



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

## **LABORATORNÍ ÚLOHA - IQRF MODUL PRO MĚŘENÍ MESH SÍTÍ**

LABORATORY EXERCISE - IQRF MODULE FOR MEASURING MESH NETWORKS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Juraj Goryl**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Lukáš Jablončík**

**BRNO 2024**



# Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

**Student:** Juraj Goryl

**ID:** 240913

**Ročník:** 3

**Akademický rok:** 2023/24

**NÁZEV TÉMATU:**

## Laboratorní úloha - IQRF modul pro měření mesh sítí

### POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce bude vytvoření funkční privátní sítě umožňující připojení různých zařízení, čidel a senzorů. Síť bude postavena na technologii IQRF. Teoretická část práce bude pojednávat o popisu použité bezdrátové technologie, jejich možnostech a porovnání s jinými LPWAN. Dále bude teoretická část obsahovat návrh vhodných prvků sítě (senzorické jednotky, gateway). Součástí teoretické části budou vypracována bloková schémata popisující chování jednotlivých částí sítě. V praktické části bude pozornost zaměřena na sestavení a zprovoznění IQRF sítě. V síti budou implementovány alespoň dvě koncová zařízení a centrální gateway. Praktickým výstupem bude vytvoření laboratorní úlohy využívající vytvořené privátní sítě.

### DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce.

**Termín zadání:** 5.2.2024

**Termín odevzdání:** 28.5.2024

**Vedoucí práce:** Ing. Lukáš Jablončík

**prof. Ing. Jiří Mišurec, CSc.**  
předseda rady studijního programu

### 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**

Táto bakalárska práca sa venuje vytvoreniu laboratórnej úlohy zaradenej do predmetu *Komunikační systémy pro IoT*. Prvú časť práce tvorí zoznámenie sa s problematikou IQRF siete, jej topológiou a možnosťami siete. V práci je porovnaná IQRF technológia s ďalšími IoT technológiami, ktoré sú často používané v tomto odvetví. Praktickú časť práce tvorí samotný návrh modulu pre IQRF vysielač TR-72DA v tvare mini PCIe karty. Ďalšia časť práce sa zaoberá vytvorením komunikácie s BPC-IoT doskou pomocou zbernice SPI. Ako príklad sa využíva meranie teploty. Poslednou časťou práce je zadanie pre samotnú laboratórnu úlohu. Prílohy obsahujú aj schémy a zdrojový kód programov BPC-IoT dosky aj IQRF vysielača.

## **KLÚČOVÉ SLOVÁ**

Internet vecí, IQRF, LoRa, ZigBee, mesh, gateway, IQMESH protokol, uzol, oneskorenie, payload, prenosová rýchlosť, dosah siete, vysielač, sieť, I2C, UART, SPI, mini PCIe, M2 key E, TR-72D

## **ABSTRACT**

This bachelor's thesis deals with creating a laboratory task integrated into the subject of "Communication Systems for IoT." The first part of the thesis consists of familiarizing oneself with the issues of the IQRF network, its topology, and the network's capabilities. The thesis compares IQRF technology with other IoT technologies commonly used in this field. The practical part of the thesis comprises the design of a module for the IQRF transmitter TR-72DA in the form of a mini PCIe card. Another part of the thesis deals with establishing communication with the BPC-IoT board using the SPI bus, with temperature measurement used as an example. The final part of the thesis is the assignment for the actual laboratory task. The appendices also include schematics and source code of programs for both the BPC-IoT board and the IQRF transmitter.

## **KEYWORDS**

Internet of Things, IQRF, LoRa, ZigBee, mesh, gateway, IQMESH protocol, node, latency, payload, data rate, network range, transmitter, network, I2C, UART, SPI, mini PCIe, M.2 key E, TR-72D



GORYL, Juraj. *Laboratorna úloha - IQRF modul pre meranie mesh sietí* . Bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024. Vedúci práce: Ing. Lukáš Jablončík,



## Vyhlásenie autora o pôvodnosti diela

**Meno a priezvisko autora:** Juraj Goryl  
**VUT ID autora:** 240913  
**Typ práce:** Bakalárska práca  
**Akademický rok:** 2023/24  
**Téma záverečnej práce:** Laboratórna úloha - IQRF modul pre meranie mesh sietí

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora\*

---

\*Autor podpisuje iba v tlačenej verzii.



## POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce Ing. Lukášovi Jablončíkovi za odborné vedenie, priateľské konzultácie, trpezlivosť a podnetné návrhy k práci. Poďakovanie tiež patrí Ing. Pavlovi Paluříkovi za zoznámenie s programom Altium Designer a prácou v ňom. Rovnako chcem poďakovať aj Ing. Michalovi Mikuláškovi za zapožičanie logického analyzátora. Rovnako ďakujem mojej mame za gramatickú korekciu, psychickú pomoc a podporu.



# Obsah

Úvod	21
<b>1 IQRF technológia a jej základné znaky</b>	<b>23</b>
1.1 Úvod do IQRF	23
1.2 Základné parametre IQRF	23
1.3 IQRF sieť	24
1.4 IQRF Gateway	25
1.5 IQRF Cloud	25
1.6 IQMESH protokol	25
1.7 Latencia - Oneskorenie	27
1.8 Prenosová rýchlosť a payload	27
1.9 Dosah siete IQRF	28
1.10 Spotreba elektrickej energie	29
1.11 Bezpečnosť IQRF siete	30
1.12 Zhrnutie parametrov IQRF siete	31
<b>2 Ďalšie riešenia IoT sietí</b>	<b>33</b>
2.1 LoRa	33
2.2 ZigBee	34
2.3 Bluetooth Low Energy	36
2.4 Wi-Fi HaLow	37
<b>3 Návrh technického riešenia modulu pre vysieláč</b>	<b>39</b>
3.1 IQRF vysieláč TR-72D	39
3.1.1 Technické parametre IQRF vysieláča TR-72D	40
3.1.2 Vnútorne zapojenie TR-72D	41
3.1.3 Vývody vysieláča TR-72D a ich funkcionality	42
3.1.4 Dosah pokrytia vysieláčom TR-72D	43
3.2 Konektor vo formáte SIM karty KON-SIM-02	43
3.3 Gateway GW-USB-06	44
3.4 Zbernica M.2 key E	45
3.5 Zbernica mini PCI-Express	46
3.6 Rozhranie I2C	47
3.7 Rozhranie UART	48
3.8 Rozhranie SPI	48
3.8.1 SPI v IQRF	49
3.9 Návrh obvodu pre IQRF modul	50

3.9.1	Napájanie . . . . .	51
3.9.2	Stabilizácia napätia (odstránenie spätnej väzby) . . . . .	51
3.9.3	Filtrácia rušenia . . . . .	52
3.9.4	Odpájanie (hardwerový reset) . . . . .	52
<b>4</b>	<b>Realizácia hardvéru laboratórnej úlohy</b>	<b>55</b>
4.1	Výroba PCB dosky modulu . . . . .	55
4.2	Úprava rozmerov dosky modulu . . . . .	55
4.3	Osadenie dosky modulu súčiastkami . . . . .	56
4.4	Osadenie BPC-IoT dosky . . . . .	57
<b>5</b>	<b>Návrh laboratórnej úlohy</b>	<b>59</b>
5.1	Základné informácie o vývojovom prostredí IQRF IDE . . . . .	59
5.2	Vytvorenie point to point komunikácie . . . . .	59
5.3	TeplotaTXIQRF.c . . . . .	60
5.4	TeplotaTXRPPico.py . . . . .	60
5.5	ExternaTeplotaTXIQRF.c . . . . .	61
5.6	ExternaTeplotaTXRPPico.py . . . . .	61
5.7	Vytvorenie IQRF mesh siete . . . . .	62
5.8	Bonus.py . . . . .	63
	<b>Záver</b>	<b>65</b>
	<b>Literatúra</b>	<b>67</b>
	<b>Zoznam symbolov a skratiek</b>	<b>71</b>
	<b>Zoznam príloh</b>	<b>75</b>
<b>A</b>	<b>Laboratórna úloha IQRF mesh siete</b>	<b>77</b>
A.1	Cieľ merania . . . . .	77
A.2	Zadanie . . . . .	77
A.3	Teoretický úvod . . . . .	77
A.4	Postup pre vypracovanie merania . . . . .	79
A.4.1	Point to point komunikácia . . . . .	80
A.4.2	IQRF mesh sieť . . . . .	82
A.4.3	Bonusová úloha . . . . .	84
A.4.4	Použité prvky na pracovisku . . . . .	84
<b>B</b>	<b>Laboratory task IQRF module for measuring of mesh networks</b>	<b>85</b>
B.1	Measurement Objective . . . . .	85

B.2	Assignment . . . . .	85
B.3	Theoretical Introduction . . . . .	85
B.4	Procedure for Measurement . . . . .	87
B.4.1	Point to point Communication . . . . .	88
B.4.2	IQRF Mesh Network . . . . .	90
B.4.3	Bonus Task . . . . .	92
B.4.4	Components used in the workstation . . . . .	92
<b>C</b>	<b>Schéma a návrh dosky modulu</b>	<b>93</b>
<b>D</b>	<b>3D model dosky modulu</b>	<b>97</b>
<b>E</b>	<b>Gerber data</b>	<b>99</b>
<b>F</b>	<b>Zdrojové kódy</b>	<b>103</b>



# Zoznam obrázkov

1.1	Dizajn IQRF siete . . . . .	24
1.2	Gateway v rôznych prevedeniach . . . . .	25
1.3	Logické adresy a virtuálne smerovacie čísla po objavovaní . . . . .	27
1.4	Porovnanie dosahu viacerých sietí z IoT odvetvia . . . . .	28
1.5	Zlúčenie dvoch sietí . . . . .	29
2.1	Bluetooth v porovnaní s BLE . . . . .	37
2.2	Pokles prenosovej rýchlosti jednotlivých technológií pre IoT v závislosti od vzdialenosti od vysielača . . . . .	38
3.1	Bloková schéma vysielača . . . . .	39
3.2	Vnútorne zapojenie IQRF vysielača TR-72D . . . . .	41
3.3	Tvar dosahu vysielača pri rôznom uložení . . . . .	43
3.4	SIM konektor pre pripojenie vysielača k modulu . . . . .	44
3.5	IQRF USB koordinátor . . . . .	44
3.6	Zbernica M.2 key E na BPC-IOT doske . . . . .	45
3.7	Zbernica mini PCI-Expres na BPC-IOT doske . . . . .	46
3.8	Architektúra I2C . . . . .	47
3.9	Architektúra UART . . . . .	48
3.10	Architektúra SPI . . . . .	49
3.11	SPI_CHECK paket . . . . .	49
3.12	Paket SPI komunikácie v IQRF . . . . .	50
3.13	Tvorba znaku PTYPE . . . . .	50
3.14	Regulátor napätia vstavaný vo vysielači . . . . .	51
3.15	Stabilizácia napätia . . . . .	52
3.16	Hardvérový reset . . . . .	53
4.1	Rozmery modulu . . . . .	56
4.2	Osadená doska modulu súčiastkami . . . . .	57
4.3	Bloková schéma zapojenia použitých zariadení . . . . .	58
4.4	Osadená BPC-IoT doska . . . . .	58
5.1	Terminal log v IQRF IDE . . . . .	60
5.2	Detail prijatej správy . . . . .	61
5.3	Zobrazenie mesh siete v záložke map view v IQRF IDE . . . . .	63
A.1	Štruktúra IQRF siete . . . . .	78
A.2	Porovnanie dosahu viacerých sietí z IoT odvetvia . . . . .	79
A.3	Bloková schéma zapojenia pracoviska . . . . .	80
A.4	IQRF IDE pre lepšiu navigáciu . . . . .	81
A.5	Sekcia Nodes info . . . . .	83
B.1	Structure of IQRF network . . . . .	86

B.2	Comparison of the range of various networks in the IoT sector . . . .	87
B.3	Block diagram of the workstation setup . . . . .	88
B.4	IQRF IDE for better navigation . . . . .	89
B.5	Section Nodes info . . . . .	91
E.1	Top layer . . . . .	99
E.2	Bottom layer . . . . .	100
E.3	Top Solder . . . . .	100
E.4	Bottom Solder . . . . .	101
E.5	Top Overlay . . . . .	101
E.6	Top Paste . . . . .	102
E.7	Bottom Paste . . . . .	102

# Zoznam tabuliek

1.1	Spotreba elektrickej energie konkrétneho IQRF modulu . . . . .	29
1.2	Prehľad parametrov IQRF siete . . . . .	31
2.1	Prehľad parametrov technológie LoRa . . . . .	34
2.2	Prehľad parametrov technológie ZigBee . . . . .	35
2.3	Prehľad parametrov technológie Bluetooth Low Energy . . . . .	36
2.4	Prehľad parametrov technológie Wi-Fi HaLow . . . . .	37
3.1	Technické parametre vysielača TR-72D . . . . .	40
3.2	Funkcionalita jednotlivých kontaktov vysielača TR-72D . . . . .	42
4.1	Parametre PCB zaslané do výroby . . . . .	55
A.1	Prehľad parametrov IQRF siete . . . . .	79
B.1	Summary of IQRF network parameters . . . . .	87



# Úvod

Aktuálne sú často uplatňované v bežnom živote inteligentné technológie, smart mestá či budovy, v ktorých sa využívajú inštalácie ako napríklad inteligentné parkovanie, osvetlenie či monitoring odpadového hospodárstva. Tieto inštalácie tvoria veľký celok nazývaný Internet vecí. Sieť internetu vecí sa skladá z viacerých zariadení ako sú senzory s dlhou výdržou pri batériovom napájaní, centrálné prvky - gateway, ktoré spájajú sieť senzorov a internetové úložisko (cloud). Bez týchto zariadení by internet vecí nemohol fungovať. V oblasti vývoja prvkov do internetu vecí sa už špecializuje mnoho firiem a jednou z nich je aj IQRF Tech s.r.o. v Jičíne, ktorá vyvíja technológiu IQRF. IQRF je bezdrôtová technológia stredného dosahu s paketovou orientáciou, dlhou životnosťou a bezúdržbovosťou.

Cieľom tejto bakalárskej práce je zdefinovať teoretický základ a vytvoriť praktické podklady pre laboratórnu úlohu *IQRF modul pre meranie mesh sietí*.

Prvou časťou bakalárskej práce je teoretický rozbor základných vlastností a parametrov IQRF technológie, ktoré sú následne v druhej kapitole porovnané s inými, frekvencovane používanými technológiami pre internet vecí.

Pre spojenie IQRF vysielča a BPC-IOT dosky, ktorá bude napájať a komunikovať s IQRF vysielčom a vykonávať merania IQRF siete, je potrebné navrhnuť modul. Tento modul bude navrhnutý pre mini PCIe kartu použitím návrhového softvéru Altium Designer a následne vyrobený firmou JLC PCB a ručne osadený súčiastkami. Do modulu bude osadený aj IQRF vysieláč naprogramovaný v jazyku C. Modul s vysieláčom bude vsunutý do mini PCIe slotu na BPC-IoT doske naprogramovanej v jazyku microPython. IQRF vysieláč bude posielať snímané údaje na koordinátor.

Laboratórna úloha sa nachádza v prílohe A, anglická verzia tejto úlohy je v prílohe B, schéma, návrh dosky a súbory pre výrobu modulu v prevedení mini PCIe karty v prílohe C, D a E. Zdrojové kódy k jednotlivým zariadeniam sú v prílohe F.



# 1 IQRF technológia a jej základné znaky

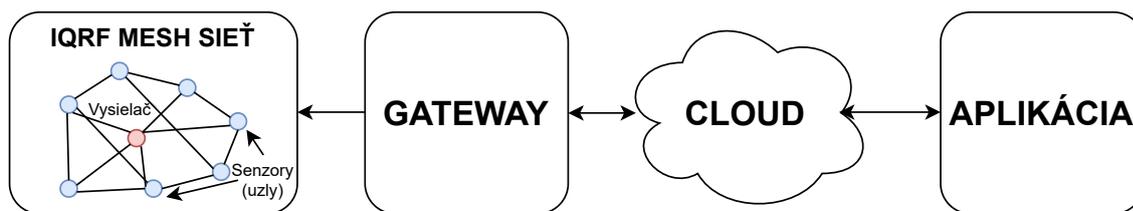
Internet vecí (Internet of Things - IoT) je koncept, ktorý má snahu prepojiť veci z bežného života s internetom pomocou senzorov a softvérov, ktoré sú takmer bezúdržbové a majú nízku spotrebu elektrickej energie. V súčasnosti sa na sféru IoT riešení špecializuje mnoho firiem, ktoré vyvíjajú technológie s rôznym spôsobom a dosahom pokrytia oblastí. Medzi tieto technológie patrí aj platforma IQRF.

## 1.1 Úvod do IQRF

IQRF (Intelligent Quick Response Frequency) je bezdrôtová technológia a platforma pre IoT orientovaná paketovo. Umožňuje posielat nízke množstvo dát s minimálnou spotrebou elektrickej energie a maximálnou spoľahlivosťou. Technológia je vhodná pre náročné aplikácie, ktoré si vyžadujú čo najnižšiu spotrebu elektrickej energie ako sú batériou napájané inštalácie, aplikácie pre IoT, Smart Cities (Inteligentné mestá). IQRF technológia bola vyvinutá v roku 2004 českou firmou MICRORISC a v roku 2017 bola založená jej sesterská spoločnosť IQRF Tech s.r.o. v Jičine, ktorá sa zaoberá vývojom IQRF technológie a jej zariadení až do dnes a je hlavným koordinátorom IQRF technológie vo svete [1], [2].

## 1.2 Základné parametre IQRF

IQRF technológia pracuje v nelicencovanom Sub-GHz ISM (industrial scientific medical) rádiovom pásme (433, 868, 915 MHz) v závislosti od kontinentu alebo štátu inštalácie. V Českej republike je používané 868 MHz pásmo. IQRF nepotrebuje externú infraštruktúru ako sú mikrovlnné prepojenia alebo dlhé káblové vedenia. Moduly IQRF technológie môžu pracovať v rôznych typológiách ako napr. bod - bod, hviezda, ale najčastejšie je používaná v mesh topológii obr.1.1. Pôvodne bola technológia IQRF využívaná v automatizácii budov, ale v posledných rokoch technológia prešla ďalším vývojom a stala sa konkurencie schopná aj vo väčších aplikáciách. Na obr.1.1 je znázornený typický dizajn IQRF siete obsahujúci jednotlivé uzly (senzory) v mesh topológii, gateway, cloud a pripojenie na internet (aplikácia pre komunikáciu so svetom) [2], [3], [4], [5].



Obr. 1.1: Dizajn IQRF siete

## 1.3 IQRF sieť

Technológia IQRF je dizajnovaná do meshovej topológie (IQMESH). Pri použití topológie mesh je využitý smerovací mechanizmus, ktorý dokáže odstrániť nevýhody topológie hviezda a zvýšiť tak spoľahlivosť celej siete. Základným prvkom siete je IQRF vysielač (transciever) tvorený malým modulom používaným k prenosu dát. Vysielač obsahuje malú mikrokontrolérovú 8-bitovú jednotku (MCU) operujúcu so vstavaným operačným systémom (OS) riadiacim prenos dát.

Jednotlivé funkcie prístupu sú do IQRF vysielača zadané dvoma možnými spôsobmi:

- Prvý spôsob je zadaný pomocou užívateľskej aplikačnej vrstvy používajúcej preddefinované funkcie v OS. Funkcie je možné naprogramovať v programovacom jazyku C, avšak táto možnosť nie je podporovaná sieťami IQMESH.
- Druhý spôsob je založený na trojvrstvovej IQRF architektúre s hardvérovým profilom, ktorý je schopný používať softvérový aplikačný plug-in. Všetky zdroje a služby môžu byť prístupné cez hardvérový profil. Hardvérový profil v realite implementuje periférny prístup k zariadeniu nad OS.

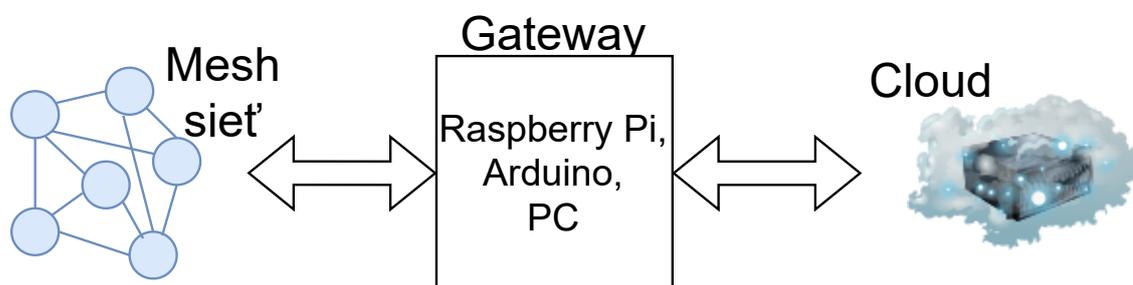
Z uvedeného vyplýva, že programovanie nie je potrebné a funkcionality je dosiahnutá jednoducho riadením cez dátový tok.

Z tohto dôvodu je možné použiť dva rôzne rádiových frekvencných komunikačných režimov:

- Sieťový - obsahujúci jeden koordinačný uzol spravujúci meshovú sieť zloženú až z 239 uzlov pre jednu sieť. V takejto sieti sú dátové pakety prístupné len pre tie uzly, ktorým sú adresované. Každý uzol preto vie slúžiť ako koordinátor alebo spoločný uzol. Takto môže v jednej sieti byť pripojených až 65.000 pracovných uzlov, ktoré vedia na pozadí pracovať ako smerovače. Vďaka tomu je podporovaných až 240 hopov na jeden paket. Cesta pre paket môže byť nájdená a vytvorená automaticky ako smerovacia štruktúra s možnosťou vyhľadávania trasy. Trasy sú vytvorené automaticky cez redundantné cesty v mesh sieti.
- Nesieťový - nepodporuje sieťové funkcie, je vhodný pre aplikáciu zhotovenú z malého množstva zariadení peer-to-peer komunikáciou. Pri maximálnom možnom rozsahu sú dátové pakety prístupné pre všetky zariadenia [2], [3], [6], [7].

## 1.4 IQRF Gateway

Nad IQRF sieťou a jej uzlami môže prevziať kontrolu cloudový server. V takom prípade je potrebné použiť v sieti bránu (gateway). Gateway poskytuje rozhranie pre spojenie a komunikáciu medzi IQRF sieťou a ostatnými sieťami podporujúcimi internetovú konektivitu. IQRF aliancia zatiaľ vyvinula Wi-Fi, GSM a ethernetové gateway. Každý gateway obsahuje senzor - uzol, ktorý slúži ako koordinátor pre ostatné uzly v sieti. Gateway potom umožňuje zber dát, prístup a kontrolu ostatných zariadení. IQRF gateway vie byť implementovaný z Linuxových zariadení ako napríklad Raspberry Pi. Okrem toho sú podporované aj zariadenia bez operačného systému ako napr. Arduino... [2], [3].



Obr. 1.2: Gateway v rôznych prevedeniach

## 1.5 IQRF Cloud

IQRF cloud je server, ktorý poskytuje okamžitú službu spojenia medzi používateľmi a sieťou IQRF s použitím klient-server komunikácie. Zároveň sa na tomto serveri vytvára úložisko údajov, ktoré sú dostupné aj po určitom čase v závislosti od kapacity pamäte servera. Dáta sa odošlú zo senzoru na gateway a odtiaľ sú priamo nahraté na server. Takto uložené dáta sú už dostupné priamo cez internetový prehliadač alebo pomocou aplikácie [3].

## 1.6 IQMESH protokol

IQMESH (Intelligent Mesh) je patentovaný meshový komunikačný protokol používaný v rámci ekosystému technológie IQRF. Je navrhnutý na sprostredkovanie bezdrôtovej komunikácie v mesh topológii, v ktorej každé zariadenie môže komunikovať s okolitými zariadeniami a potenciálne slúžiť ako preposielací uzol na rozšírenie dosahu siete. Každý paket je smerovaný uzlami na pozadí neobmedzujúc ich funkcionality. Tento proces je riadený operačným systémom a je nebadateľný pre užívateľa.

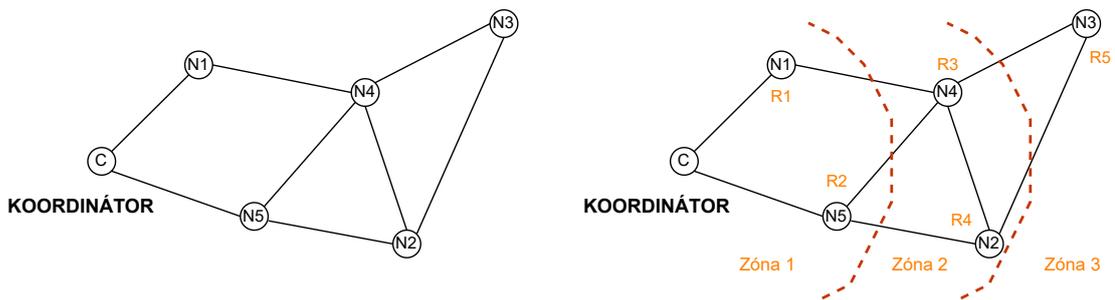
IQMESH podporuje viaceré smerovacie štruktúry ako sú: úplný mesh, redukovaný mesh, optimalizovaný mesh, strom... Smerovanie môže prebiehať staticky, a taktiež môže byť pod úplnou kontrolou užívateľa. Dynamické časovanie s výberom počtu krokov v sieti a voliteľným časovým úsekom umožňuje maximálnu priepustnosť s minimálnym rušením a spotrebou elektrickej energie. Efektivita a spoľahlivosť je dosiahnutá pomocou takzvaného záplavového mechanizmu, ktorý funguje podobne ako povodeň - informácia je na miesto doručenia poslaná viacerými možnými cestami - podobne ako voda pri povodni zaplavuje miesta kadiaľ môže pretiecť. Kolíziám v sieti sa predchádza pomocou systému Time Division Multiple Access, čo v preklade znamená viacnásobný prístup s časovým delením. Aj napriek vyššie spomenutým vyspelým funkcionalitám je IQMESH implementovateľný do jednoduchého a lacného zariadenia s limitovanými možnosťami.

Každé zariadenie v sieti má unikátne číslo - logickú adresu priradenú pri výrobe. Smerovací algoritmus Discovered Full Mesh (objavovací plne meshový algoritmus) využíva virtuálnu smerovaciu štruktúru, ktorá vzniká pri procese objavovania, kedy koordinátor prideli virtuálne smerovacie číslo do každého zariadenia, ktoré dokáže smerovať. Na obr.1.3 je v ľavej časti zobrazená štandardná statická sieť, v ktorej sú uzly adresované pomocou logickej adresy -  $N_1, N_2, \dots, N_i$ . Na pravej strane je pridaná virtuálna štruktúra (oranžovo znázornené zóny). Zóny sú vytvárané zo zariadení s rovnakým počtom potrebných hopov na prístup do koordinátora. Následne sú v zónach pridelené virtuálne smerovacie čísla. Logická adresa je používaná pre definovanie cieľového zariadenia a virtuálne smerovacie číslo je používané OS pre smerovanie. Virtuálne smerovacie číslo je pridelené systematicky podľa vzdialenosti uzla od koordinátora, čo uľahčuje prácu záplavovému algoritmu a taktiež aj iným smerovacím algoritmom. Možné je tiež smerovanie pomocou logickej adresy, pretože uzly môžu byť uložené v poradí podľa logickej adresy. V kombinácii s protizrážkovým mechanizmom je vytvorený smerovací, efektívny a bezkolízny prenos.

Počas vyhľadávania koordinátor hľadá uzly pripojené k sieti, aby im bol schopný pridelit unikátne virtuálne smerovacie číslo. Koordinátor zaháji proces vyhľadávania a všetky zariadenia, ktoré odpovedajú na výzvu koordinátora „*Odpovedzte mi*“ ich dostanú pridelené. Na základe prijatej odpovede a za predpokladu, že prepojenie medzi uzlami a koordinátorom bolo symetrické, sú objavené smerovacie cesty do uzlov a naspäť ku koordinátorovi. Všetky uzly majúce priame spojenie s koordinátorom patria do Zóny 0 a sú dostupné bez smerovania. Koordinátor postupne vytvára zóny so stúpajúcim poradovým číslom. Tieto zóny sú vytvárané tak, že všetkým uzlom z aktuálnej zóny je prikázané, aby našli svojich susedov so vzdialenosťou 1 hop. Tento algoritmus sa opakuje pokiaľ nie je dosiahnutá zóna, ktorá už nemá koho objavovať. Táto zóna sa nazýva okrajová a všetky ostatné zóny tvoria Routing Backbone (smerovaciu chrbticu). Takýto proces objavovania sa pri aplikáciách

inteligentných budov či miest vykonáva iba počas budovania siete, nakoľko pri takomto type siete nenastávajú časté zmeny v sieti. Ak by nastali, musel by proces objavovania prebehnúť opätovne.

Na rozdiel od iných technológií IQRF nepotrebuje zdieľať naprieč sieťou detailnú informáciu o čase. Smerované pakety sledujú počet uskutočnených hopov, čo zodpovedá príslušnej zóne a časovému úseku na základe virtuálneho smerovacieho čísla. Spätné poradie sa zase využíva na preposielanie odpovedí z uzlov späť na koordinátor. Tento mechanizmus s pridaním informácie o dĺžke časového okna vytvára efektívnu synchronizáciu všetkých zariadení. Podporovaný je aj iný spôsob synchronizácie a to dynamické časovanie založené na dĺžke paketu [2], [8], [9], [10], [11].



Obr. 1.3: Logické adresy a virtuálne smerovacie čísla po objavovaní

## 1.7 Latencia - Oneskorenie

Dôležitým parametrom pre každú sieť je jej oneskorenie pri smerovaní. Tento parameter siete je rozdielny podľa preferencií danej inštalácie. Oneskorenie je odlišné ak je preferovaná nízka úroveň oneskorenia s označením STD (Standard) alebo nízka spotreba energie značená LP (Low Power). Pri STD je oneskorenie do 60 ms a pri LP je oneskorenie o niečo vyššie - do 100 ms. Tieto módy s rôznym oneskorením je možné aj kombinovať (STD+LP). Podľa pomeru ich kombinácie vzniká unikátna hodnota oneskorenia z vyššie uvedeného intervalu 60 ms - 100 ms. STD+LP je najčastejšie používaným módom v IQRF sieti, pretože tento mód umožňuje používateľom najideálnejšie nastavenia siete pre ich konkrétne využitie [10], [7].

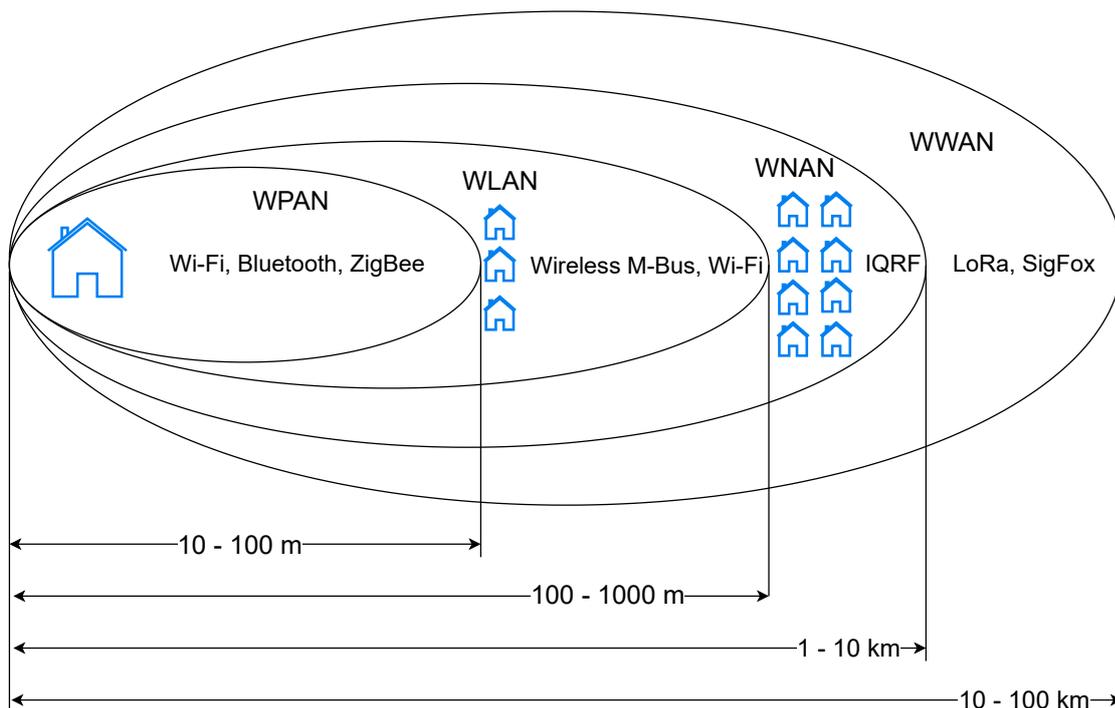
## 1.8 Prenosová rýchlosť a payload

Dáta, ktoré zozbiera senzor v IoT sieti, nie sú veľkého rozsahu. Pri prenášaní dát zo senzorov preto nie je potrebné dosahovať vysokú prenosovú rýchlosť. Prenosová rýchlosť technológie IQRF je zdanlivo nízka. V závislosti od použitia konkrétneho

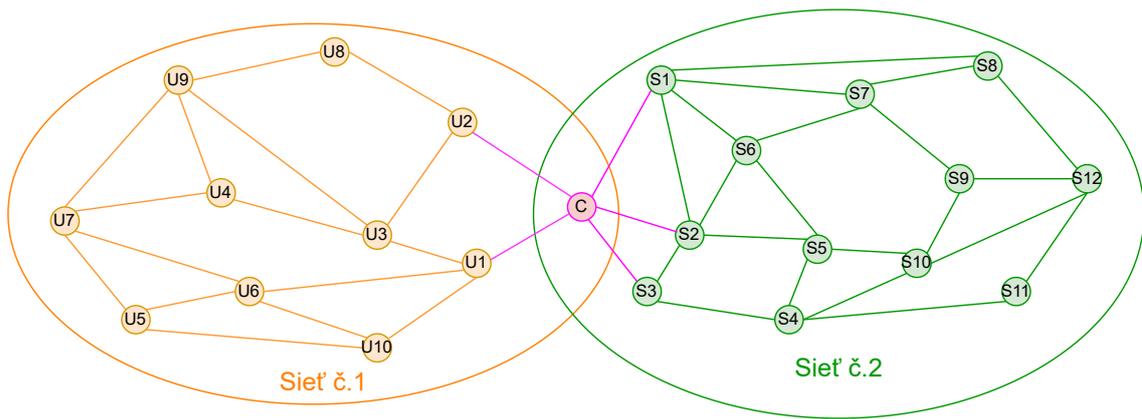
IQRF modulu sa prenosová rýchlosť pohybuje okolo 19 až 20 kbps. Táto rýchlosť je dostatočná pre aplikácie, ktoré preferujú parametre ako je vyšší dosah siete alebo nižšia spotreba elektrickej energie. Maximálna prenosová rýchlosť siete môže byť tak tiež ovplyvnená konkrétnym prostredím inštalácie alebo nastavením prvkov v sieti. V oblastiach s vysokým rušením je prenosová rýchlosť nižšia ako v oblastiach s nízkym zarušením pásma. To isté platí aj pre payload, ktorý pri ideálnych podmienkach dosahuje veľkosť správy 64 bytov [4], [12].

## 1.9 Dosah siete IQRF

Maximálny dosah jedného IQRF vysielača je certifikovaný na 500 m voľného priestranstva, čo pri použití viac ako 200 zariadení v mesh topológii predstavuje rozlohu až desiatky kilometrov štvorcových. Z tohto dôvodu IQRF sieť radíme do kategórie WWAN (Wireless Metropolitan Area Network) obr. 1.4, kde je tento dosah zobrazený v porovnaní aj s dosahmi iných IoT sietí. Pri reálnej aplikácii s prekážkami sa dosah rapídne znižuje a je potrebné prepojiť viacero IQRF sietí pomocou stredového prvku ako, je to zobrazené na obr.1.5 [4], [13].



Obr. 1.4: Porovnanie dosahu viacerých sietí z IoT odvetvia



Obr. 1.5: Zlúčenie dvoch sietí

## 1.10 Spotreba elektrickej energie

IQRF zariadenia vedia pracovať v rôznych stavoch vzhľadom na spotrebu elektrickej energie. Pri správnom použití a skombinovaní jednotlivých stavov vieme dosiahnuť tak nízku spotrebu, že zariadenia nadobudnú mnohoročnú výdrž na AAA batérii. Pre lepšiu predstavu technológia obsahuje tri základné stavy:

- **Vysielanie**, kde v závislosti na nastavenom výkone dosahuje najvyšší potrebný napájací prúd 19 mA,
- **Prijímanie**, kde maximálny odoberaný prúd je približne 12 mA,
- **Režim spánku**, ktorý dokáže spotrebu zariadenia úplne minimalizovať až na hodnoty menšie ako 1  $\mu\text{A}$  a pri hlbokom spánku až na hodnoty menšie ako 1 nA [14].

Viac režimov s detailnejšími popisom o spotrebe sa nachádza v tab.1.1

Modul TR-76D	
Stav IQRF zariadenia	Odber elektrickej energie
Vysielanie s maximálnym výkonom	19 mA
Vysielanie s minimálnym výkonom	8,4 mA
MCU beží a RF vypnuté	1,53 - 71,67 mA
MCU beží a RF úsporný režim	2,83 mA
Nízko energetické vysielanie LP	244 $\mu\text{A}$
Prijímanie s maximálnou úsporou XLP	15 $\mu\text{A}$
Prijímanie s nízkou latenciou STD	11,9 mA
Režim spánku	>1 $\mu\text{A}$
Hlboký spánkový režim	>100 nA

Tab. 1.1: Spotreba elektrickej energie konkrétneho IQRF modulu

## 1.11 Bezpečnosť IQRF siete

Každá prenášaná informácia môže byť pre niekoho cenná alebo citlivá, preto je dôležité, aby IQRF sieť mala dobrú ochranu pred útokmi alebo poškodením s cieľom dostať sa ku prenášaným informáciám.

Poškodenie koncových uzlov je jedným z najčastejších a najškodlivejších útokov na bezdrôtovú sieť so senzormi (WSN - wireless sensor network). Je to spôsobené tým, že pri takejto sieti sú uzly prítomné takmer všade (ľahko prístupné miesta bez ochrany) a majú obmedzenia v úložných a aj výpočtových oblastiach. Útočník teda môže sieť odpočúvať bez akýchkoľvek priamych prekážok. Každý takýto útok zacielený na koncové uzly môže spôsobiť zlyhanie v sieti.

Poškodenie koordinátora je ďalším častým spôsobom útoku na sieť, keďže každá informácia z koncových uzlov prechádza práve týmto bodom. Koordinátor taktiež zohráva hlavnú úlohu pri pripojení koncových bodov (senzorov) na gateway a následne na internet.

Najčastejším terčom hekerov je gateway, pretože všetky prenášané informácie z IQRF siete ňou prechádzajú. Hekeri útokom informácie ukradnú alebo ich zablokujú.

Bezdrôtová komunikácia či celý systém je vystavený potenciálnemu útoku vzduchom OTA (over-the-air). Z tohto dôvodu sa od verzie operačného systému 4.0 IQRF aliancia zamerala najmä na bezpečnosť siete. Od tejto verzie je v IQRF operačnom systéme implementovaná ochrana založená na priemyselných štandardoch a zaisťuje tak autorizovaný prístup k toku dát vzduchom. Konkrétne bol v operačnom systéme implementovaný štandard AES-128 pre šifrovanie bezdrôtovej komunikácie. Okrem šifrovania dát sa zvyšuje aj celková ochrana siete a zabráňuje sa falšovaniu jednotlivých paketov.

V IQRF sieti existujú tri hlavné úrovne ochrany, ktoré sú založené na šifrovaní AES-128:

- **Šifrovanie prístupu:** toto nezávislé šifrovanie sa vždy používa pri pripájaní všetkých zariadení k sieti. Táto úroveň ochrany zabezpečuje, že k sieti majú prístup len autorizované zariadenia a teda chráni sieť pred neautorizovaným pridávaním uzlov do siete. Keď sa zariadenie pokúsi pripojiť k sieti, prístupové šifrovanie pomáha overiť jeho autenticitu a autorizáciu.
- **Šifrovanie komunikácie v sieti:** je to bezpečnostná vrstva, ktorá používa šifrovanie na všetky komunikácie v IQRF sieti. Tým je zabezpečené, že údaje prenášané medzi jednotlivými zariadeniami v sieti sú bezpečné a nemôžu byť jednoducho odchytené alebo poškodené neautorizovanými stranami. Šifrovanie siete pomáha chrániť integritu a dôvernosc údajov vymieňanými medzi zariadeniami v sieti.

- **Šifrovanie používateľa:** poskytuje dodatočnú úroveň bezpečnosti umožňujúcu šifrovanie údajových paketov (payload) pomocou používateľom definovaného kľúča v aplikačnom kóde používateľa. Toto šifrovanie je voliteľné a môže byť implementované na skrytie obsahu vybraných údajových paketov. Umožňuje používateľom kontrolu nad šifrovaním konkrétnych údajov a poskytuje dodatočnú úroveň bezpečnosti pre citlivé informácie [10], [15], [16], [17].

## 1.12 Zhrnutie parametrov IQRF siete

V tabuľke 1.2 sú spísané najdôležitejšie parametre IQRF siete. Tieto parametre sú všeobecné a teda môžu sa odlišovať v závislosti od použitých zariadení alebo aplikovaných nastavení pre jednotlivé zariadenia napr. nastavenie vysielačieho výkonu alebo oneskorenia pre daný vysielač či uzol.

IQRF		
Veličina	Jednotka	Hodnota
Pracovná frekvencia	MHz	433, 868, 916
Prenosová rýchlosť	kbps	20
Max. dĺžka prenášanej správy	byte	64
Max. dosah vo voľnom priestranstve	m	1000
Modulácia	-	GFSK
Topológia	-	mesh
Oneskorenie	ms	60 - 100
Max. počet uzlov na gateway	-	65*103
Teoretický čas života na AAA batérii	rok	5 - 10

Tab. 1.2: Prehľad parametrov IQRF siete



## 2 Ďalšie riešenia IoT sietí

V IoT aplikáciách sa používajú okrem IQRF platformy ďalšie podobné technológie. Pre porovnanie sú v práci použité technológie s podobnými parametrami ZigBee, Bluetooth Low Energy a Wi-Fi HaLow. Taktiež zaujímavá je aj primárne úplne odlišná sieť akou je LoRa, ktorá však už vie pracovať v mesh topológii.

### 2.1 LoRa

LoRa je technológia vyvinutá firmou Semtech a Microchip, ktorá má patentovaný prístupový a kontrolný protokol LoRaWAN. Výhody technológie LoRa sú poznateľné už v názve LoRa - Long Range - v preklade do slovenčiny dlhý dosah. Dosah technológie LoRa je od 5 do 20 km na jeden gateway v závislosti od prostredia v akom sa vysiela. LoRa využíva topológie typu hviezda, v ktorej je centrálny prvok zároveň používaný ako gateway. Použitá topológia umožňuje použitie väčšieho množstva senzorov (uzlov) na jeden gateway. Nevýhody siete LoRa vyplývajú z použitej topológie. Pri poruche centrálného prvku sa celá sieť stáva nefunkčnou. Doručenie správ pri hviezdicovej topológii je taktiež menej spoľahlivé ako pri doručení správ pri topológii mesh. Z tohto dôvodu prišla LoRa aliancia s inováciou a to použitím mesh topológie alebo kombinovanej topológie mesh a hviezda.

LoRa ako bezdrôtová sieť využíva Direct Sequence Spread Spektrum (DSSS) na fyzickej vrstve, čo umožňuje vysielať s využitím nižšieho vysielačieho výkonu pre signál. Použitím FSK (Frequency Shift Keying) modulácie sa mení frekvencia pre nosné vlny podľa toho, aký bit vstupuje do modulácie (0 alebo 1) zo signálu, ktorý sa práve moduluje. LoRa je schopná demodulovať a rozoznať signály, ktoré sú niekoľko desiatok decibelov pod úrovňou šumu. Toto je možné vďaka FEC (Forward Error Correction) mechanizmu, ktorý odoslaním paritných paketov v prípade potreby opraví stratený alebo poškodený dátový tok. Jeden paritný paket je vysielaný pre skupinu N paketov.

LoRa pracuje v nelicencovanom sub GHz pásme rovnako ako IQRF čo v Európe predstavuje 863 - 870 MHz. Dáta prenáša rýchlosťou 0,25 až 50 kbps. Veľkosť správy v technológii LoRa môže byť z rozsahu 2 až 255 bajtov [18].

LoRa		
Veličina	Jednotka	Hodnota
Pracovná frekvencia	MHz	443, 868, 780, 915
Prenosová rýchlosť	kbps	0,3 - 40
Max. dĺžka prenášanej správy	byte	243
Max. dosah vo voľnom priestranstve	km	15
Modulácia	-	CSS
Topológia	-	hviezda, mesh
Oneskorenie	ms	>1000
Max. počet uzlov na gateway	-	104
Teoretický čas života na AAA batérii	rok	5 - 10

Tab. 2.1: Prehľad parametrov technológie LoRa

## 2.2 ZigBee

ZigBee je taktiež technológia pre bezdrôtovú komunikáciu s kratším dosahom podobne ako IQRF. Využíva 2,4 GHz pásmo. ZigBee je veľmi spoľahlivá technológia, keďže vo svojej architektúre implementuje zásady ako túto spoľahlivosť dosiahnuť.

Prvým bodom spoľahlivosti je implementácia štandardu IEEE 802.15.4 pre siete s kratším dosahom, ale vysokou spoľahlivosťou. Základom tohto štandardu je modulácia OQPSK (Offset Quadrature Phase Shift Keying) a DSSS (Direct Sequence Spread Spectrum), ktorých kombinácia dosahuje vynikajúce výsledky v prostrediach s veľmi nízkym odstupom signálu od šumu.

Druhým bodom je používanie CSMA-CA (Carrier Sense Multiple Access Collision Avoidance) metódy. ZigBee najskôr pred vysielaním naslúcha kanálu a až ak kanál je voľný, vysielanie môže začať. Táto metóda zabraňuje dvom zariadeniam v sieti vysielat' naraz a tak zabraňuje znehodnoteniu dát.

Tretí bod spoľahlivosti je používanie 16 bitového CRC (Cyclic Redundancy Check) na každom pakete, čo má za následok kontrolu správnosti dátových bitov. Každý paket má štyri možnosti opätovného odoslania. Ak sa ani na štvrtý krát nepodarí paket doručiť, odosielač uzol je informovaný o probléme a je na ňom, aby vyhľadal novú cestu na doručenie. V prípade ak sa nová cesta vyhľadať nedá, odosielač uzol je informovaný, aby paket už nebol viacej posielač.

Posledný bod spoľahlivosti, ktorý je používaný technológiou ZigBee, je samotná topológia mesh, v ktorej ZigBee pracuje. Táto topológia zvyšuje svoj dosah a spoľahlivosť pomocou viackrového posielania paketov. Robustnosť siete je zvýšená tak, že ak je nejaká cesta v sieti zahľtená alebo poškodená, vyhľadá sa nová cesta cez iné dostupné uzly.

Vysielenie v 2,4 GHz pásme podľa štandardu IEEE 802.15.4. je možné až s prenosovou rýchlosťou 250 kbps. Avšak po zopakovaní šifrovania a dešifrovania a po plnom využití mesh protokolu, priepustnosť siete je skôr okolo 25 kbps. Vysielače využívajú polovične duplexné spojenie, čo znamená, že rovnakou cestou je realizované aj vysielenie aj prijímanie, avšak obe funkcie nevedia byť realizované zároveň. Táto skutočnosť má taktiež za následok zníženie prenosovej rýchlosti z 250 kbps na 25 kbps. V súčasnosti je dostupný štandard IEEE 802.15.4b. Ak by si ZigBee osvojilo tento štandard, prenosová rýchlosť by mohla stúpnuť na 40 kbps.

Spotreba elektrickej energie zariadeniami v sieti je veľmi nízka, čo je výhodné pri napájaní zariadení batériou, pretože pri výdrži batérií niekoľkých rokov je prakticky údržba takéhoto zariadenia takmer nepotrebná. Nízku spotrebu elektrickej energie znásobuje aj to, že Zigbee zariadenia využívajú podobne ako IQRF rôzne režimy v závislosti na spotrebe. Taktiež komunikácia medzi zariadeniami neprebíha neustále, ale len v určitých časových intervaloch.

ZigBee je nie len spoľahlivá technológia, ale aj bezpečná, keďže implementuje štandard bezpečnosti AES-128 (Advanced Encryption Standard). Tento štandard šifruje pakety spôsobom, ktorý je veľmi náročné prelomiť. Štandard AES je považovaný za jeden z najlepších bezpečnostných štandardov, zabezpečuje aj šifrovanie paketov aj autorizáciu uzlov [19].

<b>ZigBee</b>		
Veličina	Jednotka	Hodnota
Pracovná frekvencia	GHz	2,4
Prenosová rýchlosť	kbps	250
Max. dĺžka prenášanej správy	byte	128
Max. dosah vo voľnom priestranstve	m	300
Modulácia	-	BPSK, OQPSK
Topológia	-	hviezda, mesh, strom
Oneskorenie	ms	16
Max. počet uzlov na gateway	-	65*103
Teoretický čas života na AAA batérii	rok	2

Tab. 2.2: Prehľad parametrov technológie ZigBee

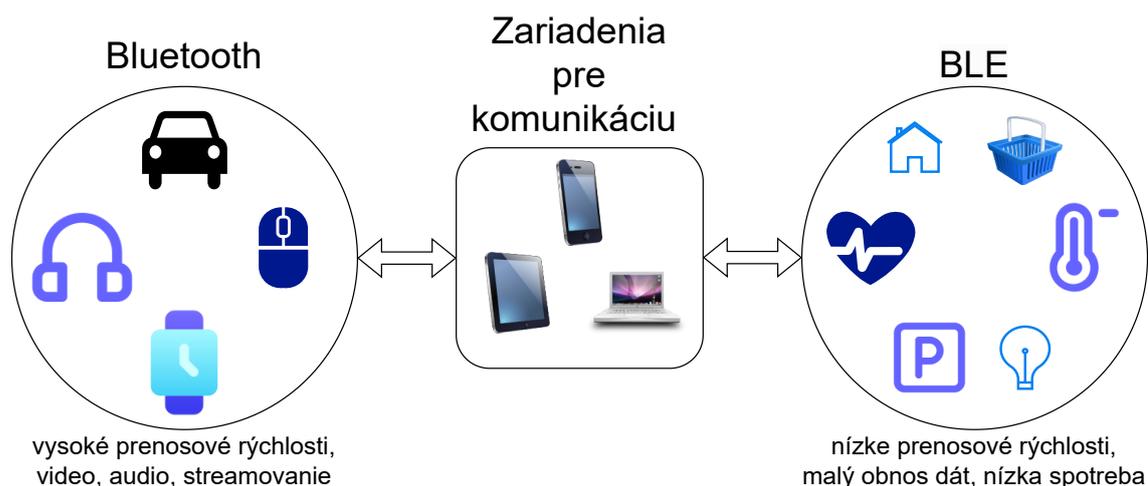
## 2.3 Bluetooth Low Energy

Bluetooth Low Energy (BLE) je bezdrôtová technológia s menším dosahom. BLE využíva 40 kanálov (37 je použiteľných pre prenos dát) v 2,4 GHz nelicencovanom frekvenčnom pásme. Táto technológia umožňuje komunikáciu s nízkou spotrebou elektrickej energie, ktorá je možná vďaka kratším dĺžkam vlny použitého rádiového signálu. BLE podporuje viacero komunikačných topológií, počínajúc jednoduchým point-to-point, broadcast, ale najčastejšie sa používa mesh. Táto topológia umožňuje, aby BLE bol schopný vytvoriť pomerne veľkú a stabilnú sieť s vysokou redundanciou. BLE poskytuje aj vyspelú lokalizačnú službu pre vonkajšie podmienky, teda vie presne zmerať a lokalizovať vzdialenosť susediacich zariadení a podľa toho uľahčiť nastavenia používateľovi. Podobne ako vyššie opisované IQRF aj BLE používa GFSK moduláciu. Nízkou spotrebu elektrickej energie podporuje aj pomerne nízky vysielač výkon, ktorý dosahuje maximálne hodnoty 100 mW. Takýto vysielač výkon je však absolútne postačujúci, pretože senzitivita prijímača je menej ako -70 dBm a v závislosti od módu až menej ako -82 dBm [20].

Bluetooth Low Energy		
Veličina	Jednotka	Hodnota
Pracovná frekvencia	GHz	2,4
Prenosová rýchlosť	kbps	2000
Max. dĺžka prenášanej správy	byte	251
Max. dosah vo voľnom priestranstve	m	400
Modulácia	-	GFSK
Topológia	-	mesh
Oneskorenie	ms	6
Max. počet uzlov na gateway	-	65*103
Teoretický čas života na AAA batérii	rok	1 - 2

Tab. 2.3: Prehľad parametrov technológie Bluetooth Low Energy

Obr.2.1 graficky znázorňuje využitie a rozdiely medzi technológiou Bluetooth a Bluetooth Low Energy. Výhodou je, že novšie verzie smartfónov, tabletov či notebookov podporujú už obe tieto technológie.



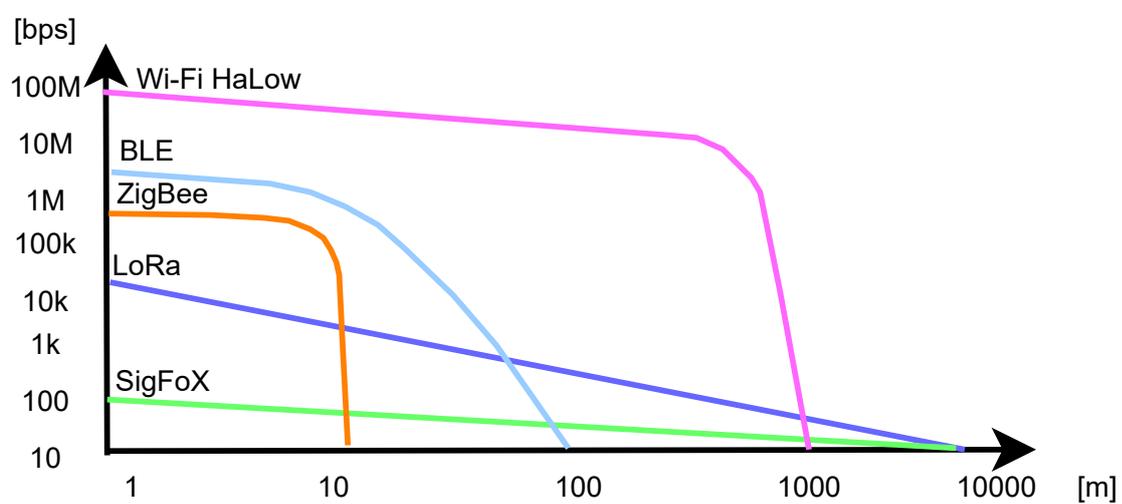
Obr. 2.1: Bluetooth v porovnaní s BLE

## 2.4 Wi-Fi HaLow

Wi-Fi HaLow je názov pre nízko rýchlostnú, úspornú Wi-Fi s veľkým dosahom. Táto technológia využíva licencované pásma s frekvenciou pod 1 GHz. V praxi to znamená dlhší dosah signálu a lepšiu priepustnosť cez prekážky až do vzdialenosti 1 km. Pri týchto využívaných frekvenčných pásmach nie je potrebné vysielat s vysokým výkonom, čo má za následok zlepšenie úspory elektrickej energie. Wi-Fi HaLow pracuje najčastejšie v mesh topológii [21].

Wi-Fi HaLow		
Veličina	Jednotka	Hodnota
Pracovná frekvencia	MHz	900
Prenosová rýchlosť	kbps	>150
Max. dĺžka prenášanej správy	byte	
Max. dosah vo voľnom priestranstve	km	>1
Modulácia	-	
Topológia	-	mesh
Oneskorenie	ms	
Max. počet uzlov na gateway	-	8000+
Teoretický čas života na AAA batérii	rok	

Tab. 2.4: Prehľad parametrov technológie Wi-Fi HaLow



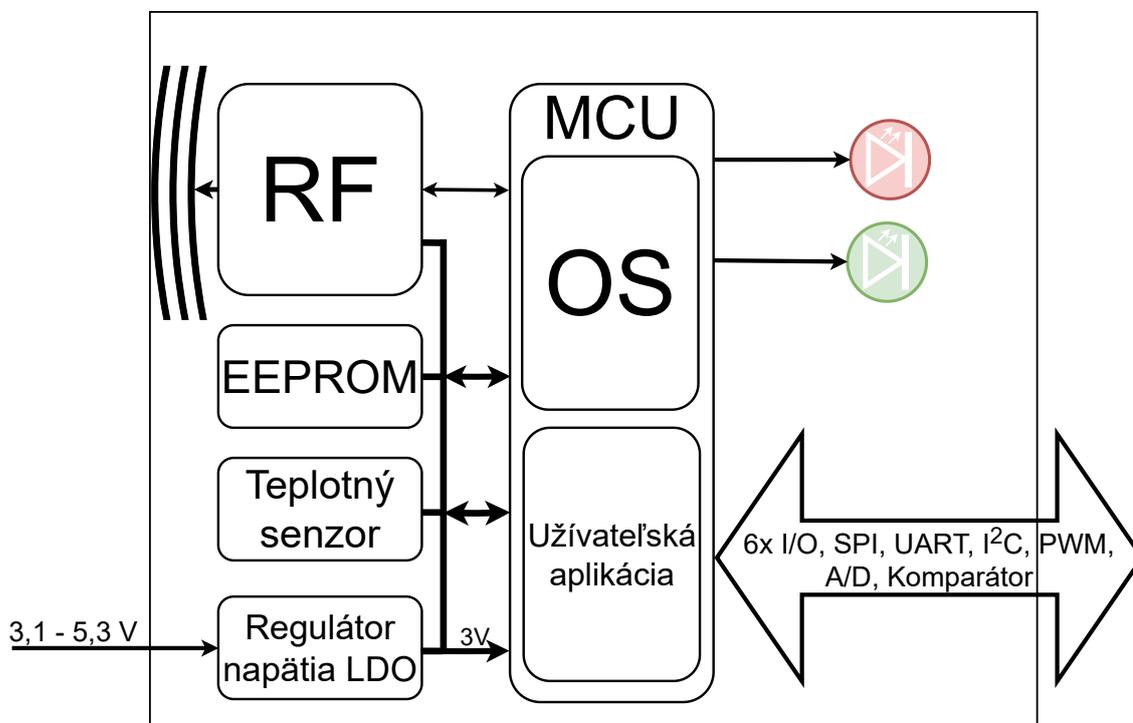
Obr. 2.2: Pokles prenosovej rýchlosti jednotlivých technológií pre IoT v závislosti od vzdialenosti od vysielača

### 3 Návrh technického riešenia modulu pre vysielateľ

Pre našu prácu bol vybraný a použitý IQRF vysielateľ TR-72D, ktorý sa na dosku aplikuje pomocou konektoru pre SIM kartu. Pre pripojenie vyrábaného modulu sú dostupné dve zbernice M2 E-key a PCI-Express. Na týchto zberniciach je možné komunikovať a prenášať informácie pomocou viacerých protokolov SPI, UART a I2C. Dostupné zbernice a protokoly sú nižšie popísané a porovnané.

#### 3.1 IQRF vysielateľ TR-72D

V module je použitý IQRF vysielateľ TR-72D s integrovanou PCB (Printed Circuit Board) anténou priamo na doske. Tento vysielateľ vysielá podľa nastavenia na frekvencii 868 MHz alebo 916 MHz. Vysielateľ je možné vsunúť do konektoru na SIM kartu, vďaka jeho malým rozmerom 31.8 mm x 14.9 mm x 3.3 mm (d x š x v). Výhodou tohto vysielateľa je, že obsahuje vstavaný regulátor napätia. Pre laboratórne použitie nám postačuje verzia s integrovanou PCB anténou, nakoľko nie je potrebné pokrytie rozsiahlych oblastí. Na obr. 3.1 je znázornená bloková schéma tohto vysielateľa [22], [23], [24].



Obr. 3.1: Bloková schéma vysielateľa

### 3.1.1 Technické parametre IQRF vysieláča TR-72D

Nižšie sú uvedené technické parametre použitého vysieláča TR-72D, ktoré udáva výrobca v technickom popise. Tieto parametre sú dôležité pre správnu voľbu vysieláča pre danú inštaláciu (napr. teplota okolia v mieste použitia vysieláča ...) a tak tiež pre tvorbu obvodu, v ktorom bude IQRF vysieláč použitý (napr. napájacie napätie ...) [24].

Veličina	Jednotka	Hodnota
Napájacie napätie	V	3.1 až 5.3
Výstupné napätie	V	3
Výstupný prúd	mA	60
Pracovná teplota	°C	-40 až +85
Odoberaný prúd počas hlbokého spánku	uA	1.7
Odoberaný prúd počas spánku	uA	2.3
RF spánok	mA	1.95
RF pripravený	mA	2.8
RX STD	mA	12.1
RX LP	uA	260
RX XLP	uA	18.5
TX mód	mA	8.3 až 25
Spotreba LED na jednu diódu	mA	2
RF pásmo pre Európu	MHz	868
RF modulácia	-	GSFK
Maximálna prenosová rýchlosť	kbps	19.8
Efektívny vyžarovací výkon	Dbm	2
Dosah pokrytia (voľné priestranstvo)	m	500

Tab. 3.1: Technické parametre vysieláča TR-72D



### 3.1.3 Vývody vysieláča TR-72D a ich funkcionlita

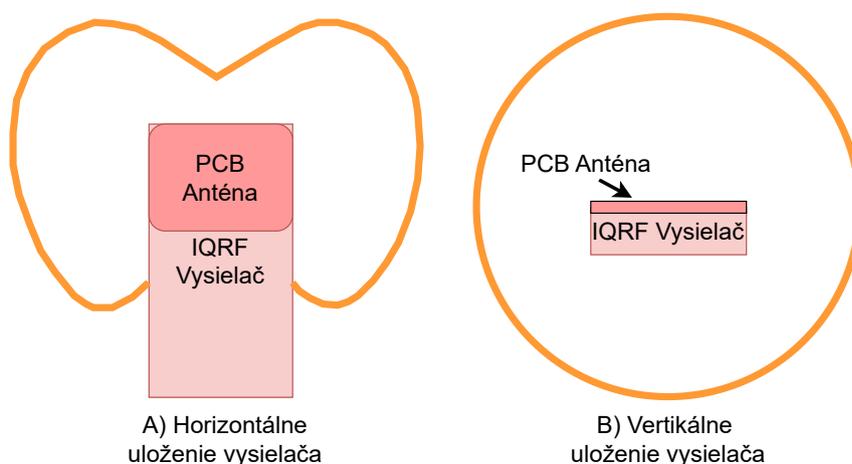
V tabulke 3.2 je uvedená funkcionlita jednotlivých pinov vysieláča TR-72D. Vlastnosti jednotlivých pinov sú taktiež dôležitou informáciou pri návrhu obvodu pre daný vysieláč. Vysieláč TR-72D disponuje pinom pre napájanie obvodu (C3), zemniacim pinom (C4). Ostatné piny slúžia na komunikáciu s vysieláčom podľa zvoleného typu komunikácie.

C1	IO/ADC/C-IN	všeobecný I/O pin analógový A/D vstup vstup komparátora
C2	IO/VOUT	všeobecný I/O pin výstup +3 V z LDO
C3	VIN	vstup pre napájacie napätie
C4	GND	zemniaci pin
C5	IO/ADC/TX/-SS/PWM/COUNT	všeobecný I/O pin SPI sleave select analógový A/D vstup výstup komparátora UART TX PWM výstup programovateľný pull-up a wake up
C6	IO/SCK/SCL	všeobecný I/O pin SPI clock vstup I2C clock
C7	IO/SDI/SDA	všeobecný I/O pin SPI dáta I2C data
C8	IO/RX/SDO	všeobecný I/O pin SPI dátový výstup UART RX

Tab. 3.2: Funkcionlita jednotlivých kontaktov vysieláča TR-72D

### 3.1.4 Dosah pokrytia vysielateľom TR-72D

Pokrytie vysielateľom sa líši v závislosti na rôznych parametroch konkrétnej inštalácie. V prípade obkolesenia vysielateľa predmetmi ako sú napríklad veľké batérie alebo kovové predmety sa vyžarovaný signál odráža naspäť a následne pôsobí proti signálu vysielateľa, a tým znižuje jeho dosah. Dosah vysielateľa je taktiež ovplyvniteľný softvérovovo. Vysielací výkon vysielateľa je možné nastaviť pomocou výberu z ôsmich úrovní. Veľký vplyv na dosah vysielateľa má orientácia a uloženie samotného vysielateľa, pretože vysielacia charakteristika antény nie je na všetky smery rovnaká ako je vyobrazené na obr.3.3 [24].

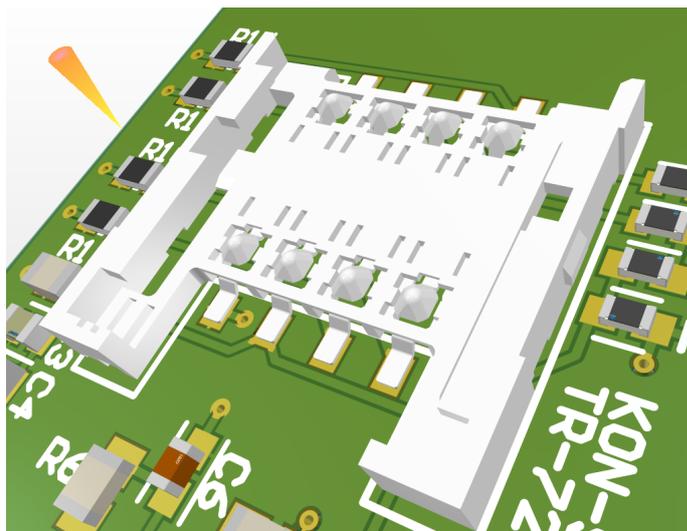


Obr. 3.3: Tvar dosahu vysielateľa pri rôznom uložení

Poznámka: Oranžová čiara znázorňuje tvar oblasti pokrytej vysielateľom.

## 3.2 Konektor vo formáte SIM karty KON-SIM-02

KON-SIM-02 je konektor určený na pripojenie IQRF vysielateľov. Je kompatibilný s mnohými modelmi, vrátane modelu TR-72Dx, ktorý je v tejto práci použitý. Konektor je určený na pájkovanie priamo na dosku. Je zhotovený z odolného plastu a výrobca deklaruje výdrž 10 000 pripojení. Kontakty sú vyrobené z bronzu s prídavkom fosforu a kontaktné plochy sú zase vyrobené z pozláteného niklu. Prechodový odpor kontaktov je určený maximálne na 100 M $\Omega$ , ale ideálna hodnota prechodového odporu je 50 M $\Omega$ . Konektor je dimenzovaný na prúdy do 0,5 A. Pracovná teplota tohto konektoru je široká a to v rozpätí od -40 až do +85 °C [25], [26].



Obr. 3.4: SIM konektor pre pripojenie vysielaca k modulu

### 3.3 Gateway GW-USB-06

Gateway GW-USB-06 je gateway alebo aj koordinátor, ktorý vychádza z TR-7xD architektúry a je priamo prepojitelný do počítača prostredníctvom USB rozhrania. Tento gateway je určený na prepojenie počítača so sieťou IQRF zariadení. Výhodou tohto zariadenia je to, že používateľ je schopný naprogramovať si vlastné špecifické funkcie pre svoje využitie. Gateway má taktiež viacero pokročilých funkcií ako sú diagnostický prípravok pre aplikácie, bezdrôtový programátor (RFPGM) pre vysielace a RTCC (hodiny s reálnym časom). Vysielací výkon koordinátora je 11 dBm a jeho maximálna prenosová rýchlosť je 19,836 kbps [27].



Obr. 3.5: IQRF USB koordinátor

### 3.4 Zbernica M.2 key E

M.2 je štandard, ktorý sa používa na pripojenie rozširujúcich kariet v počítačoch a iných zariadeniach priamo na ich základných doskách. M.2 - je náhrada za mSATA rozhranie, ktoré bolo Intelom predstavené v roku 2012. Použitie tohto rozhrania je veľmi rôznorodé, pretože je schopné prenášať rôzne signály ako sú SATA, USB alebo PCIe, čo umožňuje zariadeniam využívajúcim toto rozhranie rozšíriť svoje možnosti o siete WLAN, Bluetooth a pod. Veľmi často sa používa toto rozhranie na pripojenie SSD disku v notebooku alebo inom zariadení. Moduly M.2 sú kompaktné a majú veľmi flexibilné vlastnosti vhodné pre merania. V praxi sa v elektronických zariadeniach používa mnoho dostupných druhov konektorov tohto rozhrania označených pomocou slovíčka key (kľúč) v kombinácii s veľkým písmenom abecedy. Jednotlivé prevedenia je možné aj navzájom kombinovať. Na doske použitej v práci je dostupné prevedenie M.2 key E. Toto prevedenie kontaktov rozhrania sa skladá z 12 a 19 kontaktných plôch, ktoré sú oddelené medzerou. Označenie key E neznamená len rozdielny vzhľad konektora, ale taktiež aj podporu iného množstva a druhu prenosového signálu. M.2 key E podporuje prenos 2x PCIe a USB signál [28], [29].



Obr. 3.6: Zbernica M.2 key E na BPC-IOT doske

### 3.5 Zbernica mini PCI-Express

Mini PCI-Express (PCIe) je zbernica, ktorá je v dnešnej dobe veľmi často používaná v počítačoch a iných elektronických zariadeniach vyžadujúcich pripojenie externého zariadenia s vysokou prenosovou rýchlosťou. PCIe zbernica je využívaná na duál-simplexné, point-to-point nízkonapäťové sériové spojenia. Prenos signálu je zabezpečený vysoko-rýchlostne až 2,5 Gbit za sekundu s 8/10 bitový kódovaním, aby sa do dátového toku vložil hodinový signál. Na prenosovej strane sú paralelné dáta posunuté sériovo a na prijímacej strane sú zase dáta posunuté do registrov používaných pre paralelný výstup dát. Na tejto strane prijímač taktiež znovu obnoví hodinový signál prenášaný s dátami. Kanál PCIe je tvorený vysielačím a prijímacím nízkonapäťovým diferenciálnym signálnym párom. Pakety s dátami sú posielané simplexne s frekvenciou 2,5 GHz. Vysielač aj prijímač kanál prenášajú dáta s rýchlosťou 250 MBps, čo v kombinácii tvorí 500 MBps. Prenos môže byť tvorený viacerými kanálmi v závislosti od prevedenia x1, x4, x8, x16 a x32. Toto prevedenie má potom aj vplyv na spôsob použitia. Najčastejšie používané prevedenia sú x1 pre desktopy, x4 pre servery a x16 je často využívaná grafickými kartami. Riadiace impedančno diferenciálne stopy umiestnené na doske sú 100  $\Omega$ . Prenášaný signál dosahuje rôznu amplitúdu v závislosti od významu signálu. Prvý bit je prenášaný s amplitúdou viac ako 800 mV a každý ďalší bit má amplitúdu 500 mV. Každá matica PCIe je navrhnutá s autokonfiguráciou, to znamená, že ak je pripojené zariadenie s menšou zbernicou, piny sú automaticky rozpoznané a informácie sú distribuované správne [30].

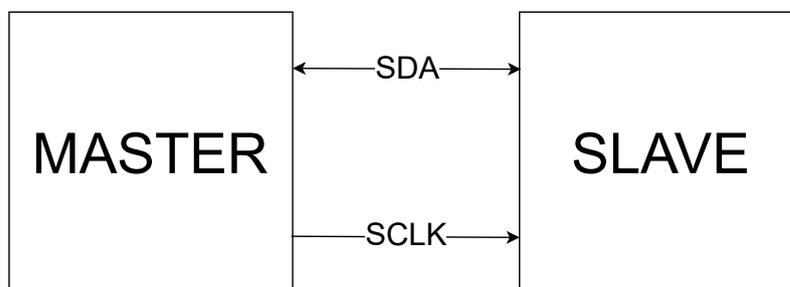


Obr. 3.7: Zbernica mini PCI-Expres na BPC-IOT doske

## 3.6 Rozhranie I2C

I2C je zbernica, ktorá bola vyvinutá firmou NXP Semiconductors. I2C je zbernica, komunikujúca za použitia dvoch vodičov, radiaca sa medzi jednoduché zbernice. Komunikácia po zbernici prebieha synchrónne, a preto je jeden z vodičov označený SCL (Synchronization CLock) a druhým vodičom sú prenášané dáta. Tento vodič nesie označenie SDA. Pripojené zariadenia pracujú v režime s otvoreným kolektorom (open drain). Pri pripojení takéhoto zariadenia sa využívajú pull-up rezistory, ktoré sú pripojené medzi napájacím napätím a jedným z vodičov tejto zbernice. Väčšinou sa využívajú rezistory s odporom 4700 ohmov (4k7). Takéto zapojenie v obvode zabezpečí to, že ľubovoľné zariadenie môže v ktoromkoľvek časovom okamihu zmeniť hodnotu na zbernici tak, že žiadne zariadenie by nemalo byť poškodené. Základná stavba tejto zbernice sa skladá z minimálne jedného master zariadenia riadiaceho komunikáciu, ktorá prebieha na zbernici a generujúceho hodinový CLK (clock) signál a jedného zariadenia typu slave, ktoré reaguje na dáta od master zariadenia. Zariadenie typu slave je väčšinou nejaké zariadenie na periférii.

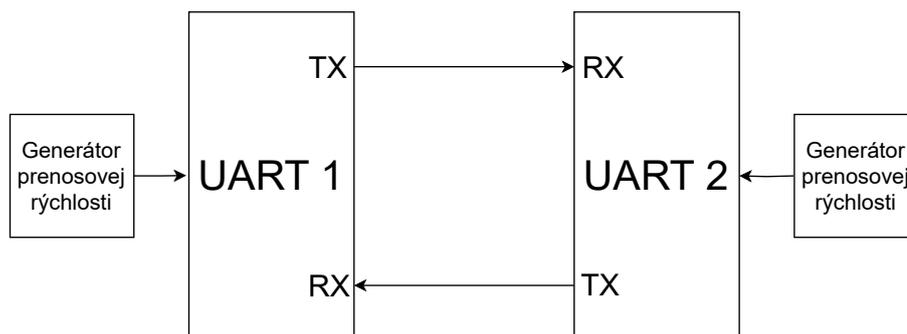
Ak master zariadenie chce začať komunikáciu vyšle štartovaciu sekvenciu, ktorá počas vysokej úrovne SCL zmení SDA do nízkej úrovne, následne sa do nízkej úrovne zmení aj SCL. Táto štartovacia sekvencia je potrebná, lebo v pokojovom režime sú oba kontakty v logickej 1, čiže vo vysokej úrovni. Následne je odoslaná 8 bitová adresná sekvencia, ktorá určí akým smerom bude komunikácia pokračovať. Master zariadenie počas deviateho hodinového impulzu nechá SDA vo vysokej úrovni. SDA je stiahnuté do nízkej úrovne až v okamihu, keď je rozpoznané nejaké slave zariadenie, ktoré je správne adresované a je pripravené na komunikáciu. Tento proces sa nazýva potvrdenie (Acknowledge), z čoho pochádza aj skratka tohto bitu (ACK). Tento bit je používaný po každom komunikačnom bite. Jeho funkciou je potvrdiť schopnosť ďalšej komunikácie, ktorá je signalizovaná nízkou úrovňou ACK. Ak by ACK neskončilo v nízkej úrovni, komunikácia bude prerušená a ukončená. Komunikácia sa ukončuje vyslaním sekvencie stop (počas vysokej úrovne SCL je SDA uvoľnené) [31].



Obr. 3.8: Architektúra I2C

## 3.7 Rozhranie UART

Komunikačný protokol UART (Universal Asynchronous Receiver-Transmitter) je asynchrónneho charakteru, ktorý vie pracovať v plne duplexnom režime. UART je často používaný protokol v telekomunikačných a kontrolných systémoch. UART komunikuje po dvoch signálových trasách, aby bolo dosiahnuté práve plne duplexné spojenie. Komunikačné cesty sú označené RXD a TXD. Architektúra komunikačného protokolu UART sa skladá z generátoru prenosovej rýchlosti, prijímacieho a vysielacieho modulu. Generátor prenosovej rýchlosti má za úlohu urýchliť asynchrónny sériový prenos dát. Generátor prenosovej rýchlosti pracuje ako rozdeľovač hodinového signálu. Prijímací modul prijíma dáta prichádzajúce na kontakty tohoto modulu. Začiatok prijímania signálu je označený prechodom vysokej úrovne 1 do nízkej úrovne 0. Vysielací modul mení dáta z paralelných na sériové a pridáva kontrolné bity. Z časovacieho diagramu UART protokolu je zrejmé, že stav nečinnosti je počas vysokej úrovne signálu. Zmena tejto úrovne na nízku značí začiatkový bit, nasledovných 8 bitov sú prenášané dáta. Po prenášaných dátach nasleduje koncový bit, ktorým je zaznačené to, že dáta už boli spracované. Po tomto bite môže zariadenie znova prejsť do stavu nečinnosti do tzv. idle mode [32].



Obr. 3.9: Architektúra UART

## 3.8 Rozhranie SPI

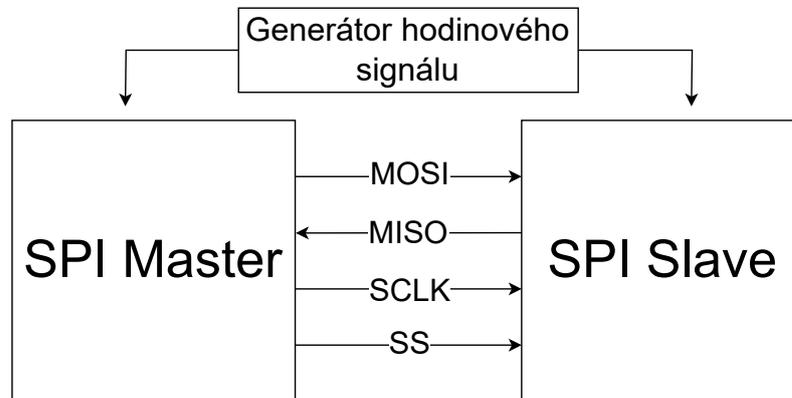
SPI (Serial Peripheral Interface) je spôsob komunikácie založený na dvoch moduloch master a slave. SPI protokol využíva 3+N ciest, kde pre N platí, že udáva počet slave modulov. Tieto cesty zastávajú funkcie MOSI (Master Out Slave In), MISO (Master In Slave Out), SCLK (Slave Clock) a SS (Slave Select).

Architektúra komunikačného protokolu SPI je rozdelená do troch blokov:

- Generátor hodinového signálu
- Master modul

- Slave modul

Vyššie uvedené cesty MOSI, SS a SCLK sú súčasťou práve Master modulu. MOSI cesta je využívaná na komunikáciu mastra so slave. MISO je jediná cesta riadená slave modulom, používaná na výstup informácií z daného modulu (senzoru). Frekvenciu poskytovania dát slave modulom určuje SCLK master [32].

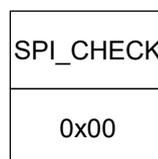


Obr. 3.10: Architektúra SPI

### 3.8.1 SPI v IQRF

Pre komunikáciu s využitím SPI v zariadeniach technológie IQRF je dôležité správne implementovať štruktúru odosielanej správy napr. od mastra. Táto štruktúra obsahuje niekoľko bajtov na začiatku každej správy, ktoré prikazujú slave zariadeniu ako má so správou postupovať v ďalšom spracovaní. Základným delením SPI paketov je paket typu SPI\_CHECK a SPI\_CMD.

SPI\_CHECK neobsahuje žiadne telo správy, iba sa dotazuje na aktivitu opačnej strany v SPI komunikácii. V praxi je tento paket používaný veľmi často a zariadenia si vymieňajú informácie o svojej aktivite približne každých 10 ms. Implementácia paketu SPI\_CHECK je pomerne jednoduchá, pretože celý paket je reprezentovaný iba jedným znakom a to v hexadecimálnom tvare čísla 0 - (0x00) vid. obr.3.11.



Obr. 3.11: SPI\_CHECK paket

SPI\_CMD je už na implementáciu o čosi náročnejší, pretože tento paket obsahuje už viacero častí vid. obr.3.12. Prvou časťou je informácia o tom, že je to read/write SPI\_CMD paket, čo je reprezentované prvým hexadecimálnym znakom (0xF0). Druhým číslom v pakete je takzvaný PTYPE, ktorý je tvorený 8 bitovým číslom vid. obr.3.13. Bit s číslom 7 má hodnotu 1, ak sa majú dáta prepisovať alebo 0 v prípade, že chceme aby tieto dáta nezmenili predošlú hodnotu údajov. Ostatných 7 bitov vyjadruje dĺžku dát v správe (SPIDLEN - 1 až 64 bitov), ktorá bude nasledovať. Ďalších n znakov je samotná správa - napríklad odosielaná teplota. Tieto znaky správy sa označujú ako:

$$DM_1, DM_2, \dots, DM_{SPIDLEN}$$

Predposledným znakom v pakete je kontrolný znak CRCM, ktorý je vypočítaný ako:

$$SPI\_CMD \text{ xor } PTYPE \text{ xor } DM_1 \text{ xor } DM_2 \text{ xor } \dots \text{ xor } DM_{SPIDLEN} \text{ xor } 0x5F$$

Takto vytvorený paket je pripravený na prenos dát z mastra do slave zariadenia, ktorý je v našom prípade realizovaný prenosom dát z Raspberry Pi pico (master) do IQRF vysielača (slave). [33]

SPI_CMD	PTYPE	DM <sub>1</sub>	DM <sub>2</sub>	...	DM <sub>SPIDLEN</sub>	CRCM
---------	-------	-----------------	-----------------	-----	-----------------------	------

Obr. 3.12: Paket SPI komunikácie v IQRF

b7	b6	b5	b4	b3	b2	b1	b0
CTYPE	SPIDLEN						

Obr. 3.13: Tvorba znaku PTYPE

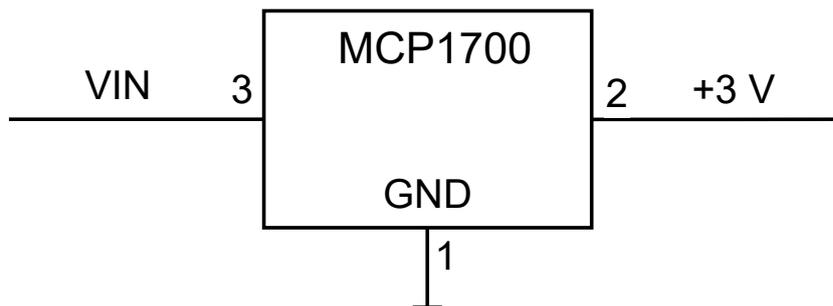
### 3.9 Návrh obvodu pre IQRF modul

V nasledujúcich podsekcích sú popísané základné bloky, ktoré sú potrebné pre správne napájanie a komunikáciu IQRF modulu a dosky BPC-IOT, na ktorej je modul osadený. Pre pripojenie IQRF vysielača a dosky BPC-IOT bola vybraná karta v prevedení mini PCIe.

Zapojenie modulu bolo navrhnuté podľa odporúčaní výrobcu. Schéma tohto zapojenia bola navrhnutá pomocou programu Altium Designer. Výsledná schéma IQRF modulu sa nachádza v prílohe C.

### 3.9.1 Napájanie

Pri napájaní je dôležité dbať na precíznosť, hlavne ak je použité napätie z DC/DC meniča alebo pulzného zdroja, pretože tým vieme zabrániť vzniku ďalších komplikácií. Použitý vysielateľ obsahuje vstavaný napäťový regulátor (LDO) s malými stratami (obr. 3.14), ktorého vstupné napätie by malo byť z rozsahu 3,1 V až 5,3 V. Výstupné napätie tohto regulátora je +3 V s toleranciou 60 mV, ktoré je určené hlavne na napájanie vnútorných obvodov vysielача. Výhodou vstavaného regulátora napätia je externý výstup napätia pre napájanie prídavných obvodov až do odberu 100 mA [34].



Obr. 3.14: Regulátor napätia vstavaný vo vysielачi

### 3.9.2 Stabilizácia napätia (odstránenie spätnej väzby)

Spätaná väzba sa odstraňuje, aby sa zabezpečilo dostatočne stabilné vstupné napätie bez nežiadúcich emisií. Napájací zdroj musí byť dostatočne stabilným, nakoľko počas vysielania a prijímania je odber elektrickej energie nestály a kolíše v závislosti na výkone. Neschopnosť zdroja dodávať konštantné napätie by ovplyvnilo celkové správanie vysielania a prijímania, ako sú napr. dosah či iné vysielacie alebo prijímacie parametre.

Stabilitu zdroja vieme zabezpečiť pripojením stabilizačných kondenzátorov na prírody napätia pre vysielateľ a uzemňovací kontakt. Stabilizačné kondenzátory by mali byť pripojené čo najbližšie ku kontaktom vysielача. Typ a veľkosť kondenzátorov je potrebné vyberať podľa typu aplikácie a kvality prívodného zdroja napätia. Najčastejšie je používaná dvojica tantalového kondenzátora s hodnotou okolo 10  $\mu\text{F}$  a keramického kondenzátora s kapacitou z rozmedzia 100 nF až 1  $\mu\text{F}$ . Ak chceme odstrániť rušivé frekvencie, nimi spôsobené prechodové prúdy a zvlnenia z frekvencií blízkych GHz, je potrebné ešte pridať kondenzátor s nízkou kapacitou - iba niekoľko desiatok pF. Pravidlom by malo ostať to, že čím nižší predradný kondenzátor je pripájaný do obvodu, tým by mal byť pripojený bližšie ku napäťovému vstupu do vysielача [34].



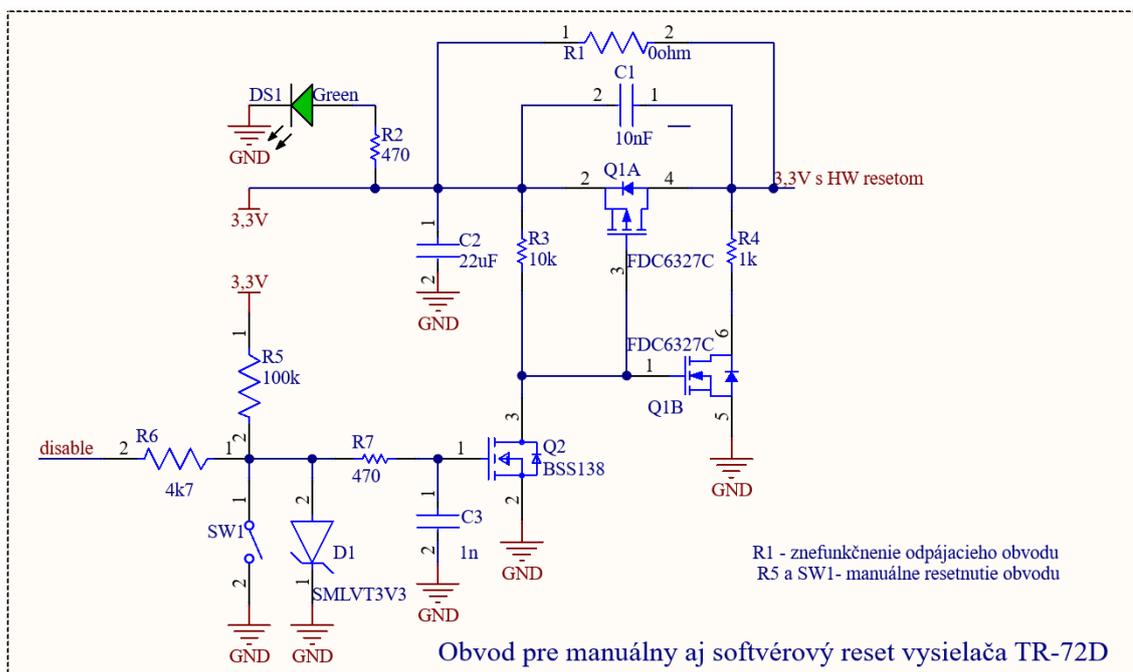
Obr. 3.15: Stabilizácia napätia

### 3.9.3 Filtrácia rušenia

Dôležitou vlastnosťou napájacieho zdroja je, aby dokázal zabrániť šíreniu prechodového prúdu výstupnej frekvencie cez vedenie elektrickej energie. Z tohto dôvodu je potrebné dodržiavať optimálne rozloženie súčiastok na doske plošných spojov. Správne filtrovaný musí byť aj napájací zdroj. Pre filtračný okruh platí zásada, že by mal byť umiestnený čo najbližšie ku vstupu napätia do vysielача. Pri napájaní zo zdroja s predpokladom už vyfiltrovaného napätia tento blok možno vynechať [34].

### 3.9.4 Odpájanie (hardwerový reset)

Ak je IQRF vysielач uložený na ťažko dostupnom mieste, pri poruche alebo novom nastavení, je vhodné doplniť modul o funkciu vzdialeného resetu. Keďže mini PCIe karta disponuje kontaktom s privedeným resetovacím signálom, dosiahnutie funkcie vzdialeného resetu je pomerne jednoduché. Na tento jednoduchý obvod použijeme komplementárne zapojenia MOSFET tranzistorového páru. Pre jednoduchosť je modul vybavený aj tlačidlom pre reset vysielача manuálne priamo na doske. Súčasťou obvodu je aj odpor s nulovou hodnotou pre znefunkčnenie daného obvodu kvôli neskorším vývojovým potrebám. V obvode je tiež zakomponovaná zelená luminiscenčná dióda pre signalizáciu funkčnosti privedeného napájacieho napätia. Zapojenie tohoto obvodu je na obr.3.16 [34].



Obr. 3.16: Hardvérový reset



## 4 Realizácia hardvéru laboratórnej úlohy

Technická realizácia laboratórnej úlohy obnáša výrobu modulu s úpravou jeho rozmerov, osadenia modulu súčiastkami, vybavenie modulu IQRF vysielačom a osadenie modulu s vysielačom do slotu mini PCIe karty na BPC-IoT doske. Táto doska musí byť správne vybavená jumper prepojkami pre funkčnosť pripojených zariadení. V neposlednom rade musí byť ku BPC-IoT doske pripojené teplotné čidlo na získanie teploty okolia.

### 4.1 Výroba PCB dosky modulu

Z návrhu v programe Altium Designer boli vytvorené potrebné súbory (NC Drill a Gerber), ktoré sú dostupné v prílohe E pre výrobu dosky modulu. Výroba bola po odporučení vedúcim práce a predošlých skúsenostiach zadaná firme JLC PCB, kam boli zaslané podklady pre výrobu modulu. Pre výrobu dosky boli zvolené nasledovné základné nastavenia:

Parameter	Hodnota	Jednotka
Základový materiál	FR-4	-
Povrchová úprava	HASL (s olovom)	-
Farba PCB	zelená	-
Farba popisu	biela	-
Rozmery dosky	30 x 51.8	mm
Hrúbka PCB	1	mm
Tolerancia rozmerov	$\pm 0,2$	mm

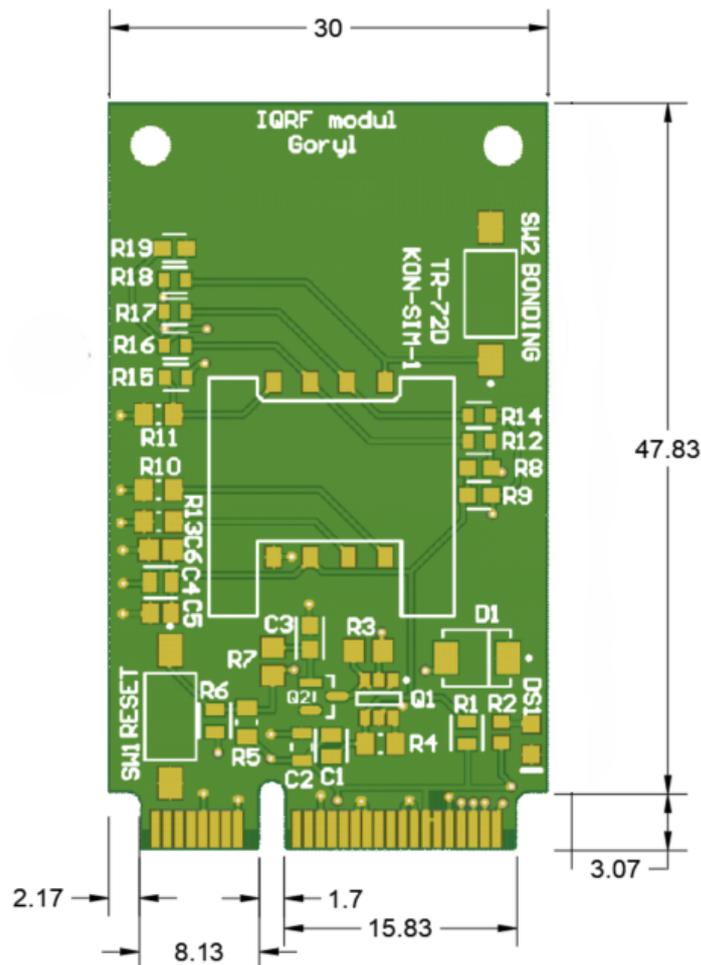
Tab. 4.1: Parametre PCB zaslané do výroby

Parametre pre výrobu dosky sa vo veľkej miere zhodujú so základnými parametrami pre výrobu PCB dosky. Jediným rozdielom pri výrobe modulu bolo použitie tenšej dosky. Zvolená hrúbka je iba 1 mm a to z dôvodu osadenia modulu do slotu mini PCIe karty, ktorý je pripravený na osadenie modulu práve s týmito rozmermi. Druhým rozdielom je znížená tolerancia rozmerov, ktorá je potrebná pre presnosť kontaktov mini PCIe karty a jej slotu.

### 4.2 Úprava rozmerov dosky modulu

Pri návrhu modulu v prevedení mini PCIe boli použité rozmery pre dosku dostupné na internete, ktoré boli následne skontrolované odmeraním inej dosky v prevedení

mini PCIe karty. Aj napriek takejto dvojnásobnej kontrole bola doska, po výrobe približne o 1 mm väčšia ako bolo možné vložiť do slotu na BPC-IoT doske. Úprava rozmerov bola vykonaná pomocou brúsneho papiera a pravouhlosť vybrúsenej časti bola zabezpečená pravouhlou sústavou na brúsnej doske. Vybrúsená časť dosky modulu bola následne ošetrená tekutým lakom na PCB dosky.



Obr. 4.1: Rozmery modulu

### 4.3 Osadenie dosky modulu súčiastkami

Pre prvú prototypovú verziu modulu bolo zvolené ručné osadenie dosky súčiastkami v študentskom laboratóriu. Použité rezistory boli z rady E24 - 1% a E12 - 5%. Po osa-

dení súčiastok bola otestovaná základná funkcionálna doska - premeranie odporu a napätia. Vykonaná bola aj kontrola spojenia komunikačných ciest SPI zbernice medzi kontaktami slotu mini PCIe karty na BPC-IoT doske a SIM konektorom na doske modulu. Prototyp modulu bol ďalej použitý pre vývoj laboratórnej úlohy a programové testy.

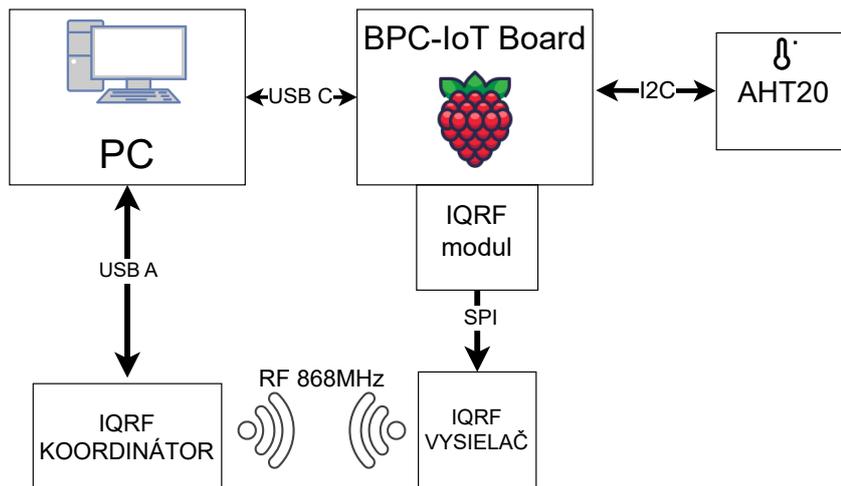


Obr. 4.2: Osadená doska modulu súčiastkami

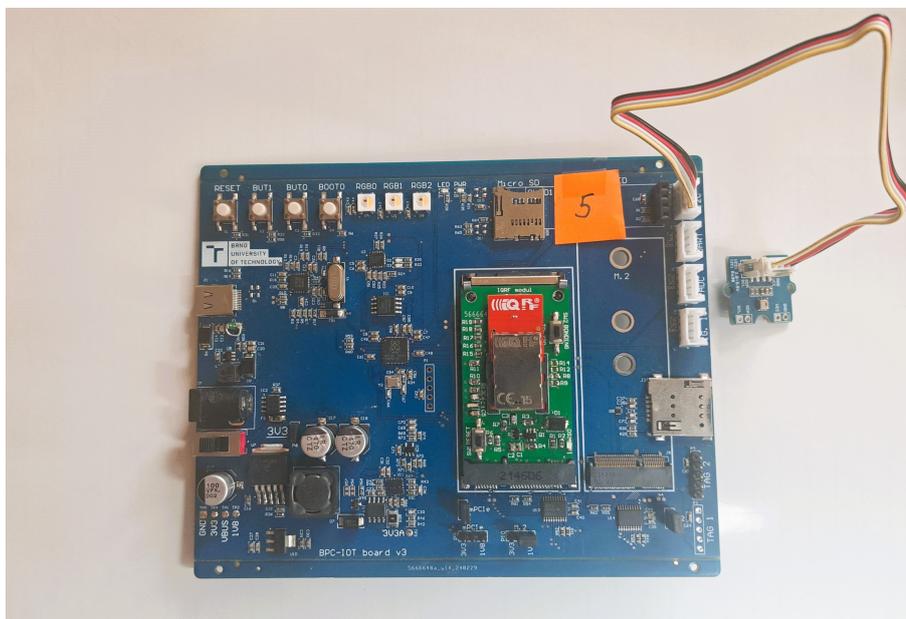
## 4.4 Osadenie BPC-IoT dosky

BPC-IoT doska je doska, na ktorej je osadený mikrokontrolér Raspberry Pi Pico. V prevedení laboratórnej úlohy je potrebné tento mikrokontrolér naprogramovať tak, aby pomocou IQRF vysielača bola odoslaná teplota z externého čidla na koordinátor. Pred samotným programovaním je nutné si ku doske pripojiť všetky využívané časti obvodu tak, aby bolo možné teplotu získať a odoslať. Na BPC-IoT dosku pripojíme teplotné čidlo AHT20. Toto čidlo s mikrokontrolérom komunikuje po zbernici I2C, z čoho jasne vyplýva aj miesto jeho pripojenia ku BPC-IoT doske. Ďalšou nevyhnutnou súčasťou pre odoslanie teploty je IQRF vysielač. Vysielač

TR-72DA vložíme najskôr do v práci navrhnutého modulu, ktorý následne celý vložíme do slotu pre mini PCIe kartu. Ako posledné je potrebné na BPC-IoT doske umiestniť jumper prepojky na správne kontakty na doske. Keďže modul je v prevedení mini PCIe karty je nevyhnutné umiestniť jumper prepojku práve na tieto piny. Druhá prepojka slúžiaca na zvolenie správnej úrovne komunikačnej logiky musí byť umiestnená do polohy 3,3 V, pretože využívaný IQRF vysielač pracuje práve s touto logickou napäťovou úrovňou.



Obr. 4.3: Bloková schéma zapojenia použitých zariadení



Obr. 4.4: Osadená BPC-IoT doska

## 5 Návrh laboratórnej úlohy

Cieľom laboratórnej úlohy je zoznámiť študentov s možnosťami a základnými funkciami technológie IQRF. Technológia IQRF ako taká neobsahuje iba konkrétny hardvér (vysielače, gateway či senzory), ale aj ovládací a programovací softvér IQRF IDE. Laboratórnu úlohu sme rozdelili na dve časti:

- Vytvorenie komunikácie bod - bod
- Vytvorenie mesh siete
- Bonusová úloha

### 5.1 Základné informácie o vývojovom prostredí IQRF IDE

Laboratórna úloha bola realizovaná vo vývojovom prostredí IQRF IDE, ktoré je orientované na vývoj projektov technológie IQRF s grafickým užívateľským prostredím. Pre vývojárov poskytuje komplexnosť, pretože IQRF IDE podporuje dizajnovanie, nahrávanie, debugovanie, testovanie a údržbu pre vyvíjané projekty.

Hlavné časti IQRF IDE:

- Debugger a konfiguračný nástroj
- Komunikačný terminál a testovací nástroj
- Správca siete IQMESH (manažment, správa, vizualizácie a testy)
- Servisný nástroj CATS (skener, tester, analyzátor)
- Nástroje na aktualizáciu operačného systému pre vysielače a firmvéru pre USB zariadenia

### 5.2 Vytvorenie point to point komunikácie

Prvá časť laboratórnej úlohy je zameraná na precvičenie point to point (bod - bod) komunikácie, teda spojenia medzi dvoma bodmi (vysielačom a prijímačom, prípadne obojsmerne medzi dvoma vysielačmi alebo medzi vysielačom/uzlom a koordinátorom). V laboratórnej úlohe sme využili nám dostupné zariadenia a bola vytvorená point to point komunikácia medzi USB koordinátorom (GW-USB-06) a vysielačom (TR-72DA), kde bol koordinátor nastavený tak, aby sa správal ako jednoduchý prijímač signálu (správ). Na vytvorenie tejto komunikácie bolo potrebné vytvoriť v programe IQRF IDE projekt a vytvoriť program (TeplotaTX) v jazyku C pre odoslanie teploty zo vstavaného teplotného čidla. Program pre prijímanie dát bol použitý z demo cvičenia E02-RX.c.

Program TeplotaTX.c bolo potrebné nahráť do vysielača pomocou programovacieho modulu a program E02-RX.c zase do USB gateway priamo z počítača. Takto vytvorené a nastavené zariadenia vedia odosielať a prijímať správy po stlačení aplikačného tlačidla (na doske označené ako bonding). V terminal logu prostredia IQRF IDE je možné vidieť prijaté správy vid. obr.5.1 a po rozkliknutí danej správy je možné vyčítať jej obsah a detail (odosielateľ, veľkosť) vid. obr.5.2. Presný návod s jednotlivými krokmi sa nachádza v prílohe A.

Line	Time	Rx/Tx	Length	Data HEX	DPA Message	Error	NADR	PNUM	PCMD
1	19:34:48.513	Rx	9	54.3D.20.32.39.2E.32.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
2	19:35:05.243	Rx	9	54.3D.20.32.38.2E.39.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
3	19:35:11.820	Rx	9	54.3D.20.32.38.2E.38.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
4	19:35:18.007	Rx	9	54.3D.20.32.38.2E.38.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
5	19:35:23.333	Rx	9	54.3D.20.32.38.2E.37.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
6	19:35:29.908	Rx	9	54.3D.20.32.38.2E.36.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
7	19:35:36.486	Rx	9	54.3D.20.32.38.2E.36.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
8	19:35:43.059	Rx	9	54.3D.20.32.38.2E.35.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
9	19:35:49.641	Rx	9	54.3D.20.32.38.2E.35.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
10	19:35:56.213	Rx	9	54.3D.20.32.38.2E.34.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
11	19:36:02.802	Rx	9	54.3D.20.32.38.2E.34.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
12	19:36:09.376	Rx	9	54.3D.20.32.38.2E.33.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
13	19:36:15.945	Rx	9	54.3D.20.32.38.2E.33.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
14	19:36:22.522	Rx	9	54.3D.20.32.38.2E.33.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
15	19:36:29.110	Rx	9	54.3D.20.32.38.2E.33.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	
16	19:36:35.679	Rx	9	54.3D.20.32.38.2E.32.80.43.	DPA Request	0x54 Reserved	0x20 User Peripheral 0x20	0x32 Unknown	

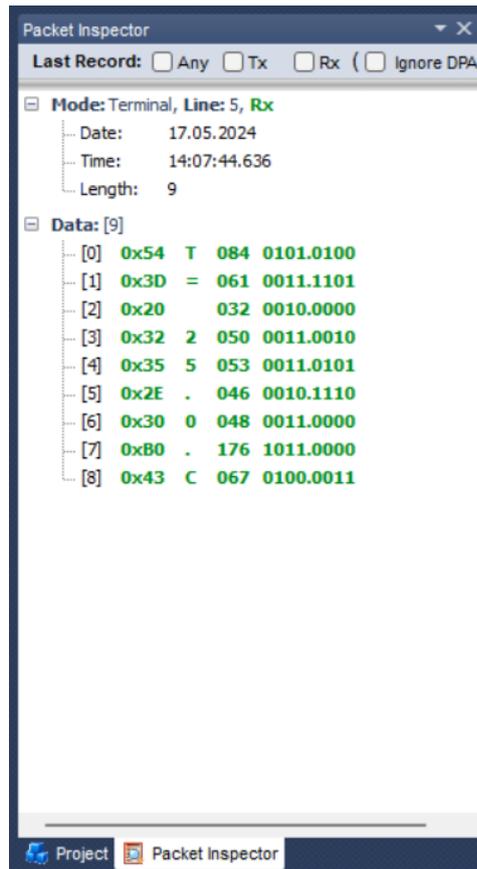
Obr. 5.1: Terminal log v IQRF IDE

## 5.3 TeplotaTXIQRF.c

Program v jazyku C využíva funkcie z IQRF OS (knižnica IQRF.h). Program načíta teplotu z integrovaného teplotného čidla, upraví načítané dáta z čidla, skonvertuje ich na teplotu a túto teplotu zaokrúhli na potrebný počet desiatinných miest. Program pripraví text správy, do ktorej je vložená teplota. Táto správa je následne odoslaná z IQRF vysielača po stlačení aplikačného tlačidla na module alebo BPC-IoT doske (BUT0). V programe sú využité funkcie na riadenie SPI zbernice (startSPI, enableSPI), kopírovanie dát z SPI buffra (bufferCOM) do odosielacieho buffra (bufferRX) a odoslanie správy prostredníctvom IQRF vysielača (RFTXpacket). Zdrojový kód tohto programu sa nachádza v prílohe F.

## 5.4 TeplotaTXRPPico.py

Program v jazyku microPython určený pre riadiacu dosku BPC-IoT, ktorá po stlačení tlačidla BUT0 vyšle signál cez SPI zbernicu na IQRF vysielač pre odoslanie správy s teplotou, ktorá je získavaná interným čidlom na IQRF vysielači. (príloha F)



Obr. 5.2: Detail prijatej správy

## 5.5 Externa Teplota TXIQRf.c

Program v jazyku C využíva funkcie z IQRf OS (knižnica IQRf.h). Program najskôr nainicializuje zbernicu SPI. V nekonečnom cykle sa testuje stav zbernice SPI. Ak nie je zbernica zaneprázdnená, v podmienke sa otestuje či sú na zbernici poslané nejaké dáta a skontroluje sa CRC suma. V prípade, že zbernica je zaneprázdnená, bliká LED. Poslané dáta zo zbernice sa prenesú do buffra na odoslanie (bufferRF) cez vysielateľ. Následne sa dáta (teplota) odošle cez vysielateľ na gateway. V prípade, že na zbernici neboli prijaté žiadne dáta do buffra na odoslanie, je prenesená správa "no data". Ak sa na zbernici dáta objavili, ale CRC suma nie je korektná, je odoslaná správa s textom "CRC err". (príloha F)

## 5.6 Externa Teplota TXRPPico.py

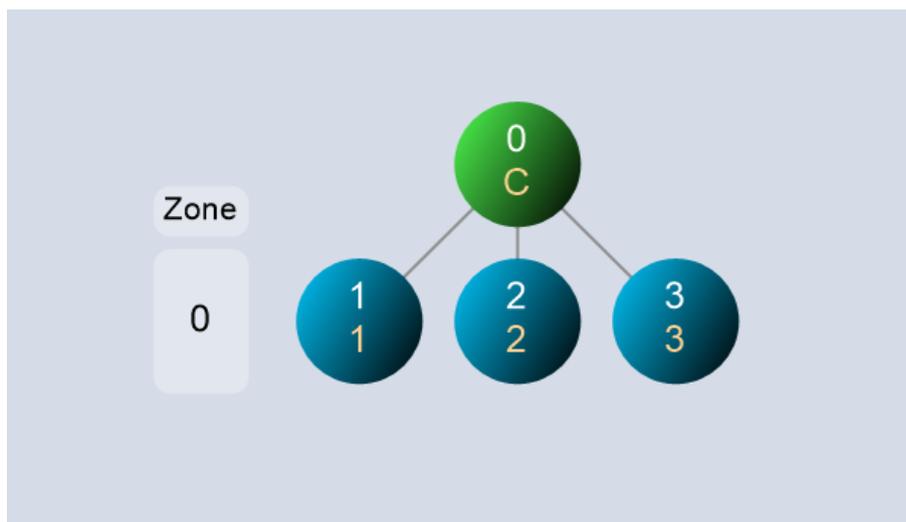
Program v jazyku microPython je určený pre riadiacu dosku BPC-IoT board s mikrokontrolérom Raspberry Pi Pico. Program načíta teplotu z externe pripojeného čidla AHT20, pripojeného ku doske cez zbernicu I2C. Poskladá SPI paket podľa

protokolu na odoslanie a vypočíta CRC sumu, ktorú vloží do SPI paketu. Celý paket prekonvertujete do poľa bajtov na odoslanie cez funkciu `spi.write`. Program využíva knižnice pre komunikáciu po SPI a I2C zberniciach. Taktiež využíva knižnicu pre komunikáciu s teplotným čidlom AHT20. V prvej časti sa zadefinujú komunikačné a ovládacie piny. Následne sa inicializujú obe zbernice a nastaví sa parametre prenosu. S využitím zbernice I2C sa nainicializuje teplotné čidlo AHT20. Po inicializácii všetkých potrebných komunikačných ciest sa načíta teplota z čidla. Táto načítaná teplota sa posiela pre overenie na konzolu. Teplota sa skonvertuje na hexadecimálny tvar, obalí sa úvodnými bajtami a CRC podľa protokolu SPI v I2C. Celý paket sa tiež vypíše do konzoly na overenie. Pred samotným odoslaním paketu s teplotou je komunikácia na SPI zbernici zahájená nastavením riadiaceho signálu na nízku úroveň (logická 0) a po odoslaní dát je táto komunikácia zase ukončená nastavením toho istého signálu na vysokú úroveň (logická 1). Toto opakované odosielanie teploty je vykonávané neustále pomocou nekonečného cyklu. (príloha F)

## 5.7 Vytvorenie I2C mesh siete

Pre vytvorenie siete mesh je potrebné použiť koordinátor a niekoľko podriadených uzlov, ktoré budú sieť tvoriť a bude u nich fungovať routovanie na pozadí. Na vytvorenie siete sa využíva proces, ktorý sa nazýva bonding alebo jednoduchšie (párovanie). Aby tento proces fungoval je potrebné sa uistiť, že uzly (vysielače), ktoré sú v sieti použité, nie sú aktuálne spárované s iným koordinátorom. Pre uistenie sa bolo na začiatok druhej časti laboratórnej úlohy vložené zresetovanie bondovacích nastavení koordinátora aj uzlov za pomoci I2C IDE v prípade koordinátora. V prípade uzlov je tento reset vykonaný použitím presnej kombinácie stlačenia tlačidiel na doske. V programe I2C IDE bolo potrebné pridať plug-iny pre koordinátor a podriadené uzly, ktoré boli nahrané do jednotlivých zariadení. Následne bola vytvorená sieť mesh a v nej prebehol algoritmus objavenia a priradenia virtuálnych smerovacích čísel.

Po vytvorení siete bola navrhnutá krátka úloha o preskúmaní parametrov siete v programe I2C IDE.



Obr. 5.3: Zobrazenie mesh siete v záložke map view v IQRF IDE

## 5.8 Bonus.py

Do laboratórnej úlohy je pridaná aj bonusová úloha, ktorá simuluje reálnu situáciu s IoT sveta. Pre vyriešenie tejto úlohy bolo potrebné upraviť program `Externa-TeplotaTXRPPico.py` tak, že funkcia čítajúca teplotu z čidla zaokrúhli jej hodnotu na jedno desatinné miesto. Pre prenos aj záporných teplôt sa použije offset  $+40$ . Takýto postup pre získanie teploty je možné využiť napríklad v našich klimatických podmienkach, v ktorých teploty v zime nie sú nižšie ako  $-40\text{ }^{\circ}\text{C}$  a v lete zase nepresahujú  $50\text{ }^{\circ}\text{C}$ . Výsledné číslo reprezentujúce teplotu je potrebné ešte vynásobiť číslom 10, aby sa nemusela prenášať desatinná čiarka. Takto upravená teplota pri odosielaní cez technológiu IQRF posielala výrazne menej dátových bajtov, čím skracuje vysielací čas a tak šetrí batériu. V laboratórnej úlohe gateway nie je naprogramovaný tak, aby si z prijatej hodnoty reverzne získal reálnu teplotu, čo však nie je problém naprogramovať pri reálnom prípade zápisu teploty napríklad do grafu na virtuálnom úložisku cloud.



## Záver

Táto bakalárska práca bola zameraná na vytvorenie laboratórnej úlohy z predmetu Komunikačné systémy pre IoT, ktorá sa venuje technológii IQRF. Pre potreby laboratórnej úlohy bol navrhnutý modul pre spojenie IQRF vysielača a dosky BPC-IoT. Vďaka tomuto modulu BPC-IoT doska dokáže komunikovať s IQRF vysielačom.

Modul bol navrhnutý tak, aby boli dodržané odporúčania výrobcu pre správne napájanie IQRF vysielača a pre jeho bezproblémovú komunikáciu s BPC-IoT doskou. Návrh modulu bol realizovaný v programe Altium Designer. Schéma zapojenia modulu a návrh dosky plošného spoja sú prílohou bakalárskej práce.

Po oboznámení sa s možnosťami IQRF vysielača a BPC-IoT dosky bola pre komunikáciu zvolená zbernica SPI, ktorá je výrazne podporovaná programovými funkciami IQRF zariadení.

Modul bol vytvorený v tvare mini PCIe karty. Toto prevedenie bolo vhodnejšie hlavne z proporčného hľadiska, pretože v práci použitý IQRF vysielač sa pripája do obvodu pomocou pomerne rozmerného SIM konektora.

Pre laboratórnu úlohu bol vybraný IQRF vysielač TR-72-DA, ktorý disponuje integrovanou PCB anténou s dosahom 500 m voľného priestranstva, čo je pre laboratórne využitie maximálne postačujúce. Vysielač je taktiež veľmi jednoducho vyberateľný z obvodu, v ktorom je inštalovaný, vďaka už spomínanému SIM konektoru. Tento spôsob pripojenia uľahčuje zmeny programu a nastavení, ktoré sú počas realizácie laboratórnej úlohy pomerne časté.

Ako koordinátor je použitý USB gateway technológie IQRF GW-USB-06, ktorý je, rovnako ako zvolený vysielač, najvhodnejšou variantou pre laboratórne využitie, pretože je pripojiteľný priamo cez USB konektor k počítaču, v ktorom je zobrazovaná prijatá správa.

Laboratórna úloha bola navrhnutá tak, aby študentov zoznámila s IQRF technológiou a jej možnosťami v sieti mesh. Počas vypracovania úlohy sa študenti oboznámia aj s vývojovým prostredím IQRF IDE, ktoré je využívané pre programovanie a správu IQRF zariadení. Taktiež sa zoznámia s možnosťou poskytovania dát pre vysielač prostredníctvom zbernice SPI, ktorá je jednou z podporovaných zberníc IQRF zariadeniami. Spracované dáta IQRF vysielačom sú následne posielané v jednoduchej sieti na koordinátor, ktorý ich premietne do počítača prostredníctvom IQRF IDE.

Súčasťou bakalárskej práce sú programy pre IQRF zariadenia v jazyku C a taktiež programy pre BPC-IoT dosku s využitím programovacieho jazyka microPython, ktoré sú použité v laboratórnej úlohe.

Laboratórna úloha je preložená aj do anglického jazyka pre potreby výučby zahraničných študentov.



# Literatúra

- [1] Vladimir Sulc, Radek Kuchta, and Radimir Vrba. Iqrf smart house - a case study. pages 103–108, 2010. doi:10.1109/MESH.2010.17.
- [2] IQRF. Iqrf - how it works, Apr. 2020. URL: <https://www.youtube.com/watch?v=CuWGUmSo7QI>.
- [3] Med Bouzidi, Yaser Dalveren, Faouzi Alaya Cheikh, and Mohammad Derawi. Use of the iqrf technology in internet-of-things-based smart cities. *IEEE Access*, PP:1–1, 03 2020. doi:10.1109/ACCESS.2020.2982558.
- [4] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/technology/rf>.
- [5] Isidro Calvo, José Miguel Gil-García, Igor Recio, Asier López, and Jerónimo Quesada. Building iot applications with raspberry pi and low power iqrf communication modules. *Electronics*, 5(3), 2016. URL: <https://www.mdpi.com/2079-9292/5/3/54>, doi:10.3390/electronics5030054.
- [6] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/technology/os>.
- [7] Radovan Hajovsky and Martin Pies. Use of iqrf technology for large monitoring systems. *IFAC-PapersOnLine*, 48(4):486–491, 2015. 13th IFAC and IEEE Conference on Programmable Devices and Embedded Systems. URL: <https://www.sciencedirect.com/science/article/pii/S2405896315008575>, doi:10.1016/j.ifacol.2015.07.082.
- [8] Petra Seflova, Vladimir Sulc, Jiri Pos, and Rostislav Spinar. Iqrf wireless technology utilizing iqmesh protocol. pages 101–104, 2012. doi:10.1109/TSP.2012.6256261.
- [9] Mohammed Bouzidi, Marshed Mohamed, Yaser Dalveren, Arild Moldsvor, Faouzi A. Cheikh, and Mohammad Derawi. Propagation measurements for iqrf network in an urban environment. *Sensors*, 22(18):7012, 2022. Copyright - © 2022 by the authors, language=English, url=<https://www.proquest.com/scholarly-journals/propagation-measurements-iqrf-network-urban/docview/2716584463/se-2>,.
- [10] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/technology/iqmesh>.

- [11] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/technology/iqmesh>.
- [12] Jan Velicka, Martin Pies, and Radovan Hajovsky. Wireless measurement of carbon dioxide by use of iqrf technology. *IFAC-PapersOnLine*, 51(6):78–83, 2018. 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018. URL: <https://www.sciencedirect.com/science/article/pii/S2405896318308772>, doi:10.1016/j.ifacol.2018.07.133.
- [13] Jiri Skovranek, Martin Pies, and Radovan Hajovsky. Use of the iqrf and node-red technology for control and visualization in an iqmesh network. *IFAC-PapersOnLine*, 51(6):295–300, 2018. 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018. URL: <https://www.sciencedirect.com/science/article/pii/S2405896318309157>, doi:10.1016/j.ifacol.2018.07.169.
- [14] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/technology/low-power-modes>.
- [15] Mohammed Bouzidi, Ahmed Amro, Yaser Dalveren, Faouzi A. Cheikh, and Mohammad Derawi. Lpwan cyber security risk analysis: Building a secure iqrf solution. *Sensors*, 23(4):2078, 2023. Copyright - © 2023 by the authors. URL: <https://www.proquest.com/scholarly-journals/lpwan-cyber-security-risk-analysis-building/docview/2779678770/se-2>.
- [16] Er. Pooja Yadav, Er. Ankur Mittal, and Hemant Yadav. Iot: Challenges and issues in indian perspective. pages 1–5, 2018. doi:10.1109/IoT-SIU.2018.8519869.
- [17] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/technology/security>.
- [18] Waheb A. Jabbar, Tan Mei Ting, M. Fikri I. Hamidun, Ajwad H. Che Kamarudin, Wenyan Wu, Jamil Sultan, AbdulRahman A. Alsewari, and Mohammed A.H. Ali. Development of lorawan-based iot system for water quality monitoring in rural areas. *Expert Systems with Applications*, 242:122862, 2024. URL: <https://www.sciencedirect.com/science/article/pii/S095741742303364X>, doi:10.1016/j.eswa.2023.122862.
- [19] Drew Gislason. *Zigbee wireless networking*. Elsevier / Newnes, Amsterdam ; Boston, 2008.

- [20] Seyed Mahdi Darroudi and Carles Gomez. Bluetooth low energy mesh networks: A survey. *Sensors*, 17(7), 2017. URL: <https://www.mdpi.com/1424-8220/17/7/1467>, doi:10.3390/s17071467.
- [21] Eric Geier. Wi-fi halow. *Network World (Online)*, Feb 27 2023. Name - Wi-Fi Alliance;. URL: <https://www.proquest.com/trade-journals/wi-fi-halow-wireless-internet-things/docview/2780434265/se-2>.
- [22] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/product-detail/tr-72d>.
- [23] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/products/transceivers>.
- [24] IQRF Tech s.r.o., Prumyslova 1275, 506 01 Jicin. *TR-72D*, 2022. URL: <https://www.iqrf.org/download/e167ae06-047e-4414-9440-1a7bd166a99e>.
- [25] IQRF Tech s.r.o. Technology for wireless. URL: <https://www.iqrf.org/product-detail/kon-sim-02>.
- [26] MICRORISC s.r.o., Prumyslova 1275, 506 01 Jicin. *KON-SIM-02*, 2015. URL: <https://www.iqrf.org/download/48f2e3ed-99f6-4cf6-a400-ace91a096f1c>.
- [27] IQRF Tech s.r.o., Prumyslova 1275, 506 01 Jicin. *GW-USB-06*, 2023. URL: <https://www.iqrf.org/download/f8207d0b-a4e2-4b62-91e6-0b5caccf5d6b>.
- [28] The M.2 interface — delock.de. [https://www.delock.de/infothek/M.2/M.2\\_e.html](https://www.delock.de/infothek/M.2/M.2_e.html). [Accessed 27-11-2023].
- [29] M.2 interface key and socket explained. URL: <https://www.atpinc.com/blog/what-is-m.2-M-B-BM-key-socket-3>.
- [30] S.K. Dhawan. Introduction to pci express-a new high speed serial data bus. 2:687–691, 2005. doi:10.1109/NSSMIC.2005.1596352.
- [31] Martin Votava. *Převodník mezi sběrnici I2C a RS232*. Masters thesis, Vysoké učení technické v Brně. Fakulta informačních technologií, Brno, June 2010. URL: <https://dspace.vutbr.cz/bitstream/handle/11012/56138/final-thesis.pdf?sequence=-1&isAllowed=y>.
- [32] Rizka Reza Pahlevi, Aji Gautama Putrada S., and Maman Abdurohman. Fast uart and spi protocol for scalable iot platform. pages 239–244, 2018. doi: 10.1109/ICoICT.2018.8528745.

- [33] IQRf Tech s.r.o., Prumyslova 1275, 506 01 Jicin. *IQRf SPI*, 2023. URL: <https://www.iqrf.org/download/f8207d0b-a4e2-4b62-91e6-0b5caccf5d6b>.
- [34] IQRf Tech s.r.o., Prumyslova 1275, 506 01 Jicin. *IQRf HW design*, 2023. URL: <https://www.iqrf.org/download/bf7c2c0d-9d6e-4450-97d9-1aac5a8ba67a>.

# Zoznam symbolov a skratiek

<b>ACK</b>	Acknowledgement
<b>A/D</b>	Analog / Digital
<b>AES</b>	Advanced Encryption Standard
<b>BLE</b>	Bluetooth Low Energy
<b>C</b>	Koordinátor
<b>CLK</b>	Clock
<b>CRC</b>	Cyclic Redundancy Check
<b>CSMA-CA</b>	Carrier Sense Multiple Access Collision Avoidance
<b>DC</b>	Direct Current
<b>DSSS</b>	Direct Sequence Spread Spectrum
<b>FEC</b>	Forward Error Correction
<b>FSK</b>	Frequency Shift Keying
<b>GND</b>	Ground
<b>IC</b>	Integrated Circuit
<b>IDE</b>	Integrated Desktop Environment
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IO</b>	Input Output
<b>IoT</b>	Internet of Things - Internet vecí
<b>IQRF</b>	Intelligent Quick Response Frequency
<b>ISM</b>	Industrial Scientific Medical
<b>LDO</b>	Low Dropout Output
<b>LP</b>	Low Power
<b>MCU</b>	Microcontroller Unit
<b>MISO</b>	Main In Slave Out

<b>MOSI</b>	Main Out Slave In
<b>N</b>	Node
<b>OQPSK</b>	Offset Quadrante Phase shift Keying
<b>OS</b>	Operačný systém
<b>OTA</b>	Ower The Ai
<b>PCB</b>	Printed Circuit Board
<b>PWM</b>	Pulse Width Modulation
<b>RF</b>	Radio Frequency
<b>RFPGM</b>	Radio Frequency Programming
<b>RTCC</b>	Real Time Clock Calendar
<b>RX</b>	Receive
<b>RXD</b>	Receive Data
<b>SCL</b>	Synchronous CLock
<b>SCLK</b>	Slave Clock
<b>SDA</b>	Serial Data
<b>SPI</b>	Serial Pheripheral Interface
<b>SS</b>	Slave Select
<b>STD</b>	Standard
<b>TX</b>	Transmit
<b>TXD</b>	Transceive Data
<b>UART</b>	Universal Asynchronous Reciever-Transmitter
<b>VIN</b>	Voltage IN
<b>VOUT</b>	Voltage OUT
<b>WLAN</b>	Wireless Local Area Network
<b>WMAN</b>	Wireless Metropolitan Area Network

<b>WPAN</b>	Wireless Personal Area Network
<b>WSN</b>	Wireless Sensor Network
<b>WWAN</b>	Wireless Wide Area Network
<b>XLP</b>	Extra Low Power



# Zoznam príloh

<b>A</b>	<b>Laboratórna úloha IQRF mesh siete</b>	<b>77</b>
A.1	Cieľ merania . . . . .	77
A.2	Zadanie . . . . .	77
A.3	Teoretický úvod . . . . .	77
A.4	Postup pre vypracovanie merania . . . . .	79
A.4.1	Point to point komunikácia . . . . .	80
A.4.2	IQRF mesh sieť . . . . .	82
A.4.3	Bonusová úloha . . . . .	84
A.4.4	Použité prvky na pracovisku . . . . .	84
<b>B</b>	<b>Laboratory task IQRF module for measuring of mesh networks</b>	<b>85</b>
B.1	Measurement Objective . . . . .	85
B.2	Assignment . . . . .	85
B.3	Theoretical Introduction . . . . .	85
B.4	Procedure for Measurement . . . . .	87
B.4.1	Point to point Communication . . . . .	88
B.4.2	IQRF Mesh Network . . . . .	90
B.4.3	Bonus Task . . . . .	92
B.4.4	Components used in the workstation . . . . .	92
<b>C</b>	<b>Schéma a návrh dosky modulu</b>	<b>93</b>
<b>D</b>	<b>3D model dosky modulu</b>	<b>97</b>
<b>E</b>	<b>Gerber data</b>	<b>99</b>
<b>F</b>	<b>Zdrojové kódy</b>	<b>103</b>



# A Laboratórna úloha IQRF mesh siete

## A.1 Cieľ merania

Cieľom laboratórnej úlohy je zoznámiť študentov so základnými funkciami a možnosťami IoT technológie IQRF, pochopiť topológiu mesh a princípy komunikácie medzi zariadeniami v nej.

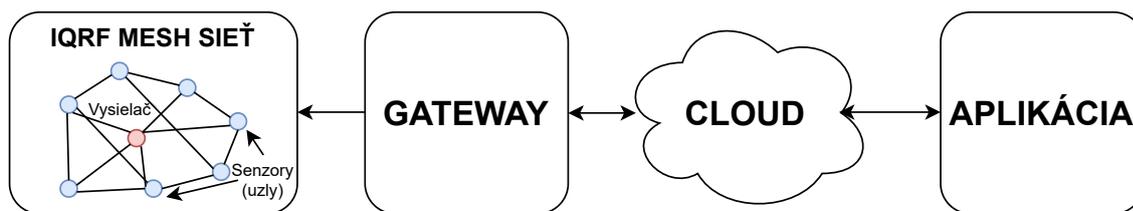
## A.2 Zadanie

- Zoznámte sa so zariadeniami na pracovisku.
- Vytvorte jednoduchú Point to Point komunikáciu medzi koordinátorom a vysielateľom. V tejto jednoduchej sieti naprogramujete dostupný mikrokontrolér Raspberry Pi Pico tak, aby prenášal informácie o teplote z uzla na koordinátor.
- Vytvorte sieť s topológiou mesh obsahujúcu koordinátor a dva uzly a zoznámte sa s vplyvom zmeny polohy uzla na signál a topológiu mesh siete.

## A.3 Teoretický úvod

IQRF (Intelligent Quick Response Frequency) je bezdrôtová technológia a platforma pre IoT orientovaná paketovo. Umožňuje posielat nízke množstvo dát s minimálnou spotrebou elektrickej energie a maximálnou spoľahlivosťou. Technológia je vhodná pre náročné aplikácie, ktoré si vyžadujú čo najnižšiu spotrebu elektrickej energie ako sú batériou napájané inštalácie, aplikácie pre IoT, smart mestá či budovy.

IQRF technológia pracuje v nelicencovanom Sub-GHz ISM (industrial scientific medical) rádiovom pásme (433, 868, 915 MHz) v závislosti od kontinentu alebo štátu inštalácie. V Českej republike je používané 868 MHz pásmo. Moduly IQRF technológie môžu pracovať v rôznych topológiách ako napr. bod – bod, hviezda, ale najčastejšie je používaná v mesh topológii. Na obr.A.1 je znázornený typický dizajn IQRF siete obsahujúci jednotlivé uzly (senzory) v mesh topológii, gateway, cloud a pripojenie na internet (aplikácia pre komunikáciu so svetom).

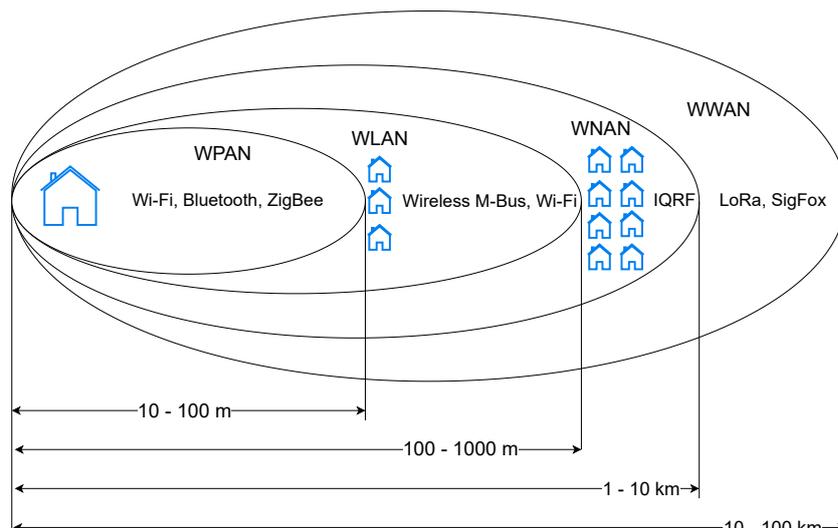


Obr. A.1: Štruktúra IQRF siete

Maximálny dosah jedného IQRF vysielača je certifikovaný na 500 m voľného priestranstva, čo pri použití viac ako 200 zariadení v mesh topológii predstavuje rozlohu až desiatky kilometrov štvorcových. Z toho dôvodu IQRF sieť radíme do kategórie WMAN (Wireless Metropolitan Area Network). Na obr. B.2 je dosah IQRF siete porovnaný s dosahmi iných IoT sietí.

IQMESH (Intelligent Mesh) je patentovaný meshový komunikačný protokol používaný v rámci ekosystému technológie IQRF. Je navrhnutý na sprostredkovanie bezdrôtovej komunikácie v mesh topológii, kde každé zariadenie môže komunikovať s okolitými zariadeniami a potenciálne slúžiť ako preposielací uzol na rozšírenie dosahu siete. Každý paket je smerovaný uzlami na pozadí neobmedzujúc ich funkcionality. Tento proces je riadený operačným systémom a je nebadateľný pre užívateľa. IQMESH podporuje viaceré smerovacie štruktúry ako sú: úplný mesh, redukovaný mesh, optimalizovaný mesh, strom... Smerovanie môže prebiehať staticky, a taktiež môže byť pod úplnou kontrolou užívateľa. Dynamické časovanie s výberom počtu krokov v sieti a voliteľným časovým úsekom umožňuje maximálnu priepustnosť s minimálnym rušením a spotrebou elektrickej energie. Efektivita a spoľahlivosť je dosiahnutá pomocou takzvaného záplavového mechanizmu, ktorý funguje podobne ako povodeň - informácia je na miesto doručenia poslaná viacerými možnými cestami - podobne ako voda pri povodni zaplavuje miesta kadiaľ môže pretiecť. Kolíziám v sieti sa predchádza pomocou systému Time Division Multiple Access, čo v preklade znamená viacnásobný prístup s časovým delením.

Ostatné technické parametre siete sú v tabuľke tab B.1



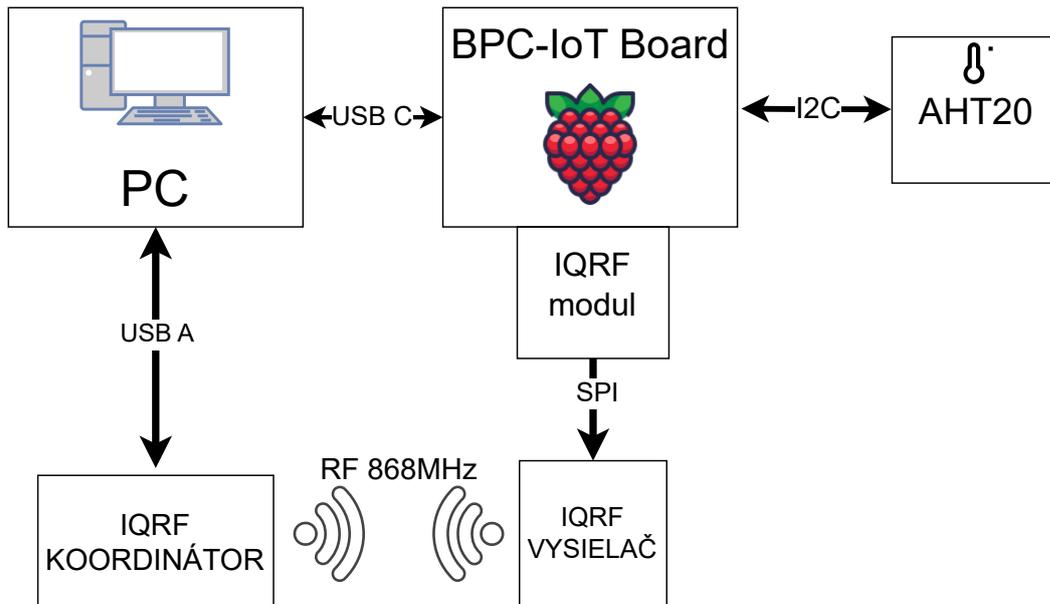
Obr. A.2: Porovnanie dosahu viacerých sietí z IoT odvetvia

IQRF		
Veličina	Jednotka	Hodnota
Pracovná frekvencia	MHz	433, 868, 916
Prenosová rýchlosť	kbps	20
Max. dĺžka prenášanej správy	byte	64
Max. dosah vo voľnom priestranstve	m	1000
Modulácia	-	GFSK
Topológia	-	mesh
Oneskorenie	ms	60 - 100
Max. počet uzlov na gateway	-	65*103
Teoretický čas života na AAA batérii	rok	5 - 10

Tab. A.1: Prehľad parametrov IQRF siete

## A.4 Postup pre vypracovanie merania

Najskôr si skontrolujeme zapojenie pracoviska podľa blokovej schémy B.3 a skontrolujeme polohu jumper prepojok prvá v polohe pre logiku 3,3 V, druhá v mieste povoľujúcom mini PCIe kartu.



Obr. A.3: Bloková schéma zapojenia pracoviska

#### A.4.1 Point to point komunikácia

Pred vytvorením point to point komunikácie je potrebné vytvoriť a nahráť do BPC-IoT dosky program, ktorý je v jazyku microPython. Tento program prijme teplotu z teplotného senzora po zbernici I2C spracuje ju a podľa IQRF SPI guide a vytvorí paket, ktorý je prijatý IQRF vysielačom.

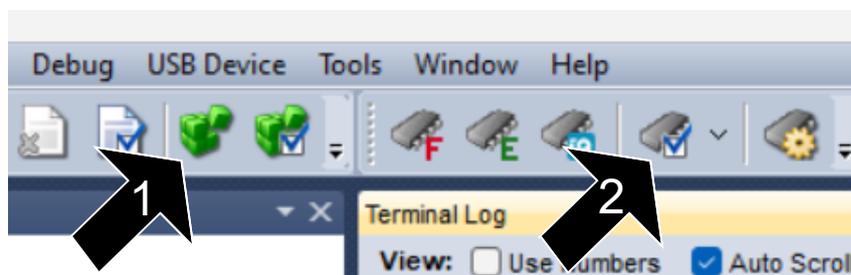
Spustíte program IQRF IDE a vytvorte si nový projekt: **New -> Non-networking Demo -> Next**. Projekt pomenujte a stlačte **Next**. TR Module nastavte na TR-72Dx a IQRF OS ponechajte na najnovšej verzii: **Finish -> Yes to All**. Do novovytvoreného projektu pridajte demo časti kódu - v ľavej časti obrazovky v záložke Project kliknite pravým tlačidlom myši na **Source -> Add -> Existing Item...-> ExternaTeplotaTXIQRF.c -> Open** a prijímaciu časť pravým tlačidlom myši kliknite na **Source -> Add -> New Non-networking Example...** V novootvorenom okne vyberte **E02-example of simple RF receiving**. Vyberte a potvrdte demo kód. -> **Next -> Finish -> Yes -> Yes to All**. V záložke Source by sa mali nachádzať časti kódu (ExternaTeplotaTXIQRF.c a E02-RX.c), ak nie, zopakujte postup.

Keď máme vytvorené demo projekty, musíme si vytvoriť výstupné HEX súbory, ktoré sa dajú nahráť do jednotlivých zariadení. Označte si krúžkom prvý projekt ExternaTeplotaTXIQRF.c a stlačte **zelenú kocku (B.4 - 1)** v hornej časti obrazovky. V záložke Output HEX (záložka pod Source) sa objaví vytvorený Hex File s názvom podľa zdroja (TeplotaTX.hex). Rovnako vytvorte HEX file aj pre druhú

časť E02-RX.c (E02-RX.hex).

Pre správnu funkciu siete (aj Point to Point) je potrebné, aby boli obe zariadenia nastavené na rovnaké frekvenčné pásmo. Preto dvojklikom otvorte záložku **TR configuration** a RF band nastavte na 868 MHz, RF channel A na 67 a RF channel B na 62. Tlačidlom **Close -> Yes -> Save** potvrdte vašu konfiguráciu. Táto konfigurácia sa objaví v záložke TR configuration na ľavej strane obrazovky.

Takto vytvorené výstupné HEX súbory a TR konfiguráciu vieme nahráť do jednotlivých zariadení. Pripojte do počítača USB koordinátor a v ľavom dolnom rohu obrazovky sa zobrazí blikajúci zelený krúžok s pomenovaním pripojeného zariadenia GW-USB-06. Ak to tak nie je, je potrebné skontrolovať, či nie je otvorené IQRF IDE vo viacerých oknách alebo odpojiť a znova zapojiť programované zariadenie. Ak už bliká zelený krúžok, všetko je pripravené na nahratie programu. Do koordinátora nahrávame prijímaciu časť projektu, preto si označte v záložke Output HEX E02-RX.hex a v záložke TR configuration si označte vami vytvorenú konfiguráciu. Následne stlačte **tlačidlo mikrokontroléra s modrou fajkou (B.4 - 2)** v hornej časti obrazovky. V dolnej časti obrazovky sa objaví potvrdzujúca hláška o úspešnom nahratí programu do GW-USB-06. Pre nahratie vysielacej časti programu najskôr zasunúť červený vysieláč do SIM slotu na programátore (modul so sivou podstavou) a pripojte ho pomocou USB kábla do počítača. Následne v dolnej časti obrazovky kliknite pravým tlačidlom myši na zelený blikajúci krúžok a zvolte si druhé zariadenie (CK-USB -04A). V záložke Output HEX vyberte TeplotaTX.hex **tlačidlom mikrokontroléra s modrou fajkou (B.4 - 2)** nahrajte program aj do druhého zariadenia. Opätovne sa objaví potvrdzujúca správa v dolnej časti obrazovky. Naprogramovaný vysieláč vložte naspäť do BPC-IoT board. Tento program prijme teplotu a po zbernici SPI a skontroluje jej správnosť a odošle ju na IQRF koordinátor. Správnosť prijatia správy skontrolujte v časti **Terminal Log**, ktorá sa nachádza v dolnej časti obrazovky.



Obr. A.4: IQRF IDE pre lepšiu navigáciu

## A.4.2 IQRF mesh sieť

Pre vytvorenie novej mesh siete je potrebné sa najskôr uistiť, že použité prvky siete (koordinátor a uzly) nie sú už spárované s inými prvkami. Preto vykonáme reset párovacích nastavení najskôr pre koordinátor potom pre uzly.

Začneme pridaním Plug-inov do projektu, ktoré bude potrebné nahráť do koordinátora a do uzlov. Pravým tlačidlom myši klikneme na záložku **Project** -> **Plug-ins** -> **Add** -> **DPA Plug-in...** v novootvorenom okne zaškrtneme DPA-Coordinator-SPI-7xD... a DPA-Node-STD-7xD... a potvrdíme výber tlačidlom **OK**. Pri nahrávaní do jednotlivých zariadení je potrebné, aby boli zaškrtnuté iba záložky Plug-ins s konkrétnou časťou pre koordinátor alebo pre uzol a v záložke TR Configuration konfigurácia z predošlého cvičenia. Rovnakým spôsobom ako v predošlej časti nahráme Plug-in aj konfiguráciu do koordinátora a do uzlov. V tomto prípade už nie je potrebné vytvárať HEX súbory.

Ak máme úspešne naprogramovaný koordinátor aj uzol, môžeme začať s resetovaním predošlých informácií o párovaní. Pre resetovanie koordinátora je potrebné najskôr koordinátor pripojiť k počítaču. Následne zvolíme záložku **IQMESH Network Manager** v dolnej časti obrazovky a v nej zase vyberieme záložku **Control** a v časti Bonding zvolíme **Clear All Bonds**.

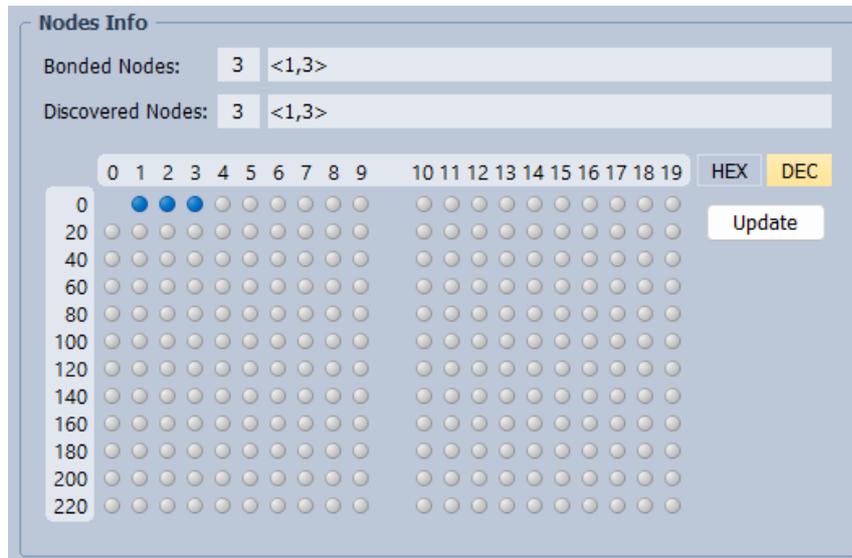
Ako ďalšie musíme resetovať uzly (vysielače). Uzly budeme resetovať pomocou kombinácie tlačidiel na doske. Stlačíme tlačidlo SW1 a držíme stlačené, počas toho krátko stlačíme tlačidlo SW2. Začne svietiť zelená LEDka. Okamžite po jej zhasnutí pustíme tlačidlo SW1. Po tomto úkone by mal vysielateľ začať blikáť na červeno približne na 10 sekúnd, potom sa uspí.

Pre nabondovanie vysieláčov ku koordinátoru sa vrátíme do IQRF IDE a na záložke Control v časti Bonding označíme Auto address a stlačíme **Bond Node**. Po stlačení sa objaví okno s načítavaním, v tej chvíli máme 10 sekúnd na to, aby sme stlačili SW2 na doske s vysielateľom. Po úspešnom nabondovaní uzla ku koordinátoru okno s načítavaním zmizne a v tabuľke uzlov sa objaví oranžová bodka. Tento proces resetovania a nabondovania zopakujeme pre všetky uzly. Ak máme nabondovaných už viacero uzlov, prepne sa do záložky **Map View** a v hornej časti obnovíme zobrazenie. Ak je všetko správne, môžeme vidieť schému koordinátora a ku nemu nabondovaných uzlov.

Táto schéma síce obsahuje koordinátor a uzly avšak neobsahuje zatiaľ cesty a spojenia medzi jednotlivými bodmi siete, preto je potrebné spustiť proces s názvom discovery, ktorý priradí jednotlivé uzly do zón a prideli im virtuálne smerovacie čísla. Pre simuláciu reálneho stavu umiestnime niektoré uzly do vzdialenejšej polohy alebo môžeme použiť tieniaci obal. Na záložke Control v sekcii bonding znížime TX power na 1, taktiež z dôvodu zhoršenia parametrov siete. Teraz spustíme vyhľadávací

algoritmus tlačidlom **Discovery**. Následne sa v karte Map View po aktualizovaní zobrazia aj cesty medzi uzlami a vytvoria sa takzvané zóny, ktoré znázorňujú logické usporiadanie siete z pohľadu koordinátora. Správnosť vykonania akcie discovery môžeme taktiež overiť tak, že v záložke Control zmodrejú bodky znázorňujúce jednotlivé uzly vid. B.5

V záložke **Table View** sa teraz nachádza tabuľka s jednotlivými bodmi našej siete, s jej parametrami ako je napríklad RSSI (Receive Signal Strength Indication) udávajúci silu prijatého signálu. Pre názornú ukážku funkčnosti mesh siete teraz umiestnime vysieláče do rôznej vzdialenosti a znova vykonáme proces discovery. V karte Map View je teraz dostupná nová topológia IQRF mesh siete. Funkčnosť komunikácie medzi vysieláčom a uzlami overíme tak, že zašleme na jednotlivé vysieláče správy, ktoré vykonajú nami zvolenú akciu (vypni-zapni LED). Pravým tlačidlom klikneme na riadok s daným vysieláčom a v menu zvolíme **LED on**. Táto správa sa prenesie sieťou a na určenom vysieláči rozsvieti LED.



Obr. A.5: Sekcia Nodes info

### A.4.3 Bonusová úloha

Upravte program `ExternaTeplotaTXRPPico.py` tak, aby bolo možné posielat teploty pod nulou s presnosťou na jedno desatinné miesto. Zároveň teplotu upravte tak, aby sa ušetrilo čo najviac prenášaných dát.

Nápoveda: pre odosielanie teploty použite 3 bajty a offset teploty.

Takto upravená správa je v technológiách IoT častejšie využívaná, pretože šetrí počet vysielaných bajtov, čo má za následok skrátenie času potrebného pre odoslanie. Ak vysielateľ vysielá kratšie, tak dochádza ku výraznej úspore spotrebovanej energie, čo je veľmi dôležitý faktor pre zariadenia napájané z batérie.

### A.4.4 Použité prvky na pracovisku

- PC s IQRF ID a programom Thonny
- BPC - IoT - doska
- Teplotné čidlo AHT20
- Koordinátor GW-USB-06
- Vysielač IQRF TR-72DA 3ks
- Programovacia doska CK-USB-04a
- Aplikačná doska DK-EVAL-04a 2ks
- Prepojovacie káble USB A do USB C a USB A do mikroUSB

# **B Laboratory task IQRF module for measuring of mesh networks**

## **B.1 Measurement Objective**

The objective of the laboratory task is to familiarize students with the basic functions and possibilities of IQRF IoT technology, understand the mesh topology and the principles of communication between devices within it.

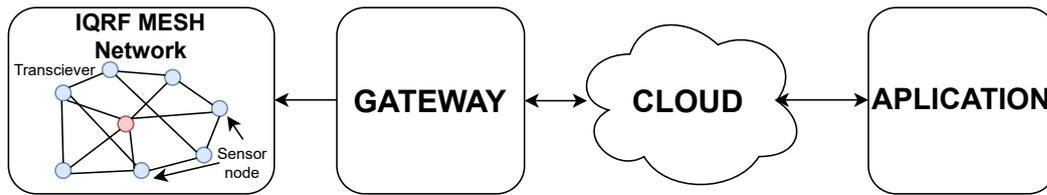
## **B.2 Assignment**

- Familiarize yourself with the devices in the workplace.
- Create a simple Point to Point communication between the coordinator and the transmitter. In this simple network, program the available Raspberry Pi Pico microcontroller to transmit temperature information from the node to the coordinator.
- Create a network with a mesh topology containing a coordinator and two nodes, and familiarize yourself with the impact of changing the position of a node on the transmitting signal and the mesh network topology.

## **B.3 Theoretical Introduction**

IQRF (Intelligent Quick Response Frequency) is a wireless technology and platform for IoT that is packet-oriented. It allows the transmission of small amounts of data with minimal power consumption and maximum reliability. The technology is suitable for demanding applications that require the lowest possible power consumption, such as battery-powered installations, IoT applications, smart cities and buildings.

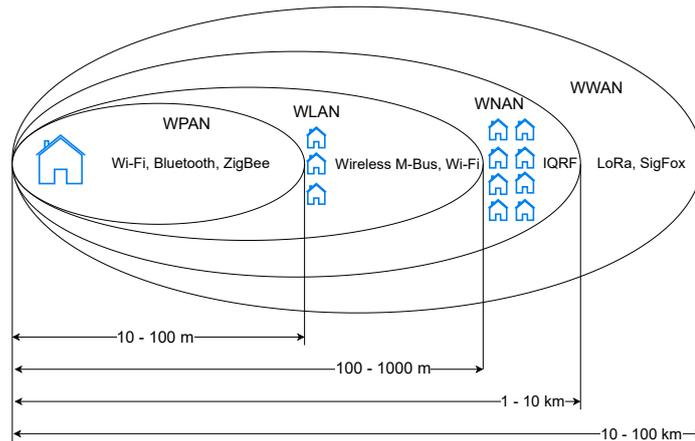
IQRF technology operates in the unlicensed Sub-GHz ISM (industrial scientific medical) radio bands (433, 868, 915 MHz), depending on the continent or country of installation. In the Czech Republic, the 868 MHz band is used. IQRF technology modules can operate in various topologies, such as point-to-point, star, but most commonly, it is used in mesh topology. Figure B.1 shows a typical design of an IQRF network containing individual nodes (sensors) in a mesh topology, a gateway, cloud, and an internet connection (application for communication with the outside world).



Obr. B.1: Structure of IQRF network

The maximum range of a single IQRF transmitter is certified at 500 m in open space, which when using more than 200 devices in a mesh topology, can cover an area of up to tens of square kilometers. For this reason, the IQRF network is classified as a WMAN (Wireless Metropolitan Area Network). Figure B.2 compares the range of an IQRF network with the ranges of other IoT networks.

IQMESH (Intelligent Mesh) is a patented mesh communication protocol used within the IQRF technology ecosystem. It is designed to facilitate wireless communication in a mesh topology, where each device can communicate with surrounding devices and potentially serve as a relay node to extend the network's range. Each packet is routed through the nodes without restricting their functionality. This process is managed by the operating system and is transparent to the user. IQMESH supports various routing structures such as full mesh, reduced mesh, optimized mesh, tree, etc. Routing can be static and can also be under the complete control of the user. Dynamic timing with a selection of the number of hops in the network and an optional time slot allows for maximum throughput with minimal interference and power consumption. Efficiency and reliability are achieved through a so-called flooding mechanism, which works similarly to a flood - the information is sent to the destination through multiple possible paths, similar to how water in a flood inundates areas it can flow through. Collisions in the network are prevented using a Time Division Multiple Access system, which means multiple access with time division.



Obr. B.2: Comparison of the range of various networks in the IoT sector

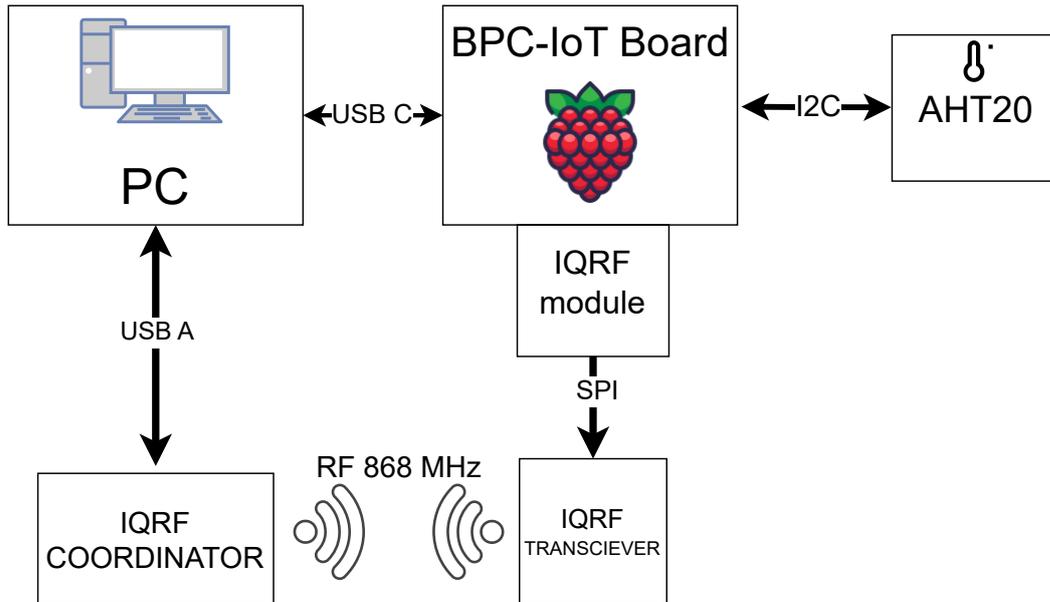
Other technical details of IQRF network are in table Tab. B.1

<b>IQRF</b>		
Parameter	Unit	Value
Operating frequency	MHz	433, 868, 916
Data rate	kbps	20
Max. message length	byte	64
Max. range in open space	m	1000
Modulation	-	GFSK
Topology	-	mesh
Latency	ms	60 - 100
Max. number of nodes per gateway	-	65*103
Theoretical lifetime on AAA battery	year	5 - 10

Tab. B.1: Summary of IQRF network parameters

## B.4 Procedure for Measurement

First, check the setup of the workstation according to the block diagram in Figure B.3 and verify the position of the jumpers. The first should be set for 3.3 V logic, and the second should enable the mini PCIe card.



Obr. B.3: Block diagram of the workstation setup

## B.4.1 Point to point Communication

Before creating point-to-point communication, it is necessary to create and upload a program in microPython language to the BPC-IoT board. This program will receive the temperature from the temperature sensor via the I2C bus, process it, and according to the IQRF SPI guide, create a packet that is received by the IQRF transmitter.

Launch the IQRF IDE program and create a new project: *New -> Non-networking Demo -> Next*. Name the project and click *Next*. Set the TR Module to TR-72Dx and keep the IQRF OS on the latest version: *Finish -> Yes to All*. Add demo code parts to the newly created project - right-click on the Project tab on the left side of the screen and select *Source -> Add -> Existing Item...-> ExternaTeplotaTXIQRF.c -> Open* for the transmission part, and for the reception part, right-click *Source -> Add -> New Non-networking Example...*. In the newly opened window, select *E02-example of simple RF receiving*. Choose and confirm the demo code. *-> Next -> Finish -> Yes -> Yes to All*. The Source tab should now contain the code parts (ExternaTeplotaTXIQRF.c and E02-RX.c); if not, repeat the procedure.

Once the demo projects are created, we need to generate output HEX files that can be uploaded to the individual devices. Highlight the first project ExternaTeplotaTXIQRF.c with a circle and press the *green cube (B.4 - 1)* at the top of the screen. The Output HEX tab (below Source) will show the created Hex File named

after the source (TeplotaTX.hex). Similarly, create a HEX file for the second part E02-RX.c (E02-RX.hex).

For proper network function (even Point to Point), both devices must be set to the same frequency band. Therefore, double-click the **TR configuration** tab and set the RF band to 868 MHz, RF channel A to 67, and RF channel B to 62. Confirm your configuration with the **Close -> Yes -> Save** button. This configuration will appear in the TR configuration tab on the left side of the screen.

The created output HEX files and TR configuration can be uploaded to the individual devices. Connect the USB coordinator to the computer, and a flashing green circle labeled GW-USB-06 will appear in the lower left corner of the screen. If not, check if IQRF IDE is open in multiple windows or reconnect the programmed device. Once the green circle is flashing, everything is ready to upload the program. Upload the receiving part of the project to the coordinator by selecting E02-RX.hex in the Output HEX tab and your created configuration in the TR configuration tab. Then press the **microcontroller button with a blue checkmark (B.4 - 2)** at the top of the screen. A confirmation message about the successful program upload to GW-USB-06 will appear at the bottom of the screen

To upload the transmission part of the program, first insert the red transmitter into the SIM slot on the programmer (module with gray base) and connect it to the computer using a USB cable. Then, in the lower part of the screen, right-click on the flashing green circle and select the second device (CK-USB -04A). In the Output HEX tab, select TeplotaTX.hex and upload the program to the second device using the **microcontroller button with a blue checkmark (B.4 - 2)**. Another confirmation message will appear at the bottom of the screen. Insert the programmed transmitter back into the BPC-IoT board. This program receives the temperature via the SPI bus, checks its correctness, and sends it to the IQRF coordinator. Verify the correctness of the received message in the **Terminal Log** section at the bottom of the screen.



Obr. B.4: IQRF IDE for better navigation

## B.4.2 IQRF Mesh Network

To create a new mesh network, you first need to ensure that the network elements (coordinator and nodes) are not already paired with other elements. Therefore, reset the pairing settings for the coordinator first, then for the nodes.

We start by adding Plugins to the project, which will need to be uploaded to the coordinator and the nodes. Right-click on the **Project -> Plug-ins -> Add -> DPA Plug-in...** tab, in the newly opened window, check DPA-Coordinator-SPI-7xD... and DPA-Node-STD-7xD..., and confirm the selection by clicking **OK**. When uploading to individual devices, ensure that only the Plugin tabs specific to the coordinator or node are checked, and in the TR Configuration tab, use the configuration from the previous exercise. In the same way as in the previous part, upload both the Plugin and the configuration to the coordinator and the nodes. In this case, it is not necessary to create HEX files.

Once the coordinator and node are successfully programmed, you can begin resetting previous pairing information. To reset the coordinator, connect it to the computer first. Then select the **IQMESH Network Manager** tab at the bottom of the screen, select the **Control** tab in it, and in the Bonding section, choose **Clear All Bonds**.

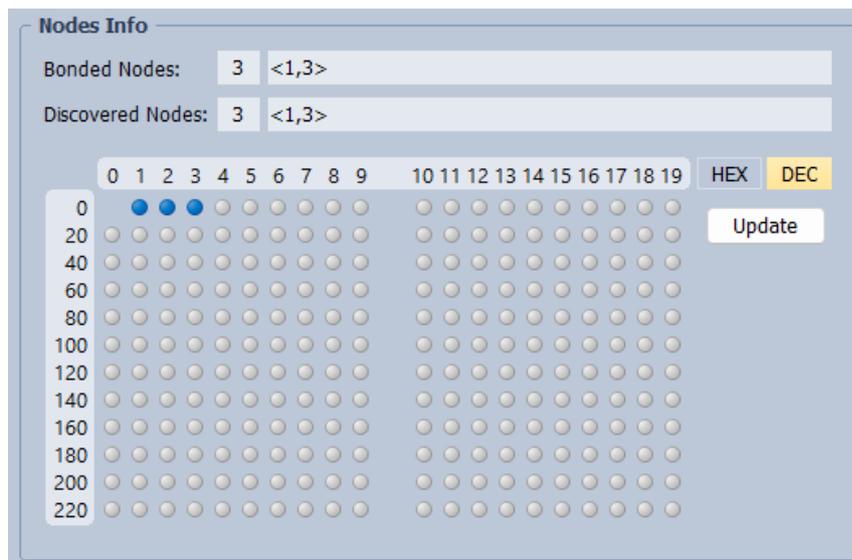
Next, we need to reset the nodes (transmitters). The nodes will be reset using a combination of buttons on the board. Press and hold the SW1 button, then briefly press the SW2 button while still holding SW1. The green LED will light up. Release SW1 immediately after the green LED turns off. After this operation, the transmitter should start flashing red for about 10 seconds before going to sleep.

To bond the transmitters to the coordinator, return to IQRF IDE and, in the Control tab, in the Bonding section, check Auto address and press **Bond Node**. After pressing, a loading window will appear; at this moment, you have 10 seconds to press SW2 on the transmitter board. Upon successful bonding of the node to the coordinator, the loading window will disappear, and an orange dot will appear in the node table. Repeat this process of resetting and bonding for all nodes. If multiple nodes are bonded, switch to the **Map View** tab and refresh the view at the top. If everything is correct, you will see a diagram of the coordinator and the bonded nodes.

This diagram contains the coordinator and nodes but does not yet show the paths and connections between the network points, so it is necessary to run a process called discovery. This process assigns the nodes to zones and gives them virtual routing numbers. To simulate a real situation, place some nodes in distant locations or use a shielding cover. In the Control tab, under bonding, reduce the TX power to 1 to worsen the network parameters. Now run the discovery algorithm by clicking **Dis-**

*covery*. Subsequently, in the Map View tab, after updating, paths between nodes will be displayed, creating so-called zones that represent the logical arrangement of the network from the coordinator's perspective. The success of the discovery action can also be verified by the blue dots representing the individual nodes turning blue in the Control tab, see B.5.

In the **Table View** tab, a table now contains the individual points of our network with their parameters, such as RSSI (Receive Signal Strength Indication), indicating the strength of the received signal. For a practical demonstration of the mesh network's functionality, place the transmitters at various distances and perform the discovery process again. In the Map View tab, the new topology of the IQRF mesh network is now available. The functionality of communication between the transmitter and nodes can be verified by sending messages to the transmitters to perform a chosen action (turn on/off LED). Right-click on the row with the given transmitter and select **LED on** from the menu. This message will be transmitted through the network, and the LED on the specified transmitter will light up.



Obr. B.5: Section Nodes info

### **B.4.3 Bonus Task**

Modify the program `ExternaTeplotaTXRPPico.py` so that it can send temperatures below zero with an accuracy of one decimal place. Additionally, adjust the temperature transmission to save as much data as possible.

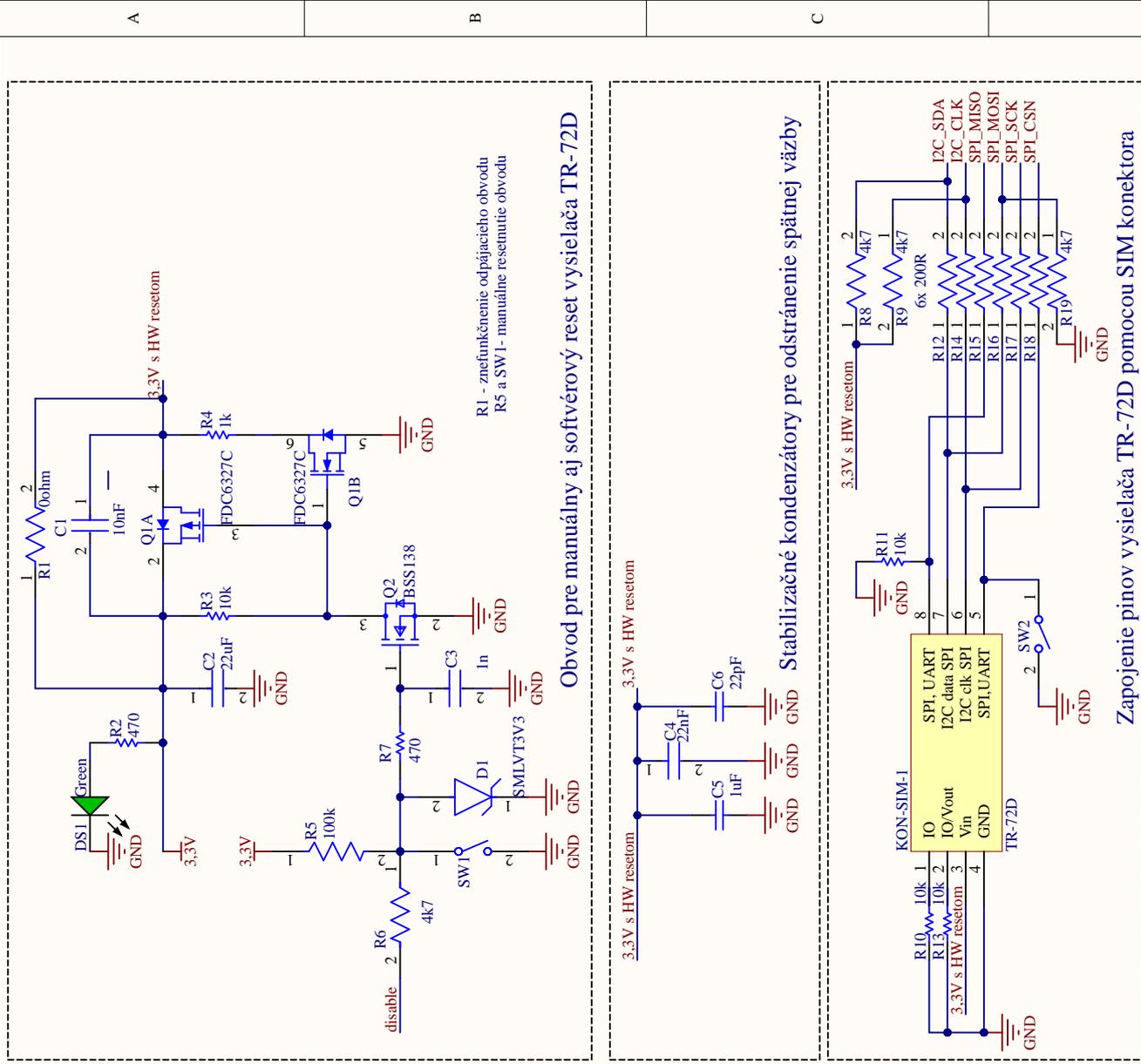
Hint: use 3 bytes and an offset for temperature transmission.

The modified message is more commonly used in IoT technologies because it saves the number of transmitted bytes, resulting in shorter transmission times. When the transmitter sends for a shorter duration, it leads to significant energy savings, which is a crucial factor for battery-powered devices.

### **B.4.4 Components used in the workstation**

- PC with IQRF IDE and Thonny program
- BPC-IoT board
- Temperature sensor AHT20
- Coordinator GW-USB-06
- IQRF TR-72DA transmitter (3 pieces)
- Programming board CK-USB-04a
- Application board DK-EVAL-04a (2 pieces)
- USB A to USB C cables and USB A to microUSB cables

## **C Schéma a návrh dosky modulu**



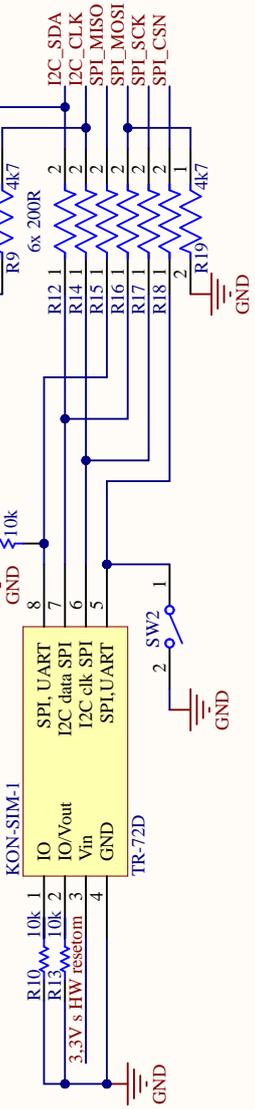
R1 - znefunkčnenie odpájacieho obvodu  
R5 a SW1 - manuálne resetnutie obvodu

**Obvod pre manuálny aj softvérový reset vysielacza TR-72D**

3.3V s HW resetom

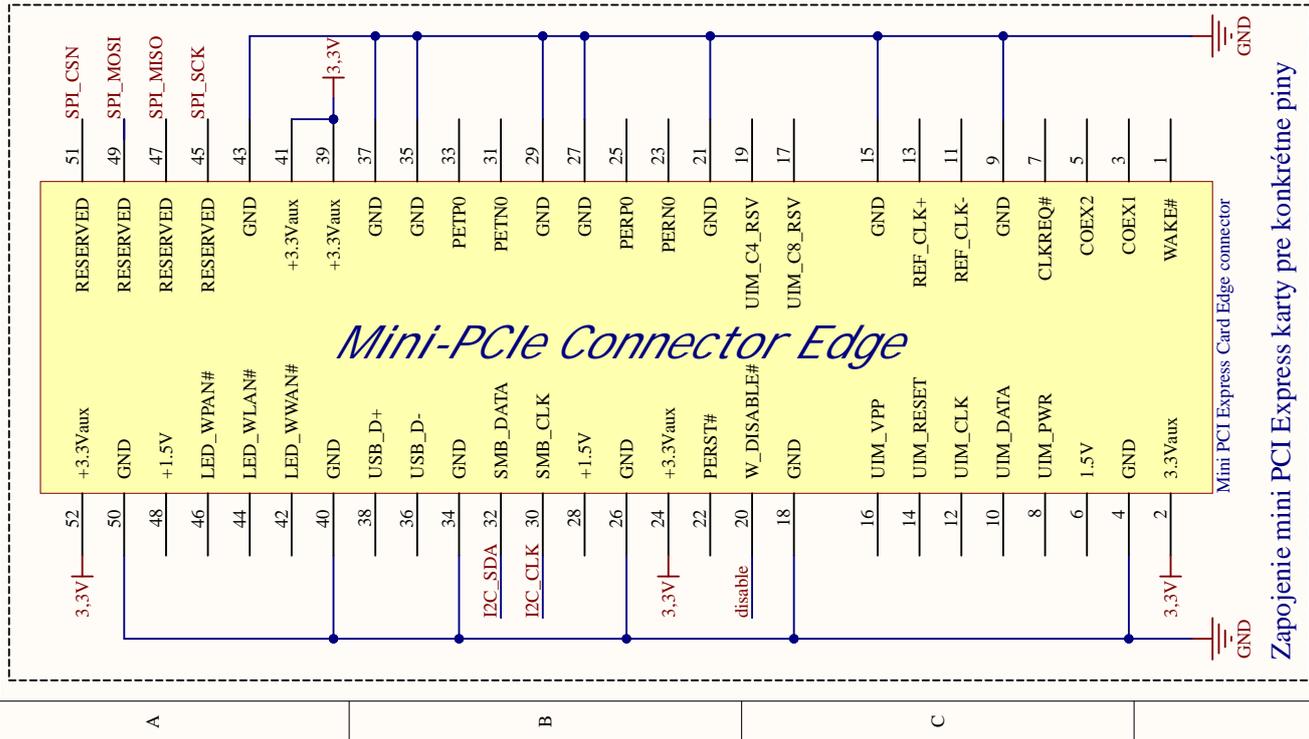
**Stabilizačné kondenzátory pre odstránenie spätnéj väzby**

3.3V s HW resetom



**Zapojenie pinov vysielacza TR-72D pomocou SIM konektora**

Title		Revision	
Size	Number	Revision	
A	1.		
Date:	11/24/2023	Sheet of 1/1	
File:	C:\Users\...MiniPCIe_SchDoc	Drawn By: Juraj Goryl	
3		4	

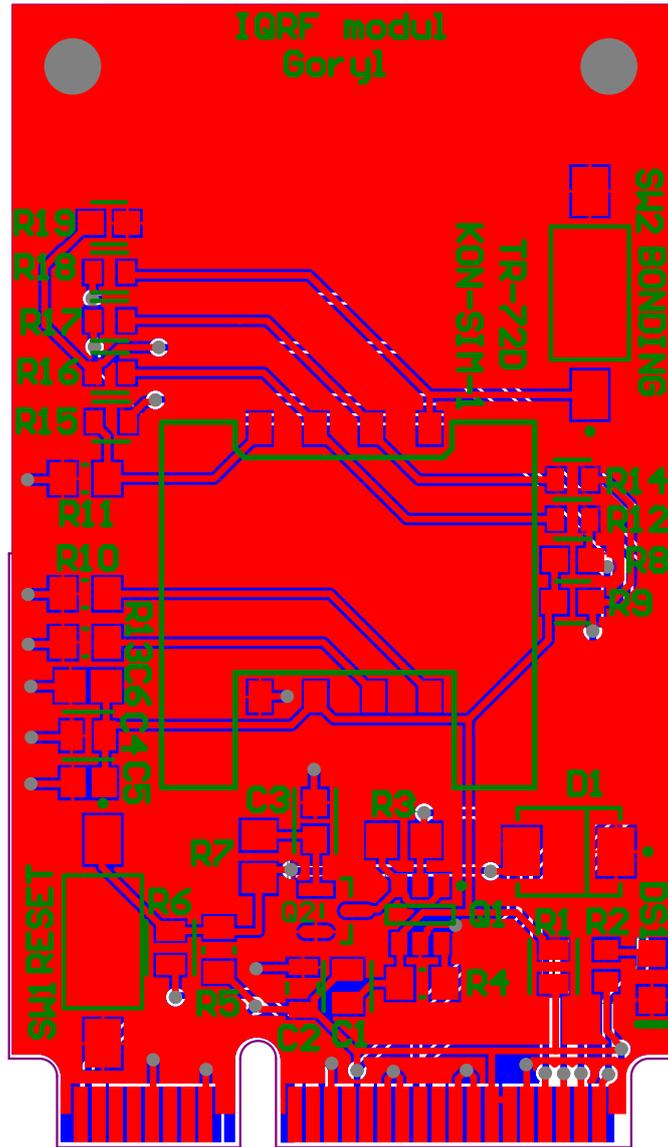


*Mini-PCIe Connector Edge*

**Zapojenie mini PCI Express karty pre konkrétne piny**

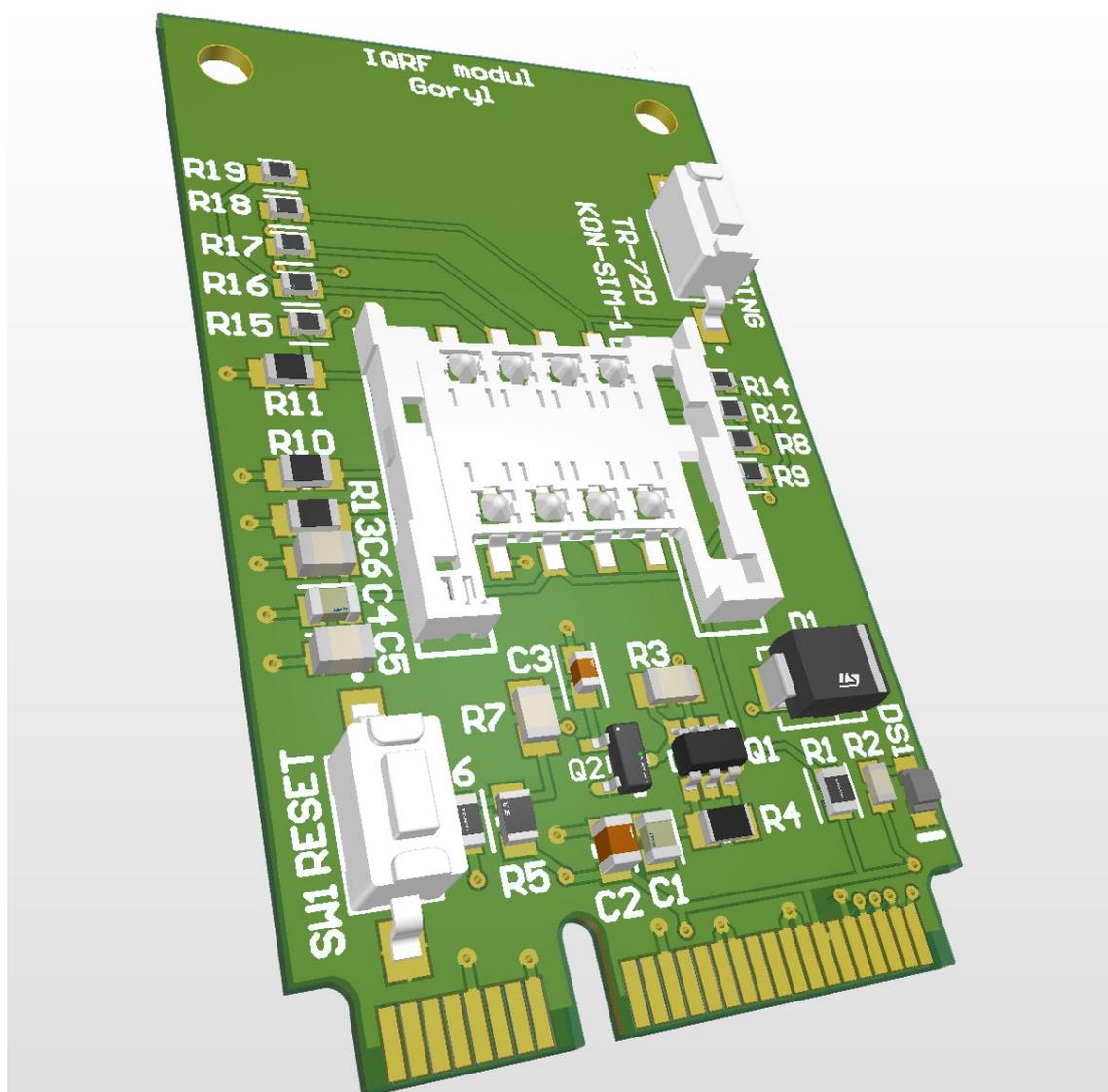
Mini PCI Express Card Edge connector

CN1





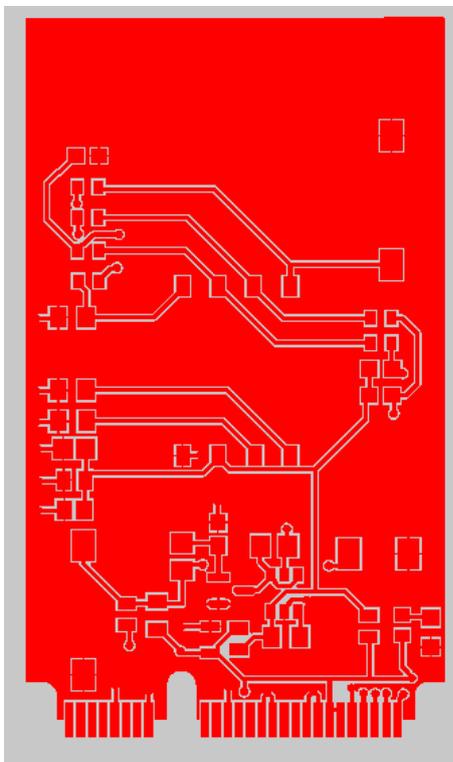
## D 3D model dosky modulu



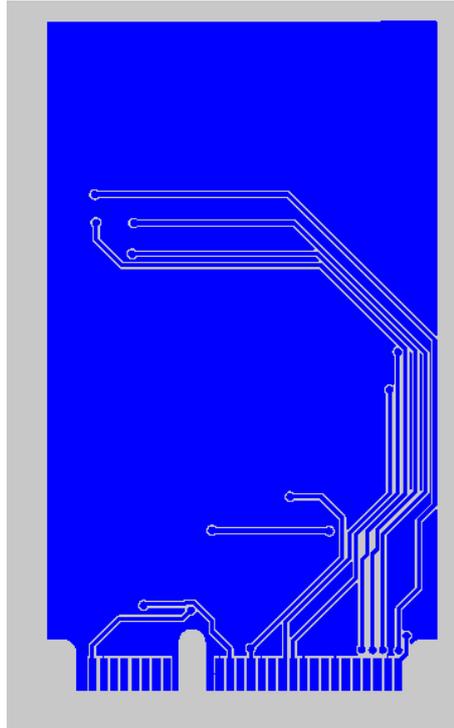


## E Gerber data

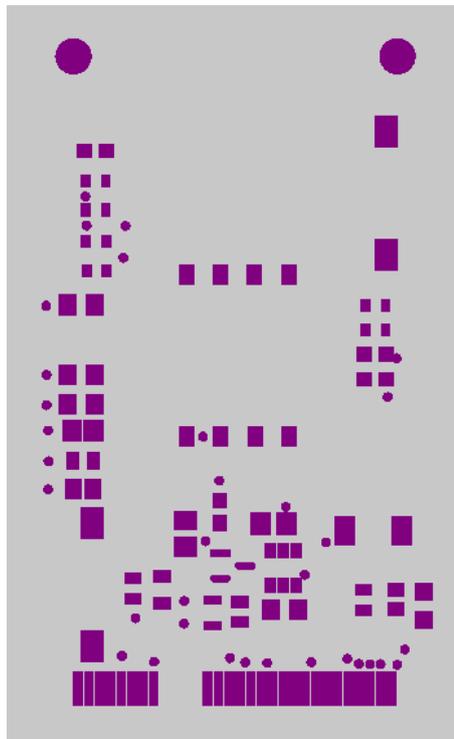
Jednotlivé gerber súbory pre výrobu modulu, ktoré sú potrebné pri zadávaní objednávky u výrobcu.



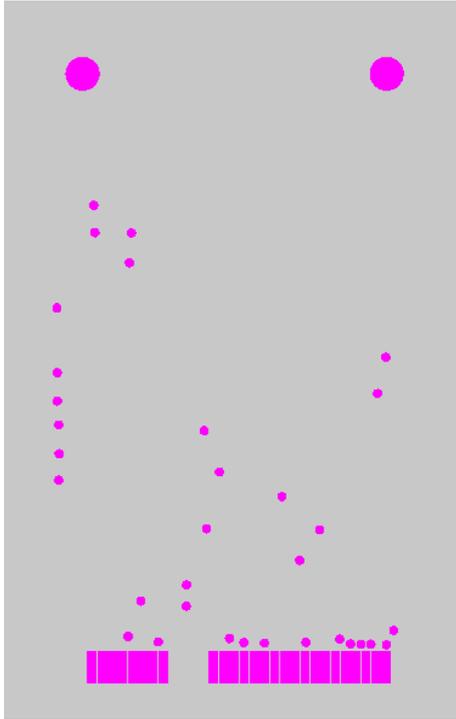
Obr. E.1: Top layer



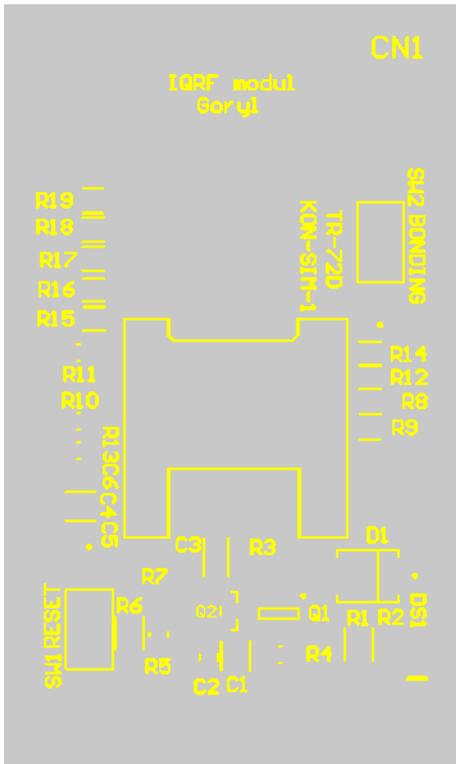
Obr. E.2: Bottom layer



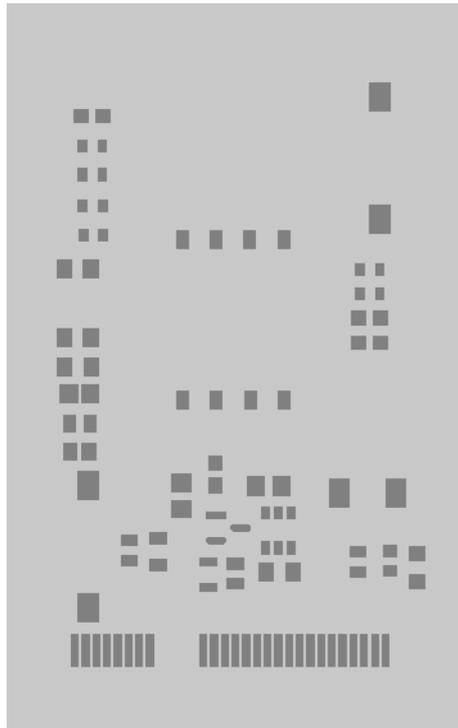
Obr. E.3: Top Solder



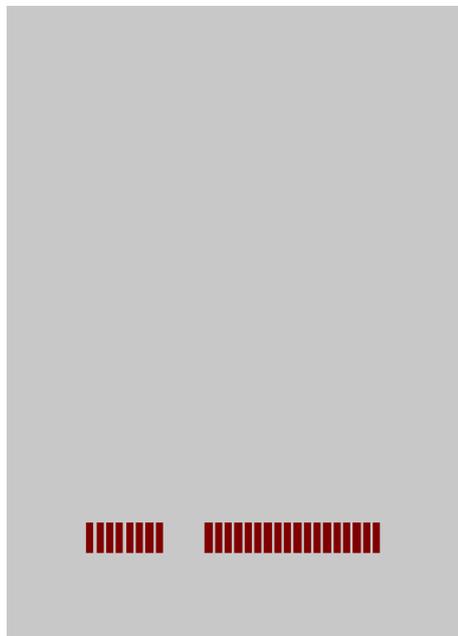
Obr. E.4: Bottom Solder



Obr. E.5: Top Overlay



Obr. E.6: Top Paste



Obr. E.7: Bottom Paste

## F Zdrojové kódy

```
1 import time
2 import struct
3 from machine import SPI, Pin, I2C
4 from ahtx0 import AHT20
5
6 # Initialization of SPI bus
7 sck_pin = Pin(10, Pin.OUT)
8 mosi_pin = Pin(11, Pin.OUT)
9 miso_pin = Pin(12, Pin.IN)
10 cs_pin = Pin(13, Pin.OUT)
11
12 spi = SPI(1, baudrate=20000, polarity=0, phase=0, sck=sck_pin, mosi
    =mosi_pin, miso=miso_pin)
13 cs_pin.value(1)
14
15 # Initialization of I2C bus
16 i2c1 = I2C(1, scl=Pin(15), sda=Pin(14), freq=100000)
17
18 # Initialization of temperature sensor
19 tmp_sensor = AHT20(i2c1)
20 tmp_sensor.initialize()
21
22 # Function to read temperature from sensor
23 def ReadTemp():
24     time.sleep(0.5)
25     temp = tmp_sensor.temperature
26     print(str(round(temp, 2)))
27     time.sleep(1)
28     return "{:.2f}".format(round(temp, 2))
29
30 # Function to calculate CRC
31 def calculate_crc(hex_string):
32     xor_result = 0xF0 ^ 0x85 ^ 0x5F # Initialize XOR
33     for i in range(0, len(hex_string), 2):
34         xor_result ^= int(hex_string[i:i+2], 16)
35     return xor_result
36
37 # Function to create SPI packet and send it
38 def SendTemp(temp):
39     hex_string = ''.join('{:02x}'.format(ord(c)) for c in temp)
40     crc_value = calculate_crc(hex_string)
41     message = f'f085{hex_string}{crc_value:02x}00'
42     bytearray = bytes.fromhex(message)
```

```

43     formatted_string = ''.join('\\x{:02x}'.format(b) for b in
        byte_array)
44     print(f"Formatted SPI Message: {formatted_string}")
45     spi.write(byte_array)
46
47 while True:
48     temp = ReadTemp()
49     cs_pin.off() # Start of SPI communication
50     time.sleep_ms(10)
51
52     SendTemp(temp)
53     time.sleep_ms(10)
54
55     print(spi.read(20)) # Reading of slave answer
56     time.sleep_ms(10)
57
58     cs_pin.on() # End of SPI communication

```

Výpis F.1: Kód pre BPC-IoT dosku odosielajúcu teplotu cez SPI zbernicu na IQRF vysielač (ExternaTeplotaTXRPPico.py)

```

1  #include "IQRf.h"
2  void _sleepWOC(void);
3  void APPLICATION()
4  {
5      enableSPI();
6      clrwdt();
7      //infinite loop of receiving message via SPI
8      while (1)
9      {
10         clrwdt();
11         sleepWOC();
12         //LED flashing if SPI bus is busy
13         if (getStatusSPI())
14         {
15             pulseLEDR();
16         }
17         else
18         {
19             if(_SPIRX)    //checking if there are available data
20             {
21                 if(_SPICRCok)    //checking if data aren't damaged
22                 {
23                     //data 5 bytes are in bufferCOM
24                     copyBufferCOM2RF();
25                     DLEN = SPIpacketLength;
26                     RFTXpacket();    //send message
27                     waitDelay(25);
28                     startSPI(0);    //SPI released for master
29                 }
30                 else
31                 {
32                     //damaged data
33                     bufferCOM[0] = 'C';
34                     bufferCOM[1] = 'R';
35                     bufferCOM[2] = 'C';
36                     bufferCOM[3] = ' ';
37                     bufferCOM[4] = 'e';
38                     bufferCOM[5] = 'r';
39                     bufferCOM[6] = 'r';
40                     copyBufferCOM2RF();
41                     DLEN = 7;
42                     RFTXpacket();    //send message
43                     waitDelay(25);
44                     startSPI(0);    //SPI released for master
45                     waitDelay(250);
46                 }

```

```

47     }
48     else
49     { //no data message
50         bufferCOM[0] = 'n';
51         bufferCOM[1] = 'o';
52         bufferCOM[2] = ' ';
53         bufferCOM[3] = 'd';
54         bufferCOM[4] = 'a';
55         bufferCOM[5] = 't';
56         bufferCOM[6] = 'a';
57         copyBufferCOM2RF();
58         DLEN = 7;
59         RFTXpacket(); //send message
60         waitDelay(25);
61         startSPI(0); //SPI released for master again
62         waitDelay(250);
63     }
64 }
65 }
66 }

```

Výpis F.2: Kod pre vysieláč prijímajúci dáta cez SPI zbernicu a posielajúci ich koordinátoru (ExternaTeplotaTXIQRf.c)

```

1 import time
2 from machine import SPI
3
4 BUT0 = machine.Pin(6, machine.Pin.IN)
5 sck_pin = machine.Pin(10, machine.Pin.OUT)
6 mosi_pin = machine.Pin(11, machine.Pin.OUT)
7 miso_pin = machine.Pin(12, machine.Pin.IN)
8 cs_pin = machine.Pin(13, machine.Pin.OUT)
9
10 spi = SPI(1, baudrate=200000, polarity=0, phase=0, sck=sck_pin,
11         mosi=mosi_pin, miso=miso_pin)
12
13 while True:
14     BUT0_state = BUT0.value()
15     BUT0_pressed = not BUT0_state
16     print(f"Button Pin State: {BUT0_state} Button Pressed: {
17           BUT0_pressed}")
18
19     if BUT0_pressed:
20         cs_pin.off()
21         time.sleep(0.05)
22         cs_pin.on()

```

Výpis F.3: Kód pre BPC-IoT dosku, ktorý odošle správu o teplote z interného čidla na IQRf vysieláči stlačením tlačidla BUT0 (TeplotaTXRPPico.py)

```

1 #include "IQRf.h"
2 void _sleepWOC(void);
3 void APPLICATION()
4 {
5     uns16 temperature, tmp;
6     uns8 i, tenths;
7
8     waitMS(100);
9     enableSPI();
10
11     while (1)
12     {
13         if (getTemperature() != -128)
14         { //sending of temperature
15             temperature = param3;
16             bufferCOM[0] = 'T';
17             bufferCOM[1] = '=';
18             bufferCOM[2] = ' ';
19             bufferCOM[5] = '.';
20             bufferCOM[7] = 0xB0;
21             bufferCOM[8] = 'C';
22
23             if (temperature.high8 & 0x08)
24             {
25                 temperature = ~temperature;
26                 temperature++;
27                 bufferCOM[2] = '-';
28             }
29             // making decimal part
30             tenths = temperature.low8 & 0x0F;
31             temperature >>= 4;
32             i = temperature.low8 / 10;
33             bufferCOM[3] = i + '0';
34             i = temperature.low8 % 10;
35             bufferCOM[4] = i + '0';
36             tmp = tenths * 625;
37             i = tmp / 1000;
38             tmp %= 1000;
39             if ((tmp >= 500) && (i < 9))
40                 i++;
41             bufferCOM[6] = i + '0';
42             startSPI(9);
43         }
44         else
45         { //error message
46             bufferCOM[0] = 'T';

```

```

47         bufferCOM[1] = '-';
48         bufferCOM[2] = 'e';
49         bufferCOM[3] = 'r';
50         bufferCOM[4] = 'r';
51         startSPI(5);
52     }
53
54     sleepWOC();
55     startSPI(1);
56
57     if (buttonPressed)
58     { //sending after button is pressed
59         copyBufferCOM2RF();
60         pulseLEDR();
61         PIN = 0;
62         DLEN = 9;
63         RFTXpacket();
64         waitDelay(25);
65     }
66 }
67 }

```

Výpis F.4: Kód pre vysielateľ odosielaajúci teplotu z interného čidla po stlačení tlačidla na IQRF koordinátor (TeplotaTXIQRf.c)

```

1
2 import time
3 import struct
4 from machine import SPI, Pin, I2C
5 from ahtx0 import AHT20
6
7 #initialization of SPI bus
8 sck_pin = machine.Pin(10, machine.Pin.OUT)
9 mosi_pin = machine.Pin(11, machine.Pin.OUT)
10 miso_pin = machine.Pin(12, machine.Pin.IN)
11 cs_pin = machine.Pin(13, machine.Pin.OUT)
12
13 spi = SPI(1, baudrate = 20000, polarity=0, phase=0, sck=sck_pin,
14         mosi=mosi_pin, miso=miso_pin)
15 cs_pin.value(1)
16
17 #initialiyation of I2C bus
18 i2c1 = I2C (1, scl = Pin(15), sda = Pin(14), freq = 100000)
19
20 #initialization of temperature sensor
21 tmp_sensor = AHT20(i2c1)
22 tmp_sensor.initialize()
23 temp=tmp_sensor.temperature
24
25 #function which reads temperature from sensor
26 def ReadTempOPT():
27     time.sleep(0.5)
28     temp=tmp_sensor.temperature
29     temp=(temp+40)*10 #optimalization of temperature
30     print ("{:d}".format(int(round(temp, 0))))
31     time.sleep(1)
32     #foramting temperature ot xxx format
33     return "{:d}".format(int(round(temp, 0)))
34
35 #calculation of CRC
36 def calculate_crc(hex_string):
37     # Initialize XOR with F0, 85, and 5F
38     xor_result = 0xF0 ^ 0x83 ^ 0x5F
39     for i in range(0, len(hex_string), 2):
40         xor_result ^= int(hex_string[i:i+2], 16)
41     return xor_result
42
43 #function which creates SPI packet and after that packet is send
44 def SendTemp(temp):
45     hex_string = ''.join('{:02x}'.format(ord(c)) for c in temp)
46     crc_value = calculate_crc(hex_string)

```

```

46     message = f'f083{hex_string}{crc_value:02x}00'
47     byte_array = bytes.fromhex(message)
48     formatted_string = ''.join('\x{:02x}'.format(b) for b in
49         byte_array)
50     print(f"Formatted SPI Message: {formatted_string}")
51     spi.write(byte_array)
52
53 while 1:
54     temp = ReadTempOPT()
55     cs_pin.off()           #start of SPI communication
56     time.sleep_ms(10)
57
58     SendTemp(temp)
59     time.sleep_ms(10)
60
61     print(spi.read(20))   #reading of slave answer
62     time.sleep_ms(10)
63
64     cs_pin.on()          # end of SPI communication

```

Výpis F.5: Program pre dosku BPC-IoT s optimalizovaným tvarom pre teplotu (Bonus.py)