

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická
Katedra řídicí techniky

DIPLOMOVÁ PRÁCE



Bc. Martina Buřčová

Modul pro monitorování stavu zařízení s rozhraním LoRa

Vedoucí práce: doc. Ing. Radislav Šmíd, Ph.D.

Studijní program: Otevřená informatika

Obor: Počítačové inženýrství

Praha 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Buřičová** Jméno: **Martina** Osobní číslo: **420177**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Otevřená informatika**
Studijní obor: **Počítačové inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Modul pro monitorování stavu zařízení s rozhraním LoRa

Název diplomové práce anglicky:

Condition Monitoring Module with LoRa Communication

Pokyny pro vypracování:

1. Navrhněte a realizujte modul pro monitorování stavu průmyslových zařízení pomocí analýzy vibrací a jeho připojení k IoT síti ThingsNetwork pomocí LoRa rozhraní.
2. Použijte low-power mikrokontroler řady STM32L07 a vytvořte pro něj software, který bude zajišťovat digitalizaci signálu z akcelerometru ADXL1002Z, předzpracování a analýzu signálu, výpočet příznaků a bezdrátový přenos do IoT cloudu.
3. Zaměřte se na implementaci pokročilejších metod analýzy signálu, implementujte výpočet kurtogramu apod.
4. Vyhodnoťte funkci zařízení pomocí simulátoru vibrací rotačních mechanismů.
5. Vytvořte demonstrační aplikaci pro vyhodnocování získaných dat.

Seznam doporučené literatury:

- [1] Randall, Robert Bond, Vibration-based Condition Monitoring, John Wiley & Sons, Ltd. 2011
- [2] J. Antoni, Fast Computation of the Kurtogram for the Detection of Transient Faults, Mechanical Systems and Signal Processing, Volume 21, Issue 1, January 2007, Pages 108-124.
- [3] Dokumentace STMicroelectronics, <http://www.st.com>
- [4] Dokumentace The Things Network, <https://www.thethingsnetwork.org/>
- [5] Dokumentace Analog Devices, <http://www.analog.com>

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Radislav Šmíd, Ph.D., katedra měření FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **16.01.2018** Termín odevzdání diplomové práce: **25.05.2018**

Platnost zadání diplomové práce: **30.09.2019**

doc. Ing. Radislav Šmíd, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studentky

Prohlášení

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

V Praze dne

Podpis autora

Anotace

Cílem této diplomové práce je navrhnout a realizovat modul pro monitorování stavu průmyslových zařízení pomocí analýzy vibrací a jeho připojení k IoT síti ThingsNetwork pomocí rozhraní LoRa. Je použit low-power mikrokontroler řady STM32L07 a pro něj je vytvořen software, který zajišťuje digitalizaci signálu z akcelerometru ADXL1002Z, předzpracování a analýzu signálu, výpočet příznaků a bezdrátový přenos do IoT cloudu. Jsou naimplementovány pokročilejší metody analýzy signálu, jako je například výpočet kurtogramu apod. Funkce zařízení je vyhodnocena pomocí simulátoru vibrací rotačních mechanismů. Na závěr je vytvořena demonstrační aplikace pro vyhodnocování získaných dat.

Annotation

The goal of this thesis is to design and implement a module for monitoring of industrial devices by vibration analysis and its connection to the ThingsNetwork using the LoRa interface. The low-power microcontroller STM32L07 is used, and software is created to provide digitization of the ADXL1002Z accelerometer signal, preprocessing, signal analysis and wireless transmission to the IoT cloud. Advanced methods of signal analysis, such as kurtogram calculations, is implemented. The function of the device is evaluated using the vibration simulation of the rotating mechanisms. Finally, a demo application is created to evaluate the data.

Poděkování

Ráda bych poděkovala vedoucímu diplomové práce panu doc. Ing. Radislavu Šmídovi, Ph.D za jeho trpělivost a vstřícnost při konzultacích, odborné rady a nasměrování vždy správným směrem.

Obsah

1	Úvod	1
1.1	Analýza vibrací	2
1.2	Monitorování zařízení s použitím internetu věcí	2
1.3	Cíl práce	4
2	Architektura systému	5
2.1	Vývojový kit P-NUCLEO-LRWAN1	6
2.1.1	Processorové jádro ARM Cortex-M0+	6
2.1.2	Paměti vývojového kitu	7
2.2	Akcelerometr	10
3	Technologie LoRa	12
3.1	Síťová architektura	12
3.1.1	Koncový uzel	13
3.1.2	Brána	13
3.1.3	Řídící server	13
3.1.4	Aplikační server	13
3.2	LoRaWan	14
4	Poruchy AC motorů	15
5	Zpracování dat z akcelerometru a výpočet příznaků	17
5.1	RMS (root mean square)	17
5.2	Crest factor	18
5.3	Kurtogram	18
5.3.1	Spectral kurtosis (spektrální koeficient špičatosti)	19
5.3.2	Dvojice frekvence a šířka pásma	20
5.4	Fast kurtogram	20
5.4.1	Filtiry FIR (finite impulse response) a IIR (infinite impulse response)	20
5.4.2	Algoritmus	21
5.4.3	Ukázka detekce vady ložiska pomocí kurtogramu	24
5.5	Kurtosis ratio	26
6	Softwarové zpracování	28
6.1	Měřicí a vyhodnocovací jednotka	28
6.1.1	Nastavení periférií	28
6.1.2	Hlavní smyčka programu	29
6.1.3	Efektivní hodnota a Crest factor	30
6.1.4	Kurtosis ratio	30
6.1.5	Kurtogram	31
6.1.6	Přístup do paměti Flash	32
6.1.7	Komunikace přes rozhraní LoRa	33

6.2	Cloudové zpracování	35
6.2.1	Popis funkčních uzlů	36
6.2.2	Uživatelské rozhraní	38
7	Testování zařízení	40
7.1	První experiment	41
7.1.1	Motor bez poruchy	41
7.1.2	Motor s poruchou	44
7.1.3	Závěr experimentu	48
7.2	Druhý experiment	49
7.2.1	Motor bez poruchy	49
7.2.2	Motor s poruchou	52
7.2.3	Závěr experimentu	55
8	Závěr	56
	Literatura	58
	Příloha A Obsah CD	61

1 Úvod

Otázka jak často, a zda vůbec, provádět kontrolu a údržbu různých strojů je důležitá snad ve všech průmyslových odvětvích. Týká se zejména bezpečnosti, avšak perioda kontrol může také ovlivňovat náklady na opravu, nebo případnou finanční ztrátu při dočasném zastavení výroby.

Jednou z nejinformativnějších referencí je kniha Roberta Bond Randalla z názvem *Vibration-based Condition Monitoring* [1]. Z této knihy jsou převzaty některé následující informace ohledně kontroly a údržby strojů.

Pouze spustit stroj a nechat ho běžet dokud se nerozbije může mít v některých případech katastrofální následky. Stále se však tato metoda v některých případech používá, například tam, kde je velké množství malých strojů a výpadek jednoho stroje na krátkou dobu nezpůsobí kritické problémy (např. šicí stroje).

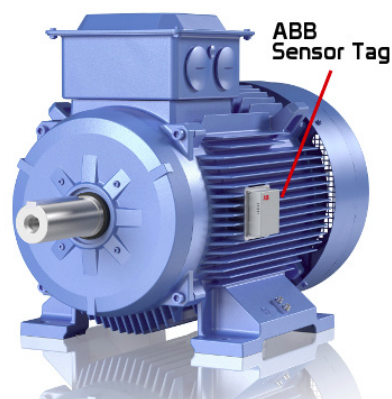
Na druhou stranu, poměrně často využívaná metoda kontroly v pravidelných intervalech může vést k vyšší spotřebě náhradních dílů a častější údržbě, než je skutečně třeba. Navíc existuje možnost, že se vyskytne neočekávaná porucha. Kontrola většinou probíhá v intervalech, které jsou kratší než očekávaný čas mezi poruchami. V drtivé většině případů je ale kontrola zbytečně častá.

Monitorování stavu zařízení (*condition-based maintenance*) je v mnoha případech výhodnější, ekonomičtější a spolehlivější. Potenciální porucha stroje je předvídána pravidelným sledováním stavu a údržba se provádí až ve chvíli, kdy je třeba. Tato metoda má největší úspěch v odvětvích, kde se očekává, že budou stroje v nepřetržitém provozu při stálém zatížení (např. výroba elektřiny, chemický průmysl).

Existuje několik typů různých poruch motorů. Některé mohou být zapříčiněny přímo činností stroje, jiné mohou být způsobeny vnějšími podmínkami, či způsobem zacházení s daným strojem. Mezi nejčastější poruchy strojů patří poruchy ložisek, dále pak poruchy v oblasti rotoru a statoru.



(a)



(b)

Obrázek 1.1: (a) Současný stav monitorování stavu motorů pomocí analýzy vibrací. Obrázek převzat z [2]. (b) Motor s ABB senzorem měřící vibrace a teplotu motoru. Obrázek převzat z [3].

Monitorování probíhá v současné době tak, že se ke kontrolovanému stroji připojí měřicí zařízení, které si uloží naměřená data do paměti a zobrazuje obsluze statistiku měřených hodnot (Obrázek 1.1(a)). Některá monitorovací zařízení umožňují danou statistiku odesílat na server, či do mobilního zařízení, příkladem takového zařízení je chytrý senzor pro motory od firmy ABB [4]. Tento senzor snímá vibrační data a zároveň teplotu. Z naměřených dat počítá efektivní hodnotu a statistická data dále umožňuje posílat přes Bluetooth technologii do mobilního zařízení. Odesílat data je možné do vzdálenosti nejvýše deseti metrů (Obrázek 1.1(b)). Jiná zařízení dokonce odesílají veškerá naměřená data na jednotku určenou ke zpracování dat. Takovýto způsob je velmi neefektivní, jelikož se posílá velké množství dat.

1.1 Analýza vibrací

K určení stavu daného zařízení v době, kdy je v provozu se nejčastěji používá analýza vibrací. Změna vibrací nám často předem signalizuje, že ve stroji dochází k nějaké závadě. Můžeme tak včas identifikovat nejen problémy s ložisky, rotory a podobně, ale je možné identifikovat i uvolněné šrouby nebo součástky, které potřebují promazat dříve než způsobí nějakou větší škodu.

Vibrace stroje jsou spjaty s dynamickým namáháním stroje a technickým stavem hřídelů, převodovek, nevyváženosti rotujících dílů, vůlemi v kluzných ložiscích, opotřebením, únavou materiálů, vznikajícími trhlinami, korozí, atd. Vibrace jsou buzeny, jak rotujícími, tak přímočaře se pohybujícími tělesy (včetně pohybu kapalin a plynů). Mechanické vibrace jsou také způsobeny rázy, při níž střetem dvou navzájem se pohybujících těles (např. pohybem poškozené strojní části, kuličkou v ložisku apod.) dochází k náhlé změně gradientu určující veličiny vibrací. Ráz způsobí přechodový kmitavý jev generující v tělese postupnou rázovou vlnu. [5]

Další možné metody k detekci stavu zařízení jsou například: analýza maziv, analýza výkonu, nebo termografie.

1.2 Monitorování zařízení s použitím internetu věcí

Internet věcí je nový trend, který se uplatňuje skoro ve všech oborech lidské činnosti. Jedná se o síť fyzických zařízení, která jsou vybavena elektronikou, softwarem a senzory, díky kterým jsou schopna snímat danou fyzikální veličinu. Tato zařízení jsou vzájemně propojená a umožňují výměnu a sdílení dat, případně informací. Data lze zpracovávat a využívat v nejrůznějších oblastech.

Podle způsobu využití se rozlišují dva směry internetu věcí a to průmyslový a spotřebitelský [6].

Průmyslový internet věcí se zaměřuje na efektivnější využívání zdrojů, bezpečnost pracovníků, snížení provozních nákladů, ale také na zvýšení pracovní produktivity. Zároveň umožňuje monitorování a včasnou údržbu průmyslových strojů. V tomto případě je velice důležitá bezpečnost, jelikož se často jedná o kritické úlohy, kde selhání může mít neblahý vliv na lidské životy. Co se způsobu komunikace týče, průmyslový internet věcí používá jak komunikaci od zařízení do vzdáleného cloudu, tak i komunikaci mezi samotnými zařízeními.

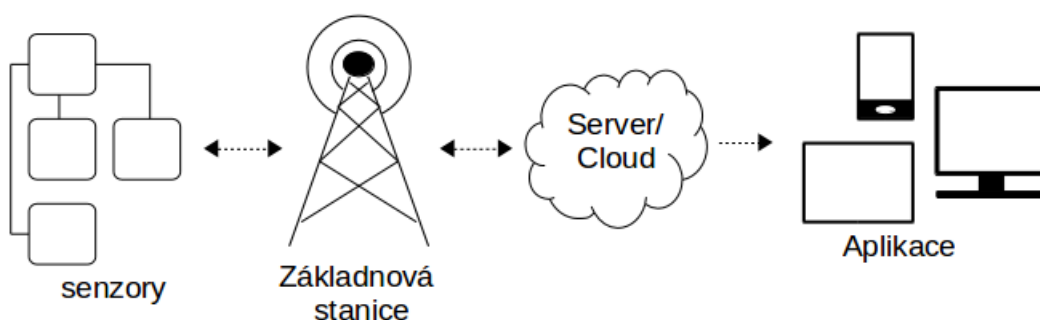
Příklady aplikace průmyslového internetu věcí je nespočet. Používá se například v průmyslové automatizaci nebo v energetice, kde může sloužit například k detekci škodlivých

látek, monitorování strojů nebo vstupů atd. Používá se také v lékařství ke sledování stavu pacientů nebo v dopravě při automaticky řízených automobilech. Mimo jiné se pomáhá k rozšíření konceptu chytrých měst.

Spotřebitelský internet věcí se zaměřuje na spotřebitele a na zjednodušení každodenního života. Hlavním zaměřením je zvýšení uživatelského zážitku. V tomto případě se nejčastěji používá komunikace od daného zařízení do cloudu.

Spotřebitelský internet věcí má využití například v chytrých domácnostech, kde může monitorovat spotřeby elektrické energie, případně vody, nebo detekovat otevření dveří, či oken. Dále se může použít u různých chytrých spotřebičů, ať už jsou to pračky, televize, či lednice. Své uplatnění najde ale i v zábavní elektronice.

V dnešní době, kdy je Internet věcí stále více a více rozšířený a dostává se do všech aspektů našeho života, je rozhodně na místě propojit monitorování stavu zařízení s vhodnou bezdrátovou technologií umožňující IoT. Internet věcí přináší nové možnosti vzájemné interakce mezi systémy nebo zařízeními, zároveň přináší nové způsoby jejich ovládání, sledování a vyhodnocování získaných dat.



Obrázek 1.2: Příklad IoT sítě a princip komunikace s použitím internetu věcí.

Existuje několik různých technologií a protokolů které umožňují bezdrátové propojení s IoT, pracující na vzdálenost několika kilometrů, jako například:

- **LoRa:** Tato technologie je stavěná na vysoké dosahy (Long Range). Je vhodná pro přenos malého množství dat. Díky nízké přenosové rychlosti má i nízký odběr. Technologie je uzpůsobena tak, aby na přímou viditelnost mohla komunikovat až na vzdálenost 70 km. V praktickém použití se však rozmisťují komunikační uzly na vzdálenost 20 km.

LoRa technologie je podrobněji popsána v Kapitole 3.

- **Sigfox:** Vyznačuje se nízkou přenosovou rychlostí a malým objemem přenesených dat na velkou vzdálenost (ve volné krajině až 50 km). Síť používající technologii Sigfox [7] je velice odolná proti rušení. Používá hvězdicovitou topologii, kde jednotlivá koncová zařízení posílají data na Sigfox bránu, které je dále posílají do Sigfox cloudu. Technologie byla založen roku 2009 ve Francii.

Sigfox technologie je velice podobná technologii LoRa, jak architekturou sítě, tak i parametry, avšak na rozdíl od LoRa technologie používá Sigfox pouze úzkopásmový přenos.

LoRa a Sigfox jsou nejčastěji používané technologie v internetu věcí, což je zřejmě dáno jejich snadnou dostupností a zároveň i nižší pořizovací cenou.

- **NB IoT** [8]: Jedná se o poměrně novou, úzkopásmovou technologii. Byla vyvinuta tak, aby umožnila efektivní komunikaci zařízení, která přenášejí pouze malé množství dat. Používá již existující infrastrukturu mobilních operátorů, kde se úpravou softwaru na vysílacích stanicích vyhradí část LTE pásma (Long Term Evolution – technologie určená pro vysokorychlostní Internet v mobilních sítích). Umožňuje komunikaci na vzdálenost 15 km.

Následující technologie pracují pouze na krátké vzdálenosti, v řádech několika desítek metrů.

- **ZigBee**: Podobně jako Bluetooth technologie je určena pro spojení nízkovýkonových zařízení v sítích PAN (personal area network) na malé vzdálenosti do 75 metrů. Na rozdíl od Bluetooth, ZigBee vzniklo za účelem průmyslové automatizace. Vyznačuje se jednoduchou a nenáročnou implementací a velmi nízkou spotřebou energie. Jedná se o standart platný od roku 2004. [9]
- **Z-Wave**: Technologie Z-Wave [10] má dosah přibližně 100 m. Nejčastěji se používá pro domácí automatizaci. Slouží k přenosu malého množství dat. Je vhodná pro zařízení napájené bateriemi. Z-Wave protokol byl vytvořen roku 2001.
- **WiFi, Bluetooth a další.**

1.3 Cíl práce

Cílem této práce je navrhnout a realizovat modul pro monitorování stavu průmyslových zařízení pomocí analýzy vibrací. Následně bude realizováno jeho připojení k IoT síti ThingsNetwork pomocí LoRa rozhraní.

V této práci bude použit low-power mikrokontroler řady STM32L07, přesněji se jedná o Nucleo kit P-NUCLEO-LRWAN1 založen na STM32L073RZ-Nucleo, který již obsahuje modul pro komunikaci pomocí LoRa rozhraní. K tomuto kitu bude připojen akcelerometr ADXL1002Z pro určení aktuálních vibrací.

Pro tento kit bude vytvořen software, který bude zajišťovat digitalizaci signálu z akcelerometru. Dále předzpracuje a analyzuje data a výpočte z nich příznaky. K tomuto účelu bude implementován výpočet kurtogramu, který dokáže detekovat a charakterizovat nestacionární části v signálu. Program bude dále zajišťovat bezdrátový přenos příznaků do sítě ThingsNetwork.

Jelikož bude samotný výpočet příznaků probíhat přímo na kitu a do sítě ThingsNetwork se budou posílat pouze příznaky, čili vypočtené a naměřené hodnoty signalizující stav zařízení, bude možné použít bezdrátovou technologii LoRa.

Zařízení bude testováno a jeho funkce bude vyhodnocena pomocí simulátoru vibrací rotačních mechanismů.

Přínosem této diplomové práce je pokročilá analýza signálu vibrací přímo na vestavěném měřicím čipu. Další invence je použití kurtogramu. Současné monitorovací systémy pouze pracují s efektivní hodnotou frekvence, případně crest factorem, či podobným nástrojem, zatímco kurtogram je komplexnější nástroj, který dokáže detekovat nestacionární chování a určit jeho frekvenci a šířku pásma, čímž umožňuje snazší identifikaci zdroje potíží. Propojení monitorování průmyslových strojů spolu s internetem věcí a použitím technologie LoRa, umožňuje odesílat data na vzdálenost až několika kilometrů.

2 Architektura systému

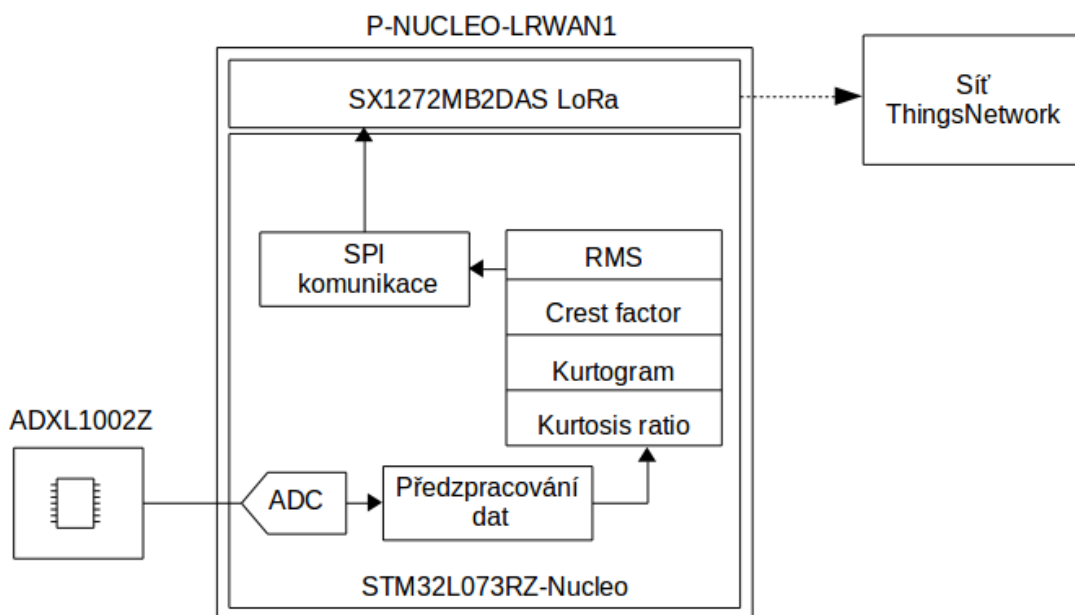
Tato kapitola popisuje architekturu systému, čili hardwarové řešení daného problému a podrobněji popisuje nejen použitý vývojový kit, ale i použitý senzor pro vyhodnocování vibrací.

Pro realizaci modulu pro monitorování stavu zařízení byl použit kit P-NUCLEO-LRWAN1, který již obsahuje modul s anténou pro komunikaci prostřednictvím LoRa technologie. Na jeho vnitřní analogově-digitální převodník je připojen jednoosý akcelerometr ADXL1002Z, který slouží k vyhodnocení vibrací monitorovaného přístroje.

Jednotlivé vypočtené parametry signálu, jako je RMS, crest factor, kurtogram a kurtosis ratio, signalizující stav zařízení, jsou dále poslány pomocí LoRa rozhraní do sítě ThingsNetwork, kde jsou dále analyzovány.

Sít ThingsNetwork je celosvětová, stále se rozšiřující síť vytvořená pro internet věcí, používající LoRa technologii a komunikující prostřednictvím LoRaWAN protokolu. Tato síť je veřejně dostupná a poskytuje webové rozhraní pro kontrolu přijatých dat do sítě z předem daného zařízení. Více než 350 komunit po celém světě pracuje na poskytování této služby svým obyvatelům. Největší počet bran (108) pro síť ThingsNetwork je v Curychu.

Konkurenční síť pro internet věcí, s technologií LoRa, poskytují České Radiokomunikace [11]. Zajišťují bezpečné ukládání dat a šifrované přenosy. Volně dostupná, bezplatná verze však umožňuje připojení do sítě LoRa pouze na jeden měsíc.



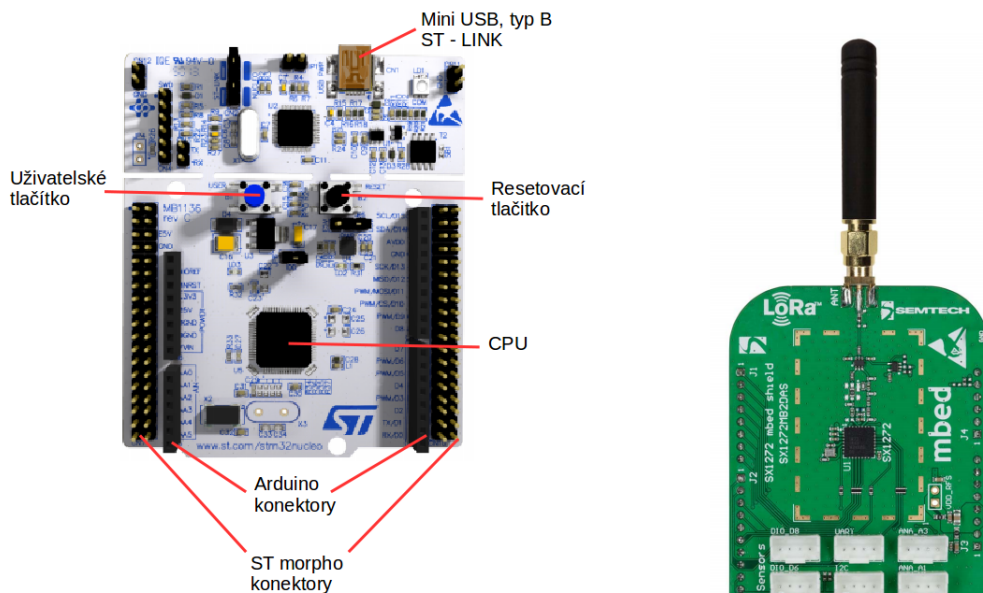
Obrázek 2.1: Blokový diagram znázorňující způsob propojení jednotlivých součástí modulu pro monitorování stavu zařízení.

2.1 Vývojový kit P-NUCLEO-LRWAN1

Vývojový kit P-NUCLEO-LRWAN1 [12], [13] je založen na STM32L073RZ-Nucleo kitu a rozšiřujícím modulu SX1272MB2DAS LoRa od společnosti Semtech. Tyto dvě vývojové desky jsou propojeny přes Arduino konektory.

Kit poskytuje 192 KB Flash paměti a 20 KB RAM paměti. Umožňuje dva způsoby napájení a to buď z USB, nebo z externího zdroje. Obsahuje dvě tlačítka, jedno resetovací a jedno uživatelské, a tři LED diody jednu pro signalizaci USB komunikace, jednu uživatelskou diodu a jednu diodu pro indikaci napájení. Dále poskytuje USB 2.0 full speed a řadič pro LCD. Vývojový kit je také vybaven ST-LINK/V2-1 debuggerem s SWD konektorem. Kromě Arduino konektorů dále umožňuje rozšíření pomocí ST morpho konektorů. Rozšiřující modul navíc obsahuje teplotní senzor a indikátor vybité baterie.

Vývojový kit STM32L073 používá 32 bitový mikrokontrolér STM32L073RZT6 (ultra low power) založen na jádru ARM Cortex-M0+.



Obrázek 2.2: Vývojový kit z přední strany a rozšiřující modul pro LoRa technologii. Původní obrázky bez popisků převzaty z [12].

Rozšiřující modul umožňuje komunikaci pomocí technologie LoRa, jejíž princip je blíže popsán v Kapitole 3.

2.1.1 Procesorové jádro ARM Cortex-M0+

Toto jádro se používá zejména pro mikrokontrolérové nenáročné aplikace, kde je vyžadována nízká cena, úspora energie a malé rozměry. Je nejmenší z řady mikrokontrolérů ARM a je postaveno na jádru ARMv6.

Podporuje instrukční sadu Thumb (16 bitové instrukce) a některé instrukce ze sady Thumb2 (32 bitové instrukce). Jádro je vybaveno dvou-stupňovou pipeline a umožňuje adresaci typu bit banding. Zároveň také povoluje až 32 fyzických přerušení. Dále obsahuje hardwarovou 32 bitovou násobičku. Využívá Von Neumannovu architekturu (má společnou paměť jak pro data, tak pro instrukce).

2.1.2 Paměti vývojového kitu

Vývojový kit disponuje 192KB prostorem Flash paměti a statickou pamětí RAM o velikosti 20 KB. Flash paměť je určena především pro programový kód a konstantní hodnoty. Do RAM paměti se ukládají veškerá dočasná data.

Paměť RAM

K této paměti lze přistupovat po bytech, 16 bitových slovech (half-word), nebo 32 bitových slovech (word). Přístup je možný na maximální frekvenci systémových hodin, bez jakéhokoli zpoždění (platí jak pro procesor, tak pro DMA). Počet čtení a zápisů není nijak omezen.

Paměť RAM je typu volatile, čili po odpojení napájení ztratí veškerá svá data.

Organizace paměti Flash

Paměťový prostor je rozdělen na dvě banky. Ty jsou dále rozděleny na sektory. Každá banka má 24 sektorů. Sektory se dále dělí na stránky. Každý sektor má 32 stránek. Velikost stránky je 128 bytů. Obsahuje ochranu proti nevyžádanému zápisu do paměti a umožňuje low-power mód. Paměť Flash má uložených šest bitů pro každé slovo, které rozpoznávají a opravují jednu chybu (ECC - Error Correction Code).

NVM	NVM addresses	Size (bytes)	Name	Description		
Flash program memory	0x0800 0000 - 0x0800 007F	128 bytes	Page 0	sector 0	Bank 1	
	0x0800 0080 - 0x0800 00FF	128 bytes	Page 1			
	-	-	-			
	0x0800 0F80 - 0x0800 0FFF	128 bytes	Page 31			
		
		
		
	0x0800 7000 - 0x0800 707F	128 bytes	Page 224	sector 7		
	0x0800 7080 - 0x0800 70FF	128 bytes	Page 225			
	-	-	-			
	0x0800 7F80 - 0x0800 7FFF	128 bytes	Page 255			
		
		
		
	-	-	-	-		
	0x0801 7F80 - 0x0801 7FFF	128 bytes	Page 767	sector 23		Bank 2
	0x0801 8000 - 0x0801 807F	128 bytes	Page 768	sector 24		
		
		
		
	0x0802 F000 - 0x0802 F07F	128 bytes	Page 1504	sector 47		
	0x0802 F080 - 0x0802 F0FF	128 bytes	Page 1505			
	-	-	-			
	0x0802 FF80 - 0x0802 FFFF	128 bytes	Page 1535			

Obrázek 2.3: Organizace paměti Flash pro kit STM32L073RZ. Převzato z [12]

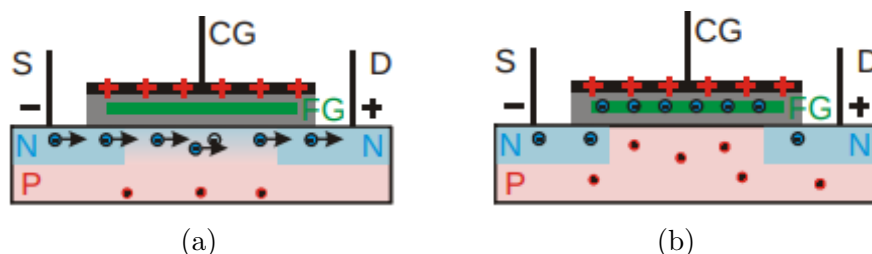
Flash paměť je typu non-volatile, což znamená, že pro činnost, jako je čtení nebo zápis, potřebuje napájení, ale při odpojení napájení se informace uložená v paměti uchovává.

Technologie Flash paměti

Jedná se o elektricky programovatelnou paměť. Výhodou této paměti je, že ji lze znovu naprogramovat, měnit její obsah, bez vyjmutí ze zařízení s použitím minima pomocných obvodů. Hlavní nevýhodou této paměti je omezený počet přepisů jednotlivých paměťových buněk. Paměť je vnitřně organizována po blocích a všechny paměťové buňky v rámci jednoho bloku musí být vymazány současně.

Je tvořena sítí řádků a sloupců, na jejichž průsečících leží jednotlivé paměťové buňky (unipolární tranzistory). Tranzistory obsahují dvě hradla, jedno řídicí (control gate – CG) a jedno plovoucí (floating gate – FG), izolované od okolí. Jelikož je FG izolované, všechny elektrony na něj přivedené jsou zde uvězněny. Tím je uložena informace.

Pokud je uvnitř plovoucího hradla uzavřen náboj, znamená to, že v buňce je uložena hodnota 0. Náboj elektronů v FG vyruší elektrické pole z CG, čímž se zvýší prahové napětí (VT1) buňky. To znamená, že nyní musí být na CG aplikováno vyšší napětí (VT2), aby byl kanál vodivý. Pro čtení hodnoty z tranzistoru se na CG použije mezifázové napětí mezi prahovými napětími VT1 a VT2. Pokud tranzistorem protéká proud v tomto mezilehlém napětí, musí být FG vybit (Pokud by uvnitř FG byl uzavřen náboj, neprotékal by tranzistorem proud, protože mezifázové napětí je menší než VT2), a proto je v bráně uložena logická hodnota 1. Pokud kanál nevede na mezilehlé napětí, znamená to, že FG je nabitý, a proto je v bráně uložena logická hodnota 0. Přítomnost logické 0 nebo 1 se zjišťuje tím, že se určuje, zda proud protéká tranzistorem, když se na řídicí hradlo připojí mezifázové napětí. [14]



Obrázek 2.4: Působení mezifázového napětí na řídicí hradlo (CG) (a) Ve FG nejsou zachyceny elektrony. Kladný náboj na CG naindukuje vodivou oblast, která umožní průchod proudu od source do drain. V buňce je uložena hodnota 1. (b) Pokud jsou ve FG zachyceny elektrony, jejich náboj vyruší působení kladného napětí na CG, čili neprochází proud mezi S a D. V buňce je uložena hodnota 0. Obrázky jsou převzaty z [15]

Mazání paměti Flash spočívá v tom, že se z plovoucího hradla odvede uložený náboj pryč.

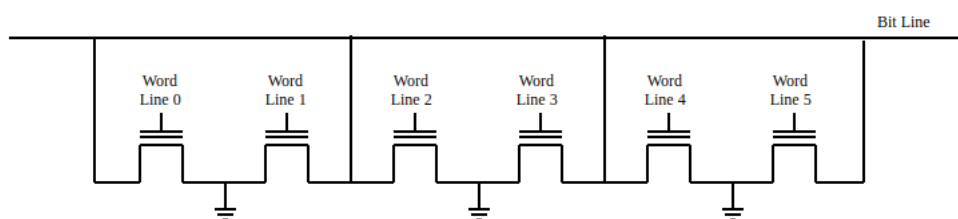
Existují dva typy buněk podle množství uchovávané informace:

- **Single-level Cell (SLC):** U tohoto typu rozlišujeme pouze, zda na FG jsou nebo nejsou elektrony, čili v buňce můžeme ukládat pouze 1 bit informace (0 nebo 1). Vyznačuje se jednodušší konstrukcí, menší citlivostí na poškození a stárnutí. Nevýhodou je, že potřebujeme více buněk.

- **Multi-level Cell (MLC):** Při čtení rozlišujeme různá množství náboje chyceného na FG. V buňce můžeme ukládat více informací. Nevýhodou je složitější konstrukce a větší citlivost na poškození a stárnutí. Je potřeba méně buněk na rozdíl od SLC.

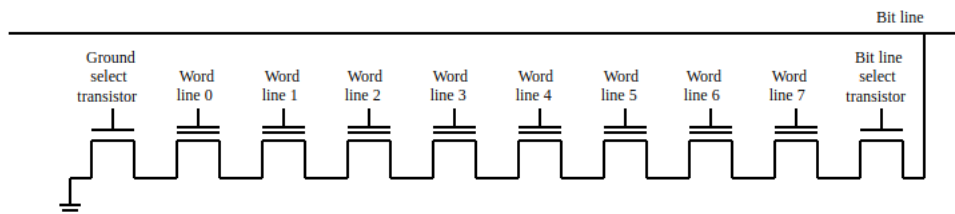
Dva hlavní typy Flash paměti jsou paměti typu NOR a NAND, lišící se zapojením paměťových buněk i principem jejich činnosti.

- **NOR Flash:** Paměťová buňka paměti NOR má tři přívody: bit line, word line a GND. Je adresována pomocí word line, díky čemuž lze programovat byte, nebo celou stránku. Chová se podobně, jak hradlo NOR; když je na jeden z word line přivedena vysoká úroveň, tak odpovídající paměťový tranzistor pošle na bit line nízkou úroveň. Mazání a programování NOR Flash paměti je pomalejší než čtení.



Obrázek 2.5: Způsob propojení paměťových buněk u NOR Flash. Převzato z [14]

- **NAND Flash:** Vždy je několik paměťových buněk zapojeno za sebou v sérii. Což značně komplikuje čtení i zápisy, protože není možné přistupovat k jednotlivým bitovým buňkám. Nejmenší adresovatelná jednotka je stránka. Používá se čtení několika tisíc bitů naráz. Dochází k častému ničení buněk, proto je důležitá správa vadných sektorů. Používají se pouze jako datové paměti, není možné je použít ke čtení instrukčního kódu.



Obrázek 2.6: Způsob propojení paměťových buněk u NAND Flash. Převzato z [14]

Každá Flash paměť musí kromě buněk, kde se uchovává daná informace, obsahovat řadič, který řídí zápis do buněk tak, aby se neustále nepřepisoval obsah jedné a té samé buňky (brzo by došlo k jejímu znehodnocení). Zároveň musí obsahovat zdroj vyššího napětí potřebného na přepisování obsahu buněk (obvyklé napájecí napětí 5 V nedostačuje). Tento zdroj často bývá nejcitlivější částí paměti [15].

2.2 Akcelerometr

Senzory vibrací se dělí na absolutní a relativní senzory.

Absolutní senzor vibrací měří veličinu mechanických vibrací vzhledem k vlastní setrvačné soustavě. Používá se tam, kde není k dispozici vhodný relativní pevný bod k upevnění vztažného bodu senzoru (jedoucí vozidla, letadla).

Relativní senzory vibrací se používají při vyhodnocení vibračního pohybu jedné části stroje vůči druhé.

V této práci byl pro účely snímání vibrací použit akcelerometr na připojovací desce EVAL-ADXL1002Z. Informace o ní pochází převážně ze stránek Analog Devices [16]. Podle předchozího rozdělení jde o absolutní senzor vibrací.

Jedná se o připojovací desku, která umožňuje rychlé připojení senzoru ADXL1002. Tyto akcelerometry jsou speciálně navrženy pro montáž na stroje, které mohou vydávat silné vibrace, a tudíž jsou vyrobeny z velmi silné desky plošných spojů. Na výstupu je k dispozici jednoduchý RC filtr typu dolní propust se zlomovou frekvencí 20 kHz (při poklesu -3 dB).



Obrázek 2.7: Fotografie připojovací desky EVAL-ADXL1002Z. Převzato z [16].

Jedná se o jednoosé akcelerometry s šířkou pásma do 11 kHz, realizované mikroelektromechanickými systémy (MEMS), které poskytují analogový výstup úměrný zrychlení pohybu mechanických vibrací.

Senzor ADXL1002 je dodáván v $5 \times 5 \times 1.8\text{mm}$ pouzdru a je schopný pracovat v teplotním rozmezí od -40°C do 125°C . Může být napájen napětím v rozmezí od 3.3 do 5.25 V. Citlivost senzoru je 40 mV/g . Při nulovém g je na výstupu polovina napájecího napětí. Měřicí rozsah je $\pm 50\text{ g}$.

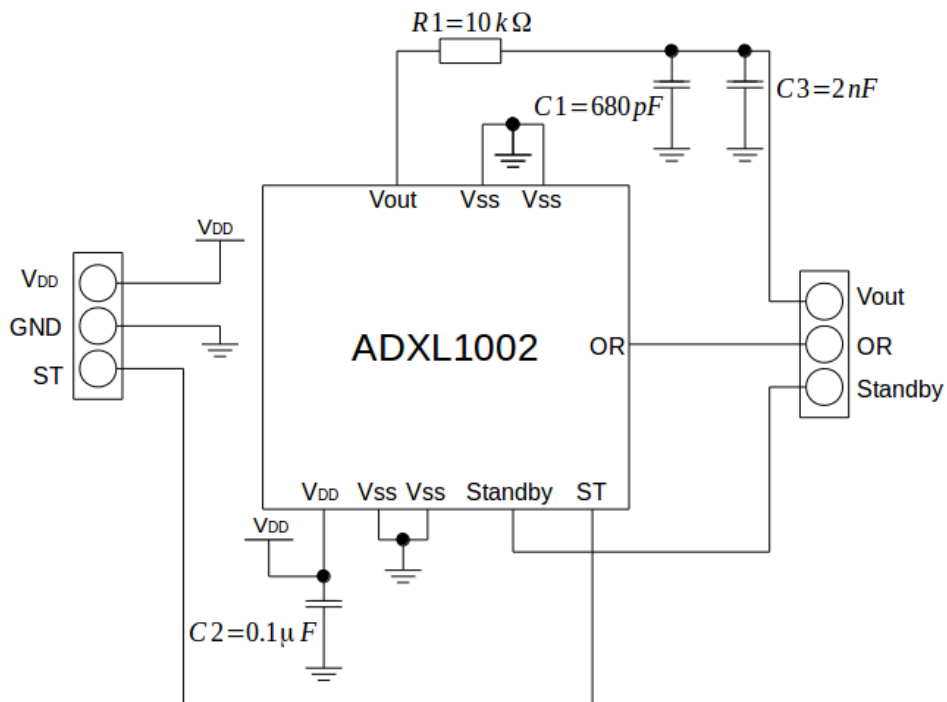
ADXL1002 má dva provozní režimy: měřicí režim a standby režim. Režim měření poskytuje nepřetržitý analogový výstup pro aktivní sledování a odebírá 1 mA (při napájení 5 V). Standby režim je neměřicí režim s nízkou spotřebou ($225\ \mu\text{A}$). Přechod z režimu pohotovostního do režimu měření trvá méně než $50\ \mu\text{s}$.

Vstupní analogový signál je převáděn a následně vyjádřen pomocí analogově-digitálního převodníku s rozlišením dvanácti bitů. Aby se při vzorkování signálu neztratila informace, musí vzorkovací frekvence splňovat Shannon-Kotělnikovu vzorkovací větu. Tato věta říká, že spojitý signál obsahující pouze frekvenční složky s frekvencemi menšími než f_{max} může být ze vzorků jednoznačně rekonstruována jen tehdy, pokud je vzorkovací frekvence větší než dvojnásobek f_{max} , neboli:

$$f_{vz} > 2 \cdot f_{max}.$$

Hraniční frekvence $f_{vz}/2$ se nazývá Nyquistova frekvence. Tato podmínka udává teoretickou hranici, v praxi je nutné pro vzorkování použít čtyř, nebo vícenásobek maximální frekvence signálu.

Pro vyloučení aliasingu byla upravena zlomová frekvence filtru typu dolní propust na 5 kHz.



Obrázek 2.8: Schéma zapojení senzoru na připojovací desce EVAL-ADXL1002Z.

3 Technologie LoRa

LoRa (Long Range) je jednou z technologií pro internet věcí. Používá se v aplikacích využívající komunikaci na velké vzdálenosti, které si zároveň vystačí s malým datovým tokem, případně se od nich ještě očekává velmi nízká spotřeba. Údaje a informace o této technologii jsou převzaty z webových stránek The things network [17], a LoRa Alliance [18].

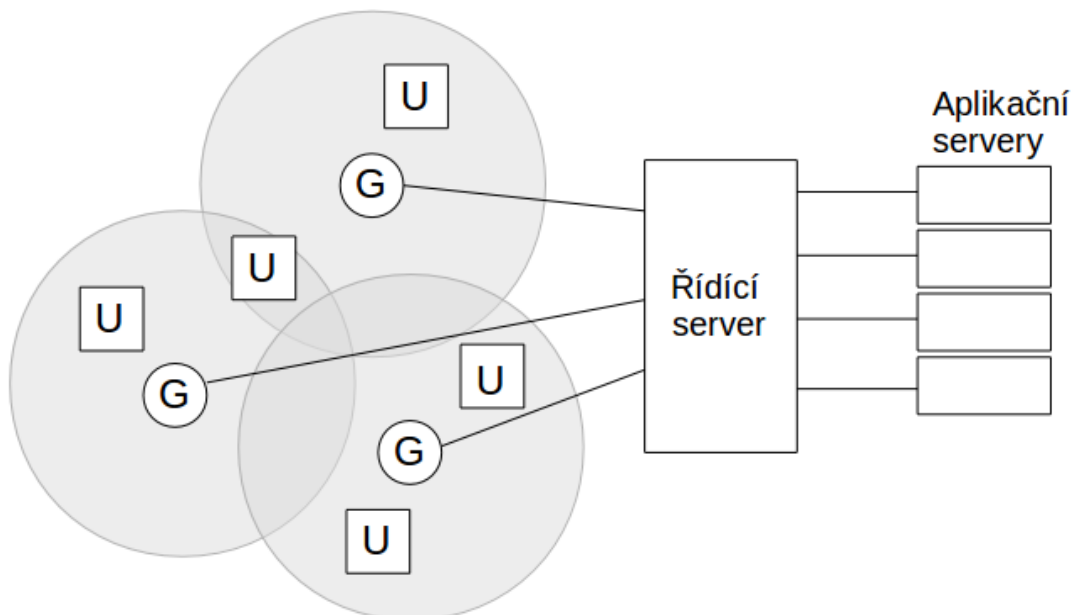
Technologie LoRa využívá radiové komunikace. Byla navržena jak pro evropské pásmo 868 MHz, tak pro to americké 913 MHz.

Rychlost přenosu dat je od 0.3 kbps do 50 kbps. Komunikace mezi branou a koncovým uzlem je umožněna v zástavbě až na vzdálenost 7 km (při přímé viditelnosti až 20 km).

Vzhledem k potenciálu využití v internetu věcí vzniklo sdružení LoRa Alliance s řadou členů, kteří poskytují podporu a zároveň stále rozšiřují síť s LoRa technologií.

3.1 Síťová architektura

Síť se skládá z koncových uzlů, bran, řídicího serveru a aplikačních serverů. Jedná se o jednoduchou hvězdicovitou topologii, kde je středem dění řídicí server. Umožňuje obousměrnou komunikaci a využívá systém potvrzování zpráv. Nepoužívá žádné opakovače, ani složité směrování v síti.



Obrázek 3.1: Síťová architektura pro LoRa technologii. Písmeno U značí koncový uzel, a písmeno G značí gateway, neboli bránu. Středem sítě je řídicí server, který zajišťuje směrování zpráv danému adresátovi.

3.1.1 Koncový uzel

Koncovým uzlem může být jakýkoliv snímač libovolné veličiny pro různá použití, musí být však vybaven radiovou jednotkou, která podporuje komunikaci systému LoRa. Tento uzel poté odesílá data na všechny brány, které jsou v dosahu.

Podle frekvence přijímání dat se rozdělují tři třídy koncových zařízení:

- **Třída A:** Má nejnižší spotřebu. Komunikace je vždy inicializována ze strany koncového uzlu a to vždy na základě nějaké události. Přijímá data pouze do dvou sekund po odvysílání svých dat. Umožňuje pouze zaslání unicastových zpráv. Tento způsob je vhodný pro zařízení napájená z baterií, kde není třeba plná obousměrná komunikace.
- **Třída B:** Tato třída se vyznačuje nižší latencí přijatých zpráv než třída A. Umožňuje příjem dat po odvysílání svých dat, podobně jako třída A, a navíc i v plánových časových intervalech. Jedná se o obousměrnou komunikaci s předem rozvrženými časovými sloty pro příjem zpráv. Server může tedy zahájit komunikaci pouze ve fixních časových intervalech. Je možné posílat unicast i multicast zprávy.
- **Třída C:** Je vhodná tam, kde není třeba řešit nízkou energetickou náročnost. Koncové zařízení je trvale připraveno k přijímání dat, kromě doby vysílání. Server může kdykoliv zahájit komunikaci. Tato třída má tedy nejvyšší proudový odběr. Může vysílat unicast i multicast zprávy.

3.1.2 Brána

Slouží k předávání zpráv mezi koncovými zařízeními a řídicím serverem. Dostává zprávy od všech koncových zařízení, která jsou v jejím dosahu. Brány jsou připojeny k řídicímu serveru prostřednictvím standardních IP připojení. V Evropě existují tři frekvenční kanály ze kterých by měli brány nepřetržitě přijímat (868.10, 868.30, 868.50).

3.1.3 Řídicí server

Díky posílání stejné zprávy více branám je zpráva doručena k řídicímu serveru typicky několikanásobně. Řídicí server vyhodnotí, že mu byla stejná zpráva (podle sekvenčního čísla zprávy) doručena více cestami a odstraní duplicity. Slouží také k nastavení provozních podmínek, kontrole zabezpečení a k předávání dat aplikačnímu serveru. Zajišťuje směrování zpráv určenému adresátovi.

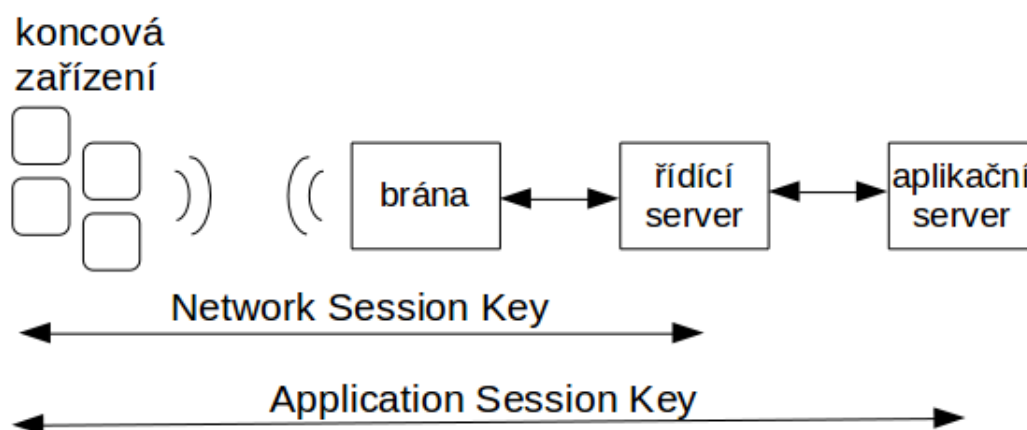
3.1.4 Aplikační server

Úkolem aplikačního serveru je komunikace s řídicím serverem a přidělování oprávnění k používání sítě. Zároveň musí dekodovat naměřené hodnoty z přenesených zpráv, které ukládá v podobě vhodné pro další analytické zpracování. Přijatá data dále zpřístupňuje pomocí standardizovaného rozhraní uživatelským aplikacím. Po analýze dat může vyvolat v uživatelské aplikaci nějakou předem definovanou akci.

3.2 LoRaWAN

Protokol LoRaWAN se zaměřuje na klíčové požadavky týkající se internetu věcí, jako jsou například bezpečná obousměrná komunikace a lokalizační služby. LoRaWAN zajišťuje zabezpečený přenos dat mezi koncovými zařízeními a aplikací běžící na vzdáleném serveru.

Každý koncový uzel má přidělený jedinečný 64 bitový identifikátor (EUI) a dva 128 bitové klíče, kterými jsou data pomocí AES (Advanced Encryption Standard) zašifrována. Jeden klíč se používá v rámci sítě (Network Session Key), druhý k zabezpečení dat mezi sítí a uživatelskou aplikací (Application Session Key). Tyto klíče jsou unikátní pro dané koncové zařízení.



Obrázek 3.2: Model síťové architektury s použitím šifrovacích klíčů.

LoRaWAN Master (řídicí server) přijímá data ze všech bran, vyhodnocuje jejich rádiové vlastnosti a přeposílá je k patřičnému aplikačnímu serveru.

Aby koncové zařízení mohlo komunikovat v LoRaWAN síti, musí být nejprve aktivováno. K tomuto účelu slouží dvě metody:

- **OTAA (Over-the-Air Activation):**

Při aktivaci OTAA je potřeba, aby koncové zařízení vyslalo požadavek o připojení do sítě. Požadavek musí obsahovat jeho unikátní EUI, appEUI dané aplikace a klíč sloužící k autentizaci. Aplikační server musí žádost akceptovat a odeslat Network Session Key a Application Session Key. Tím je aktivace ukončena.

- **ABP (Activation By Personalization):**

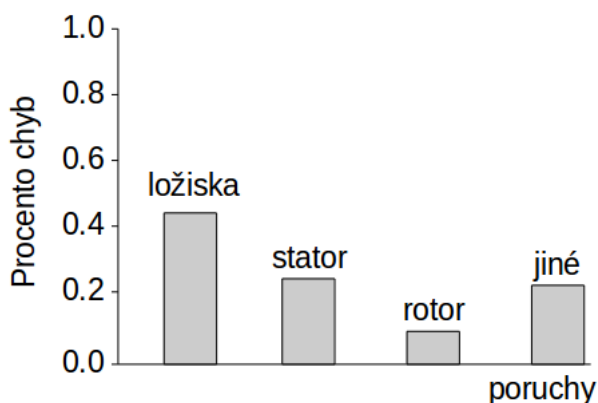
V tomto případě jsou klíče spolu s adresou uzlu vygenerovány předem a uloženy v koncovém zařízení. Zařízení je schopné ihned komunikovat bez dalších procedur.

Aktivace metodou ABP je sice rychlejší, ale je třeba dopředu určit klíče pomocí nichž budou data při přenosu šifrována. Metoda OTAA je flexibilnější, není třeba dopředu generovat a zadávat klíče síti ani koncovému uzlu. Avšak při připojení koncového uzlu do sítě dojde nejprve ke komunikaci sloužící k výměně klíčů. Tato komunikace může inicializaci zařízení v síti značně zpomalit.

4 Poruchy AC motorů

AC motory (motory napájené střídavým proudem) jsou často používané v průmyslových odvětvích, je to pro jejich spolehlivost a robustnost. V těchto odvětvích je velice důležité průběžné monitorování stroje a jeho diagnostika, pokud se dostane do stavu poruchy. Důvody selhání v rotačních elektrických strojích mají svůj původ v konstrukci, výrobní toleranci, montáži, instalaci, prostředí, povaze zatížení a harmonogramu údržby. Problematice poruch strojů, jejich monitorování a diagnostikou se zabývá článek [19].

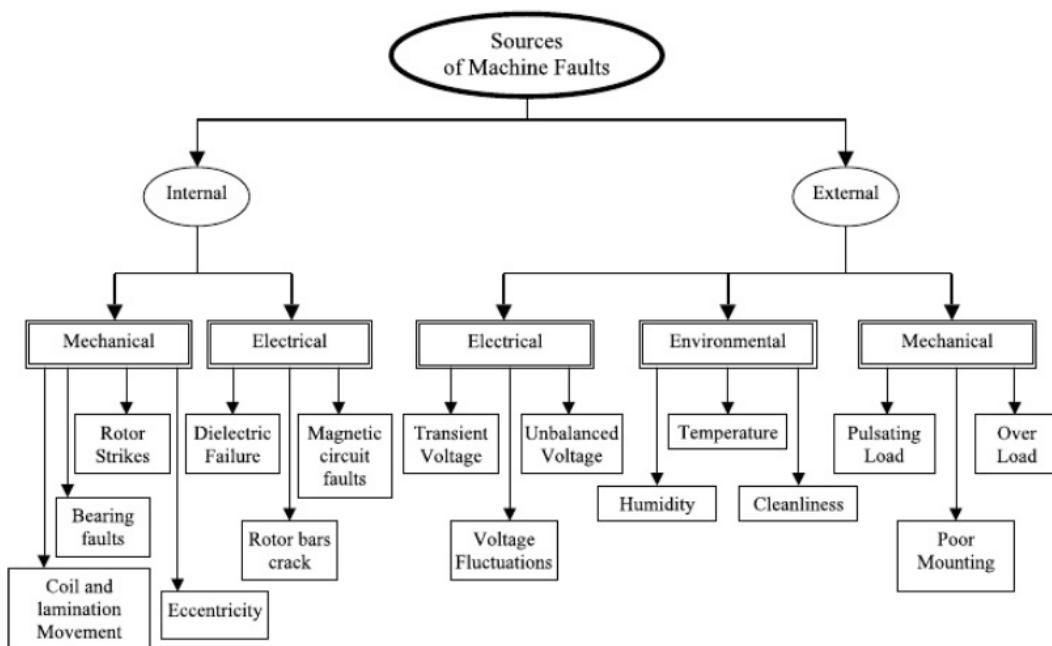
Existují případové studie o selhání strojů, aby bylo možné identifikovat ty nejslabší komponenty v elektromotorech, kde může nastat porucha. Na Obrázku 4.1 jsou vyobrazeny výsledky studie o selhání hlavních komponent motorů, která byla provedena skupinou IEEE-IGA. Studie byla provedena na základě informací, poskytnutých výrobcí motorů. Článek [19] uvádí další dvě studie zabývající se selháním strojů, jedna je od společnosti EPRI a další byla provedena firmou B&K Experts. Porucha ložisek se ukazuje, jako nejčastější porucha AC motorů, podle všech těchto studií.



Obrázek 4.1: Výsledky studie zabývající se poruchami motorů.

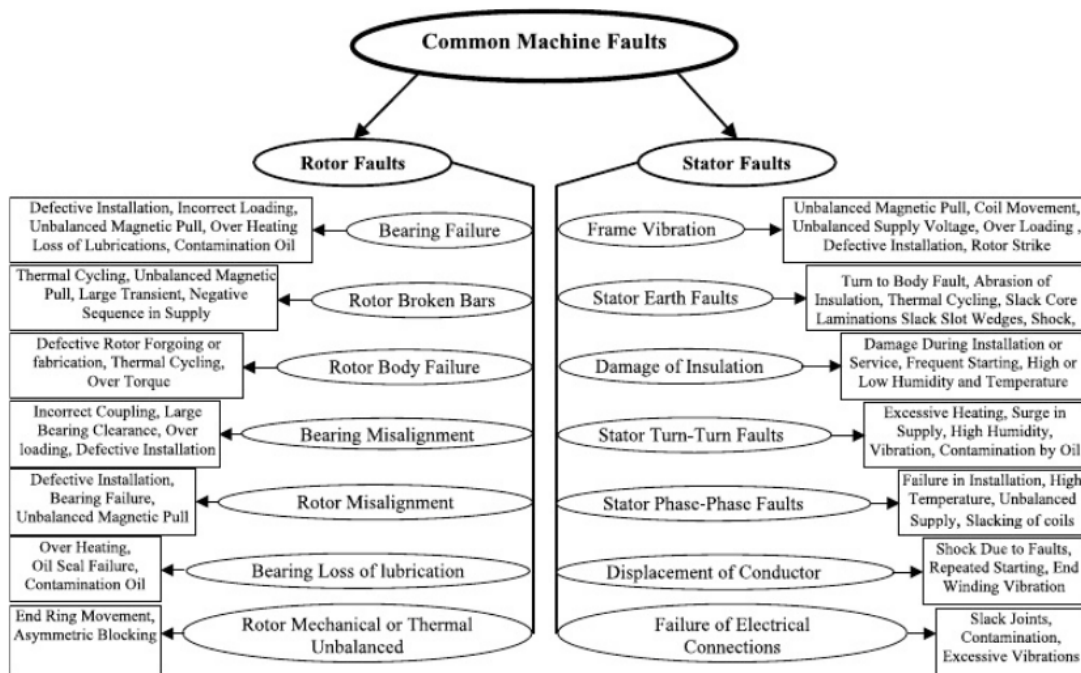
Chyby motoru lze rozdělit na dva typy: mechanické a elektrické. Častou mechanickou závadou je nesouosost, což je nesprávné usazení motoru vzhledem k poháněnému stroji. Nesouosost vyvolává třecí a deformační síly, které deformují ložiskový uzel a samotný motor. Další častou závadou je excentricita, kdy je rotor vychýlen ze středu motoru. Co se elektrických závad týče, k nejčastějším poruchám dochází na statorovém vinutí. [20]

Obecně platí, že AC motor je vystaven primárním typům poruchy a souvisejícím sekundárním poruchám. Zdroje poruch motoru mohou být interní, nebo externí, jak je znázorněno na Obrázku 4.2. Vnitřní poruchy mohou být klasifikovány s ohledem na jejich původ, tj. elektrické a mechanické, nebo jejich umístění, tj. stator a rotor. Vnější poruchy se rozdělují na elektrické, mechanické a na poruchy způsobené vnějším prostředím, ve kterém daný stroj pracuje. Na Obrázku 4.3 je znázorněna klasifikace poruch podle jejich polohy: stator a rotor.



Obrázek 4.2: Zdroje poruch u AC motorů a jejich rozdělení. Obrázek je převzat z [19]

Všechna rotační zařízení vytvářejí vibrace. Dokonce i nové nebo zdravé stroje generují určitou úroveň vibrací, avšak vyšší úrovně vibrací a rostoucí charakter jsou symptomy abnormálního výkonu stroje. Po dobu provozu strojů dochází k deformaci a poškození součástí, což způsobuje změnu dynamických vlastností a případně zvyšují úroveň vibrací stroje. Analýza vibrací je tedy nejčastěji používaný způsob pro diagnostiku motorů a případnou detekci poruchy.



Obrázek 4.3: Nejčastější poruchy a jejich příčiny ve statoru a rotoru. Obrázek je převzat z [19]

5 Zpracování dat z akcelerometru a výpočet příznaků

Pro posouzení stavu monitorovaného zařízení (motoru) je vhodné použít číselné charakteristiky, tzv. příznaky, získané ze signálů vibrací. Signály vibrací mají bohatý informační obsah a pro jeho popis je potřeba vybrat příznaky umožňující detekci poruch nebo degradací. Mezi nejčastěji používané patří příznaky v oblasti amplitudové (efektivní hodnota, crest factor) a frekvenční (spektrum). V této práci jsou použity i některé novější metody z obou oblastí: kurtosis ratio v amplitudové a kurtogram ve frekvenční.

Digitální data jsou získávána z akcelerometru ADXL1002Z, který byl připojen přes analogově-digitální převodník přímo k vývojovému kitu P-NUCLEO-LRWAN1.

Z naměřeného signálu je počítána efektivní hodnota, která se zvyšuje se zvyšujícími se otáčkami motoru, čili lze zpětně odhadnout počet otáček měřeného motoru. Obecně efektivní hodnota roste se zhoršujícím se stavem zařízení.

Dalším počítaným příznakem je crest factor [1], který měří impulsivnost naměřeného signálu. V případě poruchy, způsobené trhlinou v některé části ložiska, dochází v poškozeném místě k opakovaným nárazům během otáčení. Z tohoto důvodu bylo měření impulzivnosti signálu používáno po mnoho let jako indikátor závažnosti poruch ložisek.

Jelikož nejčastější závadou průmyslových AC motorů je porucha ložiska, byl naimplementován výpočet kurtogramu, přesněji fast kurtogramu. Kurtogram je schopný detekovat poruchy stroje, které se projevují výskytem nestacionární složky. Princip fast kurtogramu a samotný algoritmus jsou blíže popsány v [21].

Poslední počítaný parametr je kurtosis ratio. Tato hodnota vyjadřuje množství odchylek v datech. Princip a postup výpočtu kurtosis ratio je popsán v článku [22].

Všechny výpočty jsou prováděny přímo na vývojovém kitu a prostřednictvím LoRa technologie se do sítě ThingsNetwork posílají pouze vypočtené parametry naměřeného signálu.

5.1 RMS (root mean square)

Efektivní hodnota, neboli RMS je statistická veličina představující druhou odmocninu aritmetického průměru druhých mocnin daných hodnot. Je často označována jako kvadratický průměr.

Efektivní hodnota je vždy nezáporná a větší nebo rovna aritmetickému průměru. Umocnění hodnot na druhou má za následek větší váhu hodnot vzdálenějších od nuly. Je definována vztahem:

$$x_{rms} = \sqrt{\frac{1}{n} \cdot (x_1^2 + x_2^2 + \dots + x_n^2)}$$

RMS nám umožňuje určit amplitudu sledovaného signálu. Hodnota se zvyšuje se zvyšujícími se otáčkami stroje.

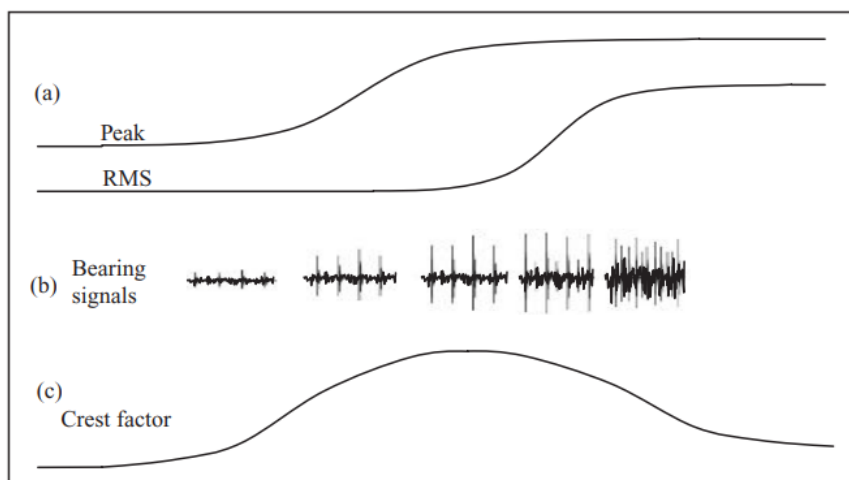
RMS se často používá v souvislosti se střídavým napětím nebo proudem. Efektivní hodnota střídavého napětí/proudu je rovna hodnotě stejnosměrného napětí/proudu, který by dával stejný průměrný výkon.

5.2 Crest factor

Porouchané ložisko typicky generuje impulsní složky v signálu, které jsou často způsobené ostrými nárazy. Crest faktor měří impulsivnost signálu. Je definován jako poměr maximální absolutní hodnoty signálu $|x_{peak}|$ ku efektivní hodnotě x_{rms} .

$$C = \frac{|x_{peak}|}{x_{rms}}$$

Problémem Crest faktoru je, že maximální hodnota není příliš stabilní a značně závisí na části analyzovaného signálu. Zvýšení hodnoty $|x_{peak}|$ beze změny efektivní hodnoty analyzovaného signálu vede ke zvýšení crest faktoru. Následné zvýšení RMS naopak vede ke snížení crest faktoru. Tento průběh je vidět na Obrázku 5.1.



Obrázek 5.1: Průběh Crest faktoru vzhledem k efektivní hodnotě a maximální absolutní hodnotě signálu. Obrázek převzat z [1].

5.3 Kurtogram

Kurtogram, se ukázal jako velmi silný a praktický nástroj při diagnostice poruch strojů. Je schopný detekovat nestacionární signál o nízké amplitudě, který je nezřetelný kvůli šumu. Díky tomu je schopný detekovat poruchy stroje, které se projevují výskytem nestacionární složky, v jejím počátku a předejít tak případným škodám. Výstupem kurtogramu je frekvence, na které se objevuje nestacionární signál a šířka pásma kurtogramu, díky tomu se dá snadněji lokalizovat zdroj poruchy.

Kurtogram je nástroj pro spektrální analýzu čtvrtého řádu, který je zpracován pro detekci a charakterizaci nestacionárních složek v signálu. Myšlenkou kurtogramu je výpočet spectral kurtosis pro každé frekvenční pásmo, abychom zjistili přítomnost skrytých nestacionárních částí a ukázali, ve kterých kmitočtových pásmech k nim dochází.

Stacionární signál má na rozdíl od nestacionárního v čase neměnné statistické vlastnosti (střední hodnotu, distribuční funkci, rozptyl atd.). Pokud se zvolí dostatečně dlouhý časový interval měření signálu, budou pro stacionární signál vypočteny stejné vlastnosti bez ohledu na posunutí na časové ose. Pokud je zvolen příliš malý časový interval, bude se i nestacionární signál zdát stacionární. Proto je u rozpoznání, zda je signál stacionární, či nikoliv, důležitý výběr dostatečně velkého časového intervalu.

5.3.1 Spectral kurtosis (spektrální koeficient špičatosti)

Tento pojem je používán ve statistice a teorii pravděpodobnosti. Jedná se o charakteristiku rozdělení náhodné veličiny, která porovnává dané rozdělení s normálním rozdělením pravděpodobnosti. Obecně je definován vztahem:

$$K[x] = \frac{\mu_4}{\rho^4} - 3,$$

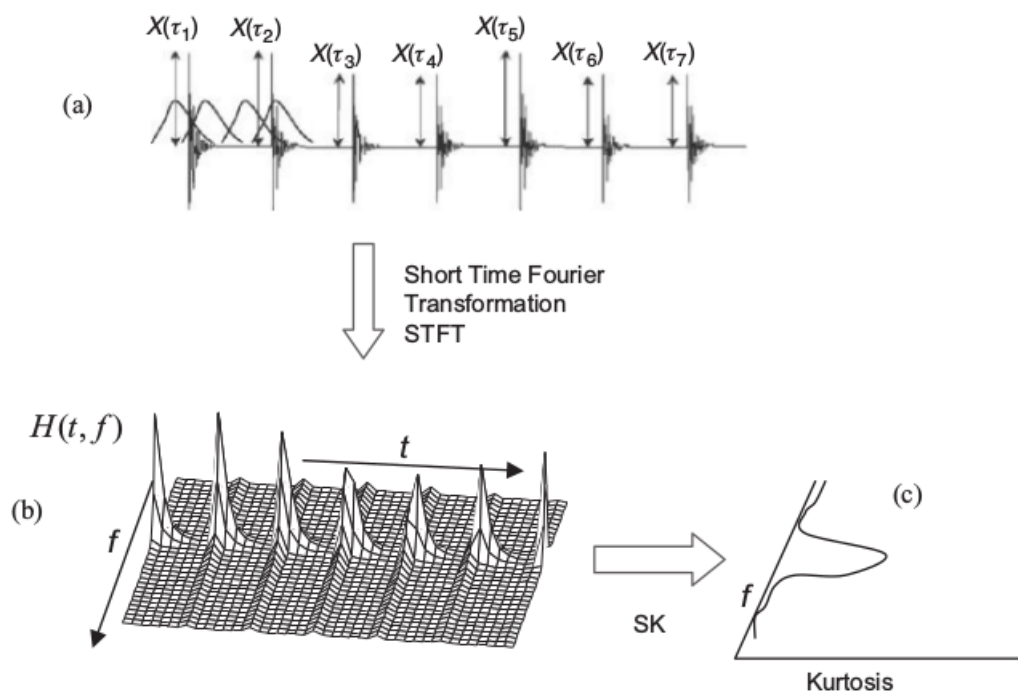
kde μ_4 je čtvrtý centrální moment a ρ je směrodatná odchylka.

Zdroje [1] a [21] udávají, že kurtosis, pro každou frekvenci f , může být vypočtena pomocí čtvrté mocniny $H(n, f)$ a následnou normalizací druhou mocninou průměru kvadrátů:

$$K_x(f) = \frac{\langle |H(n, f)|^4 \rangle}{\langle |H(n, f)|^2 \rangle^2} - 2,$$

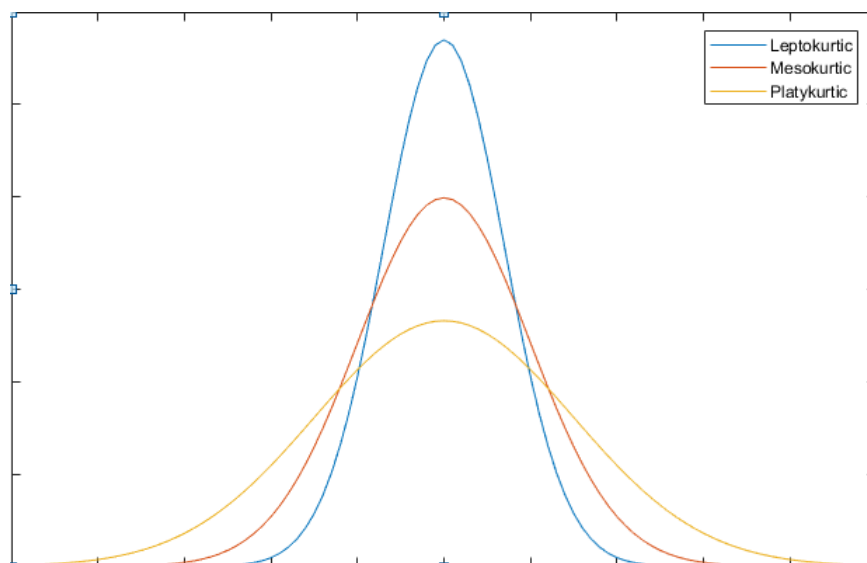
kde $\langle f(n) \rangle = \lim_{n \rightarrow \infty} N^{-1} \sum_N f(n)$ znamená časový průměr funkce $f(n)$. STFT získaná přesunem časového okna podél záznamu je zde reprezentovaná jako funkce amplitudové obálky $H(n, f)$. Konstanta -2 (namísto konstanty -3 , použité obvykle u výpočtu kurtosis) pochází ze skutečnosti, že $H(n, f)$ je komplexní.

STFT - krátkodobá Fourierova transformace je Fourierova transformace aplikovaná na funkci postupně po krátkých úsecích, které vybírá pomocí časového okna. Tím řeší problém souběžného určení času i frekvence, na kterých je rozmístěna energie signálu. Tato transformace tedy provádí časově-frekvenční analýzu.



Obrázek 5.2: Výpočet spectral kurtosis ze STFT pro simulovaný signál poruchy ložiska: a) simulovaný časový signál; b) STFT; (c) spectral kurtosis jako funkce frekvence. Převzato z [1].

Normální rozdělení (Mesokurtic) má kurtosis 0. Záporná kurtosis (Platykurtic) značí, že rozdělení je rovnoměrnější a jeho křivka hustoty je plošší nežli u normálního rozdělení. Kladná kurtosis (Leptokurtic) značí, že většina hodnot náhodné veličiny leží blízko její střední hodnoty. Křivka hustoty je špičatější, nežli u normálního rozdělení.



Obrázek 5.3: Příklady křivek pro různé tvary rozdělení: Mesokurtic, Platykurtic a Leptokurtic.

5.3.2 Dvojice frekvence a šířka pásma

Pro každý nestacionární signál existuje funkce SK (spectral kurtosis), která závisí jak na frekvenci, tak na šířce pásma. Spectral kurtosis velmi nestacionárních procesů silně závisí na volbě šířce pásma. Nekonečná šířka vede k nulovému SK. Na druhou stranu příliš úzká šířka pásma nemusí umožňovat detekci úzkopásmového kmitočtu schovaného v aditivním šumu.

Může existovat kombinace frekvence f a šířka pásma Δf , která maximalizuje SK. To vede ke konceptu dvojice frekvence a šířka pásma.

5.4 Fast kurtogram

Prozkoumání celé roviny $(f, \Delta f)$ je náročný úkol, který se těžko implementuje v embedded zařízeních. Jelikož se jedná o výpočetně složitou záležitost, zavádí [21] tzv. fast kurtogram, jehož složitost je $O(N \log N)$, kde N počet vzorků signálu. Jedná se o rychlý algoritmus pracující nad stejnou rovinou $(f, \Delta f)$.

Je dokázáno [21], že výpočet kurtogramu se zjednodušuje na výpočet kurtosis na výstupu filtrů FIR.

5.4.1 Filtry FIR (finite impulse response) a IIR (infinite impulse response)

FIR filtr je číslicový filtr s konečnou impulsní odezvou. Je vždy stabilní a má lineární fázi. Pro dosažení veliké strmosti přechodu mezi propustným a nepropustným pásmem je třeba zvolit vysoký řád filtru. Nevýhodou je nárůst zpoždění právě pro velké řády.

IIR filtr, je číslicový filtr s nekonečnou impulzní odezvou. Na rozdíl od FIR filtru nemusí být vždy stabilní a požadovanou strmost lze dosáhnout při nižším řádu nežli u FIR filtru.

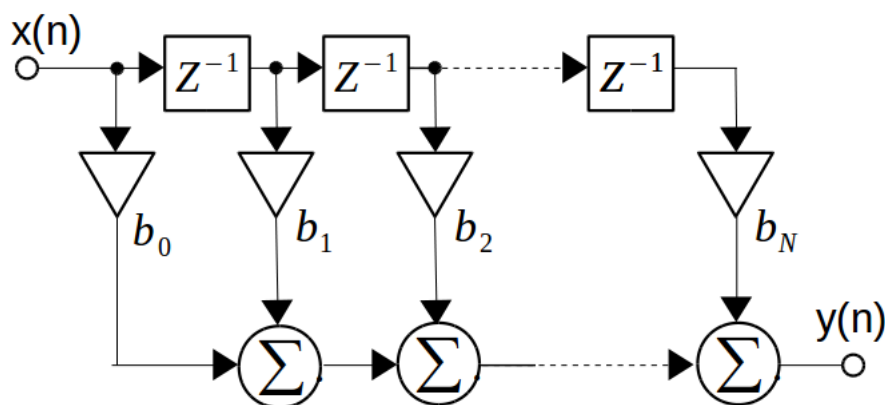
Obecný přenos filtrů FIR a IIR vyjádřený v z-transformaci je definován jako:

$$h(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n}}{1 - a_1z^{-1} - a_2z^{-2} - \dots - a_mz^{-m}}.$$

Pokud jsou $a_i \neq 0$, pak se jedná o IIR filtr. Jsou-li koeficienty a_i nulové, pak se jedná o FIR filtr a výpočet se zjednoduší pouze na:

$$h(z) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n}.$$

Výpočet filtrovaného signálu $y(n)$, získaného ze signálu $x(n)$ a aplikováním filtru FIR je znázorněn na Obrázku 5.4.



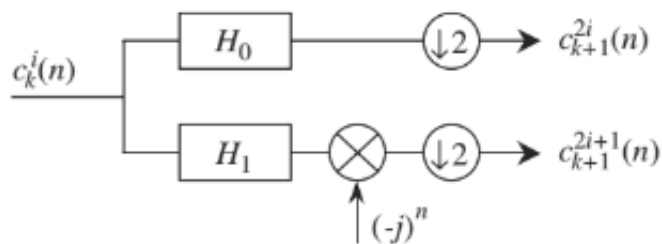
Obrázek 5.4: Struktura číslicového filtru typu FIR - přímá forma.

5.4.2 Algoritmus

Princip navrženého algoritmu pro fast kurtogram je založen na stromově uspořádané vícenásobné filtrační struktuře.

Mějme dva filtry, jeden typu dolní propust $h_0(n)$ a jeden typu horní propust $h_1(n)$ ve frekvenčních pásmech $[0; \frac{1}{4}]$ a $[\frac{1}{4}; \frac{1}{2}]$. Filtry $h_0(n)$ a $h_1(n)$ se používají k provedení elementárního rozkladu, který se opakuje pyramidálním způsobem; tak je vytvořena stromová struktura.

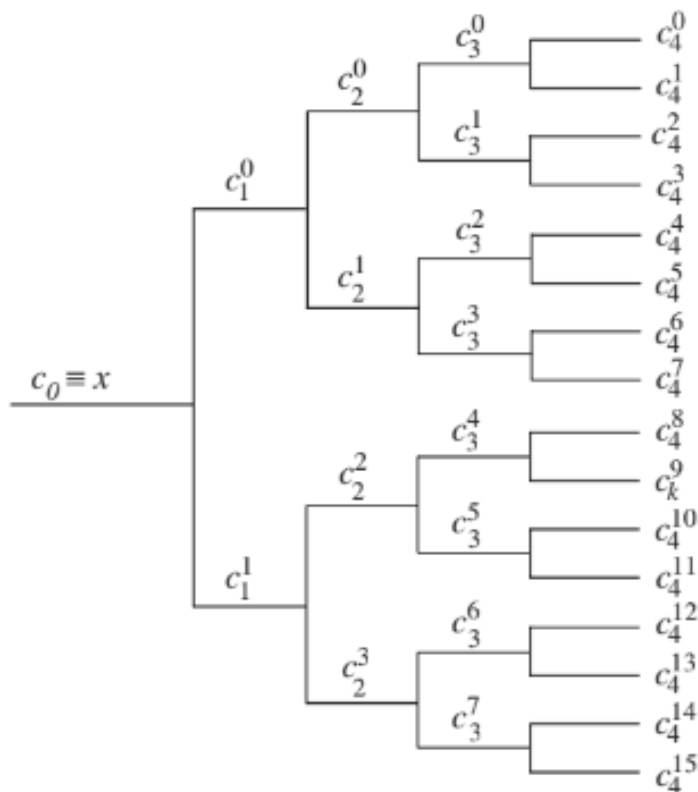
Nechť $c_k^i(n)$ je sekvence koeficientů získaných z i -tého filtru dané hladiny, $i = 0, \dots, 2^k - 1$ a k -té hladiny v dekompozičním stromě. Po průchodu filtry $h_0(n)$ a $h_1(n)$ a odstranění každého druhého prvku (down-sampling) se sekvence rozdělí na dvě nové sekvence $c_{k+1}^{2i}(n)$ a $c_{k+1}^{2i+1}(n)$ na levelu $k + 1$, jak je znázorněno na Obrázku 5.5.



Obrázek 5.5: Elementární rozklad přes filtry typu horní a dolní propust. Obrázek je převzat z [21].

Násobení $(-j)^n$ po filtraci pomocí $h_1(n)$ filtru, před odebráním vzorků (down-sampling) se děje proto, aby se sekvence s horní propustí obrátily do sekvencí s dolní propustí, a tím respektovaly uspořádání frekvence.

Toto je prováděno od hladiny $k = 0$, kdy $c_0(n) = x(n)$ až do hladiny $K-1$, kde K je maximální předem daná úroveň dekompozičního stromu. Na každé hladině se počet filtrovaných sekvencí zdvojnásobí, ale jejich délka se po každé o polovinu zmenší, takže se vždy počítá stejné množství dat. Schéma výpočtu a vytvoření stromové struktury je znázorněno na Obrázku 5.6.



Obrázek 5.6: Výpočet fast kurtogramu vytvářející schéma stromové struktury. Obrázek je převzat z [21].

Centrální frekvence $c_k^i(n)$ se dá vypočítat jako:

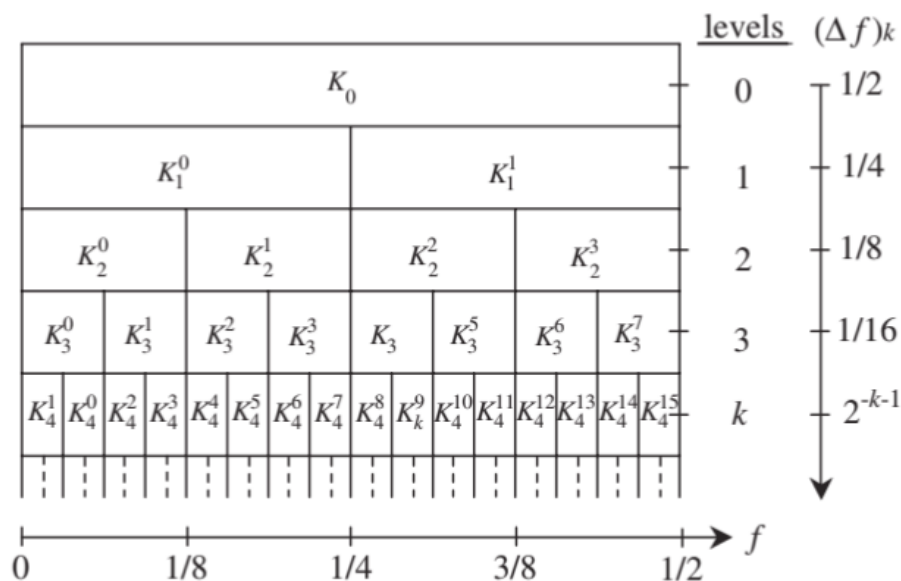
$$f_i = (i + 2^{-1}) \cdot 2^{-k-1}.$$

Šířka pásma k -tého levelu je dána vztahem:

$$(\Delta f)_k = 2^{-k-1}.$$

Nakonec je kurtogram počítán jako kurtosis všech sekvencí $c_k^i(n)$, kde $i = 0, \dots, 2^k - 1$ a $k = 0, \dots, K - 1$:

$$K_k^i = \frac{\langle |c_k^i(n)|^4 \rangle}{\langle |c_k^i(n)|^2 \rangle^2} - 2.$$



Obrázek 5.7: Repräsentace binárního kurtogramu v $(f, \Delta f)$ rovině.

Jak dále dokládají v [21] tento algoritmus je rychlejší a přesnější než například použití STFT (The short-time Fourier transformation).

Existuje rozšíření tohoto algoritmu na 1/3-binární strom, který dosahuje jemnějšího vzorkování roviny $(f, \Delta f)$ se zanedbatelným zpožděním. Avšak pro naše účely postačí pouze binární fast kurtogram.

5.4.3 Ukázka detekce vady ložiska pomocí kurtogramu

Tento příklad ilustruje použití výpočtu fast kurtogramu pro diagnostiku ložisek v motorech a případnou detekci poruchy. Zároveň naznačuje jednotlivé výpočty a předkládá význam jednotlivých výsledků.

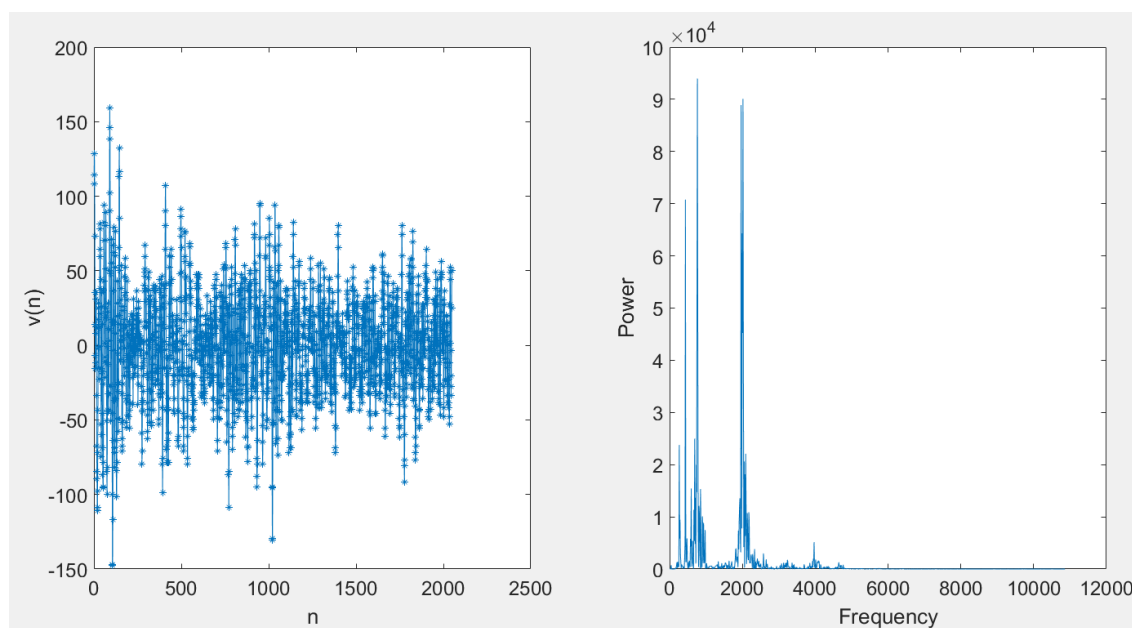
Data pro účely prezentace kurtogramu a jednotlivých výpočtů byla naměřena na simulátoru vibrací rotačních mechanismů. Jelikož nejčastější porucha na motoru je v oblasti ložisek, bylo první měření provedeno na ložisku s poruchou. Druhé měření bylo provedeno na jiném motoru, bez jakékoliv závady, taktéž na ložisku.

Z těchto dat byl bez jakýchkoliv dalších filtrací vypočten binární fast kurtogram. Vzorovací frekvence byla přibližně 22 KHz a maximální dekompoziční level kurtogramu byl nastaven na $K = 4$. U obou dvou měření se simulátor pohyboval rychlostí přibližně 2 200 otáček za minutu.

Data byla získána z akcelerometru přidělaného na jedno z ložisek. Následně byla data vyčítána přes UART rozhraní, ukládána v počítači a později analyzována pomocí fast kurtogramu v programu Matlab.

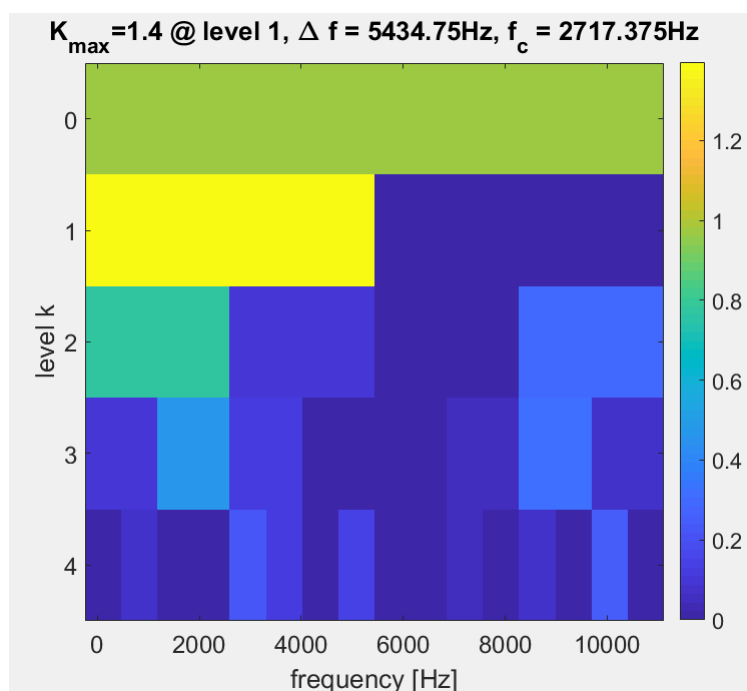
Ložisko bez poruchy

Nejprve bylo provedeno měření na motoru bez jakýchkoliv známek poruchy. Přesněji bylo provedeno na prvním ložisku. Na Obrázku 5.8 je zaznamenán průběh znormovaného signálu a frekvenční spektrum naměřených dat.



Obrázek 5.8: Normalizovaná naměřená data na ložisku bez poruchy a spektrální analýza naměřeného vibračního signálu.

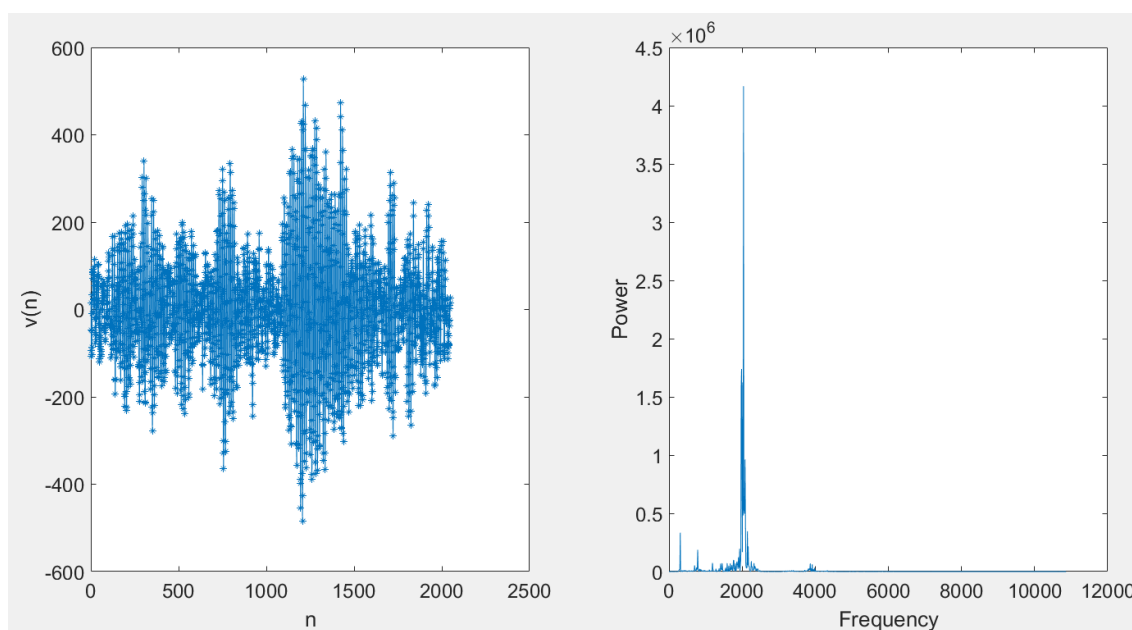
Pokud se v naměřených datech nevyskytuje žádný nestacionární signál, pak kurtogram i tak detekuje maximum na nějaké frekvenci s danou šířkou pásma. Toto maximum může být na jakékoliv hladině dekompozičního stromu. Použití samotného kurtogramu nám tedy nijak nepomáhá k určení, zda se na měřeném motoru vyskytuje nějaká vada, či nikoliv. Na Obrázku 5.9 je vykreslen vypočtený kurtogram z naměřených dat.



Obrázek 5.9: Vypočtený fast kurtogram z naměřených dat získaných z akcelerometru připojeného na ložiskem, které je součástí motoru bez vady. Kurtogram má dekompozičním levelem $K = 4$. Maximum se nachází na hladině 1 a indexu $i = 0$, čili dvojice frekvence a šířka pásma je $\{f_0; \Delta f_1\} = \{2717.375; 5434.75\}$.

Porouchané ložisko

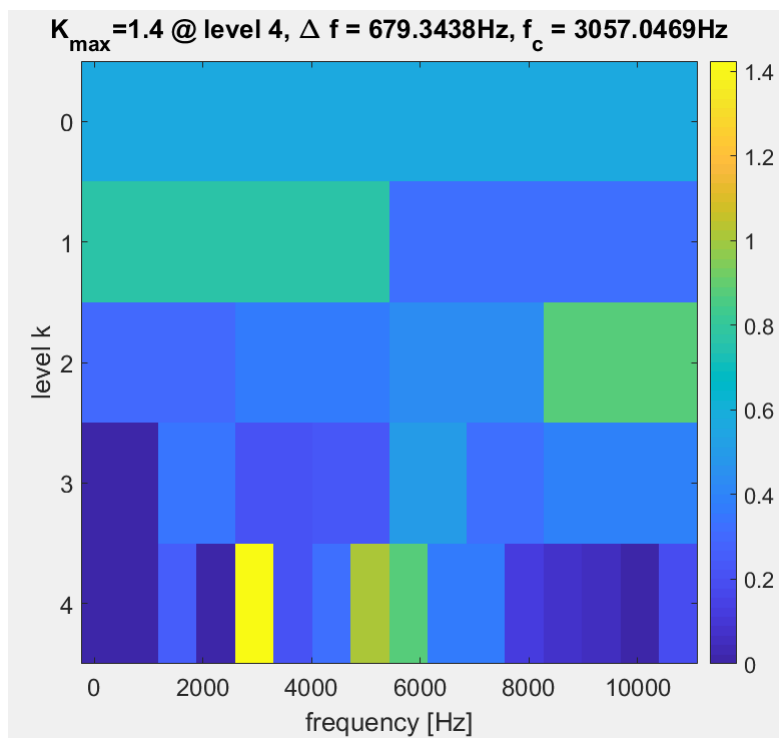
Data použitá v následujícím příkladu byla sebrána z motoru obsahující ložisko se slyšitelnou poruchou.



Obrázek 5.10: Naměřená data vibračního signálu po normalizaci na ložisku s poruchou a spektrální analýza tohoto signálu.

Použití samotného kurtogramu dává smysl v případě, že již víme, že je nějaká porucha přítomna a kurtogram nám tedy poslouží k její detekci, což je tento případ.

Pokud se v signálu vyskytuje nestacionární chování, pak kurtogram detekuje maximum při užší šířce pásma, čili ve vyšší hladině dekompozičního stromu.



Obrázek 5.11: Vypočtený fast kurtogram daného signálu z ložiska s poruchou. Maximum 1.4 je zachyceno na dvojici $\{f_4; \Delta f_4\} = \{3057.0469; 679.3438\}$.

Vypočtený kurtogram z naměřených dat (viz Obrázek 5.11) poukazuje na nestacionární aktivitu na úrovni 4 a indexu $i = 4$, kde je hodnota kurtogramu nejvyšší, a to 1,4. Tudiž $\Delta f = 2^{-k-1} = 2^{-5} \cdot f_{vz} = 679.3438$ Hz. Centrální frekvence je $f_i = (i + 2^{-1}) \cdot 2^{-k-1} = 4.5 \cdot 679.3438 = 3057.0469$ Hz, čili kolem této frekvence se projevuje nestacionární chování. Maximální kurtosis je tedy na dvojici $\{f_4; \Delta f_4\} = \{3057.0469; 679.3438\}$.

Tyto hodnoty kurtogramu by však více vypovídali pokud by společně s kurtogramem byly vypočteny i jiné příznaky, jako například efektivní hodnota, nebo crest factor. Samotné výsledky kurtogramu nic neříkají o tom, zda monitorovaný motor obsahuje poruchu, či nikoliv, ani o tom, zda se motor v době měření pohyboval nenulovou rychlostí. Pokud víme, že je porucha přítomna pomůže nám výpočet kurtogramu k její detekci.

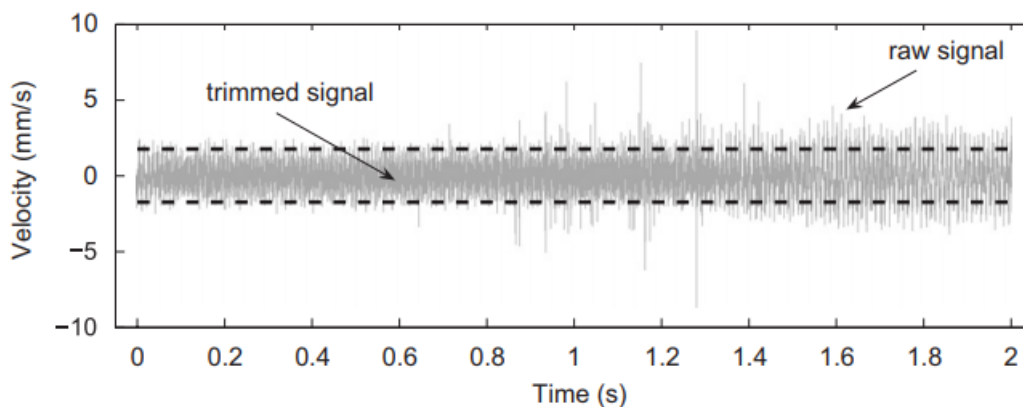
5.5 Kurtosis ratio

Jedná se o skalární vyjádření množství odchylek v datech. Informace o výpočtu kurtosis ratio byly čerpány z [22]. Kurtosis ratio je definováno jako:

$$KR(x) = \frac{K(x)}{K(x_1)},$$

kde x je naměřený signál, x_1 je oříznutý naměřený signál a $K(x), K(x_1)$ je výpočet kurtosis daného signálu.

Oříznutý signál x_1 se získá od x odstraněním vychýlených vzorků. Výsledkem toho je, že kurtosis zredukovaného signálu $K(x_1)$ není ovlivněna přítomností vychýlených hodnot a představuje tak silný odhad kurtosis.



Obrázek 5.12: Původní naměřený signál a z něj, odstraněním vychýlených hodnot, získaný zredukovaný signál. Převzato z [22]

Zredukovaný signál x_1 se nejjednodušeji získá tak, že se naměřené hodnoty seřadí a poté se ze signálu odstaní 5 - 10 % největších a nejmenších vzorků.

Pokud se v signálu nevyskytují výrazně vychýlené hodnoty jsou kurtosis $K(x)$, $K(x_1)$ srovnatelné, tudíž $KR(x) \approx 1$. Naopak $KR(x)$ pro zkreslený signál je mnohem větší než jedna, protože odlehlé hodnoty zvyšují pouze $K(x)$, zatímco robustní odhad $K(x_1)$ není ovlivněn.

6 Softwarové zpracování

6.1 Měřicí a vyhodnocovací jednotka

Kontrolní monitorovací systém byl naprogramován v jazyce C. Pro testovací a vizualizační účely byl kurtogram nejprve naimplementován v programu Matlab a poté přepsán do jazyka C a doimplementován do monitorovacího systému.

Celý projekt je napsán v AC6: SW4STM32, což je jedno z vývojových prostředí doporučených v katalogu výrobce použitého vývojového kitu, založené na programu Eclipse.

Pro jednodušší implementaci byli použity HALové knihovny a vzorové příklady ze stránek STMicroelectronics [23].

Pro ladící účely je naprogramován pomocný výpis přes rozhraní UART, který vypisuje parametry LoRa komunikace a informace o měřených datech z akcelerometru na terminál připojeného počítače pomocí sériového rozhraní. Dále umožňuje výpis jednotlivých chybových hlášení, která slouží pro diagnostiku monitorovacího systému.

Monitorovací zařízení odesílá data do sítě ThingsNetwork každých pět minut. Vždy deset sekund po odeslání dat se procesor probudí a naměří data z akcelerometru. Po uložení a zpracování dat se procesor opět uspí až do doby odesílání dat. Po odeslání vypočtených hodnot se procesor znovu uspí.

6.1.1 Nastavení periférií

Systémové hodiny monitorovacího systému jsou nastaveny na 32 MHz.

Pro přenos dat z akcelerometru slouží periférie ADC1 (analog to digital converter) zabudovaná přímo na vývojovém kitu. Periférie je inicializovaná s 12 bitovým rozlišením a data zarovnáva doprava. Pro účely ADC1 periférie byl nakonfigurován pin A1 jako analogový vstup.

Pro správné určení vzorkovací frekvence a snadné vzorkování je použit DMA přenos, který se stará o přenos dat z ADC periférie do paměti bez účasti procesoru. Není nastaven do cyklického módu, jelikož opakování dma přenosu je řízeno timerem. Naměřené data se ukládají jako 16 bitový unsigned integer.

Pomocí DMA přenosu se navzorkuje přesně 2048 dat (jedná se o mocninu dvou, tudíž se data úsporně zarovnají do paměti). Výrazně větší množství dat z akcelerometru by se nevešlo do paměti RAM.

Rozhraní UART používá pro přenos 8 datových bitů a jeden stop bit. Není nastavena žádná paritní kontrola. Baud rate je nastaven na 115200. Je nakonfigurován pouze pro přenos dat z monitorovacího zařízení přes sériovou linku do připojeného počítače, tudíž monitorovací systém neumožňuje přijímat zprávy z terminálu připojeného počítače.

Pro přesné časování sběru dat, je použit timer, přesněji je nakonfigurována periférie TIM2.

DMA přenos má největší prioritu (priorita 0). Stejně tak UART pro ladící účely má prioritu 0, je to z toho důvodu, aby se při chybě, nebo nečekaném chování vypsala na terminálu daná chybová hláška. Pokud by měl výpis menší prioritu mohl by být přerušen interruptem s vyšší prioritou a ladící výpis by nefungoval. O jedna menší prioritu (1) má

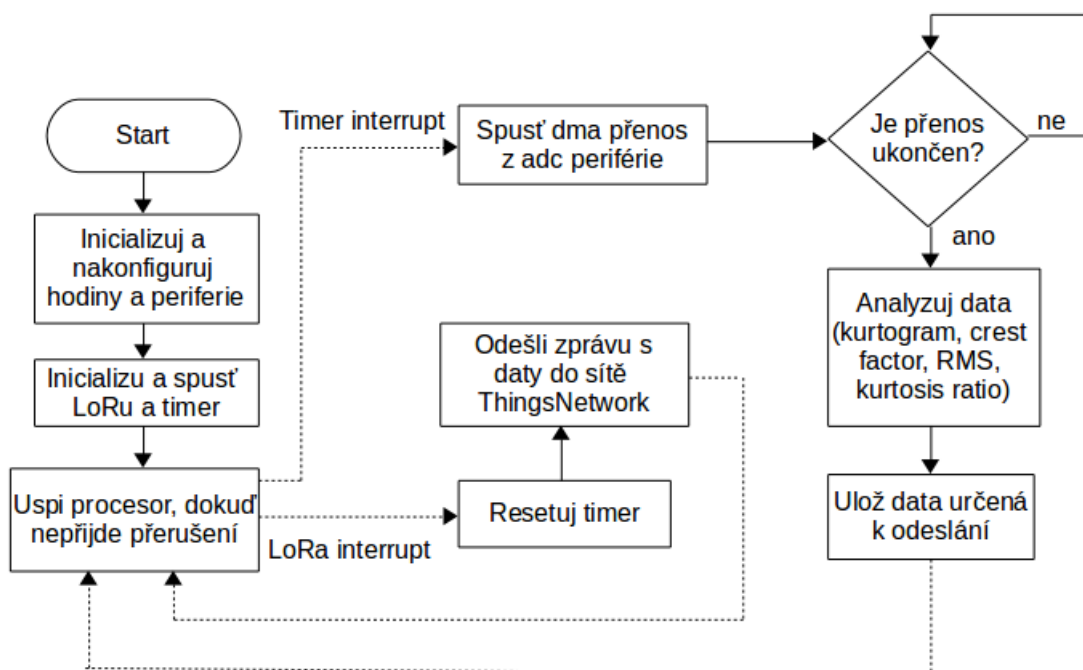
RTC interrupt, který slouží pro časování LoRa komunikace. Zároveň má prioritu 1 timer, který slouží k časování sběru dat. Kdyby měl timer větší prioritu než DMA přenos, pak by nikdy obsluha timeru nebyla přerušena interruptem indikující konec DMA přenosu, což je v našem případě nežádoucí.

6.1.2 Hlavní smyčka programu

Program začíná funkcí `int main(void)`. Na začátku se nainicializují a nakonfigurují hodiny a jednotlivé periférie. Dále se inicializuje LoRa Stack a periférie pro komunikaci s připojeným modulem pro rozhraní LoRa. Ještě před vstupem do programové smyčky se inicializuje, nastaví a spustí timer.

Popis hlavní programové smyčky spolu s jednotlivými přerušeními je zobrazen na Obrázku 6.1.

Timer slouží k časování sběru dat z akcelerometru. Pokud přijde přerušení od timeru, vyvolá se přerušovací rutina `static void MeasurementStartTimerIrq()`. V této funkci se nejprve spustí DMA přenos z ADC periférie, tím jsou navzorkována data z akcelerometru. Program ve smyčce čeká, než se přenos dokončí, což signalizuje interrupt a vyvolání handleru `static void done_cb(DMA_HandleTypeDef* dh)`. Po dokončení přenosu se data zpracují, spočítá se kurtogram, efektivní hodnota, crest factor a kurtosis ratio.



Obrázek 6.1: Vývojový diagram popisující základní funkčnost programu a obsluhu jednotlivých přerušení.

Při přerušení od LoRa se nejprve resetuje timer pro vzorkování dat z akcelerometru, což slouží k přesnému časování vzorkování, poté se odešlou určená data přes LoRa rozhraní. Podrobnější popis implementace LoRa rozhraní je v Sekci 6.1.7.

Po počáteční inicializaci a mezi jednotlivými přerušeními od timeru a od LoRa se procesor uspí pro účely menšího odběru energie.

6.1.3 Efektivní hodnota a Crest factor

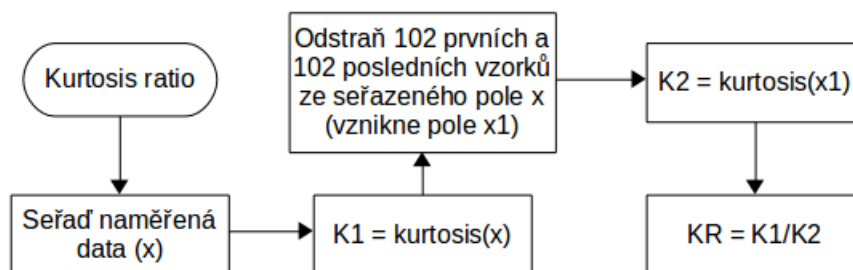
Data z akcelerometru jsou před jakýmkoliv výpočtem nejprve normalizována. Naměřené hodnoty jsou čísla z analog-digital převodníku, čili jsou ovlivněny rozlišením použitého ADC. V případě, že akcelerometr naměří nulové g , je na výstupu (podle datasheetu výrobce) polovina napájecího napětí, což při 12 bitovém rozlišení ADC znamená hodnotu 2048.

Pokud jsou data normalizována, zbaví je to stejnosměrné složky. O to se stará funkce `void normalization()`, která nejprve spočítá průměr z naměřených dat a následně od každého vzorku vypočtený průměr odečte.

Výpočet RMS a crest factoru je realizován ve funkci `void compute_crest()`. Nejprve nad naměřenými daty spočte RMS, následně je určen maximální absolutní vzorek a z těchto dvou hodnot je vypočten crest factor. Tato funkce používá pro dílčí výpočty funkce s knihovny `arm_math.h`, což je volně dostupná matematická knihovna pro práci s daty v procesorech ARM.

6.1.4 Kurtosis ratio

Tento údaj zajišťuje funkce `void kurtosis_ratio()`, která částečně setřídí pole naměřených dat, vypočte kurtosis z navzorkovaného signálu pomocí funkce `float kurtosis(uint32_t *x_real, uint32_t *x_imag, int size, int first)`, poté odstraní 102 dat z počátku a 102 z konce setříděného pole a vypočte kurtosis oříznutého pole (102 dat přibližně odpovídá 5 %). Z těchto dvou hodnot vypočte kurtosis ratio. Pole je částečně setříděno pomocí `shaker sortu`.



Obrázek 6.2: Průběh výpočtu kurtosis ratio.

Shaker sort je třídící algoritmus se složitostí $O(n^2)$. Je podobný bubble sortu, ale prochází oběma směry. Princip bubble sortu spočívá v tom, že porovnává sousední prvky a pokud jsou ve špatném pořadí, tak je prohodí. Po jednom průchodu tedy setřídí jeden (největší/nejmenší) prvek. Shaker sort třídí oběma směry, každá iterace se skládá ze dvou fází: při dopředné stoupá nejmenší prvek vzhůru, při zpětné klesá největší prvek ke dnu. Shaker sort je naimplementován tak, aby setřídil pouze 102 největších a nejmenších prvků. Setřizovat celé pole není pro účely výpočtu kurtosis ratio třeba.

Shaker sort byl vybrán z důvodů nízkých paměťových požadavků. Umožňuje třídít prvky přímo v předem daném poli a nepotřebuje vytvářet žádné další pomocné pole. Použití rychlejších třídících algoritmů jako je merge sort, nebo quick sort, by vyžadovalo další paměťový prostor. Jelikož pole větších rozměrů se již nevejdou do paměti RAM, musely by být prvky odkládány do paměti Flash.

6.1.5 Kurtogram

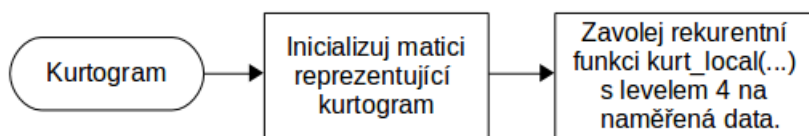
Byl zvolen dekompoziční level $K = 4$ a to z toho důvodu, že máme 2048 navzorkovaných dat a pro dekompoziční level musí podle [21] platit podmínka:

$$K \leq \log_2(\text{lenght}(x)) - 7.$$

Tato podmínka je způsobena použitím filtru FIR o daném množství koeficientů (v našem případě 17 koeficientů pro filtr typu dolní propust a 16 pro filtr typu horní propust). Pokud by při menším množství navzorkovaných dat byl použit příliš velký level dekompozice (tak, že by neplatila podmínka), pak by na vyšších levelech dekompozice bylo menší množství dat, než je koeficientů filtru.

Jednotlivé koeficienty filtru FIR byli převzaty ze stránek [24], jelikož se jedná o komplexní koeficienty, jsme nuceni počítat kurtosis v komplexních číslech. Díky tomu jsou data uložena ve dvou polích, jedno pole uchovává reálnou složku čísla a druhé pole uchovává zase imaginární část.

O výpočet kurtogramu se stará funkce `void kurtogram()`, která nejprve inicializuje matici o velikosti $K \times 2^K$. A poté pomocí rekurze počítá jednotlivé hladiny dekompozičního stromu, na což je naimplementována funkce `void kurt_local(uint32_t *x_real, uint32_t *x_imag, int size, int level, int begin, int first)`, kde `x_real` a `x_imag` jsou reálné a imaginární části dat, `size` je velikost pole `x` a `level` označuje dekompoziční level kurtogramu. Proměnná `begin` určuje sloupec v matici reprezentující kurtogram, od kterého se budou zapisovat vypočtené hodnoty kurtosis a proměnná `first` pouze značí první volání a slouží k omezení přístupu do Flash paměti.



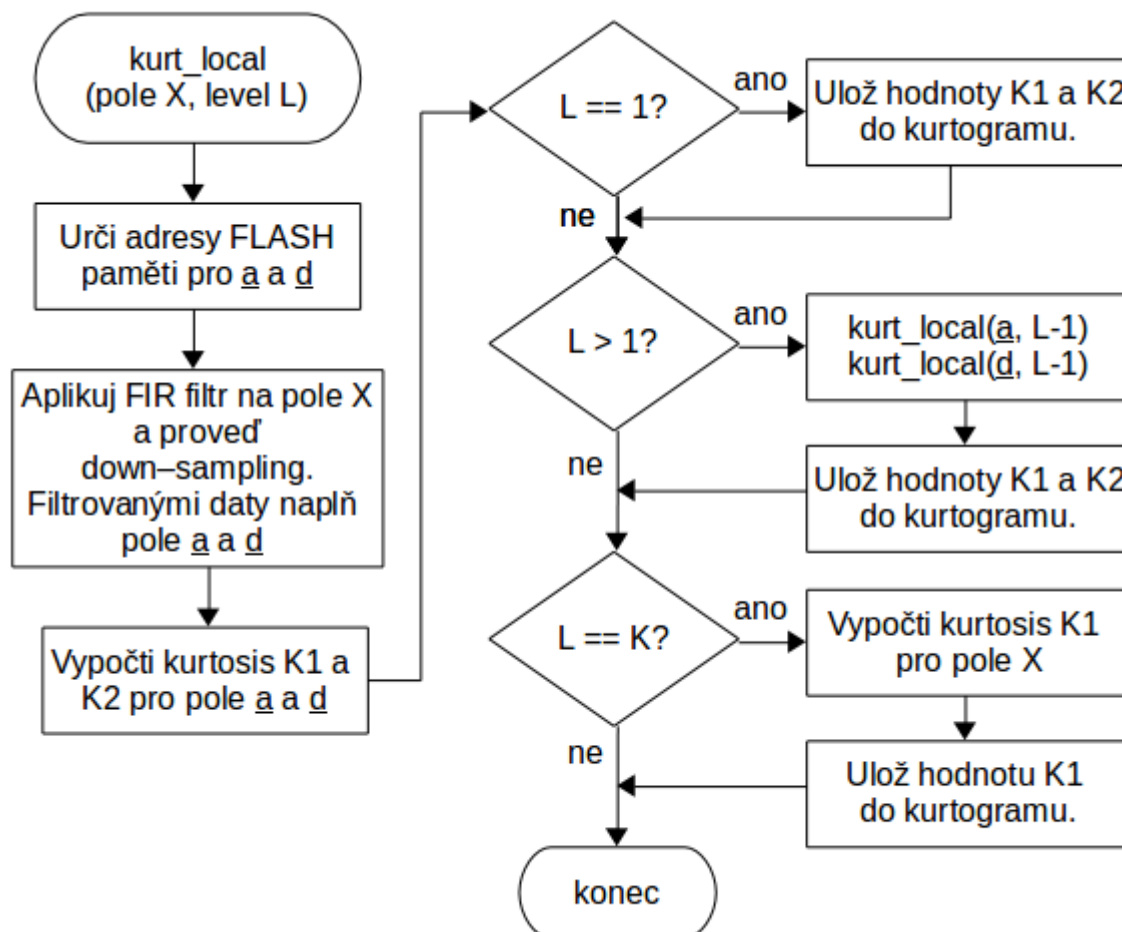
Obrázek 6.3: Vývojový diagram pro funkci `void kurtogram()`.

Funkce `kurt_local(...)` je rekurentní funkce, která nejprve určí adresní prostor na Flash paměti pro dvě poloviční pole dat a a d . Poté zavolá funkci `DBFB(...)`, která aplikuje FIR filtry (dolní propust, horní propust) na posloupnost X , se kterou je volaná, vykoná down-sampling a nakonec filtrovanými daty naplní pole a a d . Pomocí funkce `float kurtosis(uint32_t *x_real, uint32_t *x_imag, int size, int first)` vypočte kurtosis z polí a a d . Nakonec zapíše hodnoty kurtosis do matice reprezentující kurtogram. Rekurentně se opakuje volání funkce `kurt_local(...)` se vstupními poli a a d s dekompozičním levelem o jedna menší než jeho předchůdce, dokud je level větší než jedna. Průběh funkce `kurt_local(...)` je naznačen na Obrázku 6.4.

Funkce `void DBFB(uint32_t *res_real, uint32_t *res_imag, uint32_t *x_real, uint32_t *x_imag, const float *f_real, const float *f_imag, int size, int size_filter, int first)` má jako výstupní parametry pointery na pole `res_real` a `res_imag`, kam se uloží profiltrovaná data po down-samplingu. Vstupními parametry jsou `x_real` a `x_imag` což jsou data, na která chceme aplikovat filtr. Koeficienty filtru jsou uloženy v proměnných `f_real` a `f_imag`. Proměnná `size` určuje velikost pole `x` a `size_filter` udává počet koeficientů filtru.

Funkce `DBFB(...)` mimo jiné zapisuje profiltrovaná data do paměti Flash. Podrobnosti o Flash paměti jsou rozepsány v Sekci 6.1.6.

Po naplnění kurtogramu hodnotami kurtosis je zavolána funkce `void maxK()`, která najde v kurtogramu největší hodnotu a určí její souřadnice, díky kterým se následně dá určit centrální frekvence nestacionárního chování a šířka pásma kurtogramu.



Obrázek 6.4: Schéma rekurentního průběhu funkce `void kurt_local(...)`, která se stará o postupné plnění jednotlivých hladin dekompozičního stromu (postupně zaplňuje jednotlivé položky v matici reprezentující kurtogram).

6.1.6 Přístup do paměti Flash

Jelikož se rekurentní data používaná pro výpočet kurtogramu, do kterých se ukládala profiltrovaná data, nevešla do RAM paměti (20 KB), musela se odkládat do Flash paměti (192 KB). Jelikož Flash paměť je určena především pro ukládání programového kódu a konstantních hodnot, tak je zapisování do paměti Flash velice pomalé. Z tohoto důvodu byl program upraven tak, aby se data odložená do Flash paměti měnila co možná nejméně, díky čemuž mimo jiné vznikla proměnná `int first`, používaná skoro ve všech funkcích jako vstupní parametr, signalizuje první volání rekurze a tím zabraňuje zbytečnému přístupu do paměti.

Pro zapsání do Flash paměti je třeba nejprve povolit přístup, poté vymazat daný úsek paměti zarovnaný na stránky. Následně jsou do paměti zapisována 32 bitová slova. Na

konec je třeba uzamknout paměť pro nepovolený přístup. Zápis je naznačen na Obrázku 6.5.

Čtení jednotlivých hodnot z pole uloženého do Flash paměti je možné pomocí 32-bitového pointeru na počáteční adresu. Poté se pomocí pointerové logiky dají indexovat jednotlivé prvky pole.

Pro jednodušší manipulaci s Flash pamětí byli pro mazání a zápis do paměti použity funkce z HALové knihovny.



Obrázek 6.5: Schéma zápisu do Flash paměti po 32 bitových slovech.

6.1.7 Komunikace přes rozhraní LoRa

Implementace LoRy byla částečně převzatá z příkladů uvedených na webové stránce výrobce STMicroelectronics [13]. Komunikace probíhá na frekvenci 868 MHz.

Monitorovací zařízení je naimplementováno jako zařízení třídy A, které má nejnižší spotřebu a vždy zahajuje komunikaci. Pro účely testování byla použita aktivační metoda ABP, kdy jsou šifrovací klíče předem vygenerovány a zapsány do koncového zařízení. Tento způsob aktivace je rychlejší než aktivační metoda OTAA, kdy se koncové zařízení musí nejprve dohodnout s řídicím serverem na klíších a parametrech komunikace.

Pro sledování zpráv posílaných přes LoRa rozhraní byl vytvořen účet na síti ThingsNetwork [17]. Po přihlášení je třeba vytvořit aplikaci, zadat její popis a zaregistrovat na ní handler. V našem případě jsme aplikaci zaregistrovali na handler ttn-handler-eu. V aplikaci je pak nutné registrovat dané zařízení s jeho unikátním ID.

Nejprve je potřeba nastavit parametry LoRa komunikace v souboru `comissioning.h` tak, aby souhlasily odpovídající parametry v síti ThingsNetwork. Prvním parametrem je Device EUID, což je unikátní ID zařízení. Vývojový kit toto unikátní ID vypíše na hyper

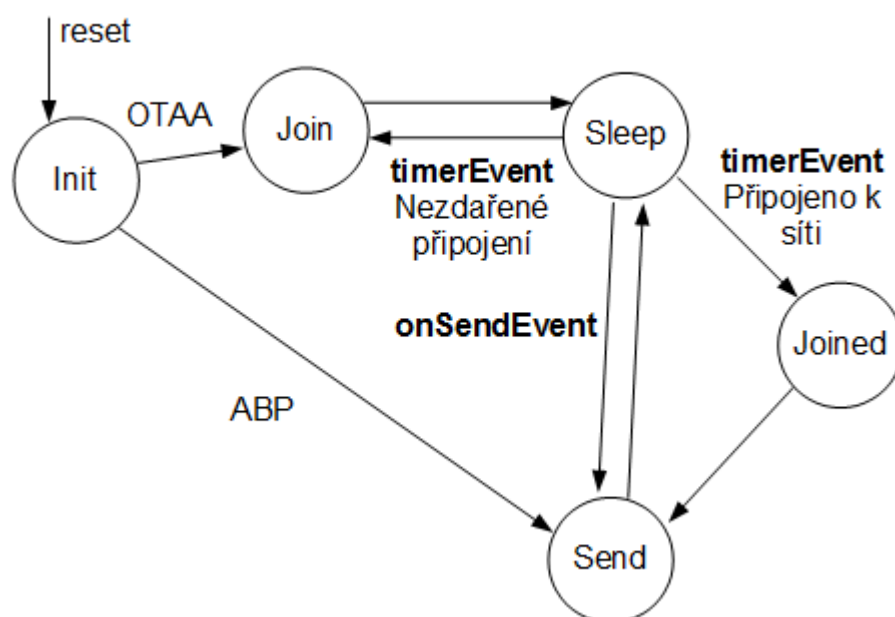
terminál připojeného počítače pomocí rozhraní UART při prvotním připojení. Dalšími parametry jsou Application EUID a Device Address, jedná se o ID LoRaWAN aplikace a adresu zařízení. Obě dvě hodnoty vygeneruje síť ThingsNetwork při registraci zařízení. Na závěr je třeba zapsat klíče Network Session Key a App Session Key, vygenerované sítí ThingsNetwork, do koncového zařízení.

K softwareové implementaci LoRa rozhraní, byly použity následující hardwareové komponenty:

- Radio: Slouží k odesílání a přijímání dat. Jeden GPIO pin je použit jak radio reset, další pin slouží k ovládání přepínače antény pro nastavení do režimu odesílání, nebo přijímání.
- SPI: Radio registry jsou dostupné přes rozhraní SPI.
- RTC: Poskytuje centralizovanou časovou jednotku, která pokračuje v provozu i v režimu nízké spotřeby. Alarm RTC se používá k probouzení systému v určitých časových úsecích. Výstup RTC je omezen na 32 bitový časovač.

Všechny tyto periférie jsou inicializované pouze jednou a to na začátku programu ve funkci `main()`.

Komunikace přes LoRa rozhraní je vedena stavovým automatem, jak je znázorněno na Obrázku 6.6.



Obrázek 6.6: Stavový automat popisující chování systému při komunikaci prostřednictvím LoRa.

Odesílání dat má dva možné režimy: `timerEvent` a `onSendEvent`. `TimerEvent` umožňuje posílat data na událost vyvolanou časovačem. Tento režim byl použit v této práci. Režim `onSendEvent` odesílá data na externí událost.

Při resetu po inicializaci zařízení se systém dostane do počátečního stavu, definovaného jako `'Init'`.

Při použití aktivační metody OTAA přejde zařízení do stavu 'Join', kde odesílá žádost o připojení k síti, dále přejde do stavu 'Sleep'. Při události timerEvent se pokusí zařízení připojit k síti. Při nezdářílém pokusu se vrátí do stavu 'Join' a pokusí se opětovně připojit. Pokud se zařízení v pořádku připojí, přejde do stavu 'Joined' a následně do stavu 'Send'.

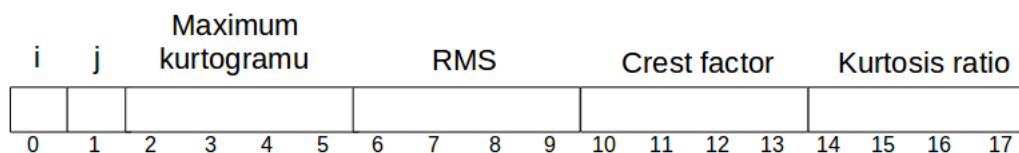
Při aktivaci ABP, je přeskočena fáze připojování a zařízení rovnou přejde do stavu 'Send'.

Ze stavu 'Sleep' se dostane zařízení událostí onSendEvent do stavu 'Send'.

Ze stavu definovaného jako 'Send' se systém vrátí zpět do stavu 'Sleep', aby bylo možné čekat na další událost odpovídající dalšímu naplánovanému paketu, který má být odeslán.

Obsah rámce

Pomocí LoRa technologie odesílá monitorovací modul rámec o velikosti 18 bytů.



Obrázek 6.7: Struktura rámce a obsah jednotlivých bytů odesílaných do sítě ThingsNetwork.

Všechny odesílané hodnoty (kromě indexů i a j , které jsou 8 bitové) jsou reprezentovány jako 32 bitový unsigned integer. byty v rámci jsou ukládány jako little endian, což znamená, že na nejnižší index rámce, patřící dané hodnotě, se uloží nejméně významný byte (LSB) a za něj se ukládají ostatní byty až po nejvíce významný bajt (MSB).

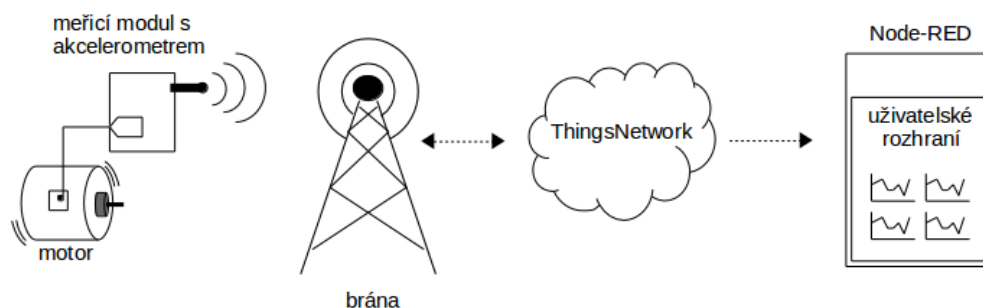
Maximální hodnota kurtogramu se vyskytuje na sloupci i a řádku j , což jsou první dva byty v rámci. Na následujících čtyřech bytech je samotná maximální hodnota kurtogramu. Od indexu 6 do indexu 9 je odesílaná efektivní hodnota. Následuje Crest factor, který je na indexech 10-13. Poslední čtyři byty uchovávají hodnotu kurtosis ratio.

6.2 Cloudové zpracování

Data, která jsou poslána do sítě ThingsNetwork, jsou dále zpracovávána pomocí programovacího nástroje Node-RED, kde jsou pomocí user interface zobrazená statistická data z naměřených a vypočtených hodnot. Obrázek 6.8 zobrazuje měření a odesílání dat přes jednotlivé funkční bloky.

Node-RED je programovací nástroj založený na toku dat, byl vyvinutý týmem IBM [25].

Programování založené na toku je způsob, jak popisovat chování aplikace jako síť black-boxů nebo "uzlů", jak se nazývají v Node-RED. Každý uzel má dobře definovaný účel. Uzlu se předají údaje, ten je zpracuje a pak je předá dál. Síť je odpovědná za tok dat mezi uzly.



Obrázek 6.8: Schéma funkčnosti od naměření dat až po uživatelské rozhraní.

Node-RED využívá Node.js, který umožňuje ve webovém prohlížeči přístup k editoru toku dat. V prohlížeči lze vytvořit aplikace přetažením uzlů z palety do pracovního prostoru a následným pospojováním.

Programovací nástroj Node-RED umožňuje přidávat palety s různými uzly. Jednou z těchto palet je paleta `node-red-contrib-ttn`, která poskytuje uzly pro komunikaci se sítí ThingsNetwork. Díky této paletě je možné odchyťovat rámec přicházející do sítě ThingsNetwork a data v něm následně zpracovat.

Dále byla kromě základní nabídky použita paleta `node-red-contrib-mdashboard`, která umožňuje vykreslení grafů a měřidel v uživatelském rozhraní.

6.2.1 Popis funkčních uzlů

Veškeré následující funkce byly naprogramovány v jazyce JavaScript a přidány do funkčních bloků pomocí nástroje Node-RED. Aplikace parsující data z rámce a vytvářející grafy v uživatelském rozhraní je znázorněná na Obrázku 6.9.

Jakmile přijde do sítě ThingsNetwork rámec s daty, je odchyten nástrojem Node-RED.

Uzel s funkcí `kurtogram values` parsuje z rámce informaci o maximální hodnotě kurtogramu a jejím umístění, čili index sloupce a řádku, kde se tato hodnota nachází. Z těchto dat vypočte frekvenci a šířku pásma kurtogramu. Tyto vypočtené údaje spolu s maximální hodnotou kurtogramu zobrazí v grafech v uživatelském rozhraní. Tento uzel se také stará o vypisování aktuální maximální hodnoty kurtogramu s danou frekvencí a šířkou pásma v horní části uživatelského rozhraní.

Uzel `bearing failure` představuje funkci, která si ukládá do paměti poslední tři hladiny kurtogramu ve kterých bylo nalezeno maximum a poslední tři hodnoty crest factoru. Pokud jsou poslední tři hladiny větší než jedna (což znamená, že šířka pásma je menší než 2000 Hz) a zároveň jsou poslední tři hodnoty crest factoru větší než 3,5 pak nastaví proměnnou signalizující poruchy na `true` a tento údaj pošle funkcím `email` a `status`.

Funkce `email` se stára o odeslání varovného emailu obsluze a funkce `status` zobrazuje v horní části uživatelského rozhraní zda je možné, že nastala porucha ložiska.

Funkční uzel `statistics` parsuje a zobrazuje v grafech vypočtené údaje o efektivní hodnotě, crest factoru a kurtosis ratio. Následné funkce `max of RMS`, `max of crest factor` a `max of kurtosis` uchovávají a zobrazují maximální hodnoty, které byli vypočteny za dobu běhu programu.

Funkce `connection parameters` parsuje metadata z přijaté zprávy. Vyčítá frekvenci a modulaci, na které komunikuje monitorovací zařízení se sítí ThingsNetwork. Záro-

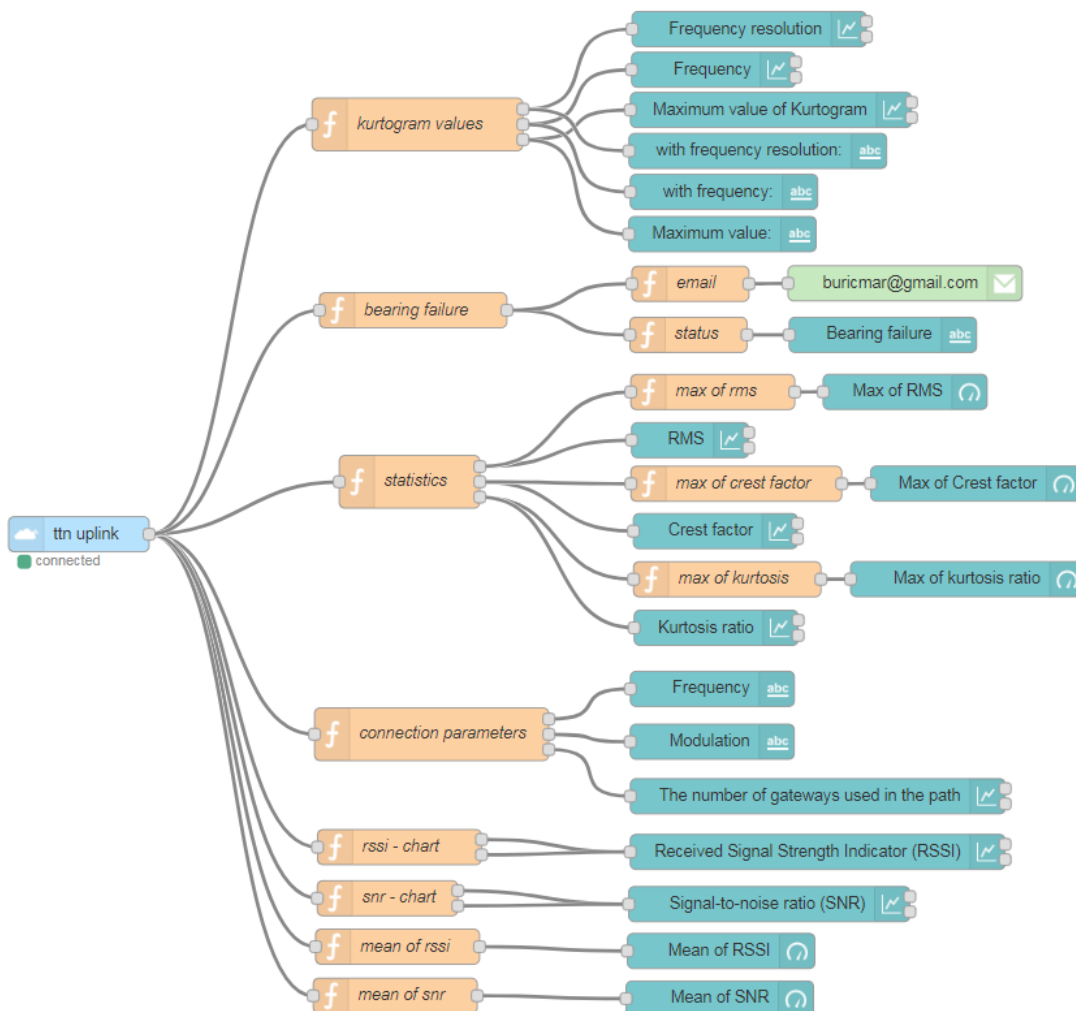
veň z metadat vyčte počet bran, které byly použity při posílání rámce do sítě ThingsNetwork a zobrazuje tuto informaci v grafu.

Funkce `rssi-chart` a `snr-chart` použijí z metadat informaci o RSSI a SNR prvních dvou bran, přes které se posílala data do sítě ThingsNetwork. Tyto údaje zobrazují v grafech. Pokud byla použita pouze jedna brána zobrazí se v grafech pouze jeden údaj. Pokud je použito více bran, pak se zobrazí hodnoty z prvních dvou bran. První dvě brány jsou určeny podle času, kdy k nim dorazil daný rámec.

RSSI (Received Signal Strength Indication) je indikátor síly přijímaného signálu. Udává se v dBm. Vypočítá se jako rozdíl výkonu užitečného přijatého signálu a poměru přijaté energie a energie přijatého šumu. [26]

SNR (Signal-to-noise ratio) je poměr výkonu signálu k výkonu šumu. Tento poměr se udává v dB. [27]

Funkce `mean of rssi` a `mean of snr` počítá a zobrazuje průměrnou hodnotu RSSI a SNR po celou dobu běhu programu.



Obrázek 6.9: Aplikace vytvořená v Node-REDu pomocí definování funkcí jednotlivých uzlů a určením toku mezi vytvořenými uzly. Tmavě modré uzly jsou prvky v uživatelském rozhraní, béžové jsou funkce pro parsování dat z rámce, světle modrý je příjem rámce ze sítě ThingsNetwork a zelená označuje email údržby.

6.2.2 Uživatelské rozhraní

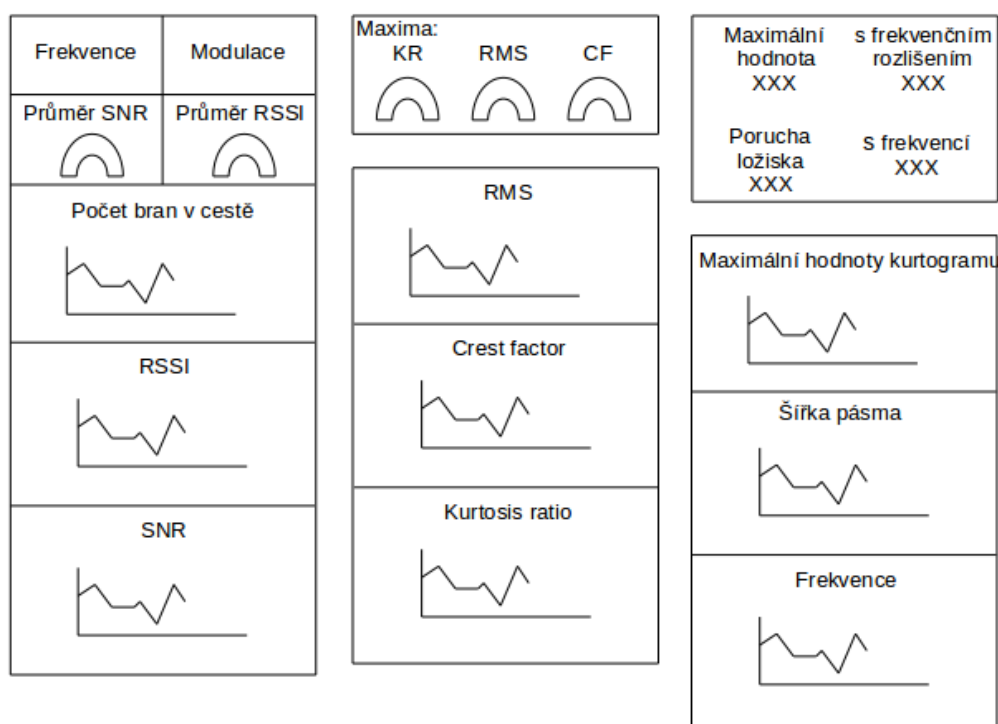
Programovací nástroj Node-RED poskytuje uživatelské rozhraní což je snadný způsob vizualizace naměřených dat monitorovacího zařízení. Toto rozhraní zobrazuje grafy naměřených a vypočtených hodnot a jejich jednoduché statistické údaje.

Uživatelské rozhraní je rozděleno na tři části: parametry spojení, hodnoty kurtogramu a blok s RMS, crest factorem a kurtosis ratio. Rozdělení a umístění jednotlivých údajů je vyobrazeno na Obrázku 6.10.

V levém sloupci jsou parametry spojení. Jako první je vypsána informace o frekvenci a modulaci. Pod těmito údaji se vykreslují měřidla s průměrným SNR a s průměrným RSSI. Poté následuje graf s informací o počtu bran použitých v cestě. Podle tohoto parametru se dále zobrazuje buď jedna nebo dvě křivky v následujících grafech obsahujících RSSI a SNR. Přičemž jako první brána v cestě je označována ta z nich, kde se rámeček vyskytl dříve.

Uprostřed uživatelského rozhraní jsou v horní části měřidla s maximálními hodnotami kurtosis ratio, RMS a crest factoru, které se objevily za dobu běhu programu. Ve spodní části jsou pak jejich grafy vývoje v čase.

V pravém sloupci jsou informace o vypočteném kurtogramu. Horní část obsahuje aktuální maximální hodnotou kurtogramu, spolu s frekvencí a šířkou pásma této maximální hodnoty. V této části je zároveň výpis signalizující možnou poruchu. Pod tímto úsekem jsou vyobrazeny grafy maximálních hodnot kurtogramu, frekvence a šířky pásma kurtogramu vyvíjejících se v čase.



Obrázek 6.10: Schéma uživatelského rozhraní. Parametry spojení jsou v levém sloupci, hodnoty kurtogramu uprostřed a v pravé části je RMS, crest factor a kurtosis ratio.

Všechny zobrazované grafy si uchovávají data za posledních 24 hodin.

Pokud je crest factor delší dobu větší než 3.5 a maximum kurtogramu je po něja-

kou dobu detekováno na větším levelu dekompozičního stromu, pak se odešle informační email na předem danou emailovou adresu. Tato služba je odzkoušená pro přihlášení z účtu Gmail (což je bezplatná e-mailová služba provozovatele vyhledávače Google). Pro správnou funkcionální je třeba v emailové schránce, v sekci zabezpečení, povolit méně bezpečné aplikace. Poslední nutné opatření pro funkční odesílání emailových zpráv je nedovolit blokování odesílání zpráv antivirovým programem na zařízení, kde běží Node-RED.

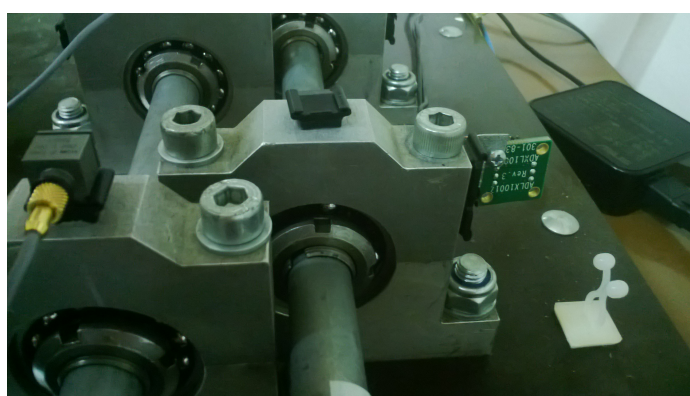
7 Testování zařízení

Pro účely testování monitorovacího zařízení byli provedeny dva nezávislé experimenty na simulátoru vibrací (Obrázek 7.1). Každý experiment se skládal z měření na motoru s porouchaným ložiskem (Motor 1) a na motoru bez vady (Motor 2).



Obrázek 7.1: Simulátor vibrací použitý při ověření funkčnosti monitorovacího zařízení.

V obou případech byl akcelerometr připojen k ložisku, který je umístěn blíže motoru a to tak, že osa akcelerometru byla rovnoběžná s podložkou simulátoru, jak je ukázáno na Obrázku 7.2. Akcelerometr byl připojen k vývojovému kitu, na kterém běžel vytvořený software pro digitalizaci signálu, předzpracování, určení příznaků a jejich následné odeslání.



Obrázek 7.2: Způsob umístění akcelerometru na ložisko. Osa akcelerometru je rovnoběžná s podložkou simulátoru vibrací.

U obou experimentů byly příznaky posílány do sítě ThingsNetwork každých pět minut a data navzorkována vždy deset sekund po odeslání příznaků.

Ve všech případech měření (na motoru s poruchou i na motoru bez poruchy, v prvním i druhém experimentu) byl prvních deset minut motor v nečinnosti, poté se počet otáček

zvýšil na 500 otáček za minutu. Po každých deseti minutách se počet otáček zvýšil o tisíc. Při 3 500 otáčkách za minutu motor běžel patnáct minut, poté se počet otáček snížil na 2 000, po dalších deseti minutách na 1 000 otáček za minutu. Na závěr se opět zvýšil na 2 500 otáček, kdy motor běžel deset minut a poté se měření ukončilo.

Při testování monitorovacího zařízení se poměrně často stávalo, že síť ThingsNetwork některý rámeček s daty nepřijala i přes to, že data byla z monitorovacího zařízení odeslána.

Z každého měření je výstup v podobě zpráv přijatých do sítě ThingsNetwork a jejich následné zobrazení v uživatelském rozhraní pomocí nástroje Node-RED.

7.1 První experiment

Experiment byl nejprve prováděn na motoru bez poruchy (Motor 2), kdy byl akcelerometr přimontován na ložisko blíže motoru, poté na motoru s porouchaným ložiskem (Motor 1). Při prvním experimentu byla vzorkovací frekvence přibližně 11 kHz.

V prvním experimentu byla indikátorem vady ložiska pouze vypočtená hladina dekompozičního stromu, kde bylo nalezeno maximum kurtogramu. Indikátor byl aktivní pokud byla hladina, kde se našlo maximum kurtogramu větší než dva, čili šířka pásma byla menší než 1 000 Hz, což se ukázalo, jako nedostatečné řešení.

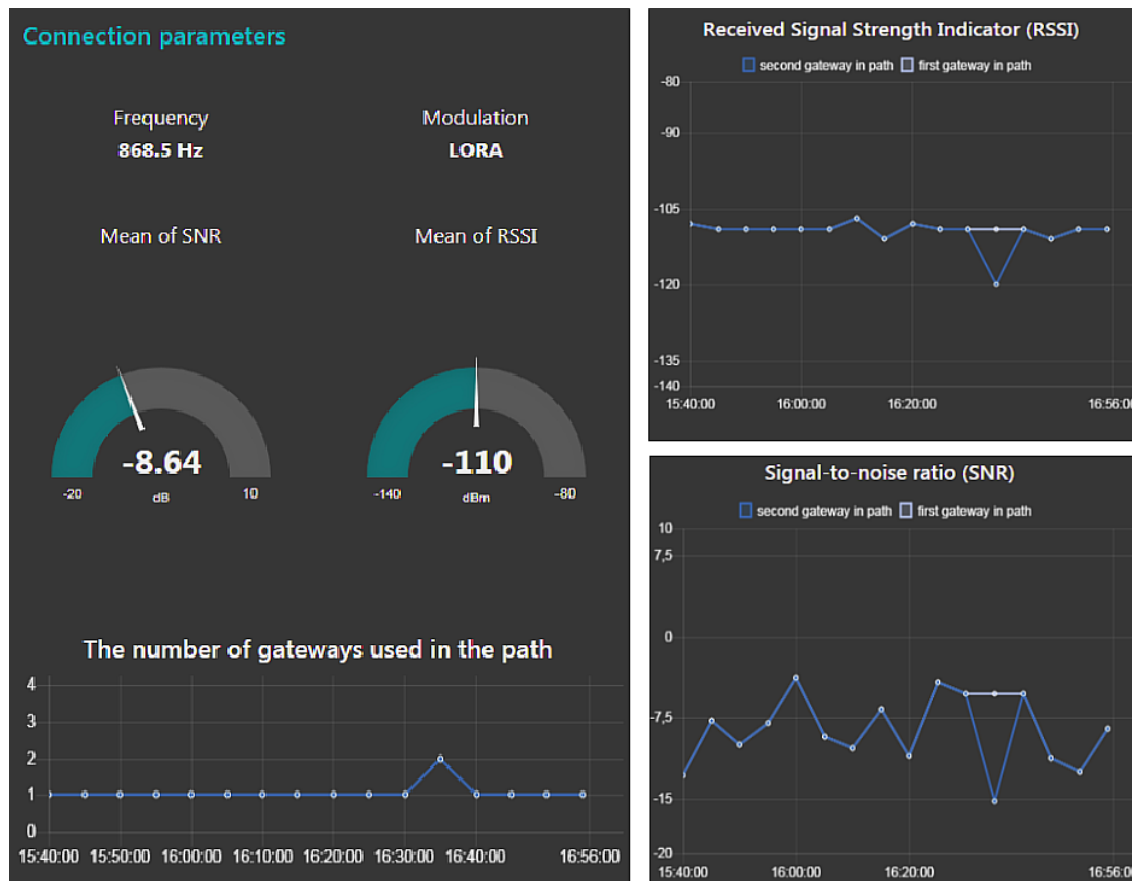
7.1.1 Motor bez poruchy

Na Obrázku 7.3 je soupis přijatých zpráv z prvního měření ze sítě ThingsNetwork s časem, kdy byl daný rámeček přijat. Celkem bylo odesláno 17 rámečků. Podle hodnoty counteru je poznat, že poslední rámeček nebyl sítí ThingsNetwork přijat. Položka payload obsahuje přijatá data, význam jednotlivých bytů je popsán v Kapitole 6.1.7.

time	counter	port	payload
▲ 16:55:04	15	2	payload: 01 00 20 09 42 3F E5 D8 2A 42 A6 30 4E 40 1B D8 BB 3F
▲ 16:50:04	14	2	payload: 01 00 74 BC 43 3F 8C AB 72 41 FA 17 61 40 AE 6D AF 3F
▲ 16:45:03	13	2	payload: 04 0A 74 FA 6E 3F F6 5E 50 41 81 E4 55 40 54 BD AC 3F
▲ 16:40:03	12	2	payload: 01 00 E0 37 B0 3E B7 9B EC 41 EB 02 4E 40 04 79 A1 3F
▲ 16:35:03	11	2	payload: 01 00 44 33 25 3F 6B E6 EF 41 29 01 3A 40 0E AB A3 3F
▲ 16:30:03	10	2	payload: 04 0E 58 4D E5 3E 46 3A 48 42 A3 19 51 40 FD 0C AA 3F
▲ 16:25:03	9	2	payload: 03 08 CC 8D 3B 3F 8D F4 2A 42 78 75 48 40 B8 7B AB 3F
▲ 16:20:03	8	2	payload: 01 00 40 82 CE 3E 83 16 4E 42 3D 6D 5D 40 FD FC A8 3F
▲ 16:15:03	7	2	payload: 04 06 88 D9 EF 3E 74 7B 28 42 5F 66 3E 40 BB 35 A0 3F
▲ 16:10:03	6	2	payload: 02 0C 70 1C 1C 3F D9 DA 29 42 7F 38 46 40 8F 10 9D 3F
▲ 16:05:03	5	2	payload: 01 00 00 30 0A 3F 79 6A A0 41 64 E1 4A 40 55 55 B0 3F
▲ 16:00:02	4	2	payload: 04 06 28 EB 8F 3F 62 64 A7 41 D7 C6 5A 40 F9 13 B8 3F
▲ 15:55:02	3	2	payload: 01 08 14 63 7B 3F 98 46 09 41 59 98 4E 40 3B E4 A8 3F
▲ 15:50:02	2	2	payload: 01 08 24 37 51 3F F1 97 F5 40 21 F8 40 40 23 A7 9E 3F
▲ 15:45:02	1	2	payload: 00 00 98 93 B9 3F D3 24 83 3F 24 43 B2 40 C0 64 CD 3F
▲ 15:40:02	0	2	retry payload: 01 08 AA 9C 1F 40 A7 BB 97 3F FB F5 C2 40 31 B5 E0 3F

Obrázek 7.3: Data přijatá sítí ThingsNetwork.

Na následujícím obrázku 7.4 jsou zobrazeny parametry spojení. Jak je vidět většinou byla pro přenos použita pouze jedna brána. V čase 16:35 byly použity dvě brány pro přenos dat, jejich rozdílné SNR a RSSI je vidět v daných grafech. Průměrné SNR počítané po dobu měření je -8.64 dB a průměrné RSSI činí -110 dBm. V uživatelském rozhraní jsou tyto bloky zobrazeny pod sebou v levém sloupci.



Obrázek 7.4: Parametry spojení. Počet bran v cestě a průměrná hodnota SNR a RSSI za dobu běhu motoru při měření (vlevo) a jejich změna v čase (vpravo).

Na Obrázku 7.5 jsou vidět maximální vypočtené hodnoty, za dobu měření, a jednotlivé časové průběhy RMS, crest factoru a kurtosis ratio. V uživatelském rozhraní jsou tyto grafy pod sebou uprostřed.

Maximum crest factoru bylo 6.09 a maximum kurtosis ratio bylo 1.76, což jsou hodnoty získané z prvního přijatého rámce, tudíž, když byl motor vypnutý. Tyto první hodnoty značně ovlivňují statistické údaje zobrazované pomocí měřidel. Maximální vypočtená efektivní hodnota byla $51.52 \text{ rel } m \cdot s^{-2}$, která byla naměřená při běhu motoru rychlostí 3 500 otáček za minutu.

U průběhu efektivní hodnoty je vidět, jak se měnili otáčky monitorovaného stroje v čase. První dvě hodnoty reprezentují vypnutý motor, hodnota RMS je v tomto případě blízká nule. Poté se otáčky vždy po dvou naměřených datech zvyšovali. Na nejvyšších otáčkách byli naměřeny tři hodnoty a poté došlo ke snižování. Na závěr se počet otáček opět zvýšil. Poslední rámec s daty nebyl v síti ThingsNetwork přijat proto zde graf končí zvětšením hodnoty RMS pouze s jedním údajem.

Při nulových otáčkách je crest factor vychýlen až na hodnotu kolem 6. Při běhu motoru

se bez ohledu na počet otáček crest factor pohybuje přibližně v rozmezí mezi 3 - 3.5.

Hodnoty kurtosis ratio také vykazují vyšší hodnoty při nulových otáčkách. Zde je však hodnota větší oproti běhu motoru v řádu desetin. Hodnota kurtosis ratio se při nevypnutém motoru pohybuje v rozmezí od 1.2 do 1.5.



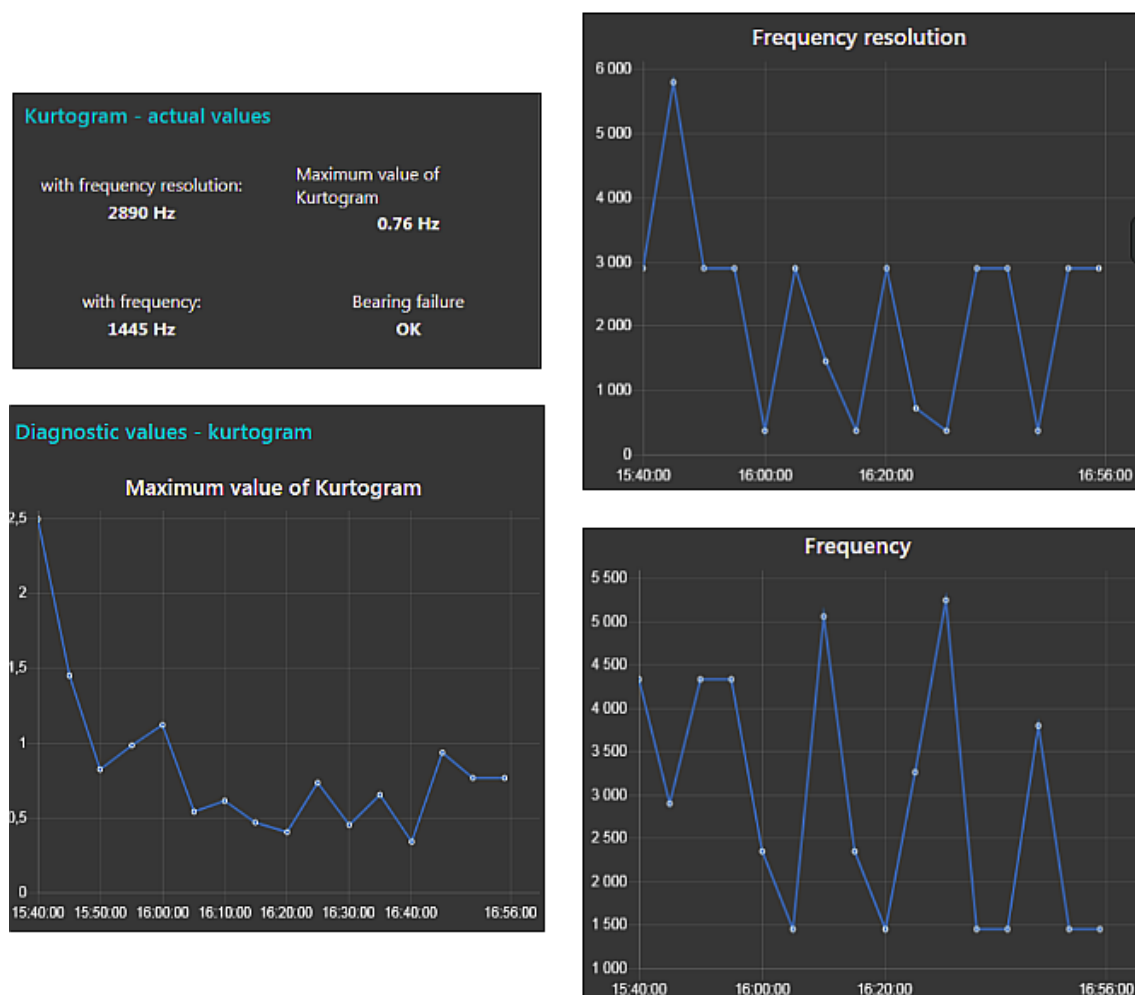
Obrázek 7.5: Časový průběh RMS, crest factoru a kurtosis ratio a jejich maximální hodnoty za dobu měření zobrazované pomocí měřidel.

Na závěr jsou zobrazeny grafy s parametry získanými z kurtogramu měnící se v čase (Obrázek 7.6).

Z grafu frekvence a šířky pásma je vidět, že kurtogram má své maximum pokaždé na jiných pozicích. Navíc často dochází k detekci maxima kurtogramu s šířkou pásma větší než 1 000 Hz.

Jelikož indikátor poruchy se určuje pouze podle hladiny dekompozičního stromu, tak informativní email údržbě byl odeslán celkem pětkrát, jak signalizuje graf průběhu šířky pásma v čase (Obrázek 7.7).

V uživatelském rozhraní se zobrazují poslední naměřená data získaná z kurtogramu v horní části v pravé oblasti. Zároveň s těmito údaji se ve stejném bloku zobrazuje i potencionální možnost poruchy ložiska. V případě, že indikátor je aktivní, vypíše se pod "Bearing failure" text "NOT OK". V případě, že není detekována vada vypíše se pouze "OK".



Obrázek 7.6: Průběh maximální hodnoty kurtogramu, frekvence, kde byla nalezena a šířky pásma v čase a jejich poslední vypočtené údaje.

Condition monitoring system - Hodnota kurtogramu přesáhla toleranční mez.	16:45
Condition monitoring system - Hodnota kurtogramu přesáhla toleranční mez.	16:30
Condition monitoring system - Hodnota kurtogramu přesáhla toleranční mez.	16:25
Condition monitoring system - Hodnota kurtogramu přesáhla toleranční mez.	16:15
Condition monitoring system - Hodnota kurtogramu přesáhla toleranční mez.	16:00

Obrázek 7.7: Ukázka odeslaných emailů na předem zadanou adresu.

7.1.2 Motor s poruchou

Druhé měření bylo provedeno na motoru s porouchaným ložiskem (Motor 1). Akcelerometr byl připojen přímo na porouchané ložisko.

Na Obrázku 7.8 je zachycena přijatá zpráva i se svými metadaty do sítě ThingsNetwork. Metadata obsahují informace o bránách použitých pro přenos dat. V tomto případě byla použita pouze jedna brána na souřadnicích 50.1 stupňů zeměpisné šířky a 14.3 stupňů zeměpisné délky s názvem "cesgw1". Z metadat lze také vyčíst RSSI, SNR a čas, kdy daný rámeček k dané bráně dorazil.

Na dalším Obrázku (7.9) následuje opět výpis přijatých dat do sítě ThingsNetwork i s časy. Jak je vidět, při tomto měření se ztratili datové rámce s čísly 1,2 a 11.

```

Payload
03 0A A4 77 A0 3F 0B E5 99 42 96 1B 5D 40 16 8D B2 3F

Fields
no fields

Metadata
{
  "time": "2018-05-14T13:18:28.139860017Z",
  "frequency": 868.5,
  "modulation": "LORA",
  "data_rate": "SF12BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "cesgw1",
      "gtw_trusted": true,
      "timestamp": 2107112532,
      "time": "2018-05-14T13:18:27.079486Z",
      "channel": 2,
      "rssi": -108,
      "snr": -7.5,
      "rf_chain": 1,
      "latitude": 50.10169,
      "longitude": 14.391644,
      "altitude": 242
    }
  ]
}

```

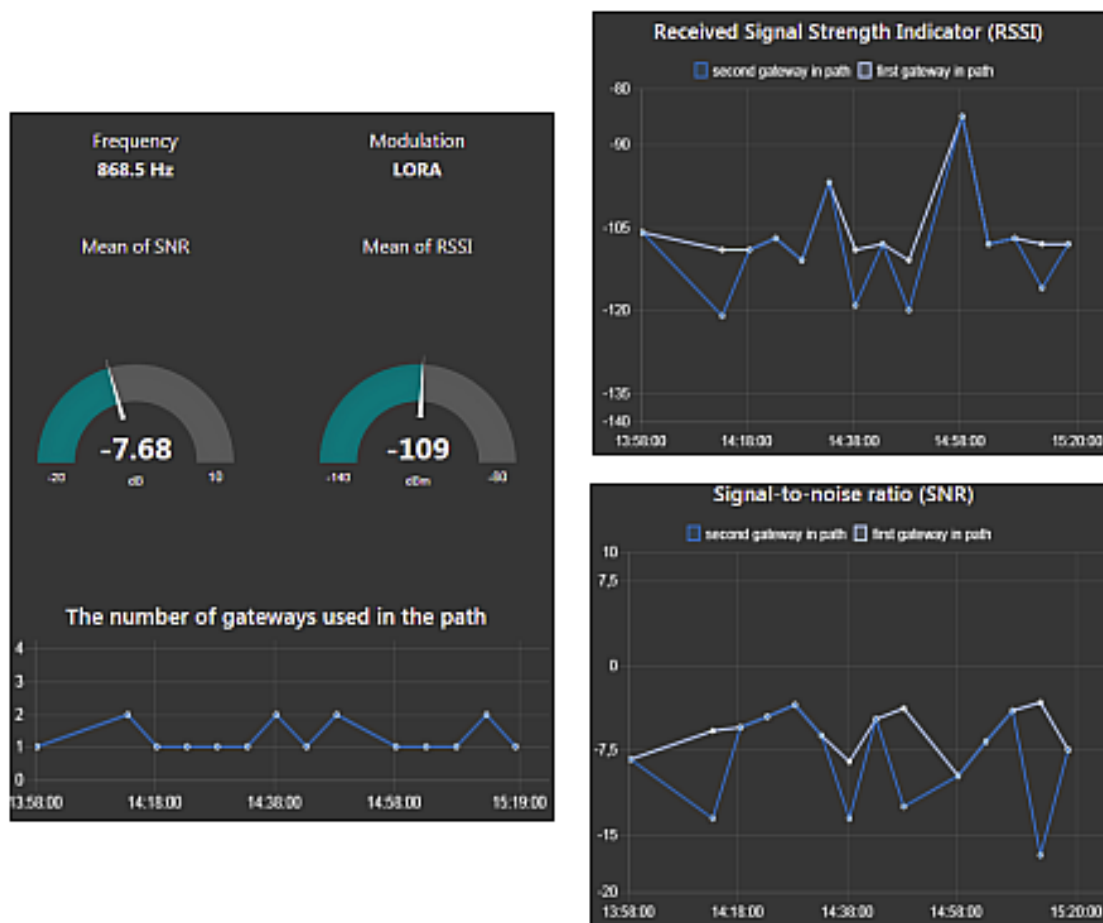
Obrázek 7.8: Výpis metadat z jedné přijaté zprávy ze sítě ThingsNetwork.

▲ 15:18:28	16	2	payload: 03 0AA4 77 A0 3F 0B E5 99 42 96 1B 5D 40 16 8DB2 3F
▲ 15:13:28	15	2	payload: 03 0C 82 73 A9 3F 4C 6F A0 42 6D 91 7F 40 E4 82 C6 3F
▲ 15:08:27	14	2	payload: 04 07 0E 41 26 40 BE 64 11 42 D2 00 82 40 3AA6 BA 3F
▲ 15:03:27	13	2	payload: 04 07 C4 63 19 40 E3 78 1F 42 D5 D6 95 40 98 AA 0C 40
▲ 14:58:27	12	2	payload: 04 07 86 0B 60 40 08 98 91 42 A6 2A 57 40 18 7C C1 3F
▲ 14:48:27	10	2	payload: 01 00 4C 06 1B 3F 0B C4 30 43 7E 38 71 40 21 F1 B9 3F
▲ 14:43:27	9	2	payload: 04 07 10 AA 7E 3F 10 65 3D 43 9A 56 37 40 6A 94 A2 3F
▲ 14:38:27	8	2	payload: 04 07 10 73 3D 3F 1F 8D 4B 43 A8 6C 60 40 51 65 C4 3F
▲ 14:33:27	7	2	payload: 04 07 B8 FE 11 40 BE 7E D0 42 C7 C4 82 40 B2 04 D8 3F
▲ 14:28:26	6	2	payload: 04 07 8E 0A 29 40 CA EB EF 42 18 8E 82 40 AED1 DE 3F
▲ 14:23:26	5	2	payload: 04 07 1F 14 48 41 CF C8 A1 42 27 C7 69 40 6D 17 BB 3F
▲ 14:18:26	4	2	payload: 04 04 68 1E 8F 40 28 32 AF 42 76 67 67 40 9A E3 C4 3F
▲ 14:13:26	3	2	payload: 04 07 F4 43 96 40 80 55 CD 41 F1 A1 A8 40 8E 50 20 40
▲ 13:58:26	0	2	retry payload: 00 00 66 57 6D 40 F1 33 3D 3F 86 C1 C5 40 0E 3C 49 40

Obrázek 7.9: Data přijatá sítí ThingsNetwork.

Následuje Obrázek 7.10 s parametry spojení pro druhé měření. Průměrné SNR bylo -7.65 dB a průměrné RSSI bylo -109 dBm. Údaje o frekvenci a modulaci se nemění. V tomto případě byla data čtyřikrát poslána přes více než jednu bránu. V případě, že bylo použito více bran, pak jsou v grafech, vykreslující RSSI a SNR, dvě křivky, přičemž světle modrá linka označuje tu bránu, kam rámeček dorazil jako první, tmavě modrá pak označuje hodnoty pro druhou použitou bránu v cestě. Která brána je první, je určeno pomocí časové značky uložené v metadatech.

V tomto případě došlo ke ztrátě tří datových rámců, čili tyto hodnoty nejsou v grafech uvedeny.



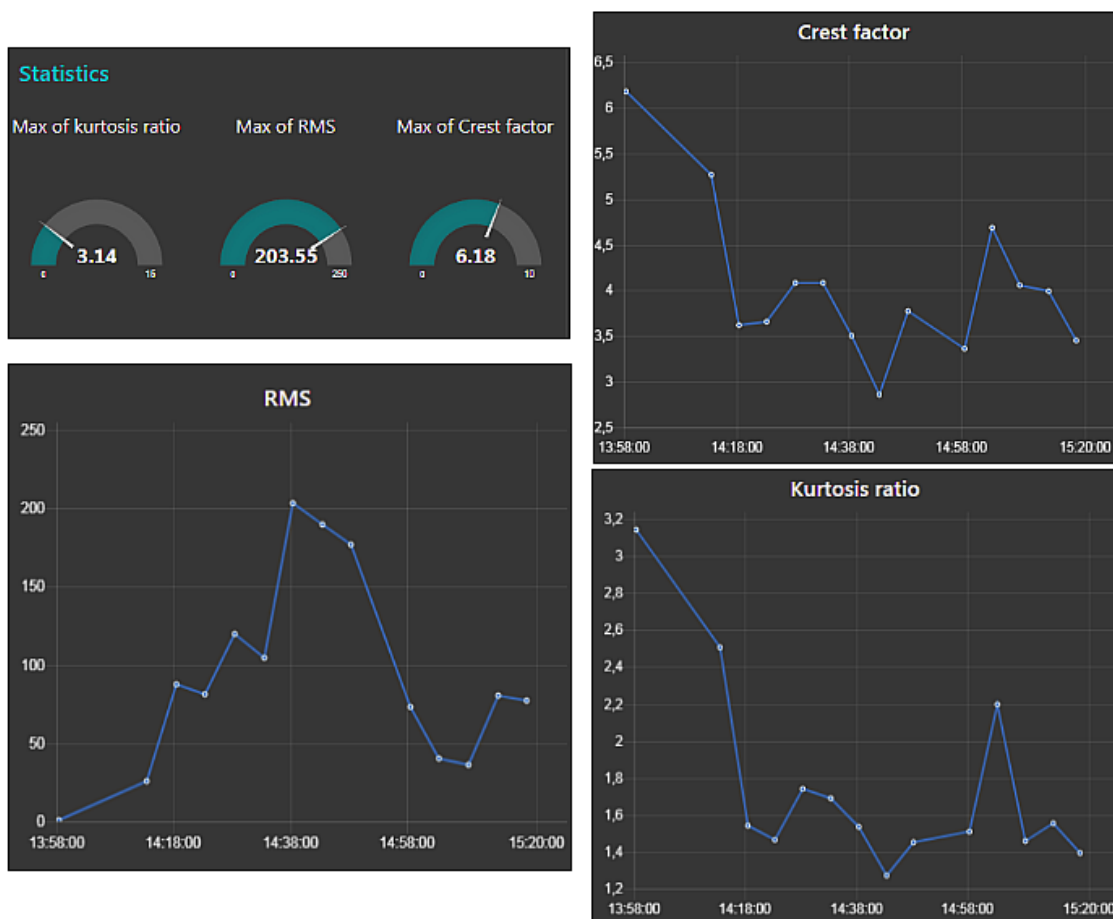
Obrázek 7.10: Parametry spojení pro druhé měření. Počet bran v cestě a průměrná hodnota SNR a RSSI za dobu běhu motoru při měření (vlevo) a jejich změna v čase (vpravo).

Při tomto měření byla maximální naměřená hodnota kurtosis ratio 3.14. Maximum RMS bylo $203.55 \text{ rel } m \cdot s^{-2}$ a maximum crest factoru bylo 6.18. Maximální hodnoty a průběhy těchto veličin v čase jsou vykresleny na Obrázku 7.11. Maximum efektivní hodnoty je podstatně větší než u prvního měření.

První dvě hodnoty u kurtosis ratio a crest factoru jsou opět vychýleny, stejně jako v prvním měření. Opět se jedná o stav, kdy motor neběžel, tudíž byli nulové otáčky. Dále se křivka crest factoru pohybuje ve vyšších hodnotách, než v případě měření na motoru bez poruchy, kdy se hodnota crest factoru pohybovala přibližně v rozmezí 3 - 3.5. V měření

na porouchaném ložisku se crest factor pohybuje v rozmezí (až na výjimky) od 3.4 do 4.1. Drobné zvýšení hodnot nastalo i u kurtosis ratio, kdy se hodnoty pohybují od 1.2 do 1.8.

Křivka RMS opět kopíruje změnu otáček motoru. Při zvýšení otáček křivka stoupá a při snížení křivka klesá. Avšak pohybuje se taktéž ve vyšších hodnotách než v prvním měření.

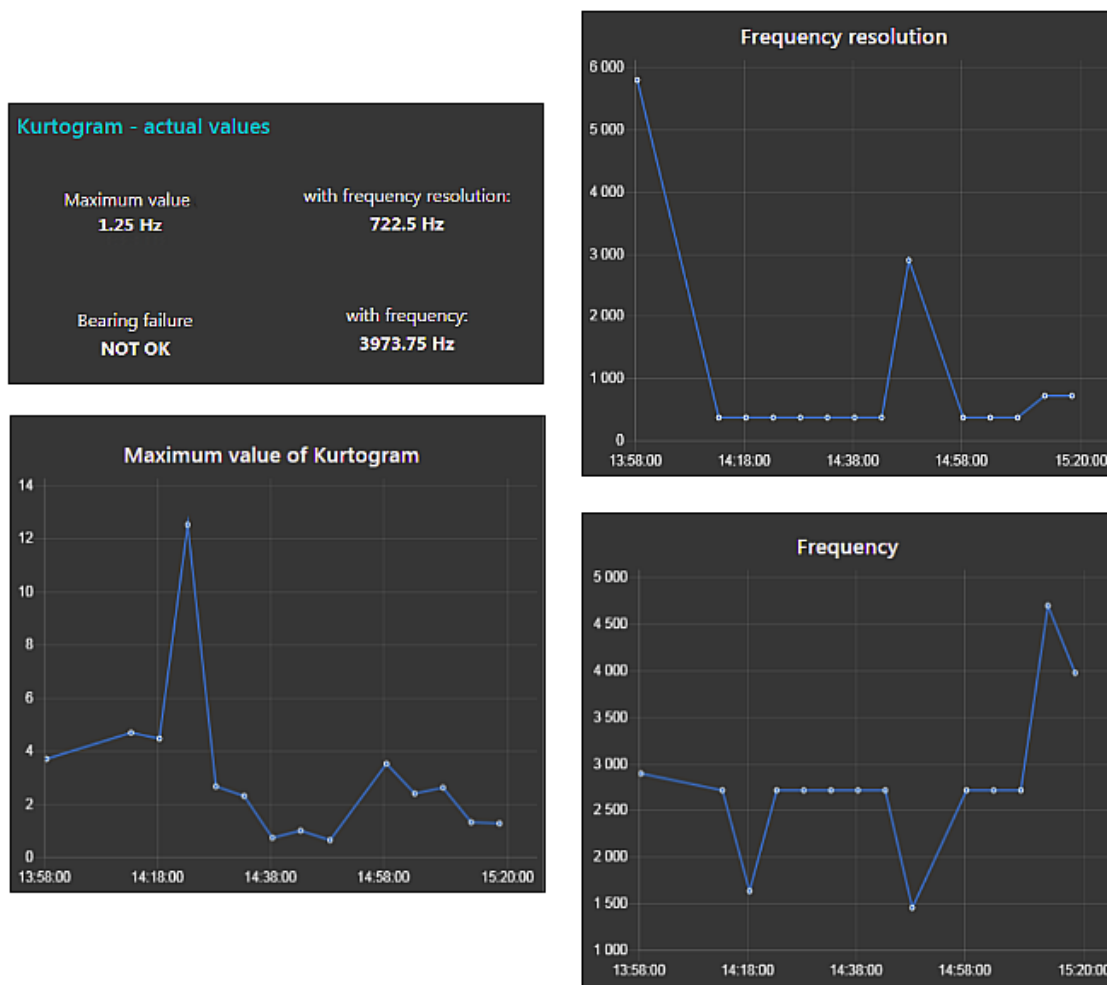


Obrázek 7.11: Časový průběh RMS, crest factoru a kurtosis ratio a jejich maximální hodnoty za dobu měření zobrazované pomocí měřidel.

Hodnoty z kurtogramu jsou zobrazeny na Obrázku 7.12. Je zde vidět, že poloha maxima v kurtogramu je stále v podobných místech, díky čemuž se frekvence a šířka pásma kurtogramu v čase tolik nemění. Maximum bylo nejčastěji nalezeno na nejvyšším levelu dekompozičního stromu (4). Zároveň došlo ke zvýšení samotných nalezených maximálních hodnot z kurtogramu oproti předchozímu měření. V tomto měření je největší maximální hodnota přibližně 12.5, zatímco v prvním měření na motoru bez porouchaného ložiska byla nejvyšší maximální hodnota kurtogramu přibližně 2.5.

Informativní email obsluze o možné poruše byl odeslán celkem dvanáctkrát, vždy když šířka pásma byla menší než 1 000. Struktura emailové zprávy je stejná jako v prvním měření.

Jelikož poslední hodnota šířky pásma je 722.5 Hz, tak indikátor možnosti poruchy ložisek udává "NOT OK".



Obrázek 7.12: Průběh maximální hodnoty kurtogramu, frekvence, kde byla nalezena a šířky pásma v čase a jejich poslední vypočtené údaje.

7.1.3 Závěr experimentu

Bylo provedeno nezávislé měření na dvou různých motorech. První měření bylo provedeno na motoru, kde se neměla vyskytovat žádná porucha. Druhé měření pak bylo uskutečněno na motoru, kde se naopak měla vyskytovat závada na ložisku. Tento rozdíl v motorech byl i při měření slyšitelný.

V druhém měření se příznaky crest factor i RMS pohybovaly ve značně vyšších hodnotách než v prvním měření. Hodnoty kurtosis ratio byli srovnatelné. Zároveň nalezená maxima v kurtogramu byla značně větší než v prvním měření.

Co se týče frekvence a šířky pásma kurtogramu, v prvním měření tyto veličiny vykazují výraznější kolísání, zatímco v druhém měření jsou hodnoty víceméně konstantní a šířka pásma se pohybuje více pod hranicí 1 000 Hz.

Při srovnání výsledků z prvního a druhého měření je vidět, že druhé měření bylo provedeno na motoru s blíže nespecifikovaným druhem poruchy, což byl očekávaný výsledek.

Detekce poruchy pouze za pomoci kurtogramu se opravdu projevila jako nedostačující. Bylo odesláno pět falešně alarmujících emailů a možné poruše při měření na motoru bez poruchy. Určovat indikátor na základě více příznaků by vedlo k lepšímu rozpoznání defektu na motoru. Není však vyloučené, že i na motoru, kde se žádná závada vyskytovat neměla,

opravdu žádná závada nebyla, avšak podle ostatních zobrazovaných příznaků a výsledků z druhého měření na porouchaném motoru je to velmi nepravděpodobné.

7.2 Druhý experiment

Podobně jako první experiment i druhý byl nejprve proveden na motoru bez poruchy (Motor 2) a poté na motoru, kde se měla vyskytovat porucha (Motor 1). Při druhém experimentu byla taktéž vzorkovací frekvence přibližně 11 kHz.

Na rozdíl od prvního experimentu byl v druhém indikátor aktivní v případě, že třikrát po sobě byla hladina kurtogramu, kde bylo nalezeno maximum větší než jedna, čili šířka pásma byla menší než 2 000 Hz a zároveň průměrný crest factor byl větší než 3.5.

Tento experiment byl proveden v jiný den, než první experiment.

Bohužel v průběhu tohoto experimentu při měření na motoru s porouchaným ložiskem došlo k závadě na samotném motoru, díky čemuž motor přestal běžet. Tuto závadu se nepodařilo odstranit. Z tohoto důvodu nebylo druhé měření dokončeno a nasbíraná data jsou neúplná.

7.2.1 Motor bez poruchy

První měření bylo uskutečněno na motoru, na kterém se nevyskytovala závada. Data která byla přijata sítí ThingsNetwork jsou vidět na Obrázku 7.13.

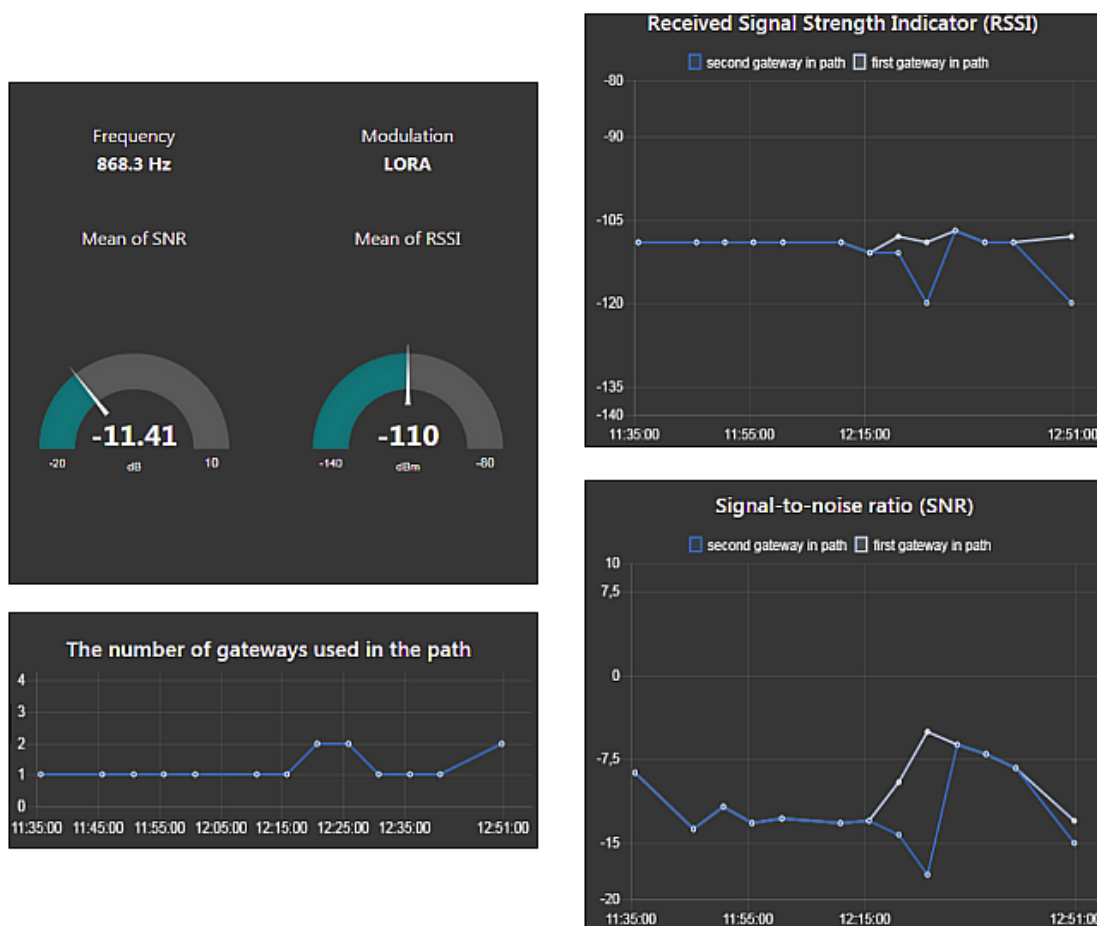
time	counter	port	payload
▲ 12:50:47	15	2	payload: 01 00 C4 F6 08 3F D9 E9 30 42 EB 00 6D 40 63 AB AA 3F
▲ 12:40:47	13	2	payload: 04 0F 92 82 B0 3F B6 BA 83 41 92 06 54 40 9F 9A A9 3F
▲ 12:35:47	12	2	payload: 04 06 54 3C 61 3F 53 9C CF 41 B8 48 35 40 05 A1 97 3F
▲ 12:30:47	11	2	payload: 04 0F 74 AA 2D 3F 27 55 00 42 9E 1E 45 40 24 38 A7 3F
▲ 12:25:47	10	2	payload: 03 08 3C 5F 1E 3F 3A EE 34 42 B1 F2 4A 40 C9 C8 A8 3F
▲ 12:20:46	9	2	payload: 04 09 C0 88 4D 3F 6ADA 44 42 F8 90 64 40 00 12 B6 3F
▲ 12:15:46	8	2	payload: 01 00 74 AD 00 3F 23 E5 40 42 56 E7 57 40 80 24 AB 3F
▲ 12:10:50	7	2	payload: 01 00 D4 DC 38 3F FAD 6 30 42 4C F3 4C 40 BE 04 AE 3F
▲ 12:00:46	5	2	payload: 04 0C 98 02 10 3F 88 D FAA 41 B5 8E 4F 40 DD 3B A2 3F
▲ 11:55:46	4	2	payload: 01 00 8C 62 3B 3F 04 E0 9A 41 E5 F0 66 40 E6 E1 AD 3F
▲ 11:50:46	3	2	payload: 04 06 E4 17 01 3F 95 04 10 41 07 A8 42 40 FC EB A0 3F
▲ 11:45:46	2	2	payload: 01 00 A4 61 8C 3F 77 C5 F1 40 D2 FB 89 40 96 51 C0 3F
▲ 11:35:45	0	2	retry payload: 03 00 E4 5B 14 3F ED 7F 86 3F 28 4D 83 40 A1 5E A5 3F

Obrázek 7.13: Data přijatá sítí ThingsNetwork. Je zde vidět čas přijetí daného rámce, pořadové číslo rámce a jeho obsah.

Opět došlo ke ztrátě několika rámců. Jmenovitě rámce číslo 1, 6, 14 a 16. Z monito-

rovacího zařízení byla data odeslána každých pět minut. Čas přijetí daných dat je vidět v levém sloupci.

Parametry spojení monitorovacího zařízení se sítí ThingsNetwork jsou zaznamenány na Obrázku 7.14. Oproti minulému experimentu se změnila frekvence z 868.5 Hz na 868.3 Hz. Průměrná hodnota SNR je -11.41 dB. Průměrné naměřené RSSI je -110 dBm. I v tomto měření byla několikrát použita více než jedna brána. V těchto případech se opět křivka v grafu rozvětví na dvě a ukazuje dvě nezávislé hodnoty dané veličiny naměřené na první a druhé bráně.



Obrázek 7.14: Parametry spojení. Počet bran v cestě a průměrná hodnota SNR a RSSI za dobu běhu motoru při měření (vlevo) a jejich změna v čase (vpravo).

Maximální hodnota kurtosis ratio byla 1.50. Maximální efektivní hodnota byla 49.21 rel $m \cdot s^{-2}$. Crest factor měl maximální hodnotu 4.31. Tyto hodnoty jsou zobrazeny spolu s časovým průběhem těchto příznaků na Obrázku 7.15.

Průběh efektivní hodnoty má velice podobný tvar i vypočtené hodnoty pro jednotlivé počty otáček, jako při prvním experimentu na neporouchaném motoru. Hodnota RMS se pohybovala v rozmezí od 0 do 50 rel $m \cdot s^{-2}$.

Crest factor i kurtosis ratio se také pohybují v podobném rozmezí jako v měření při stejných podmínkách u prvního experimentu. Crest factor se, až na výjimky, pohyboval v rozmezí 2.8 až 3.6. Kurtosis ratio byl v rozmezí přibližně 1.2 - 1.5.



Obrázek 7.15: Časový průběh RMS, crest factoru a kurtosis ratio a jejich maximální hodnoty za dobu měření zobrazované pomocí měřidel.

Šířka pásma kurtogramu, z tohoto měření, opět vykazuje střídání nalezeného maxima na širším a užším pásmu, jak je vidět na Obrázku 7.17. Podobnou, rychle se měnící, tendenci má i křivka reprezentující změnu vypočtené frekvence z polohy maxima kurtogramu v čase.

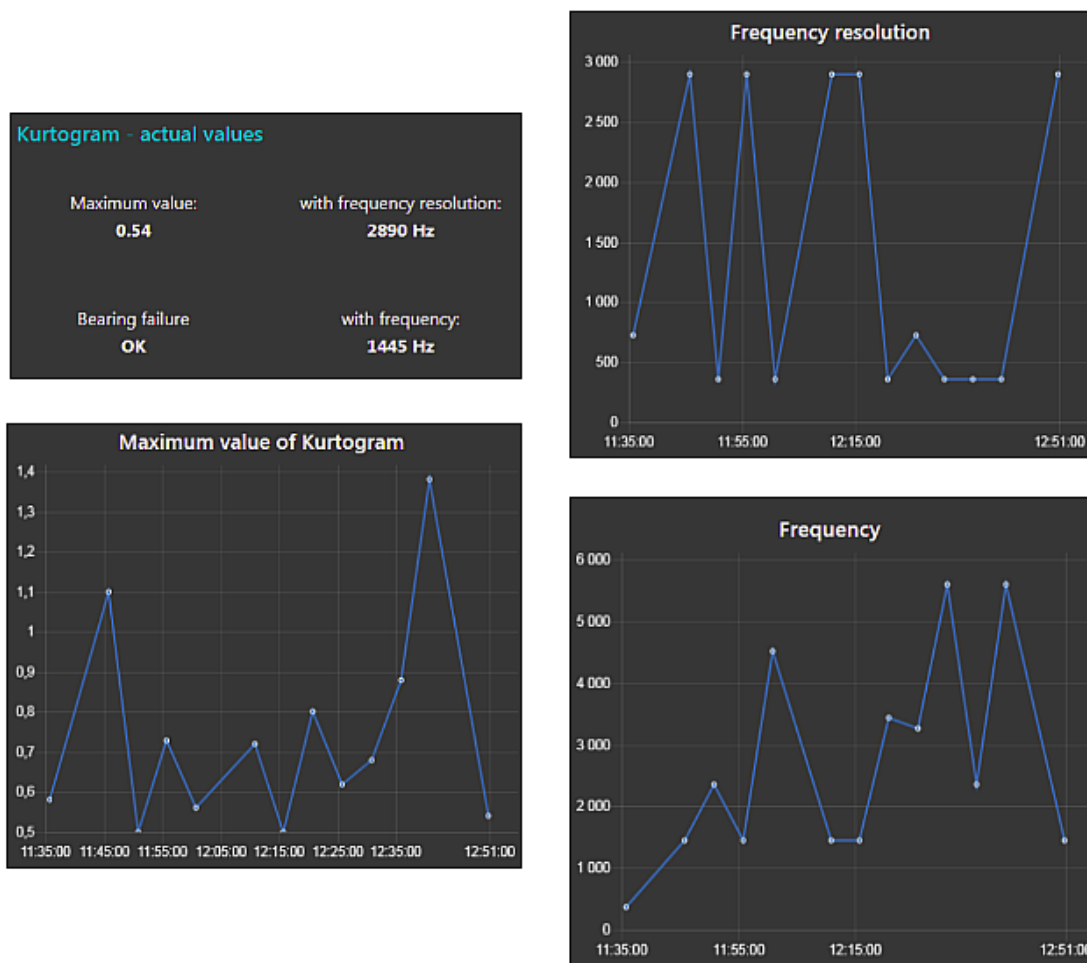
Maximální hodnoty kurtogramu se pohybují v hodnotách od 0.6 do 1.4.

Emailová zpráva o možné poruše byla v tomto případě odeslána celkem třikrát. Je to z toho důvodu, že třikrát po sobě bylo maximum kurtogramu nalezeno na hladině dekompozičního stromu větší než jedna a průměrná hodnota crest factoru byla v tuto dobu vyšší než 3.5.

Condition monitoring system - Podezření na poruchu monitorovaného přístroje.	12:40
Condition monitoring system - Podezření na poruchu monitorovaného přístroje.	12:35
Condition monitoring system - Podezření na poruchu monitorovaného přístroje.	12:30

Obrázek 7.16: Struktura odesílaných emailových zpráv s časem přijetí.

Poslední zobrazený indikátor ukazoval nepravděpodobnost poruchy, jelikož z posledních přijatých dat se vyčetlo maximum kurtogramu na hladině jedna, což porušilo podmínku tří po sobě jdoucích hodnot, indikující hladinu kurtogramu, kde je maximum, větších než jedna.



Obrázek 7.17: Poslední údaje o nalezeném maximu a průběh maximální hodnoty kurtogramu, frekvence, kde byla nalezena a příslušné šířce pásma v čase.

7.2.2 Motor s poruchou

V této části experimentu došlo při zvýšení rychlosti motoru na 3 500 otáček za minutu k závažné poruše, která zapříčinila následnou nečinnost monitorovaného motoru. Z tohoto důvodu bylo měření ukončeno a k následnému snižování a zvyšování rychlosti motoru již nedošlo.

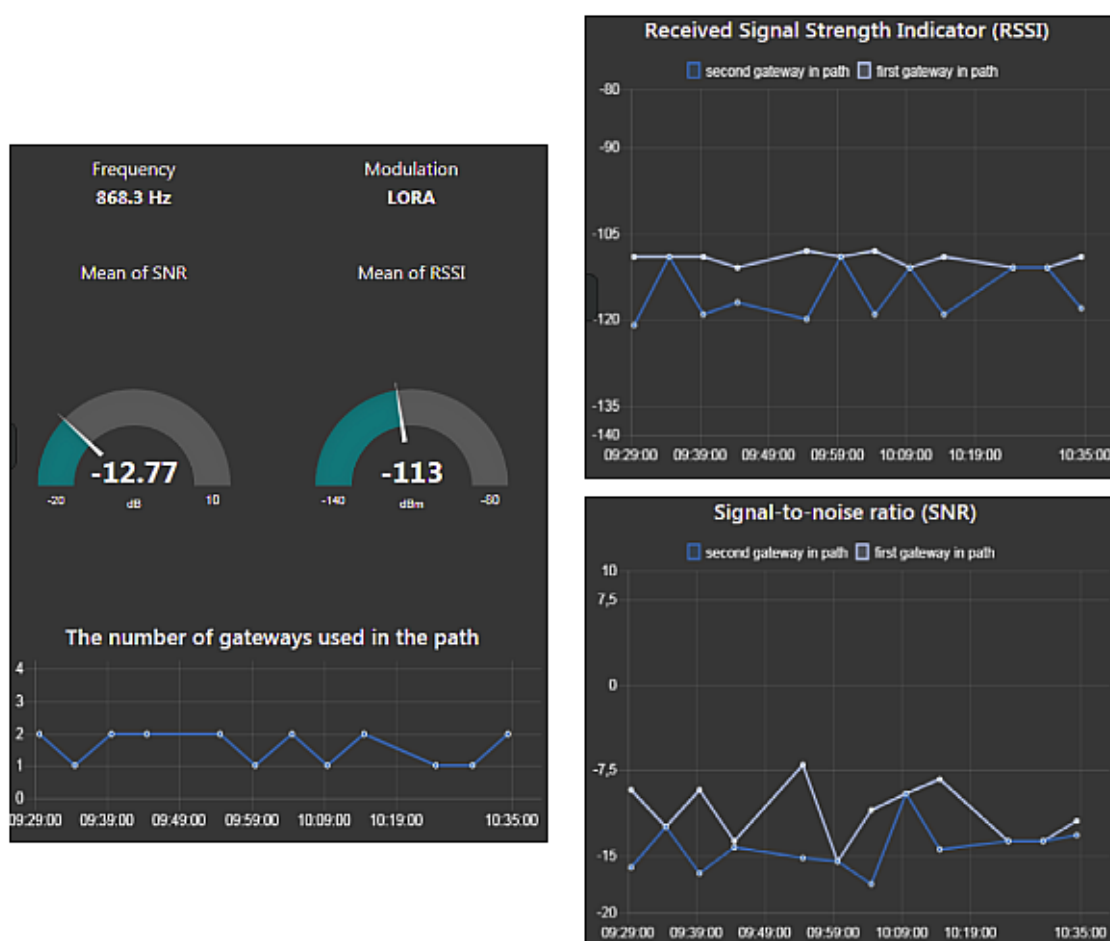
Z výpisu dat ze sítě ThingsNetwork je vidět, že bylo přijato celkem deset různých rámců s daty. Poslední čtyři přijaté rámce jsou naměřeny při vypnutém motoru (Obrázek 7.18).

Na Obrázku 7.19 jsou zachyceny informace o spojení. Průměrná hodnota SNR byla v tomto případě -12.77 dB a průměrné RSSI bylo -113 dBm. Při tomto měření bylo výrazně častěji spojení provedeno přes dvě brány.

Je pozoruhodné, že i přes to, že bylo sítí ThingsNetwork přijato pouze deset zpráv s daty, tak uživatelské rozhraní nástroje Node-RED zobrazuje v grafech dvanáct přijatých a naparsovaných rámců s daty. Jmenovitě se jedná o data přijatá přibližně v 9:54 a v 9:59. Tyto dva rámce nejsou ve výpisu dat přijatých do sítě ThingsNetwork.

time	counter	port	payload
10:34:33	13	2	payload: 03 02 40 69 48 3E C7 6B 87 3F 3DC15A 40 12 A1 9E 3F
10:29:33	12	2	payload: 00 00 E4 6C 8F 3F 2D 75 81 3F 56 A4A8 40 9F F6 9F 3F
10:24:33	11	2	payload: 00 00 D4EC 9F 3F F5 C5 7F 3F FC EE AB 40 AA 50 DA 3F
10:14:32	9	2	payload: 01 00 58 72 1A 3F 04 A97D 3F 1A 8B 81 40 B6 17 B9 3F
10:09:32	8	2	payload: 04 09 18 07 68 3F 49 2B F9 42 13 BB 82 40 EE 54 BC 3F
10:04:32	7	2	payload: 02 08 68 02 5D 3F 94 7A B0 42 9BCC 4B 40 78 0B A0 3F
09:44:32	3	2	payload: 04 07 79 82 9B 40 35 1A EB 41 F4 41 60 40 3C 25 B5 3F
09:39:32	2	2	payload: 03 06 AAA6 1C 41 F0 15 E6 41 52 82 8C 40 64 08 F0 3F
09:34:31	1	2	payload: 01 08 34 7F 64 3F 9A 02 7D 3F ADAC 84 40 84 7B C1 3F
09:29:31	0	2	retry payload: 01 00 9C 81 44 3F 7C 84 85 3F BECC 9B 40 8B 7A 93 3F

Obrázek 7.18: Přijatá data sítě Thingsnetwork.



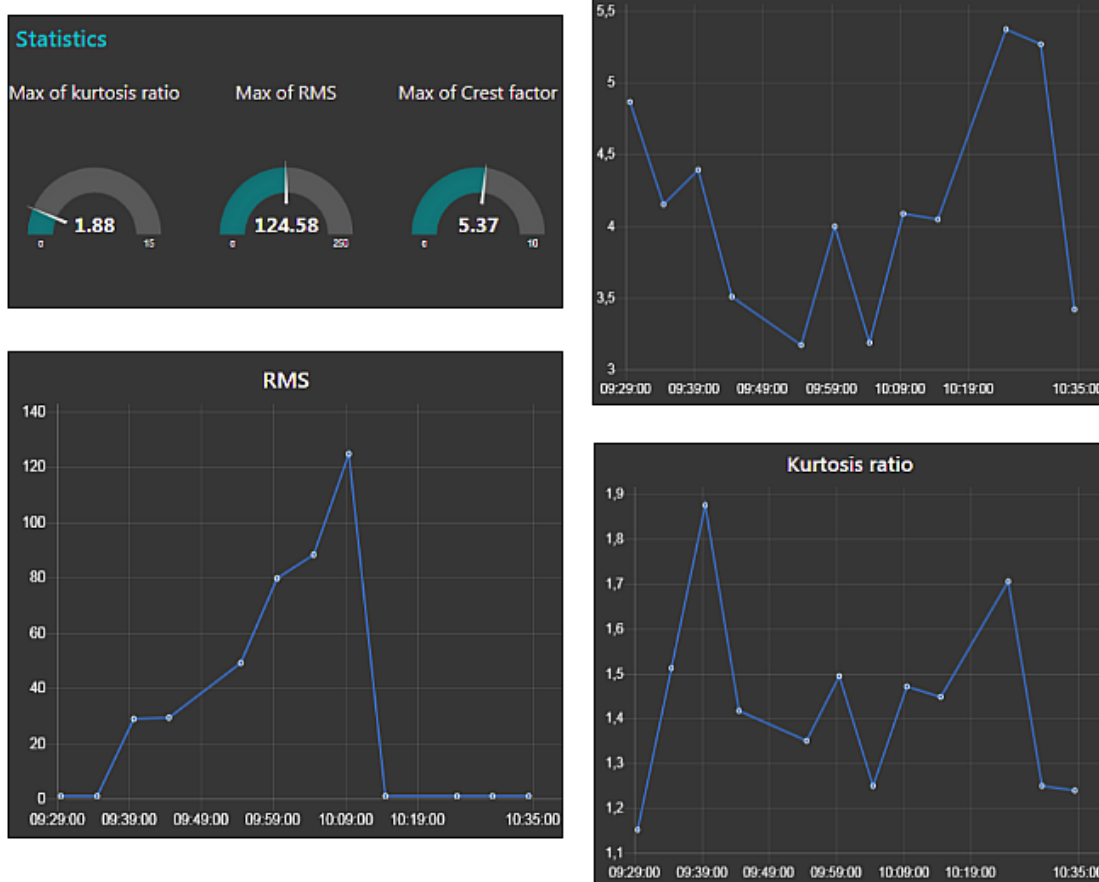
Obrázek 7.19: Parametry spojení. Počet bran v cestě a průměrná hodnota SNR a RSSI za dobu běhu motoru při měření (vlevo) a jejich změna v čase (vpravo).

Náhlé vypnutí motoru také reflektuje Obrázek 7.20. Jak je vidět na průběhu RMS, přibližně v čase 10:09 - 10:14 došlo k závažné poruše, která zapříčinila snížení rychlosti motoru na 0 otáček za minutu.

Velikost efektivní hodnoty se v tomto případě pohybuje v o něco nižších hodnotách, než při prvním experimentu, kdy byly podmínky měření obdobné, avšak stále jsou podstatně vyšší, než při měření na motoru bez závady. Nejvyšší hodnota RMS byla $124.58 \text{ rel } m \cdot s^{-2}$.

Nejvyšší hodnota crest factoru byla 5.37. Crest factor se pohyboval v hodnotách od 3.2 do 4.4, nepočítaje dobu, kdy byl monitorovací stroj vypnutý (čili první dvě a poslední čtyři hodnoty).

Kurtosis ratio mělo maximální hodnotu za dobu běhu 1.88. Rozmezí hodnot kurtosis ratio bylo 1.2 - 1.88.



Obrázek 7.20: Časový průběh RMS, crest factoru a kurtosis ratio a jejich maximální hodnoty za dobu měření zobrazované pomocí měřidel.

Na Obrázku 7.21 je vidět časový průběh frekvence, šířky pásma kurtogramu a maximálních hodnot, nalezených ve vypočteném kurtogramu. Je zde vidět, že šířka pásma byla po dobu běhu motoru (v době nenulových otáček) menší než 2 000 Hz. Celkem čtyřikrát byla odeslána emailová zpráva na určenou emailovou adresu.

Neočekávané vypnutí motoru mělo sice za následek zvětšení crest factoru, avšak maximum kurtogramu bylo nalezeno na nižší hladině kurtogramu. Tudíž indikátor nenaznačoval možnou poruchu.



Obrázek 7.21: Poslední údaje o nalezeném maximu a průběh maximální hodnoty kurtogramu, frekvence, kde byla nalezena a příslušné šířce pásma v čase.

7.2.3 Závěr experimentu

I přes závažnou poruchu jednoho z monitorovaných motorů, kdy motor přestal pracovat, je z dostupných dat patrné podobné chování, jako v prvním experimentu. Měření na motoru s vadným ložiskem vykazovalo vyšší RMS i crest factor než měření na druhém motoru. Navíc šířka pásma, vypočtená z polohy maxima kurtogramu, se pohybovala v užších pásmech, podobně jako v prvním experimentu.

Tyto výsledky opět dokazují, že druhé měření bylo provedeno na motoru s defektem, což mohl být i důvod naprostého selhání motoru.

Indikátor poruchy založený na šířce pásma kurtogramu a na hodnotě crest factoru se jeví jako přijatelný. Avšak nebylo vhodné používat průměrný crest factor, který je výrazně ovlivněn hodnotami naměřenými při nulových otáčkách. Bylo by vhodnější, místo průměrné hodnoty, kontrolovat například jen poslední tři hodnoty crest factoru.

8 Závěr

V této práci bylo vytvořeno zařízení pro monitorování stavu průmyslových zařízení pomocí analýzy vibrací.

K účelu měření a výpočtu příznaků byl použit mikrokontrolér STM32L073RZ-Nucleo na desce P-NUCLEO-LRWAN1, jejíž součástí je i modul pro komunikaci prostřednictvím LoRa rozhraní. K tomuto kitu byl připojen akcelerometr ADXL1002Z, pro naměření vibračních dat.

Byla zprovozněna komunikace pomocí bezdrátové technologie LoRa, používající se v internetu věcí. Jako aktivační metoda byla použita metoda ABP (Activation By Personalization) pro rychlejší inicializaci spojení. Pomocí této technologie monitorovací zařízení odesílá vypočtené příznaky do sítě ThingsNetwork.

Dále byl naimplementován fast kurtogram, jako pokročilejší metoda zpracování signálu, umožňující detekovat nestacionární složku v signálu. Výpočet fast kurtogramu byl testován a vizualizován v prostředí Matlab. Kromě fast kurtogramu byl také naimplementován výpočet efektivní hodnoty pro určení amplitudy. Dalšími vypočtenými příznaky byly crest factoru a kurtosis ratio. Tyto příznaky slouží k odhadu možné poruchy stroje.

Data z akcelerometru byla získána pomocí analogově-digitálního převodníku. Tato data byla normalizována a použita pro výpočet příznaků, které byly dále odeslány do sítě ThingsNetwork prostřednictvím rozhraní LoRa.

Monitorovací zařízení zároveň vypisuje diagnostická data přes rozhraní UART na hyperterminál připojeného počítače.

Navíc byla vytvořena aplikace pomocí programovacího nástroje node-RED. Aplikace poskytuje uživatelské rozhraní, které obsahuje grafické znázornění příznaků a jejich průběh v čase. Navíc zobrazuje technické parametry spojení. Kromě snadno dostupného a intuitivního uživatelského rozhraní je naimplementováno odesílání informační emailové zprávy na předem danou emailovou adresu v případě detekce zhoršujícího se stavu monitorovaného zařízení.

Zařízení bylo testováno na simulátoru vibrací. Byli provedeny dva nezávislé experimenty, vždy na dvou různých motorech. První z monitorovaných motorů obsahoval závadu na ložisku. Druhý motor byl bez známého defektu. Jak je z výsledků patrné, efektivní hodnota a crest factor mají výrazně vyšší hodnoty na datech získaných z motoru s poruchou než při měření na motoru bez poruchy. Hodnoty kurtosis ratio jsou v našem případě srovnatelné pro všechna měření. Zároveň maximum kurtogramu se projevuje na vyšších hladinách dekompozičního proudu při existenci poruchy na ložisku.

V našem případě je indikována porucha pouze na základě posledních tří hodnot crest factoru a hladiny kurtogramu, kdy bylo nalezeno maximum.

Pro přesnější indikátor poruchy by bylo třeba provést sadu nezávislých experimentů na motoru, kde je přítomna vada a na motoru, kde se žádná závada nevyskytuje. Bohužel toto nemohlo být provedeno, jelikož při druhém experimentu motor s porouchaným ložiskem přestal pracovat, tudíž nemohli být uskutečněny další experimenty. V nástroji Node-RED se však velice snadno dá podmínka pro aktivaci indikátoru poruchy pozměnit na základě získaných dat.

Při testování se ukázalo, že na rozdíl od klasických komunikačních sítí nezaručuje síť ThingsNetwork, která je orientována na minimalizaci energie potřebné k přenosu, stoprocentní doručení. Úspěšnost doručení kolísala mezi jednotlivými dny od přenosu jedné z šesti zpráv až po doručení všech.

Literatura

- [1] R. B. Randall, *Vibration-based condition monitoring*. John Wiley & Sons, Ltd, 2011.
- [2] Camphus.com, “Motor vibration monitoring.” <http://camphus.com/motor-vibration-monitoring/kirby-risk-corporation.html>, 2018. [Online; Dostupné 20-5-2018].
- [3] Nelson Publishing, Inc, “Motor sensor tells when it’s time for a service.” <http://www.designfax.net/cms/dfx/opens/article-view-dfx.php?nid=4&bid=562&et=featurearticle&pn=02>, 2016. [Online; Dostupné 20-5-2018].
- [4] ABB, “Smart sensor for motors.” <http://new.abb.com/motors-generators/service/advanced-services/smart-sensor/smart-sensor-for-motors>, 2018. [Online; Dostupné 20-5-2018].
- [5] M. Kreidl, R. Šmíd, *Technická diagnostika*. Nakladatelství BEN, 2006.
- [6] P. Pohanka, “Internet věcí.” <http://i2ot.eu/internet-of-things/>, 2017. [Online; Dostupné 20-5-2018].
- [7] A. Vojáček, “Sigfox - princip, struktura, protokol, použití.” <https://vyvoj.hw.cz/sigfox-princip-struktura-protokol-pouziti.html>, 2017. [Online; Dostupné 20-5-2018].
- [8] Wikipedie, “Narrowband IoT.” https://en.wikipedia.org/wiki/Narrowband_IoT, 2018. [Online; Dostupné 20-5-2018].
- [9] Wikipedie, “ZigBee.” <https://cs.wikipedia.org/wiki/ZigBee>, 2017. [Online; Dostupné 20-5-2018].
- [10] Wikipedie, “Z-Wave.” <https://en.wikipedia.org/wiki/Z-Wave>, 2018. [Online; Dostupné 20-5-2018].
- [11] České Radiokomunikace a.s., “Objevte svět IoT.” <https://www.cra.cz/objevte-svet-iot>, 2017. [Online; Dostupné 20-5-2018].
- [12] STMicroelectronics, “P-NUCLEO-LRWAN1.” <http://www.st.com/en/evaluation-tools/p-nucleo-lrwan1.html>, 2018. [Online; Dostupné 20-5-2018].
- [13] STMicroelectronics, “NUCLEO-L073RZ.” http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo/nucleo-l073rz.html, 2018. [Online; Dostupné 20-5-2018].

- [14] Wikipedie, “Flash memory.” https://en.wikipedia.org/wiki/Flash_memory, 2018. [Online; Dostupné 20-5-2018].
- [15] M. Krynický, “Flash paměti.” <http://www.realisticky.cz/ucebnice/02%20Fyzika%20S%C5%A0/04%20Elekt%C5%99ina%20a%20magnetismus/08%20%C4%8Cinnost%20elektrick%C3%BDch%20a%20elektronick%C3%BDch%20za%C5%99%C3%ADzen%C3%AD/01%20Flash%20pam%C4%9Bti.pdf>. [Online; Dostupné 20-5-2018].
- [16] Analog Devices, Inc, “EVAL-ADXL100X.” <http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-adxl100x.html>, 2018. [Online; Dostupné 20-5-2018].
- [17] The Things Network, “Documentation.” <https://www.thethingsnetwork.org/wiki/Home>, 2018. [Online; Dostupné 20-5-2018].
- [18] LoRa Alliance, “What is the lorawan specification.” <https://loro-alliance.org/about-lorawan>, 2018. [Online; Dostupné 20-5-2018].
- [19] G. K. Singh and S. A. S. Al Kazzaz, “Induction machine drive condition monitoring and diagnostic research - a survey,” *Electric Power Systems Research*, vol. 64, pp. 145–158, 2003.
- [20] V. Bednařík, “Diagnostika poruch rotoru asynchroních motorů,” Master’s thesis, Vysoké učení technické v Brně, 2012.
- [21] J. Antoni, “Fast computation of the kurtogram for the detection of transient faults,” *Mechanical Systems and Signal Processing*, vol. 21, pp. 108–124, 2007.
- [22] J. Vass, R. Šmíd, R.B. Randall, “Avoidance of speckle noise in laser vibrometry by the use of kurtosis ratio: Application to mechanical fault diagnostics,” *Mechanical Systems and Signal Processing*, vol. 22, p. 647–671, 2008.
- [23] STMicroelectronics, “STM32CubeL0.” http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-mcu-packages/stm32cubel0.html, 2018. [Online; Dostupné 20-5-2018].
- [24] J. Antoni, “Fast kurtogram.” <https://www.mathworks.com/matlabcentral/fileexchange/48912-fast-kurtogram?ue>, 2015. [Online; Dostupné 20-5-2018].
- [25] Node-RED, “Flow-based programming for the internet of things.” <https://nodered.org/>. [Online; Dostupné 20-5-2018].
- [26] Wikipedie, “Received signal strength indication.” https://cs.wikipedia.org/wiki/Received_Signal_Strength_Indication, 2017. [Online; Dostupné 20-5-2018].

- [27] Wikipedie, “Signal-to-noise ratio.”
https://en.wikipedia.org/wiki/Signal-to-noise_ratio, 2018. [Online;
Dostupné 20-5-2018].

A Obsah CD

```
/
├── Condition_monitoring_system
│   ├── readme.txt ..... Informace o obsahu přikládaného CD
│   ├── DP_text
│   │   ├── DP_Buricova_2018.pdf .....Elektronická verze textu diplomové práce
│   │   └── DP_Buricova_2018.zip ..... Zdrojový kód textu k diplomové práci
│   └── source_codes
│       ├── kurtogram
│       │   ├── demo_Fast_kurtogram.m ... Skript v Matlabu pro spuštění výpočtu
│       │   │   kurtogramu a dopočítání dalších příznaků
│       │   └── fast_kurt.m ..... Skript s výpočtem a vizualizací kurtogramu
│       ├── CMS
│       │   └── Condition_monitoring_system.zip Výsledná aplikace v jazyce C
│       └── node_red
│           └── node_red_application.json .Aplikace vytvořená pomocí nástroje
│               Node-RED uložená ve formátu json
```