí mimo jiné i upravovat citlivost na příjem rádiových vln. [14]

Dle výrobce, IQRF Technologie umožňuje vysílači komunikovat v následujících režimech: [14]

1. Bez síťování: Jedná se o zapojení dvou a více zařízení do struktury klient-klient (rovný s rovným). Každé zařízení je přístupné všem zařízením ve svém rádiovém dosahu. Jedná se o jednoduché zapojení, které nepodporuje mimo jednoduché adresování, žádné jiné síťové funkce.
2. Síťování: Síť je řízena jedním koordinátorem, který spravuje až 239 uzlů. Tento režim povoluje používat síťové funkce, jako je například, automatické směrování paketů ve smíšené síti přes redundantní cesty. Služba DPA, může být funkční pouze v tomto režimu síťování.

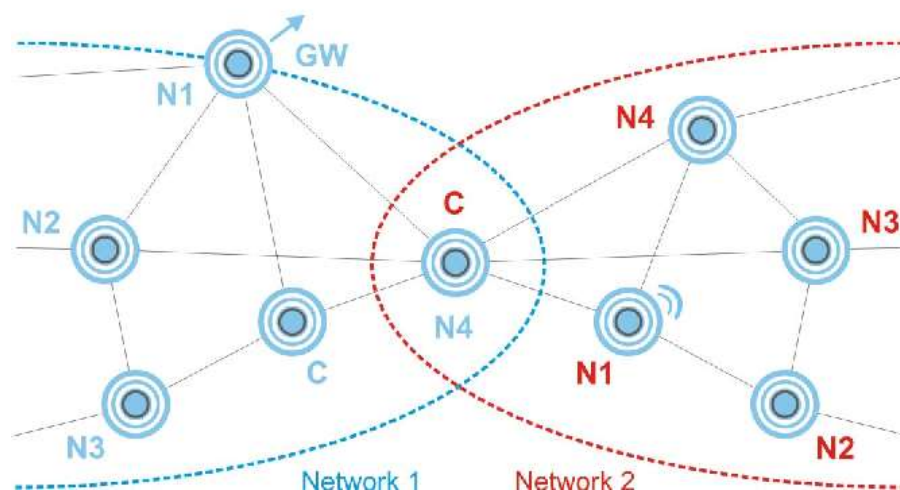
5.2 IQMESH Protokol

„IQMESH (Intelligentní MASH) protokol byl stanoven v roce 2005 jako základní komunikační protokol pro IQRF zařízení. Typické vlastnosti pro tento protokol jsou nízká spotřeba energie při adresaci a nízká přenosová rychlost. IQRF využívá několik unikátních, za patentovaných funkcí, a právě tento protokol byl vytvořen, aby všechny tyto funkce podporoval. „[15]

IQRF vlastní například patent na metodu zřetězení (*angl. Multi-bonding*). Pomocí této metody může jeden vysílač pracovat jako uzel nebo koordinátor ve dvou a více bezdrátových sítích. [15]

Na obrázku 6, jsou vyobrazeny dvě navzájem nezávislé bezdrátové IQRF sítě, Network 1 a Network 2. Každá síť je tvořena několik uzly(N) a vždy jedním koordinátorem(C), se kterým jsou uzly stejné barvy spárované. Pro zajištění obousměrné komunikace mezi Network 1 a Network 2 je použit Multi-bonding mechanismus. Tento mechanismus umožňuje modrému vysílači N4 zastávat funkci koordinátoru v síti Network 2 a zároveň funkci uzlu v síti Network 1. Koordinátor v modré síti tak může přijmout požadavek, který bude přesměrován pomocí modrého uzlu N4 na jiný uzel v červené síti. [15]

Pomocí Multi-bonding mechanismu, lze shromažďovat pakety, které směřují do opačné sítě a zaslat je do cílené sítě společně. Dojde tak k odlehčení síťové aktivity. [18]



Obrázek 6: diagram IQRF Bezdrátové sítě [15]

5.3 Základní informace o IQMESH

IQMESH využívá smíšenou (*angl. mesh*) topologii, ve které jsou pakety do cíleného zařízení zasílány buďto přímo, nebo přes další zařízení v síti směrovacím mechanismem. Tato topologie řeší potencionální problémy s topologií hvězda, ale výpočet nejefektivnější cesty je velmi náročný. [16]

IQMESH podporuje až 240 vysílačů v jedné síti (obvykle 1 koordinátor a až 239 uzlů). Adresa každého vysílače je reprezentována hodnotou o velikosti jednoho bajtu. Každý koordinátor nebo uzel může v síti figurovat jako brána, a tak rozšířit síť o přístup k datům z internetu. Implementace IQMESH protokolu má několik výhod, mezi které patří funkce Automatická síť a Objevovat. [17]

5.4 Aplikační přístupy pro IQRF vysílače

5.4.1 Programovací přístup

Programovací přístup využívá přímo operační systém s několika předdefinovanými funkcemi a všechny ostatní funkce, které uživatel požaduje, musí být naprogramovány přímo do uživatelské paměti v zařízení. Funkce, které jsou uživatelem naprogramovány, běží pod operačním systémem a využívají funkce operačního systému. Tento princip má IQRF technologie patentovaný. [18]

Jediný jazyk, který lze k programování funkcí využít, je jazyk C. Celá funkcionální vysílače, je závislá na znalostech a schopnostech programátora. Tento přístup nepodporuje využívat IQMESH síť. [18]

Při tomto přístupu se komunikuje v režimu klient-klient. Zařízení se při přijímání a odesílání dat chová následovně: [19]

- Při vysílání: Při zmáčknutí tlačítka nebo jiného spouštěče se zařízení probudí z režimu spánku a odešle informaci na všechna zařízení v radiovém dosahu. Poté opět přejde do režimu spánku.
- Při příjmu: Přijímač na zařízení přijímá radiové pakety a posílá je prostřednictvím SPI ke zpracování. Každý úspěšný příjem pak lze indikovat například bleskem led diody.

5.4.2 DPA – přímý periferní přístup

Tento přístup se také nazývá bezprogramový přístup a je právě druhým z možných aplikačních přístupů. Tento přístup využívá DPA protokolu a zařízení je tak ovládáno tokem dat. V tomto přístupu je pro komunikaci využit IQMESH protokol a není tak potřeba programovat komunikaci mezi vysílači. Uživateli je v tomto přístupu také dovoleno doprogramovat vlastní funkcionální, ale pomocí vlastního obslužného programu DPA (tzv. CustomDPAHandler). K ovládání zařízení slouží pouze UART terminál, pomocí kterého lze odesílat příkazy a přijímat odpovědi. [20]

5.5 Architektura DPA

Architekturu IQRF zařízení v DPA režimu lze charakterizovat třemi vrstvami, viz *obrázek 7*, stejně tak jako u programovacího přístupu, je základem vrstva s operačním systémem, zde je ale operační systém bez předpřipravených funkcí a všechny funkce řídí až druhá vrstva. [21]

Druhou vrstvou je DPA vrstva. Součástí této vrstvy, je již od výrobce zařízení předpřipravený hardwarový profil (HWP). Jedná se o funkci, která zajišťuje, aby zařízení bylo připojeno ke všem zařízením a službám, které uživatel v síti spravuje. Tato funkce usnadňuje instalaci hlavně ve velké IQMESH síti. Tento hardwarový profil obsahuje již na programované funkce pro připojení a zprovoznění IQMESH sítě na vysílači. [21]

Třetí a nepovinnou vrstvou je vlastní obslužný program DPA, tuto vrstvu v zařízení si může aktivovat uživatel, který si přeje doprogramovat funkcionalitu do vysílače. Při programování nových funkcí, dochází k rozšíření funkcí, které HWP poskytuje. [22]

Tento koncept architektury je za patentovaný a patří k dalším celosvětově unikátním řešením, na této platformě. [22]

5.5.1 Struktura DPA příkazu

Tabulka 1: Struktura DPA příkazu [22]

Název	NAND	PNUM	PCMD	HWPID	PDATA
Popis	Sít'ová adresa zařízení	Číslo periferie	Příkaz pro příslušnou periferii	Identifikátor hardwaru	Volitelná data

Popis tabulky 1: [22]

- **Pole NADR** (2B) – síťová adresa – může obsahovat hodnoty 0x00 (koordinátor), 0x01-EF (adresy uzlů), 0xFF (broadcast), 0xFE (dočasná adresa), 0xFC („místní zařízení“), ostatní adresy jsou rezervované.
- **Pole PNUM** (1B) – číslo periferie – např. 0x02 – Operační systém, 0x03 – EEPROM, 0x04 – EEPROM, 0x05 – RAM, 0x06 – LEDR, 0x07 – LEDG, 0x08 – SPI, 0x09 – IO, 0x0A – teploměr, 0x0D – FRC, 0x20-6F – uživatelské periferie.
- **Pole PCMD** (1B) obsahuje kód příkazu, který přísluší zvolené periférii. Může obsahovat hodnoty 0x00-0x3E.
- **Pole HWPID** (2B) slouží k filtrování cílového hardwaru. V případě, že je zde uvedena hodnota 0xFFFF, tak k filtrování nedochází.
- **Pole PDATA** může obsahovat až 56 bajtů uživatelských dat (v aktuální DPA verzi).

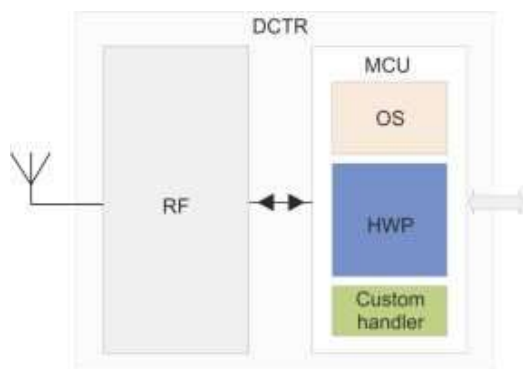
Tento odstavec byl plně převzat [22]

5.6 IQRF zařízení

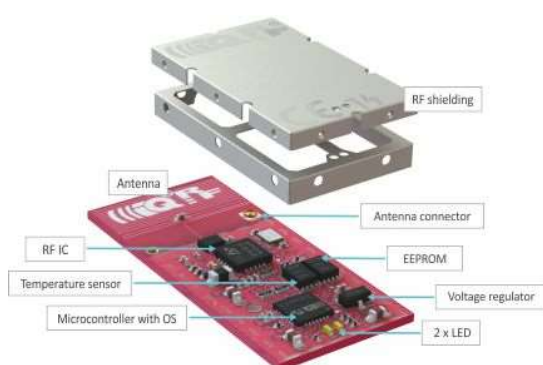
5.6.1 IQRF vysílač

IQRF vysílač (TR) je elektronická deska s kompletním elektrickým obvodem potřebným pro implementaci bezdrátové RF konektivity, viz *obrázek 8*. Jedná se o základní komunikační komponentu platformy IQRF, která se používá ve všech IQRF bránách, směrovačích atd. [23]

Každý vysílač je vybaven dvěma LED diodami, teplotním senzorem a integrovanou anténou. [23]



Obrázek 7: Schéma vysílače s podporou DPA [21]

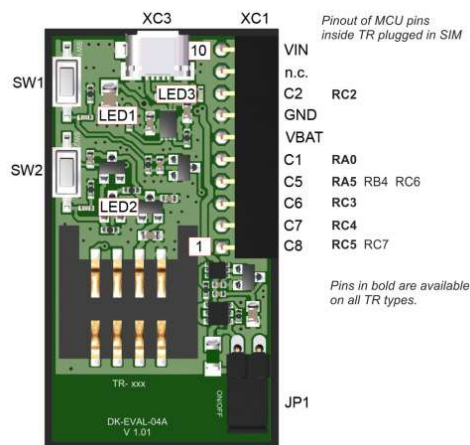


Obrázek 8: IQRF vysílač [23]

5.7 IQRF Vývojářské moduly

5.7.1 DK-EVAL-04A

Tento modul slouží výhradně pro vývoj bezdrátových aplikací a k aplikaci a testování rozsáhlých IQMESH sítí. Modul obsahuje SIM slot pro IQRF vysílač, Li-Pol baterii, mini USB konektor pro napájení a dvě tlačítka, viz *obrázek 9*. První tlačítko slouží k probuzení zařízení a na druhé tlačítko je možné naprogramovat vlastní funkcionalitu. Výstupy tvoří celkově tři indikační diody a možnost připojit až šest vstupní-výstupních zařízení. Tento modul má možnost napájet až 5 dalších modulů a lze k němu připojit i další nízko energetické senzory. Modul je kompatibilní i s jinými moduly pomocí IQRF DDC (Vývojový řetězec Daisy). Maximální teploty, při kterých je tento modul funkční jsou od -20 °C do +60 °C. [24]



Obrázek 9: Schéma uzlového modulu [24]

5.7.2 DDC-SE-01

Tento senzorový modul neobsahuje slot pro vysílač, ale připojuje se pomocí výstupního IQRF DDC (Vývojový řetězec Daisy) konektoru k jinému modulu, viz *obrázek 10*. Tento senzorový modul v základní konfiguraci obsahuje senzory pro měření teploty, fotorezistor pro měření intenzity světla a potenciometr pro měření napětí. Modul dále obsahuje I2C teplotní senzor a I2C konektor, který slouží pro připojení externí periférie. K tomuto modulu jde pomocí vstupního IQRF DCC konektoru připojit i další modul. Maximální teploty, ve kterých lze modul provozovat jsou od -40 °C do +85 °C. [25]



Obrázek 10: Senzorový modul [25]

5.8 IQRF Cloud Server

IQRF Cloud Serveru je služba, která má za cíl sběr a ukládání dat z koncových bezdrátových zařízení. Nejčastěji je tato komunikace realizována mezi Cloudem a bránou. V případě IQRF Cloudu lze využít obousměrné komunikace. IQRF Cloud umožňuje uživateli zaslat DPA příkaz do brány, která jej následně zpracuje, stejně jako např. příkaz zasláný z vývojového prostředí. Tato služba také umožňuje zobrazení a uložení všech příchozích a odchozích příkazů. Kapacita záznamů, kterou IQRF Cloud nabízí v bezplatné verzi, je 1000 záznamů. Při pokusu o uložení nového příkazu, který je přes limit, je tento záznam uložen a nejstarší záznam je v databázi smazán. [26]

Tyto komunikační možnosti, mohou být realizovány pomocí uživatelského rozhraní na webové stránce („www.cloud.iqrf.org“), které umožňuje zaslat příkaz na bránu a které umožňuje exportovat uložená data do souboru (např. do CSV). [26]

Druhou metodou, jak tyto komunikační možnosti aplikovat, je pomocí IQRF Cloud API. Tato přístupová metoda je aplikována pomocí naprogramované aplikace. Tento přístup je vhodný při programování vlastního řešení (automatické dozování se uzlů, na data ze senzorů). [26]

6 Ukázka využití IQRF v domácnosti

(V této kapitole jsou využity znalosti, zjištěné z IQRF dokumentace [27])

V této části bakalářské práce je nastíněno praktické využití IQRF technologie. Pro demonstraci jsou využity tři vývojářské moduly (DK-EVAL-04A), jeden senzorový modul (DDC-SE-01), jeden programovací modul (CK-USB-04A) a brána (GW-ETH-02A). Vývojářské moduly, programovací modul a brána jsou osazeny vždy jedním IQRF vysílačem. O tom, jestli bude vysílač zastávat funkci koordinátoru, nebo uzlu, rozhodne až cílová konfigurace. Všechny tyto vysílače jsou na začátku stejné, mají po celou dobu testování verzi operačního systému 3.08D a využívají DPA verzi 2.xx. K naprogramování těchto vysílačů, je využito vývojové prostředí IQRF IDE 4.50. Toto vývojové prostředí slouží také ke kompletnímu ovládní IQMESH sítě (koordinátoru).

Pro správnou funkčnost IQMESH sítě a vrstvy DPA, jsou využity soubory a programy od společnosti IQRF. Písemné povolení k využívání těchto zdrojových kódů za účelem vypracování této bakalářské práce, je součástí příloh pod číslem 1.

K vývoji praktického řešení je použit programovací přístup DPA. Tento přístup je doporučený od výrobce, neboť, jak je popsáno v kapitole č. 5.4.2 a č. 5.5, DPA přístup umožňuje vytvořit a využívat IQMESH síť.

IQRF platforma má několik zásad, jak vytvořit program, aby byl po zkompilování do čitelné podoby pro vysílače správný. Soubor je zkompilován do „HEX“ souboru a nahrán do paměti vysílače. Zkompilování a nahrání souboru se provádí ve vývojové prostředí IQRF IDE.

6.1 Vytvoření IQRF IoT Sítě

Prvním krokem je vytvoření IQMESH sítě, kde bude demonstrováno využití IQRF technologie v domácnosti. Její vytvoření a správa bude probíhat ve vývojovém prostředí IQRF IDE.

Základní vlastnosti nové sítě jsou vydefinovány v záložce: „OS“, která se nachází v souboru: „DPA-config.xml“. Určující jsou zde charakteristiky: rádiová frekvence a rádiový kanál, které musí být shodné na všech vysílačích v síti. Rádiová frekvence je nastavena dle všeobecného oprávnění vydané Českým telekomunikačním úřadem č. VO-R/10/11.2016-13 na 868 MHz.

6.1.1 Nahrání základních komponent do IQRF vysílačů

Konfigurace jednotlivého vysílače je přizpůsobena k roli, kterou vysílač v síti zastává. Součástí konfigurace je výběr vhodného plug-in souboru, což je soubor, který obsahuje mimo jiné i operace, bez kterých by nebylo možné docílit komunikace pomocí IQMESH protokolu. Dále je zapotřebí správně nastavit zdroje v záložce: „HWP“, která se nachází v souboru: „DPA-config.xml“. Většina z nabízených zdrojů jako jsou LED diody, interní a externí EEPROM, MCU RAM, vstupní a výstupní zařízení nebo teploměr jsou pro všechny role zapnuty. V této záložce je i subkategorie RF, kde lze například omezit sílu rádiové antény apod.

V případě vysílače-koordinátoru, je nahrán do jeho paměti Plug-in soubor s názvem GeneralHWP-Coordinator.iqrf. Tento soubor je vhodně vybrán, protože obsahuje všechny operace, které potřebuje ke své práci koordinátor a navíc obsahuje i některé operace, které jiné dodávané Plug-in soubory neobsahují a nebylo by tak možné

zprovoznit vlastní CustomDpaHandler. V záložce HWP je pro koordinátory doporučeno využít FRC funkcionality, která umožňuje sbírat data z vícero uzlů zároveň.

V případě vysílače-uzlu je do jeho paměti nahrán Plug-in soubor s názvem: GeneralHWP-Node.iqrf. Tento soubor obsahuje operace, potřebné pro komunikaci s koordinátorem v IQMESH síti.

Do paměti vysílače je možné nahrán soubor z kategorie: „CustomDpaHandler“, který obsahuje naprogramovanou funkcionalitu (např: čtení dat z externích senzorů). V záložce HWP je poté pro vysílače s externími funkcionalitami vybrána položka „Custom DPA Handler“, aby vysílač věděl, že obsahuje i funkce, nad rámec základních (obsažených v plug-in souboru).

6.1.2 Zprovoznění IQMESH sítě

Zprovoznění IQMESH sítě je provedeno u koordinátoru připojenému k vývojovému prostředí nebo bráně připojené k vývojovému prostředí. Ve vývojovém prostředí se nachází okno s názvem: “IQMESH Network Manager“. V tomto okně, na záložce „IQMESH“ dochází nejprve k připojení jednotlivých uzlů ke koordinátoru nebo bráně. Po fyzickém rozmístění uzlů na místa, kde budou snímat hodnoty, je spuštěna funkce „Discover“, ta zajistí vytvoření nejlepší trasy mezi uzly a koordinátorem (bránou).

Po připojení uzlů ke koordinátoru (bráně), je možnost využít ve vývojovém prostředí tabulkového pohledu na všechny připojené uzly a zjistit si například stav baterie nebo adresu jednotlivého uzlu.

6.2 Rozsvícení diod na zařízeních, při nízké hladině světla

Tato ukázka praktického využití je rozdělena do dvou částí. Se spojením tvoří danou funkcionalitu. První část zahrnuje informace o čtení dat z fotorezistoru. A druhá část obsahuje informace o rozesílání paketu, na příslušné uzly v síti.

Zdrojové kódy využití v této ukázce, jsou součástí příloh, pod číslem 2.

6.2.1 Založení základní struktury CustomDPAHandleru

Program, který je v této podkapitole popsán, se jmenuje „UzelFotorezistor.c“. A nachází se v příloze č. 2. Ve složce PrvniUkazka\PraktickaUkazka\DPA\CustomDPAHandlers.

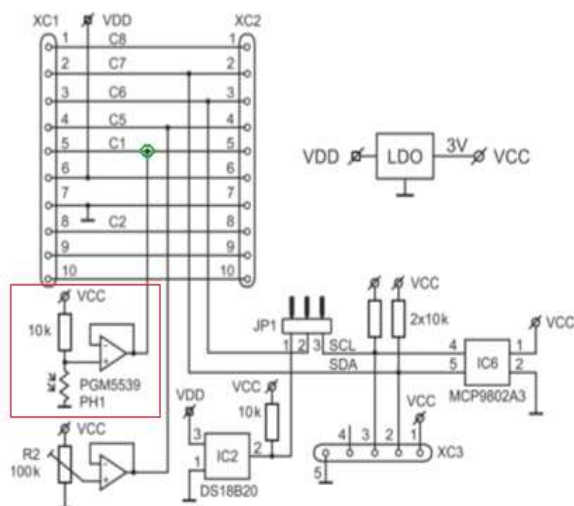
Aby byla dodržena pravidla pro správnou strukturalizaci kódu, je využita šablona, která je ke stažení na internetové stránce s dokumentací. Na této šabloně lze nalézt importování základních IQRF a DPA datových struktur, viz *obrázek 11*. Tyto struktury lze chápat jako předdefinované proměnné, předdefinované struktury příkazů, a také přednastavené základní hodnoty. Jedná se tak o zjednodušení pro vývojáře, neboť například každý příchozí DPA příkaz je pomocí těchto datových struktur automaticky rozdělen do proměnných.

```
#include "template-basic.h"
#include "DPA.h"
#include "DPAcustomHandler.h"
```

Obrázek 11: Ukázka implementace základní struktury [27]

6.2.2 Snímání dat ze sensorového modulu

Na *obrázku 12*, je zjednodušené schéma sensorového modulu, na kterém je červeně označen fotorezistor a zeleně označen pin, kterým fotorezistor do vývojářského modulu zasílá data (C1). Toto schéma slouží ke zjištění, které konektory je potřeba na uzlu probudit pro snímání dat.



Obrázek 12: Zjednodušené schéma sensorového modulu [28]

Z důvodu importování hlavičky "DPA.h", jsou všechna spojení, mezi sensorovým modulem a vývojářským modulem vypnuta (hodnota všech bitů v proměnné TRISA, která spravuje IQRF DDC je 0).

Z tohoto důvodu je vytvořena eventualita „DpaEvent_Init“, která se spustí vždy při zapnutí zařízení. V této eventualitě dojde k nastavení požadovaných bitů v proměnné TRISA na 1, viz *obrázek 13*. Tímto krokem je zapnut fotorezistor, a později z něj půjdou číst data.

```
case DpaEvent_Init:
    TRISA.0 = 1;
    TRISA.5 = 1;
```

Obrázek 13: Ukázka zapnutí požadovaného portu na IQRF DDC

Zdroj: vlastní zpracování

Před samotným čtením dat ze sensorů, musí být ještě ověřena správnost DPA příkazu, viz *obrázek 14*. Každý DPA příkaz, který si žádá číst data ze sensorů, má od výrobce předurčený formát. Jestliže by nebyla splněna jakákoli podmínka, nebyl by kód spuštěn (např. v případě rozdílného _PNUM) nebo by byla v konzoli zobrazena chybová hláška se špatnou hodnotou v příslušném parametru.

```
if ( _PNUM == PNUM_USER)
{
    if ( _PCMD != 0 )
        DpaApiReturnPeripheralError( ERROR_PCMD );
    if ( _DpaDataLength != 0 )
        DpaApiReturnPeripheralError( ERROR_DATA_LEN );
}
```

Obrázek 14: Kontrola přijatého DPA příkazu) [27]

Jestliže obdrženy DPA paket je strukturálně v pořádku, a je určen pro tento program, pokračuje inicializace analog-digitálního převodníku. Ten se nachází na adrese ANSELA.0, dle dokumentace musí být nastaven analog-digitální převodník na příslušné hodnoty, viz *obrázek 15*.

Po nastavení převodníku je uzel nucen vyčkat přesně 2 sec., aby fotorezistor zpracoval dopadající světlo. Následně se spustí analog-digitální převodník, který přečte hodnotu dopadajícího světla na fotorezistor.


```
ANSELA.0 = 1;  
ADCON0 = 0b0.00000.01;  
ADCON1 = 0b1.10100.00;
```

Obrázek 15: Zapnutí a nastavení analog-digitálního převodníku) [27]

Po dokončení práce analog-digitálního převodníku jsou zaslány data z paměti převodníku do konzole. Data, která jsou na výstupu z převodníku, mají celkovou velikost 10 bitů, a proto jsou rozdělena do dvou datových pamětí (prvních 8 bitů je uloženo v proměnné ADRESL a zbývající 2 bity jsou uloženy v proměnné ADRESH). Při čtení ADRESH, jsou z paměti této proměnné, požadovány pouze první 2 bity (dekadicky se jedná o číslo 3). Tyto data jsou umístěna do paměti DPA příkazu, viz *obrázek 16*. Tento příkaz je následně odeslán jako odpověď na příchozí příkaz z koordinátoru.

```
_DpaMessage.Response.PData[1] = ADRESH & 0x03;  
_DpaMessage.Response.PData[0] = ADRESL;  
_DpaDataLength = sizeof( uns16 );
```

Obrázek 16: Zapnutí a nastavení analog-digitálního převodníku) [27]

6.2.3 Analýza dat na koordinátoru

Program, který je v této podkapitole popsán se jmenuje „Koordinator.c“. A nachází se v příloze č. 2. Ve složce PrvniUkazka\PraktickaUkazka\DPA\CustomDPAHandlers.

Zdrojový kód pro analýzu dat z fotorezistoru je do koordinátoru vložen jako CustomDPAHandler funkcionalita. Koordinátor tuto funkcionalitu spouští vždy na začátku (když je spuštěn) a poté může být tato funkcionalita znovu vyvolána DPA příkazem, například z vývojového prostředí. V této ukázce je tato funkcionalita v kódu zacyklena, aby se po dokončení znovu spustila.

Stejně jako v případě CustomDPAHandleru, který je vytvořen pro snímání dat z fotorezistoru, musí být implementovány potřebné hlavičkové soubory. V případě programování funkcionality pro koordinátor je potřeba zavést oproti uzlovému záhlaví ještě speciální hlavičku pro koordinátory.

Po implementaci hlavičky jsou zavedeny stavové ukazatele. Jedním z ukazatelů je hodnota „stav_ZaslatDotaz“. Tento ukazatel usměrňuje algoritmus, aby následující akce vedla k zaslání dotazu na uzel s fotorezistorem. Po zaslání dotazu se využije

stavový ukazatel „stav_CekatNaOdpoved“, který slouží k vymezení časomíry do kdy musí uzel odpovědět.

Stejně jako v případě uzlu, je operace CustomDpaHandler() umístěna jako první operace v programu. Společně s ní jsou nadefinovány i některé proměnné, viz *obrázek 17*. Každá proměnná má v programu jiný účel. Všechny mají ale stejný datový typ. Obecně lze využít u proměnných dva datové typy. Uns8 s 8 bity nebo uns16 s 16 bity přidělené paměti.

```
bit CustomDpaHandler()
{
    clrwdt();

    // Adresa uzlu s fotorezistorem
    static uns8 adresaUzlu;
    // Stav koordinátoru
    static uns8 stav;
    // Počet světlých chvil
    uns8 pocetSnimaniSvetla;
    // Počet zamračených chvil
    uns8 pocetSnimaniTmy;
```

Obrázek 17: Vydefinování proměnných

Zdroj: vlastní zpracování

Program je v této metodě rozvětven příznakem switch a case. Tento příznak se rozhoduje podle události, která nastala na vysílači (přijetí paketu, zapnutí vysílače, ...), viz *obrázek 18*.

První podmínkou která nastane při spuštění vysílače, je: „DpaEvent_Init“. Tato větev programu slouží k inicializaci základních proměnných.

```
switch ( GetDpaEvent() )
{
    case DpaEvent_Init:
        ...
        break;
```

Obrázek 18: Příznak switch a case [27]

Další událost, která může na koordinátoru nastat a rozvětvit algoritmus, je: „DpaEvent_Idle“. Vysílač na prvním řádku této větve ověří dokončení časovače. Časovač je reprezentován datovou strukturou „DpaTicks“. Pokud je bit „DpaTicks.15“ jiný než „0“, tak časovač stále běží a není proveden žádný z kroků v této větvi

programu. Po vyčkání na konec časovače, je vytvořen opět další příznak switch a case, který se rozhoduje podle stavové proměnné „stav“.

Jestliže si stav proměnné vyžaduje zaslat dotaz na uzel s fotorezistorem, je ověřena podmínka, že uzel s adresou č. 1 je připojen do IQMESH sítě. Pokud ano, je zaslán požadavek na poskytnutí dat z fotorezistorového senzoru. Po odeslání DPA příkazu se změní stav na „stav_CekatNaOdpoved“ a nastaví se časovač na přibližně 25 sec., viz *obrázek 19*. Tento časovač zajišťuje pozdržení odeslání dalšího DPA příkazu na tento uzel.

Může nastat i situace, že uzel není připojen ke koordinátoru, v takovém případě se vyčká 10 sec. a program se vrátí zpět na začátek a pokusí se znovu zaslat požadavek o hodnotu naměřenou na fotorezistoru.

Jestliže nebude neobdržena odpověď od uzlu, změní se stav kontroléru na „stav_ZaslatDotaz“ a program se vrací zpět na zasílání dotazu.

Poslední událost kterou koordinátor zachycuje, je: DpaEvent_ReceiveDpaResponse.

Tato událost nastane, když koordinátor obdrží DPA příkaz.

Prvním krokem před analýzou dat z fotorezistoru je ověření, zda se jedná o DPA příkaz, který byl zaslán z uzlu č. 1 a jestli hlavníčka příkazu obsahuje správné náležitosti. Jestliže je toto ověření v pořádku, následuje zpracování příchozích dat.

Data z fotorezistoru mají celkovou velikost 10 bitů. Pro uložení těchto dat je vytvořena proměnná „hodnotaSvetla“ o velikosti 16 bitů, která data z fotorezistoru pojme. Zajímavostí je naplnění jednotlivých bitů v proměnné. Prvních 8 bitů z levé strany je naplněno po dodání „high8“ za proměnou a následující 8 bitů je naplněno jako dodatek „low8“, viz *obrázek 20*.

Pozorováním bylo určeno, že ideální hodnota, při kterém je vhodné zapnout nebo vypnout světlo je přibližně 800. Aby však rozsvícení světel a zhasínání světel nebylo náchylné na ojedinělé jevy (např. nasměrování rozsvícené kapesní svítilny na foto rezistor), je zapotřebí, aby byla naměřena vysoká nebo nízká hodnota světla alespoň 6x krát.

```

case DpaEvent_Idle:
    if ( DpaTicks.15 != 0 ){
        switch ( stav )
        {
        case stav_ZaslatDotaz:
            adresaUzlu = 1;
            _NADR = adresaUzlu;
            _NADRhigh = 0;
            _PNUM = 32;
            _PCMD = 0;
            _HWPID = 65535;
            _DpaDataLength = 0;
            DpaApiRfTxDpaPacketCoordinator();
            ...
        case stav_CekatNaOdpoved:
            if ( DpaTicks.15 != 0 )
                stav = stav_ZaslatDotaz;
            break;

```

Obrázek 19: Odeslání požadavku na uzel s fotorezistorem

Zdroj: vlastní zpracování

```

case DpaEvent_ReceiveDpaResponse:
    ...
    uns16 hodnotaSvetla;
    hodnotaSvetla.low8 = _DpaMessage.Response.PData[0];
    hodnotaSvetla.high8 = _DpaMessage.Response.PData[1];

```

Obrázek 20: Naplnění 16 bitové proměnné hodnotou z fotorezistoru

Zdroj: vlastní zpracování

6.2.4 Zprovoznění 1. ukázky

Každý vysílač je bezpečně připojen do programátorského modulu a po nastavení, jak je popsáno v kapitole 6.1.1 a *tabulce 2*, je přeprogramován a vložen do příslušného modulu. *Tabulka 2* slouží k sumarizaci, jaká zařízení jsou v ukázce použity a jakou mají tyto zařízení činnost v systému.

Po naprogramování všech vysílačů se připojí do programovacího modulu koordinátor a vytvoří se IQMESH síť, dle kapitoly 6.1.2. Zařízení jsou přidávána postupně, aby obdržela příslušnou adresu dle *tabulky 2*.

Tabulka 2: Zařízení v IQMESH síti 1. Ukázka

Adresa	Role (činnost)	Modul	Použité soubory
0	Koordinátor	CK-USB-04A	GeneralHWP-Coordinator.iqrf, „Koordinator.c“
1	Uzel (snímač), Senzor	DK-EVAL-04A, DDC-SE-01	GeneralHWP-Node.iqrf, „UzelFotorezistor.c“
2	Uzel (světlo)	DK-EVAL-04A	GeneralHWP-Node.iqrf
3	Uzel (světlo)	DK-EVAL-04A	GeneralHWP-Node.iqrf

Zdroj: vlastní zpracování

6.3 Ukázka inteligentní regulace teploty

(V této podkapitole jsou využity znalosti IQRF Cloud. [26] A IQRF Gateway. [29])

V této ukázce jsou snímány teploty z teploměrů na dvou vysílačích. Tato data jsou pomocí brány zasílána do vzdálené databáze (IQRF Cloud). Z této databáze jsou následně programem napsaným v PHP stahována a analyzována. Výstupy této analýzy jsou, graf teplot a regulace led diod na uzlu č. 3. Led diody zastávají funkci topného tělesa (pro znázornění zapnutí (svítí zelená led dioda) nebo vypnutí (svítí červená led dioda)).

Na rozdíl od první ukázky, nebude analýza probíhat na koordinátoru, ale na domácím počítači, na kterém je nainstalován program XAMPP.

Program XAMPP byl pro tuto ukázkou zvolen, protože umožňuje spouštět webové funkcionality napsané v jazyce PHP a zároveň obsahuje i MySQL databázi.

Zdrojové kódy týkající se této ukázky, se nachází v příloze č. 2. Ve složce DruhaUkazka.

6.3.1 Konfigurace Uzlů a Brány

Uzly, které jsou použity pro tento pokus, jsou nakonfigurovány, dle znalostí z kapitoly 6.1.1 a tabulky 3.

Brána je před nastavením ve vývojovém prostředí IDE připojena síťovým kabelem do lokální sítě. Tato lokální síť zároveň poskytuje přístup k internetu, který je potřebný pro další část této ukázky.

Při nastavení brány se postupuje dle návodu od výrobce a všechny vlastnosti se ponechávají ve výchozím nastavení. Na kartě „Operation mode“ se nachází důležitá nastavení pro správnou funkčnost brány. Zde musí být mód brány nastaven na „Datalogger“ a povolen přístup na IQRF Cloud.

Na kartě „Lan“ se ve vlastnosti „Host name“ nachází jméno brány. Pro konfiguraci plug-in souboru musí být do internetového vyhledávače zadáno jméno brány a po přihlášení se v záložce „TR upload“ nachází prostor pro nahrání plug-in souboru. Plug-in soubor, který je do brány nahrán, je totožný, jako v případě koordinátoru v kapitole 6.1.1. Po nahrání tohoto souboru do brány, je brána připojena k vývojovému prostředí a proběhne vytvoření IQMESH sítě. Uzly jsou do IQMESH připojovány dle *tabulky 3*.

Tabulka 3: Zařízení v IQMESH síti 2. Ukázka

Adresa	Role (činnost)	Modul	Použité soubory
0	Brána	GW-ETH-02A	GeneralHWP-Coordinator.iqrf
1	Uzel (snímač), Senzor	DK-EVAL-04A, DDC-SE-01	GeneralHWP-Node.iqrf, UzelFotorezistor.c
2	Uzel (snímač)	DK-EVAL-04A	GeneralHWP-Node.iqrf
3	Uzel (regulátor)	DK-EVAL-04A	GeneralHWP-Node.iqrf

Zdroj: vlastní zpracování

6.3.2 Komunikace mezi IQRF Cloudem a bránou

Pro navázání komunikace mezi IQRF Cloud serverem a bránou, musí být nejprve brána připojena k IQRF Cloud a poté může přijímat příkazy z tohoto serveru a zasílat odpovědi na tento server. Při obou komunikacích, si IQRF Cloud do své databáze ukládá čas přijetí příkazu, strukturu DPA příkazu a identifikační číslo zasláné nebo přijaté zprávy.

V dokumentaci, kterou IQRF Alliance na webových stránkách nabízí, se nachází struktura PHP kódu, který vytváří HTML příkazy pro stažení dat nebo nahání data na IQRF Cloud pomocí IQRF Cloud API.

Jestliže se jedná o požadavek na obdržení dat nebo odeslání dat, je struktura zdrojového kódu velmi podobná. V obou požadavcích se nachází parametry „requestParams“, „apiKey“ a „publicIp“. Zbytek struktury kódu se stará o zakódování dat a vytvoření validního HTML požadavku, který je zaslán IQRF Cloud API.

Parametr „requestParams“ je pole hodnot, které obsahuje klíčové parametry, jak má IQRF Cloud zpracovávat přijatý příkaz. Mezi parametry, které jsou u obou požadavků stejné patří: verze DPA (č. 2), uživatelské přihlašovací jméno (Mrnin) a identifikátor brány (12003FAB). Naopak rozdílný je parametr „cmd“, který udává, jak má být HTML příkaz zpracován.

Dalším parametrem je „apiKey“, tento parametr je snadno dohledatelný na IQRF Cloud webové stránce v sekci „profil uživatele“ a slouží k jednoznačné identifikaci uživatele. Parametr „publicIp“, je zde pro ověření, že vygenerovaný požadavek na Cloud půjde přijmout jen z konkrétní počítačové sítě. Oba tyto parametry jsou povinné pro zajištění komunikace mezi domácím počítačem a IQRF serverem.

Pro generování HTML příkazu se musí spojit proměnné „requestParams“, „apiKey“, „publicIP“ a aktuální čas. Toto spojení je dále zakódováno pomocí MD5 šifry a připojeno na statickou HTML adresu od IQRF Cloud API. Takto vygenerovaný příkaz je funkční pouze následujících 10 sekund.

6.3.2.1 Odeslání požadavku na bránu pomocí IQRF Cloud

Pro vytvoření a odeslání HTML požadavku, který zašle na IQRF Cloud příkaz, aby se brána dotázala uzlů na data ze senzorů, slouží třída „UpdateSensorData“. Tato třída obsahuje metodu „UpdateSensorData“, která při volání vyžaduje pole hodnot s požadavky (vstupní parametr). V algoritmu této metody se vytvoří cyklus, který jednotlivé požadavky ze vstupního pole odešle.

Pro získání dat ze senzorů, slouží třída „UpdateStaticData.php“. Tato třída při volání vrátí pole hodnot, obsahující DPA příkazy, které musí být z brány odeslány na jednotlivé uzly. Toto pole slouží jako vstupní parametr při volání metody „UpdateSensorData“, viz *obrázek 21*.

```
return array('01000A00FFFF', '02000A00FFFF', '01002000FFFF'); ...
$requestParams = array(
    'ver' => '2',
    'cmd' => 'uplc',
    'uid' => 'Mrnin',
    'gid' => '12003FAB',
    'data' => $data,);
```

Obrázek 21: Struktura pole s DPA příkazy a parametr „requestParams“

Zdroj: vlastní zpracování

6.3.2.2 Získání dat ze senzorů uložených na IQRF Cloud

K získávání dat z Cloudu slouží třída „SensorData.php“ a její operace „getAllData“. Při volání této metody je vyžadován vstupní parametr „count“, kterým je určeno, kolik příchozích příkazů, má být z Cloudu načteno. Jestliže by tento parametr nebyl uveden, IQRF Cloud by standartně zobrazil všechny příchozí pakety.

Tato metoda ve svém průběhu vytváří HTML příkaz. Po spuštění tohoto HTML příkazu je zobrazeno požadované množství příchozích příkazů. Tyto příchozí příkazy jsou součástí jedné věty, kdy jednotlivé hodnoty jsou odděleny středníkem.

Tato metoda načte obsah této stránky a tuto dlouhou větu postupně rozdělí na jednotlivé DPA příkazy, které byly přijaté ze senzorů. Pro přehlednost, je v této třídě implementována třída „SensorDataFileStructure.php“, která reprezentuje datovou strukturu jednoho příkazu.

Brána při přijmutí příkazu z Cloudu vždy potvrdí, že operace byla odeslána na uzel. Později brána odpoví Cloudu znovu, jestliže obdrží zprávu od uzlu zpět. Tato operace filtruje všechny potvrzující zprávy, které byly z brány na Cloudu odeslány a zpracovává pouze ty, které jsou odpověďmi ze senzorů.

Všechny odpovědi z prvního uzlu a druhého uzlu, jsou uloženy do pole, které obsahuje instance třídy „SensorDataFileStructure.php“. Toto pole je při dokončení této metody i výstupní hodnotou.

6.3.3 Analýza dat ze senzorů

Aby mohla být snímaná data analyzována, musí existovat zdroj dat, se kterými budou snímaná data porovnávána. Zdrojová data pro porovnávání, jsou uložena v MySQL databázi, kterou spravuje program XAMPP. V této databázi existuje pouze jeden list

s názvem „sensor_days“, který obsahuje sloupce s názvem „id“, „hod“, „data“ a „senzor“. Soubor pro vytvoření této ukázkové databáze je také součástí přílohy č. 3, ve složce DruhaUkazka pod názvem „sensor_days.sql“.

Sloupeček „id“ slouží k jednoznačné identifikaci daného řádku a zastává tak funkci primárního klíče. Hodnoty ve sloupečích „hod“ a „data“ se týkají vymezení požadované teploty („data“) v konkrétní hodinu („hod“). Tabulka tak obsahuje pro každou hodinu jeden řádek (konkrétní hodnoty jsou od 0 do 23). Každý řádek s hodinou obsahuje hodnotu („data“), které udává minimální požadovanou teplotu v místnosti. Místnost je definována čtvrtým sloupečkem („senzor“), který je vyplněn na každém řádku hodnotou 1, neboť právě tento senzor se stará o měření teplot v místnosti, viz *tabulka 4*.

Tabulka 4: První 3 záznamy v listu sensor_days

id	hod	data	senzor
1	0	22.50	1
3	1	28.00	1
4	2	27.00	1

Zdroj: vlastní zpracování

Analýza dat probíhá v třídě „Graph.php“. Tato třída zpracovává i grafickou vizualizaci snímaných dat, které se věnuje následující podkapitola. Za účelem analýzy senzorových dat, tato třída obsahuje metodu „compareData“. Tato funkce při zavolání odešle požadavek, který byl zmiňován v kapitole 6.3.2.1 a za účelem analýzy si načte posledních 50 příchozích příkazů (využije funkci popsanou v kapitole 6.3.2.2). Po načtení těchto dat, si program zjistí, kolik je aktuálně hodin a do databáze je odeslán požadavek na zjištění, jaká je minimální požadovaná teplota v tuto hodinu. Příchozí požadavek z databáze je načten do konkrétních proměnných (například pro minimální teplotu je proměnná „dateSensor“). Tento krok je zobrazen na *obrázku 22*.

Po načtení požadovaných dat, program vyhledá poslední přijatý příkaz ze senzoru č. 1, za podmínky, že hodina, kdy byl příkaz přijat, je stejná jako aktuální hodina. Do proměnné „dataFromSensor“ se načte hodnota „Pdata“ z nalezeného příkazu. Jestliže není načten žádný příkaz, ponechá se proměnná na maximální velikosti (PHP_INT_MAX), aby se v následujícím kroku topení „nezapnulo“. Program následně porovná naměřenou hodnotu s požadovanou hodnotou. Pokud je naměřená hodnota

menší, tak rozsvítí diodu na uzlu č. 3 zeleně, a jestliže je větší, tak rozsvítí diodu na uzlu č. 3 červeně.

```
$timeNow = date("G");  
$data = $this->DbConnection()->query("SELECT * FROM sensor_days WHERE  
hod = $timeNow");  
$row = $data->fetch_assoc();  
...  
$dateSensor = $row['senzor'];
```

Obrázek 22: Načítání minimální požadované teploty v aktuální hodině

Zdroj: vlastní zpracování

6.3.4 Vizuální reprezentace dat ze senzorů

Data ze senzorů jsou zobrazena ve dvou grafech. První z nich reprezentuje data, která naměřil fotorezistor na uzlu číslo 1., viz *obrázek 24*. Druhý graf slouží k reprezentaci naměřených teplot ze senzorů číslo 1. a číslo 2, viz *obrázek 25*. Každý graf obsahuje dvě osy. Osa horizontální slouží k reprezentaci času a osa vertikální slouží k reprezentaci naměřené teploty nebo svítivosti. Graf je vytvořen za pomoci knihoven „Chart.js“ a „jquery-3.3.1.js“.

Vstupy do těchto grafů tvoří několik polí s hodnotami. Do prvního grafu je ze zadní části programu zasíláno pole s časy a pole s naměřenou svítivostí. Do druhého grafu je zasíláno pole s teplotou naměřenou na senzoru číslo 1., pole s teplotou naměřenou na senzoru číslo 2. a pole s časy, kdy byly jednotlivé hodnoty naměřeny.

Zadní část této části programu probíhá v třídě „Graph.php“ a konkrétně v její metodě „loadData“. V této funkci jsou nejprve vytvořena jednotlivá pole, které tvoří vstupy do grafů, viz *obrázek 23*. Následuje načtení posledních 500 příchozí příkazů z IQRF Cloudu (je použita operace popsána v kapitole 6.3.2.2) a roztřídění těchto příkazů do konkrétních polí.

```
$label_1_Light = array();  
$data_1_Light = array();  
$label_Temperature = array();  
$data_1_Temperature = array();  
$data_2_Temperature = array();
```

Obrázek 23: Vydefinování polí, sloužících jako vstupní parametr do grafů

Zdroj: vlastní zpracování

Pole s časy jsou u obou grafů rozdílná. V případě reprezentace dat z fotorezistoru, je pole s časy vyplňováno současně s polem obsahující naměřené hodnoty z fotorezistoru. V případě grafu s teplotami je potřeba se vypořádat se situací, kde jeden ze senzorů může hodnotu zaslat i s několika sekundovým zpožděním. Z tohoto důvodu, je čas přijetí aktuálně zpracovávaného příkazu porovnáván s poslední hodnotou v poli s časy. Jestliže jsou hodnoty rozdílné o více než mez, která je stanovena na 60 sec., je do pole s časy přidán čas, který nese zpracovávaný paket. Poté je do pole s teplotami u zpracovávaného senzoru přidána teplota, kterou nese zpracovávaný příkaz a do pole, které spravuje druhý senzor, je přidána prázdná hodnota.

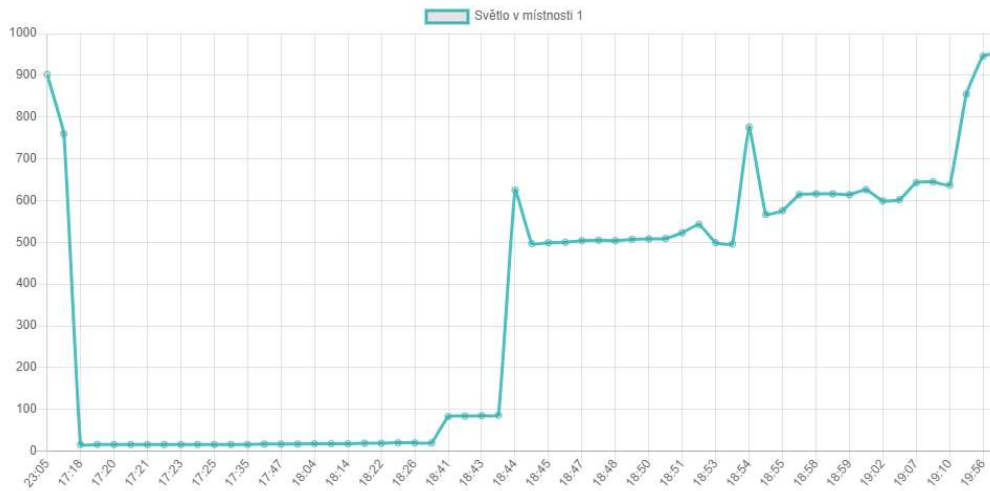
Jestliže ale rozdíl mezi teplotou zpracovávaného příkazu a teplotou v poli teplot je nižší jak mezní hodnota, znamená to, že tento čas už je v poli již obsažen (tento čas byl nejspíše přidán při zpracovávání příkazu z opačného senzoru). Program na tuto akci zareaguje odstraněním poslední informace z pole teplot u zpracovávaného senzoru. Tento krok je proveden, neboť při přidávání nové časové hodnoty jsou vyplněna obě pole. Na závěr je přidána naměřená teplota do pole, které souvisí se zpracovávaným senzorem.

Po zpracování všech příkazů, jsou odeslány příslušná pole do grafů.

Na *obrázku 25*, jsou znázorněné naměřené teploty v době testování programu. Venkovní senzor byl v rámci testů dotázán o svou teplotu o 120 sec. dříve, než byly později dotázány oba senzory současně (venkovní senzor a domácí senzor). Jelikož, tento rozdíl je vyšší než mez (60 sec.), je v čase 20:03 na grafu pouze naměřená teplota z venkovního teploměru.

Dalším zajímavým okamžikem byl čas od 17:24 do 17:37, v tento okamžik byl venkovní senzor vypnut.

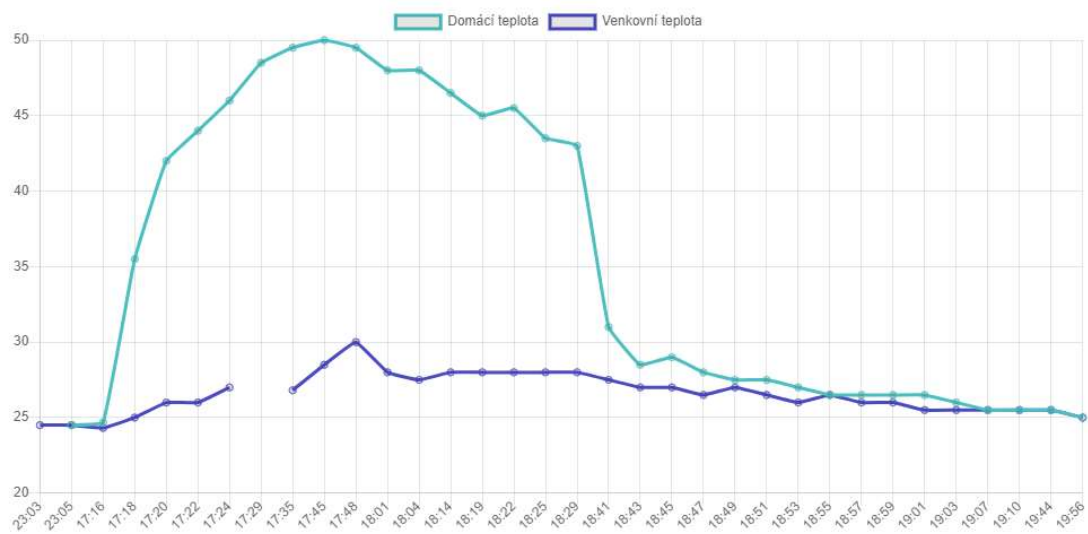
Graf Světla



Obrázek 24: Výstupní graf sensorických hodnot z fotorezistoru)

Zdroj: vlastní zpracování

Graf Teplot



Obrázek 25: Výstupní graf sensorických hodnot z teploměru)

Zdroj: vlastní zpracování

7 Shrnutí výsledků

V praktické části byl první pokus zaměřen na zapínání a vypínání led diod uzlů, podle hodnoty naměřené na fotorezistoru. Při tomto pokusu bylo využito lokální řízení uzlů a programátorský přístup DPA. Pro vytvoření této ukázky bylo zapotřebí zjistit, jak funguje operační systém IQRF z dokumentace. V dokumentaci jsou jednotlivé funkce DPA přístupu s podrobným popisem.

První ukázka demonstrovala typ přístupu DPA, zde byla ověřena programovatelnost těchto vysílačů. Popis celé struktury byl zjištěním funkcionality přístupu DPA. Na základě výsledků z tohoto procesu, by nebyl sestavený program vhodný pro praktické využití. Technické prvky z prvního pokusu byly vybrány tak, aby sestavený přístup DPA fungoval a demonstroval tuto metodu.

V druhé ukázce je nastíněno využívání IQRF Cloud Serveru s přenesením logiky na domácí počítač (globální řízení). V tomto pokusu se podařilo zprovoznit základ inteligentního termostatu. Ten byl regulován podle teploty v místnosti a předem požadované teploty. Teplota byla nadefinována v externí MYSQL databázi.

Součástí tohoto řešení byla také vizuální prezentace dat ze senzorů.

V tomto pokusu se nachází příležitost, jak ukázku vylepšit. Dané vylepšení spočívá ve vytvoření programu pro editaci požadovaných teplot. Stávající struktura programu tuto možnost nenabízí a má pouze editační strukturu ve správci databáze.

8 Závěry a doporučení

Internet věcí je velmi zajímavá technologie, kde nachází uplatnění v mnoha odvětvích. Využití jednoho typu senzoru lze nalézt ve vícero aplikacích. Tato technologie se velmi rychle rozšířila z průmyslové sféry až po rodinné domy a domácnosti. Brzy se tato technologie ještě více rozšíří do měst a dopravy, kde bude využita ke zvýšení bezpečnosti osob a plynulosti navazujících infrastruktur.

Při vydefinování požadavků na uzly, byla v uživatelských omezeních nastíněna potřeba řešit bezpečnost komunikace. Touto problematikou by se mohla zabývat navazující práce. Narušení nebo elektronické odposlouchávání bezdrátových sítí by potenciálně mohlo zkomplikovat zavádění této technologie například do státní sféry (hlídání hranic s nežádoucím pohybem osob a tím ochrany státu).

Obsah čtvrté a páté kapitoly ukázal, že existují technologie, které mají podobné vlastnosti jako technologie IQRF. Technologické principy však těchto platforem jsou odlišené. IQRF technologie nabízí řídit uzly za pomoci programovacího DPA přístupu. Tento přístup umožňuje vytvářet a spravovat IQMESH síť. Ta podporuje několik patentovaných funkcí, jako například Discovery (pro vytvoření spojení mezi uzly a koordinátorem) nebo Multi-bonding mechanismus (pro možnost využití vysílače jako uzlu v jedné IQMESH síti a jako koordinátoru v jiné IQMESH síti).

Provedené pokusy byly demonstrací chytrých řešení na platformě IQRF. Ukázaly její technickou funkcionalitu. U obou pokusů se nepodařilo vyřešit uspávání uzlů, kdy musely být velmi často dobíjeny. V budoucnu by bylo užitečné tento problém vyřešit. Tím se otevře možnost dlouhodobé časové zkoušky. Tento nedostatek neměl na výsledky práce vliv a ukázal ověření celé programové smyčky. Senzory se využívaly pouze krátkodobě. Poskytnutá zařízení splnila svůj účel jako základní testovací balíček IQRF platformy.

9 Seznam použité literatury

- [1] HUSSAIN, Fatima. Internet of things: building blocks and business models. New York, NY: Springer Science+Business Media, 2017. ISBN 978-3-319-55404-4.
- [2] Alioto M. Enabling the internet of things. New York, NY: Springer Science+Business Media, 2017. ISBN 978-3-319-51480-2.
- [3] CHEN, Min. RFID technologies for internet of things. New York, NY: Springer Science+Business Media, 2016. ISBN 978-3-319-47354-3.
- [4] Rayes A., Salam S. Internet of things--from hype to reality. New York, NY: Springer Berlin Heidelberg, 2016. ISBN 978-3-319-44858-9.
- [5] Luntovskyy A., Spillner J. (2017) Smart Grid, Internet of Things and Fog Computing. In: Architectural Transformations in Network Services and Distributed Systems. Springer Vieweg, Wiesbaden
- [6] Zemědělství – IoT portál. IoT portál – Brána do světa internetu věcí [online]. Copyright © 2019 [cit. 21.04.2019]. Dostupné z: <https://www.iot-portal.cz/2016/04/18/zemedelstvi/>
- [7] I. WATER RESOURCE ISSUES AND AGRICULTURE. Food and Agriculture Organization of the United Nations [online]. Dostupné z: <http://www.fao.org/3/t0800e/t0800e0a.htm>
- [8] What is ZigBee Technology. Electronics notes [online]. Dostupné z: <http://www.electronics-notes.com/articles/connectivity/zigbee/what-is-zigbee-technology-tutorial.php>
- [9] FARAHANI, Shahn. ZigBee wireless networks and transceivers. Boston: Newnes/Elsevier, c2008. ISBN 0750683937.
- [10] Christopher W. Badenhop, Scott R. Graham, Benjamin W. Ramsey, Barry E. Mullins, Logan O. Mailloux: The Z-Wave routing protocol and its security implications - ScienceDirect. ScienceDirect.com | Science, health and medical journals, full text articles and books. [online]. Copyright © [cit. 21.04.2019]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0167404817300792>
- [11] Jiří Pech : IoT technologie: LoRa a LoRaWAN (3/5) [online]. Copyright © eMan s.r.o. 2019 [cit. 21.04.2019]. Dostupné z: <https://www.eman.cz/blog/iot-technologie-lora-a-lorawan-3-5/>
- [12] IQRF About - IQRF. IQRF - Technology for wireless [online]. Copyright © IQRF Tech s.r.o. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/iqrfabout>
- [13] IQRF Alliance Do it wireless [online]. [cit. 10.08.2018]. Dostupné z: <http://www.iqrfalliance.org/alliance>
- [14] RF. IQRF - Technology for wireless [online]. Copyright © IQRF Tech s.r.o. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/technology/rf>
- [15] VRBA, Radimir, Smart Wireless Communication Platform IQRF, In: researchgate.net [online] KUČHTA, Radek, SULC, Vladislav 1.1.2010, [cit. cit. 12.08.2018] Dostupné z <https://www.researchgate.net/publication/221905403_Smart_Wireless_Communication_Platform_IQRF>. ISSN 1080-2711.
- [16] IQMESH. IQRF - Technology for wireless [online]. Copyright © IQRF Tech s.r.o. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/technology/iqmesh>

- [17] IQMESH features. IQRF - Technology for wireless [online]. Copyright © IQRF Tech s.r.o. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/technology/iqmesh/iqmesh-features>
- [18] Operating system. IQRF - Technology for wireless [online]. Copyright © IQRF Tech s.r.o. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/technology/operating-system>
- [19] IQRF - Technology for wireless. IQRF - Technology for wireless [online]. Copyright © IQRF Tech s.r.o. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/technology/operating-system/os-illustrative-examples/remote-controller>
- [20] Technology - IQRF. IQRF - Technology for wireless [online]. Copyright © IQRF Tech s.r.o. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/technology>
- [21] DPA přístup podrobně – přeloženo a upraveno z: <https://www.iqrf.org/technology/dpa>
- [22] SPURNÁ, Ivona, Bezdrátová technologie IQRF (3), In: atpjournal.sk [online]. 17.6.2016, [cit. 10.08.2018]. Dostupné z: https://www.atpjournal.sk/rubriky/prehladove-clanky/bezdratova-technologie-iqrf-3.html?page_id=23653
- [23] TR-72D. IQRF - Technology for wireless [online]. Copyright © IQRF Tech s.r.o. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/products/transceivers/tr-72d>
- [24] Copyright © IQRF Tech s.r.o., DK-EVAL-04A - User's Guide, [online]. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/products/development-tools/development-kits/dk-eval-04>
- [25] Copyright © IQRF Tech s.r.o., DDC-SE-01- User's Guide, [online]. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/products/development-tools/development-kits/ddc-se-01>
- [26] Copyright © IQRF Tech s.r.o., IQRF Cloud Server – Technical guide, [online]. [cit. 21.04.2019]. Dostupné z: <https://www.iqrf.org/technology/iqrf-cloud>
- [27] IQRF DPA Framework - Technical Guide. IQRF - Technology for wireless [online]. Dostupné z: <https://www.iqrf.org/DpaTechGuide/302/>
- [28] Copyright © IQRF Tech s.r.o., DDC-SE-01- User's Guide, [online]. [cit. 10.08.2018]. Dostupné z: <https://www.iqrf.org/products/development-tools/development-kits/ddc-se-01>
- [29] Copyright © IQRF Tech s.r.o., GW-ETH-02A– User's Guide, [online]. [cit. 21.04.2019]. Dostupné z: <https://www.iqrf.org/products/gateways/gw-eth-02a>

10 Seznam použitých zkratech

IoT	Internet věcí (<i>angl. Internet of Things</i>)
RFID	Identifikace na rádiové frekvenci (<i>angl. Radio Frequency Identification</i>)
RF	Rádiová komunikace
MEMS	MikroElektroMechanický Systém
TCP/IP	Primární přenosový protokol/protokol síťové vrstvy
LAN	místní síť (<i>angl. Local Area Network</i>)
MAC	Media Access Control
WAN	rozlehlá síť (<i>angl. Wide Area Network</i>)
IEEE	Institut pro elektrotechnické a elektronické inženýrství
ITU	Mezinárodní telekomunikační unie
UART	Synchronní / asynchronní sériové rozhraní
HWP	Hardwarový profil
DPA	Přímý periferní přístup (<i>angl. Direct Peripheral Access</i>)
USB	Univerzální sériová sběrnice
DDC	Vývojový řetězec Daisy
IDE	Vývojové prostředí
PHP	Hypertextový preprocesor
API	Rozhraní pro programování aplikace (<i>angl. Application Programming Interface</i>)
HTML	Hypertext Markup Language

11 Přílohy

- [1] Povolení od společnosti IQRF k využívání zdrojových kódů
- [2] CD se zdrojovými kódy (Pro spuštění první ukázky je zapotřebí software IQRF IDE. Pro spuštění druhé ukázky, je zapotřebí software IQRF IDE, upravit zdrojové kódy (ty, které se starají o komunikaci s IQRF Cloud API), software s podporou spuštění PHP souborů a MySQL databázový server)

Dohoda o použití zdrojů informací.

Firma IQRF Tech s.r.o. povoluje panu Tomáši Mrňákovi použít v jeho bakalářské práci Chytrá řešení na platformě IQRF, informace z technické dokumentace a ostatních veřejně dostupných materiálů k technologii IQRF a p.Mrňák se zároveň zavazuje uvést toto ve zdrojích bakalářské práce, poskytnout bakalářskou práci v elektronické formě firmě IQRF Tech s.r.o., a umožnit případné čerpání informací a citovat z této práce firmě IQRF Tech s.r.o.

V Jičíně 18.3.2019


IQRF Tech s.r.o.
DIČ: CZ06317375
Průmyslová 1275, 506 01 Jičín
tel. +420 493 538 125

Jan Vašta, obchodní ředitel
IQRF Tech s.r.o.



Tomáš Mrňák



IQRF Tech s.r.o. Průmyslová 1275, 506 01 Jičín, Czech Republic sales@iqrf.com tel.+420 493 538 125
www.iqrf.com
VAT nr.: CZ06317375 registered by the Regional Court in Hradec Králové Section C, Insert 40027
IQRF Tech s.r.o. is MICROBIS's technological spin-off and successor for all IQRF™ related activities. IQRF Tech s.r.o. is the future of IQRF™