

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická
Katedra materiálů a technologií

DIPLOMOVÁ PRÁCE

Návrh a realizace distribuovaného CMD systému

Autor práce: **Bc. Václav Hlinomaz**
Vedoucí práce: **Ing. Jan Leffler**

2024

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Václav HLINOMAZ**
Osobní číslo: **E22N0057P**
Studijní program: **N0714A060013 Elektronika a informační technologie**
Specializace: **Výkonová elektronika**
Téma práce: **Návrh a realizace distribuovaného CMD systému**
Zadávací katedra: **Katedra materiálů a technologií**

Zásady pro vypracování

1. Identifikujte kritické subsystémy elektrických strojů točivých z hlediska poruch.
2. Zvolte relevantní signály pro hodnocení stavu, provozních podmínek, spolehlivosti a zbývajících životnosti elektrických strojů.
3. Zvolte vhodné metody pro analýzu získaných dat a stanovení diagnózy a prognózy.
4. Navrhněte distribuovaný CMD systém pro neinvazivní sledování zvolených relevantních signálů a jejich hodnocení.
5. Fyzicky sestavte tento CMD systém, proveďte zapojení všech komponentů, zahajte komunikační procedury, v případě potřeby kalibrujte senzory, implementujte vyhodnocovací algoritmy a vytvořte grafické uživatelské rozhraní.
6. Proveďte validační měření k ověření správnosti funkce.
7. Otestujte celkovou funkční integritu tohoto CMD systému.

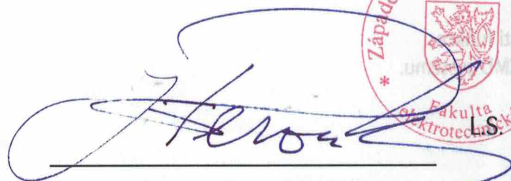
Rozsah diplomové práce: **40 – 60**
Rozsah grafických prací:
Forma zpracování diplomové práce: **elektronická**

Seznam doporučené literatury:

1. STONE, Greg C., et al. Electrical insulation for rotating machines: design, evaluation, aging, testing, and repair. John Wiley & Sons, 2004.
2. TAVNER, Peter; RAN, Li; PENMAN, Jim. *Condition monitoring of rotating electrical machines*. IET, 2008.
3. MENTLÍK, Václav. Diagnostika elektrických zařízení. Praha: BEN – technická literatura, 2008. ISBN 978-80-7300-232-9.
4. MENTLÍK, Václav. Spolehlivostní aspekty elektrotechnologie. Praha: BEN – technická literatura, 2011. ISBN 978-80-7300-412-5.

Vedoucí diplomové práce: **Ing. Jan Leffler**
Katedra materiálů a technologií

Datum zadání diplomové práce: **6. října 2023**
Termín odevzdání diplomové práce: **24. května 2024**



Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan



Prof. Ing. Aleš Hamáček, Ph.D.
vedoucí katedry

V Plzni dne 6. října 2023

Abstrakt

Tato diplomová práce se zabývá návrhem vibrodiagnostického systému pro krátkodobý a střednědobý monitoring se zaměřením na využití MEMS akcelerometrů s cílem minimalizace finanční náročnosti měřicího zařízení včetně senzorů k němu připojitelných. Teoretická část se zabývá obecnou problematikou diagnostiky a poruch elektrických strojů. Praktická část poté zahrnuje výběr vhodných hardwarových součástí, návrh DPS pro vysílací část a také kompletní návrh softwaru pro celý systém. Cílem práce je oživení CMD systému a jeho zhodnocení na základě provedení validačních měření a testování.

Klíčová slova

Vibrace, Analýza, Monitoring, Diagnostika, MEMS, Elektrický stroj

Abstract

This thesis deals with the design of an alternative vibrodiagnostic system for short-term and medium-term monitoring, focusing on the use of MEMS accelerometers to make the system and attached sensors cost-effective.. The theoretical part addresses the general issues of diagnostics and faults in electrical machines. The practical part includes the selection of suitable hardware components, the design of the PCB for the transmitter device, and the complete software design for the entire system. The aim of the thesis is to build the CMD system and evaluate it based on the validation measurements and testing.

KeyWords

Vibration, Analysis, Monitoring, Diagnostics, MEMS, Electric machine

Poděkování

Rád bych poděkoval především panu Ing. Janu Lefflerovi za ochotu a odbornou pomoc při vedení práce. Dále mé poděkování patří panu Ing. Oldřichu Turečkovi, Ph.D. za poskytnutí možnosti provedení měření na vibrační stoličce a za odbornou pomoc při tomto měření. Rovněž děkuji panu Ing. Zdeňku Frankovi za umožnění provedení měření na soustrojí s devítifázovým strojem.

Obsah

Úvod.....	- 1 -
1 Teoretická část.....	- 2 -
1.1 Diagnostické přístupy a metody.....	- 2 -
1.2 Kritické subsystemy z hlediska poruch.....	- 2 -
1.3 Signály pro vyhodnocování stavu, spolehlivosti a zbývající životnosti točivých strojů - degrační faktory.....	- 4 -
1.4 Snímače vibrací.....	- 6 -
1.4.1 Možnosti montáže akcelerometrů na měřený objekt.....	- 8 -
1.5 Metody analýzy dat a stanovení diagnózy a prognózy ve vibrodiagnostice..	- 9 -
1.5.1 Vyhodnocení v časové oblasti.....	- 9 -
1.5.2 Vyhodnocení ve frekvenční oblasti.....	- 10 -
2 Návrh.....	- 13 -
2.1 Koncepce systému.....	- 13 -
2.2 Návrh distribuovaného CMD systému.....	- 14 -
2.2.1 Obecné schéma systému.....	- 14 -
2.2.2 Volba komponent.....	- 15 -
3 Realizace a ladění.....	- 19 -
3.1 Hardware.....	- 19 -
3.1.1 Vysílač.....	- 19 -
3.1.2 Přijímač.....	- 26 -
3.2 Software.....	- 27 -
3.2.1 Vysílač.....	- 30 -
3.2.2 Přijímač.....	- 35 -
3.2.3 PC – MATLAB aplikace.....	- 39 -
3.3 První validační měření.....	- 44 -
3.4 Upravená koncepce systému.....	- 44 -
3.5 Významné části softwaru upraveného systému.....	- 46 -
3.6 Uživatelské rozhraní Webového Serveru.....	- 48 -
4 Testování.....	- 50 -

4.1	Druhé validační měření	- 50 -
4.2	Třetí validační měření	- 51 -
4.3	Čtvrté validační měření	- 53 -
4.4	Testování funkční integrity	- 54 -
	Zhodnocení a závěr	- 57 -
	Literatura.....	- 59 -
	Přílohy.....	I

Seznam symbolů a zkratek

Symboly

\ddot{a}	m/s ²	Zrychlení
g	m/s ²	Gravitační zrychlení
v	m/s	Rychlost
s	m	Dráha (výchylka)
m	kg	Hmotnost
\vec{F}	N	Síla
B	T	Magnetická indukce
P_j	W	Jouleovy ztráty
U	V	Elektrické napětí
I	A	Elektrický proud
ϵ_0	F/m	Permitivita vakua
ϵ_r		Relativní permitivita materiálu
f	Hz	Frekvence
T_s, T_{vz}, dt	s	Doba vzorkování, časový krok
θ	rad	Úhel natočení vektoru

Zkratky

ACK	Acknowledge Potvrzení (většinou příjmu dat)
A/D	Analog/Digital Spojitě/Číslicový (převodník)
AP	Access point Přístupový bod (v síti)
CAD	Computer aided design Počítačem podporované projektování
CMD	Condition monitoring and diagnostics Monitorování stavu a diagnostika
BMS	Battery management systém Systém řízení baterie
DPS	Deska plošných spojů
DUT	Device under test Testované zařízení
EEPROM	Electrically Erasable Programmable Read-Only Memory elektricky mazatelná programovatelná paměť pouze pro čtení
EMC	Electromagnetic compatibility Elektromagnetická kompatibilita

FIFO	First in, First out Elektronická paměť
G	Gravitační zrychlení 9,81 m/s ²
GPIO	General purpose input/output vstup/výstup obecného použití
GPT	Generative Pre-trained Transformer Pokročilý model umělé inteligence
HW	Hardware Fyzická část zařízení
HTML	HyperText Markup Language Značkovací jazyk pro hypertext
FDM	Fused Deposition Modeling Modelování depozice taveniny
FFF	Fused Filament Fabrication Metoda 3D tisku (stejná jako FDM)
ID	Identifier Jedinečný identifikátor
IDE	Integrated Development Environment Integrované vývojové prostředí
IMU	Inertial measurement unit Jednotka měření inerciálních veličin
I ² C	Inter-Integrated circuit Sériové komunikační rozhraní
IoT	Internet of things Internet věcí
MAC	Media Access Control Fyzická adresa zařízení
LCD	Liquid crystal display Displej s technologií tekutých krystalů
LDO	Low dropout regulátor Lineární regulátor napětí
LED	Light emitting diode Svítivá dioda

LSB	Least significant bit Nejméně významný bit
MEMS	Micro electromechanical system Elektromechanický systém vytvořený zpravidla na jednom čipu
PC	Personal computer Osobní počítač
PID	Proporcional – Integral – Derivative Proporcionálně integračně derivační regulátor
RX	Receiver Přijímač
SMD	Surface mounted device Součástka určená k montáži na povrch DPS
SCL	Serial clock Hodinový signál sběrnice I ² C
SDA	Serial data Označení vodiče I ² C přenášejícího data
SPI	Serial peripheral interface Sériové rozhraní pro komunikaci procesoru s ostatními zařízeními
SPIFFS	SPI Flash file systém Souborový systém pro flash paměť na SPI
TEAM	Thermal, Electrical, Ambient, Mechanical stresses Tepelné, Elektrické a mechanické namáhání a namáhání okolními vlivy
THT	Through hole technology Technologie montáže součástek do DPS
TWI	Two-wire interface Alternativní označení pro I ² C
TX	Transmitter Vysílač
UART	Universal Asynchronous receiver-transmitter Rozhraní pro asynchronní sériovou komunikaci
UPS	Uninterruptible Power Supply Záložní napájecí zdroj
USB	Universal serial bus Univerzální sériové komunikační rozhraní

Úvod

Vibrodiagnostika je jedním z nejvýznamnějších oborů technické diagnostiky. Nachází totiž uplatnění na nejrůznějších rotačních i jiných pohybujících se uzlech strojů vyznačujících se mechanickým chvěním (kmitavým pohybem, vibracemi). [1] To je velmi důležité, zvláště pokud se jedná o stroje, jejichž porucha by mohla znamenat například výpadek dodávky elektrické energie a v horším případě i neopravitelné poškození celého stroje. Netřeba dodávat, že takové poruchy vedou velmi vysokým finančním nákladům za následné opravy. Z toho důvodu je výhodné na taková zařízení nasadit monitorovací a diagnostické systémy, které jsou schopné sledovat aktuální provozní hodnoty a odhalit blížící se poruchu a umožní tak s dostatečným předstihem naplánovat odstávku a údržbu.

Moderní vibrodiagnostické systémy jsou často velmi drahé. Jejich cena bývá sice v porovnání s cenou oprav v případě poruchy většinou mizivá, ale přesto existují i provozy, kde by se úspora mohla hodit. Toto je zvláště významné, pokud je nutné provádět diagnostiku na mnoha místech testovaného zařízení. Například se může jednat o výrobní linku, která zahrnuje více různých motorů, převodovek, či pásových dopravníků a profesionální měřicí zařízení o mnoha vstupech by svou cenou nemuselo dosahovat požadované návratnosti investice.

Naproti tomu moderní technologie výroby jednoduchých zařízení, potenciálně využitelných pro vibrodiagnostiku však v posledních letech značně postoupila a do popředí se dostávají mnohá levná elektromechanická zařízení vytvořená na jednom čipu, známá jako MEMS.

Hlavní myšlenkou této práce je proto navržení vibrodiagnostického systému s využitím levných MEMS akcelerometrů a otestování jeho funkčnosti v praxi.

Cílem by pak mělo být zhodnocení funkce navrženého měřicího zařízení a vyvození závěru, zda je tato metoda měření vibrací v praxi vyhovující a jaké poskytuje výhody a nevýhody v porovnání s konvenčními diagnostickými metodami.

1 Teoretická část

1.1 Diagnostické přístupy a metody

Existuje mnoho metod a přístupů sloužících k provádění diagnostiky různých strojů. Tyto metody mají za úkol *sledování života technických zařízení – provozní diagnostika*. V této oblasti má význam nejen sledování vývoje parametrů zařízení, ale také ukládání dat či tvorba cenných databází hodnot popisujících vlastní vývoj chování systémů. Na základě těchto informací pak lze zpracovat předpověď chování systému v dalším období. [9] Toto nazýváme elektrotechnologická prognostika. [9]

Rozeznáváme dva základní diagnostické přístupy: Fenomenologický, kde nás zajímá, jak diagnostikovaný objekt reaguje na vnější diagnostické signály a strukturální přístup kde nás zajímá, co se děje ve struktuře diagnostikovaného systému. [9]

Dále lze diagnostické metody dělit podle jejich destruktivnosti. Ačkoliv destruktivní metody mají zpravidla značně vyšší výpovědní hodnotu o diagnostikovaném systému, logicky většinou nedává smysl jej aplikovat na složitější a dražší zařízení. Pro ně volíme jiné diagnostické metody, které jsou nedestruktivní. [9]

Další významné dělení přístupů je z hlediska postupu samotné diagnózy. Zde máme dva přístupy a to on-line a off-line. Rozdíl tkví v tom, zda je během testování dané zařízení v provozu, či nikoliv. [9, 10]

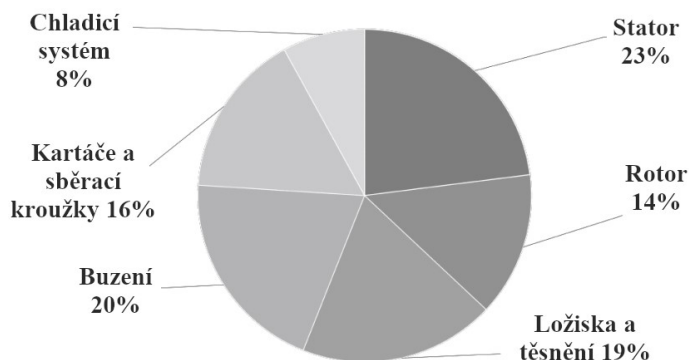
Práce využívá nedestruktivní online fenomenologický diagnostický přístup.

1.2 Kritické subsystémy z hlediska poruch

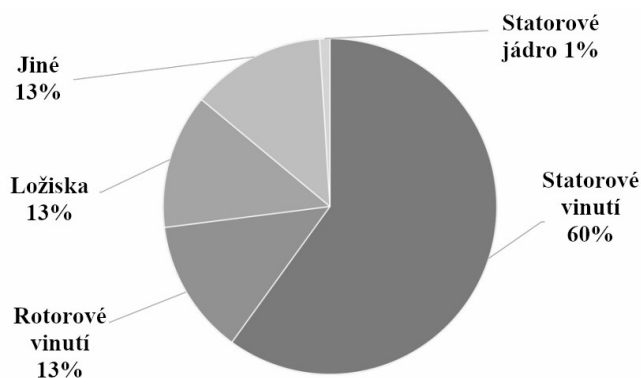
Aby bylo možné účinně monitorovat stav zařízení a předpovídat jeho poruchy je nutno definovat jeho kritické subsystémy. Tyto subsystémy jsou zcela zásadní pro správnou funkci zařízení a jejich selhání znamená většinou kritickou poruchu, která vede zpravidla k úplné ztrátě funkčnosti zařízení, v krajním případě k jeho nevratnému poškození.

Mohlo by se zdát, že nejkritičtějším subsystémem točivých strojů bude jejich hřídel. Hřídel je vždy uložen v ložiskách, kde vykonává otáčivý pohyb. Působí na něj tedy stejně jako na rotor velmi silné odstředivé síly. Pokud je hřídel ohnutý nebo je jakýmkoliv způsobem nevyvážený rotor, pak vždy při provozu zařízení vznikají kmity odpovídající první harmonické otáčivé rychlosti a jejím násobkům, avšak tímto tématem se podrobněji zabývá citace článku [1] v kapitole 1.5. Velmi přesně lze tyto problémy odhalovat pomocí metody měření výchylky v časové oblasti a pomocí kinematické dráhy hřídele. Tato

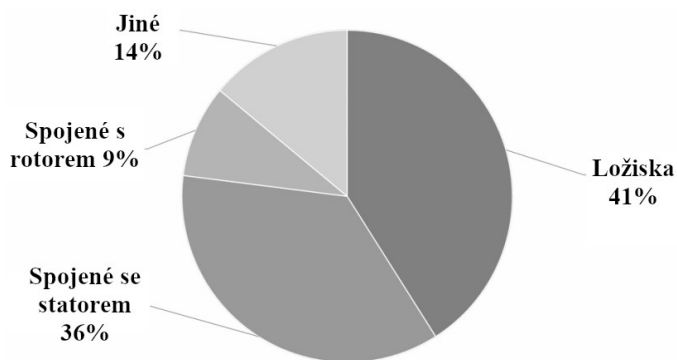
metoda je rovněž popsána v kapitole 1.5. Avšak navzdory těmto skutečnostem dle statistiky uvedené ve článku [13] hřídel stroje není zdaleka tak problematický, jak by se mohlo zdát.



Obr. 1 Statistika poruch subsystémů turbogenerátorů [13]



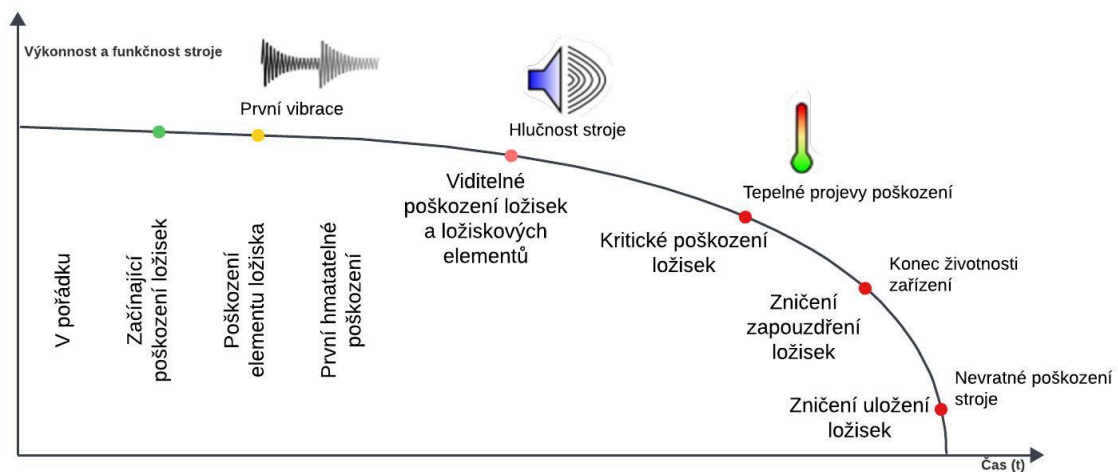
Obr. 2 Statistika poruch synchronních strojů obecně [13]



Obr. 3 Statistika poruch motorů obecně [13]

Z těchto grafů vyplývá, že zdaleka nejporuchovějším subsystémem točivých strojů je obecně jejich stator a co se statoru týká, pak nejporuchovější je evidentně statorové vinutí strojů. Statorové vinutí je namáháno jednak tepelně vlivem Jouleových ztrát a také mechanicky působením sil na vodiče vinutí protékaných proudem v magnetickém poli dle Ampérova zákona. Dále je samozřejmě vinutí namáháno i elektricky a má na něj vliv i jeho okolní prostředí.

Dalším velmi významně poruchovým subsystémem strojů bývají jejich ložiska, které jsou v případě menších strojů s valivými ložisky dokonce majoritní komponentou, která selhává, proto se na ně tato diplomová práce zaměřuje. Nesou totiž hřídel stroje a zajišťují maximální snížení třecí síly vznikající při otáčivém pohybu hřídele. Existuje mnoho druhů ložisek. U točivých strojů se zabýváme výhradně rotačními ložisky. Ty lze dále rozdělit na kluzná, valivá a magnetická. Měření vibrací proto provádíme zpravidla na ložiskových štítech nebo ložiskových stojanech motoru a často souběžně s tímto měřením měříme i teplotu ložisek. V případě, že je ložisko vadné, většinou začne ztrácet svoji schopnost účinně snižovat tření, čímž se začne ložisko nadměrně ohřívat. Tomu, jakým způsobem poté probíhá degradace stroje, popisuje následující obrázek. [12]



Obr. 4 Vývoj poškození ložisek v čase [12]

1.3 Signály pro vyhodnocování stavu, spolehlivosti a zbývajících životnosti točivých strojů - degradační faktory

Tato problematika je velmi dobře popsána v knize *Electrical Insulation for rotating Machines* [10].

Autoři zde hovoří o skupině namáhání označované jako “TEAM stresses“ Zkratka TEAM zde odkazuje na klíčová slova Thermal, Electrical, Ambient a Mechanical, tedy tepelné, elektrické, mechanické namáhání a namáhání okolními vlivy. Tyto vlivy mají za následek degradaci subsystémů točivých strojů. Na základě znalosti těchto dějů lze přesným navýšením jednoho nebo více těchto faktorů provádět tzv. zrychlené stárnutí vzorků za účelem provedení jejich diagnostiky v laboratoři. Tím lze získat lepší povědomí o tom, jak se bude skutečný stroj chovat v reálném provozu za delší dobu a lze predikovat jeho selhání. [10]

Jedna věc je tedy znalost chování materiálů vystavených různým druhům namáhání, která byla popsána výše a věc druhá je znalost jak se projevují závady stroje, pokud již nastávají vlivem degradace jeho kritických subsystémů, které byly popsány v kapitole 1.2.

Jak je již zmíněno v kapitole 1.2. Počínající poruchy točivých strojů mají různé projevy a tyto projevy jsou měřitelné. Výsledky takto naměřených vlastností stroje pak mohou vstupovat do výpočtu aktuální spolehlivosti stroje $Q(t)$: [15]

$$Q(t) = \text{geneze zařizení}^K \quad (1.1)$$

Kde součinitel K vyjadřuje superpozici všech vlivů působících na dané zařízení, lze ho tedy vyjádřit s respektováním skutečnosti, že se pohybujeme v oblasti „určitého“ spolehlivostního řetězce sledovaného zařízení, jako průnik výstupů diagnostických šetření v provozu a laboratorních zkoušek na vzorcích – např. formou životnostních křivek, což lze obecně vyjádřit rovnicí (1.2). [15]

$$K = K_{el\ model} \wedge K_{mech\ model} \wedge K_{čv} \wedge K_{vibrace} \wedge K_{otáčky} \wedge K_{rozběhy/odstávky} \quad (1.2)$$

Do tohoto výpočtu tedy vstupují tato data a informace které můžeme určitým způsobem získat nebo změřit pomocí různých diagnostických metod, které jsou dále uvedeny v knize [15].

Mimo jiné se jedná také o mechanické vibrace, které se rovněž ze své podstaty projevují akusticky. Často již pouhým poslechem jsme schopni odhadnout stav stroje, avšak tato metoda nemusí být přesná a postrádá jakoukoliv kvantizaci výsledků. A snadno tak může dojít k omylu. Z toho důvodu je vhodné na točivé stroje, jejichž porucha by měla významné ekonomické dopady nasadit různé měřicí diagnostické systémy. Je vhodné používat online neinvazivní diagnostické metody, k nimž typicky patří vibrodiagnostické metody a metody měření teploty, jak je popsáno dále:

K diagnostickým účelům se v oblasti mechanického chvění nabízí jednak frekvence (kmitočet) kmitání a dále některá ze tří dostupných charakteristických veličin – výchylka, rychlost nebo zrychlení kmitavého pohybu –, které lze navzájem přepočítávat s použitím operací derivace nebo integrace. [1]

$$a = \frac{dv}{dt} \quad (1.3)$$

$$v = \frac{ds}{dt} \quad (1.4)$$

Pokud již máme hodnoty získané ve frekvenčním spektru, lze použít vzorec A.2 uvedený v normě ČSN ISO 20816-1:

$$v_{rms} = 10^{-3} \pi \sqrt{\frac{1}{2} [(s_1 f_1)^2 + (s_2 f_2)^2 + \dots + (s_n f_n)^2]} = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2} = \frac{10^3}{2\pi} \sqrt{\left(\frac{a_1}{f_1}\right)^2 + \left(\frac{a_2}{f_2}\right)^2 + \dots + \left(\frac{a_n}{f_n}\right)^2} \quad (1.5)$$

Konkrétním způsobům, jakými jsou měřeny tyto veličiny, se věnuje kapitola 1.4. Dalším velmi významným signálem, který může poukazovat na nastávající poruchu je zvýšená teplota. To se týká především ložisek. Proč tomu tak je již popsáno v kapitole 1.2.

1.4 Snímače vibrací

V technické praxi se můžeme setkat s více druhy snímačů vibrací. Ty můžeme rozdělit na snímače polohy (výchyly), snímače rychlosti a snímače zrychlení.

Snímače polohy bývají zpravidla bezdotykové a opět existuje více různých typů. Dělíme je většinou podle jejich principu na indukčnostní, kapacitní, magnetické nebo optické.

Snímače rychlosti jsou „učebnicový příklad“ snímače vibrací, protože jeho elektrodynamický princip byl již v minulosti vcelku snadno realizovatelný a na jeho výstup – rychlost kmitavého pohybu (vibrací) – přímo odkazuje množství norem hodnotících vibrace v časové oblasti. [1] Jejich princip spočívá v pohybu cívky v magnetickém poli magnetu. Platí zde známý vzorec pro závislost indukovaného napětí na délce l a rychlosti v vodiče pohybujícího se v magnetickém poli charakterizovaném magnetickou indukcí B :

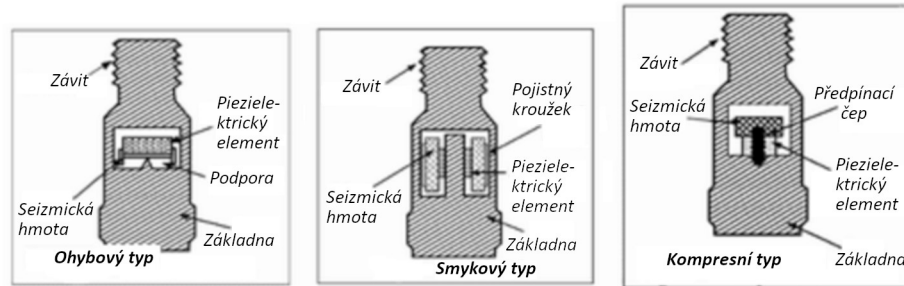
$$U_i = - \frac{d\phi}{dt} = B \cdot l \cdot v \quad (1.6)$$

Snímače zrychlení – často označované jako akcelerometry. Jejich princip se zakládá na silovém působení na seizmickou hmotu, které vzniká vlivem působení zrychlení. Tato síla je lineárně úměrná tomuto zrychlení dle vzorce:

$$F = m \cdot a \quad (1.7)$$

Tuto sílu měříme pomocí piezoelementu a to většinou pouze v jedné ose. Uspořádání však mohou být různá podle namáhání piezoelementu:

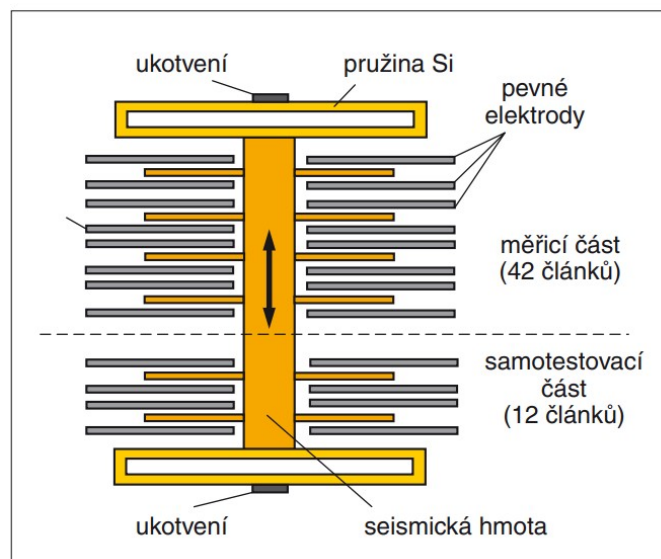
- Smykové (Shear)
- Ohybové (Flexural)
- Kompresní (Compression mode)



Obr. 5 Typy piezoelektrických elementů [2]

Namáháním piezoelementu vzniká náboj, který je nutné převést na napětí, které odpovídá zrychlení. Tento převod bývá buď realizován již přímo v akcelerometru pomocí integrovaného nábojového zesilovače, avšak tento zesilovač může být i externí. Je to proto, že v případě interního nábojového zesilovače může docházet k omezení pracovní teploty kvůli tomuto zesilovači. Těmito snímači však nelze měřit konstantní zrychlení. Minimální frekvence u těchto snímačů bývá okolo 0.1 Hz.

Vedle Piezoelektrických snímačů existují také takzvané MEMS snímače. Ty bývají velmi často používané především ve spotřební elektronice, mobilních zařízeních, herních ovladačích, dronech apod. Zkratka MEMS vychází z anglického označení Micro electro-mechanical system a zahrnuje celou řadu elektromechanických zařízení, vyráběných křemíkovými technologiemi, která lze integrovat do jednoho pouzdra. Typicky se pak jedná o různé snímače či dokonce mikročerpadla a mikroventily.



Obr. 6 Principiální schéma hřebenového jednoosého MEMS akcelerometru [4]

Struktura a funkce MEMS akcelerometru je založena na proměnné kapacitě třielektrodového vzduchového kondenzátoru. [3]

$$C = \varepsilon_0 \varepsilon_r \frac{S}{d} \quad (1.8)$$

Kde ε_0 je permitivita vakua, která má hodnotu přibližně $8,85 \cdot 10^{-12}$ F/m, ε_r je relativní permitivita materiálu. S je plocha elektrod a stejně jako dvě předchozí uvedené veličiny je v MEMS akcelerometru konstantní. Proměnná veličina je zde d vyjadřující vzdálenost elektrod.

I v případě MEMS akcelerometrů platí rovnice (1.7). Seizmická hmota je zavěšena na pružinách o tuhosti k . Pro výchylku s , která odpovídá změně vzdálenosti elektrod Δd lze psát:

$$\Delta d = s = \frac{F}{k} = \frac{m \cdot a}{k} \quad (1.9)$$

Při uvážení rovnice (1.8) vyplývá, že kapacita je lineárně a nepřímo úměrná zrychlení.

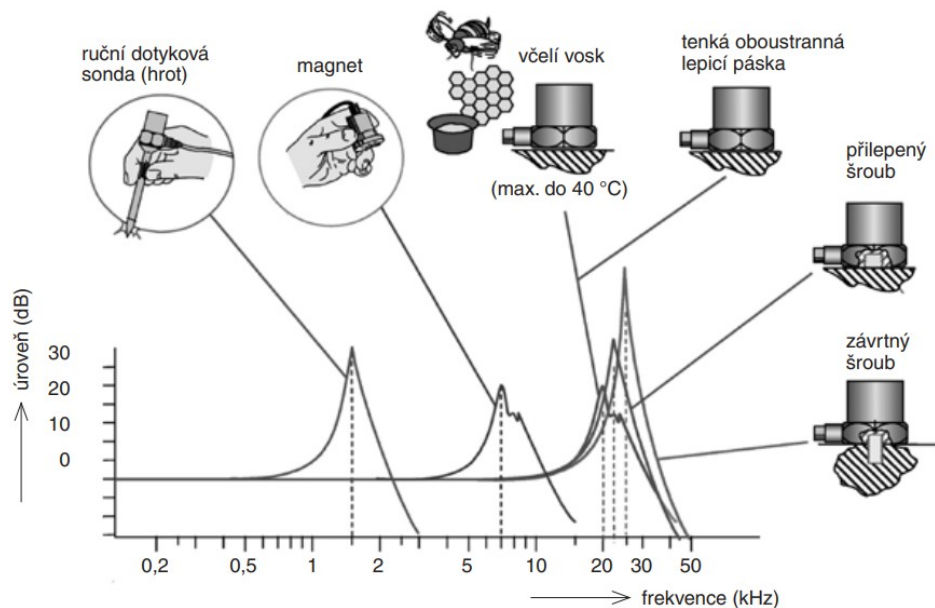
K měření kapacity se využívá metody spínaných kondenzátorů řízené číslicovou logikou a generátorem spínacího hodinového signálu. Dochází tak k převodu změny kapacity na změnu napětí. To je následně linearizováno a filtrováno opět obvody se spínanými kondenzátory a nakonec se provádí kompenzace vlivu teploty. Výsledkem je lineární, zesílený a kompenzovaný napěťový signál s definovanou převodní konstantou - citlivostí podávající informaci o kolik se musí změnit hodnota měřeného zrychlení, aby došlo ke změně výstupního napětí o 1 V (hodnota g/V). Vše se provádí zvlášť pro každou osu snímání (kanál). [3]

Mnohé MEMS akcelerometry mají přímo napěťový výstup. Jiné obsahují i A/D převodníky a mají digitální výstup. Jejich citlivost je pak udávána v LSB/G. To znamená, že máme-li akcelerometr s citlivostí například 8 LSB/G, pak změna zrychlení o 1 G, tedy přibližně 9,81 m/s, způsobí změnu bitu s nejnižší vahou 8x. Z toho lze určit, že akcelerometr je schopen rozlišit jednu každou osminu G. V praxi se však lze setkat s výrazně vyššími citlivostmi. Řádově tisíce LSB/G, běžně tedy mají akcelerometry rozlišení na tisícinu G a vyšší.

1.4.1 Možnosti montáže akcelerometrů na měřený objekt

O montáži akcelerometrů na měřený objekt by bylo možné napsat opravdu velmi mnoho. Metody montáže v zásadě dělíme na dvě základní skupiny: invazivní a neinvazivní. Ovšem pozor, není to totéž co destruktivní a nedestruktivní měřicí metody o kterých se zmiňuje kapitola 1.1. Zde je řeč pouze o montáži a ani invazivní metoda montáže nezasahuje do funkčnosti testovaných zařízení. Každá dále uvedená metoda montáže má své výhody a nevýhody. Obecně jde zpravidla o kompromis mezi

proveditelností či jednoduchostí montáže a výsledným použitelným frekvenčním rozsahem. Tuto problematiku nejlépe přímo vystihuje následující obrázek:



Obr. 7 Metody montáže akcelerometrů [11]

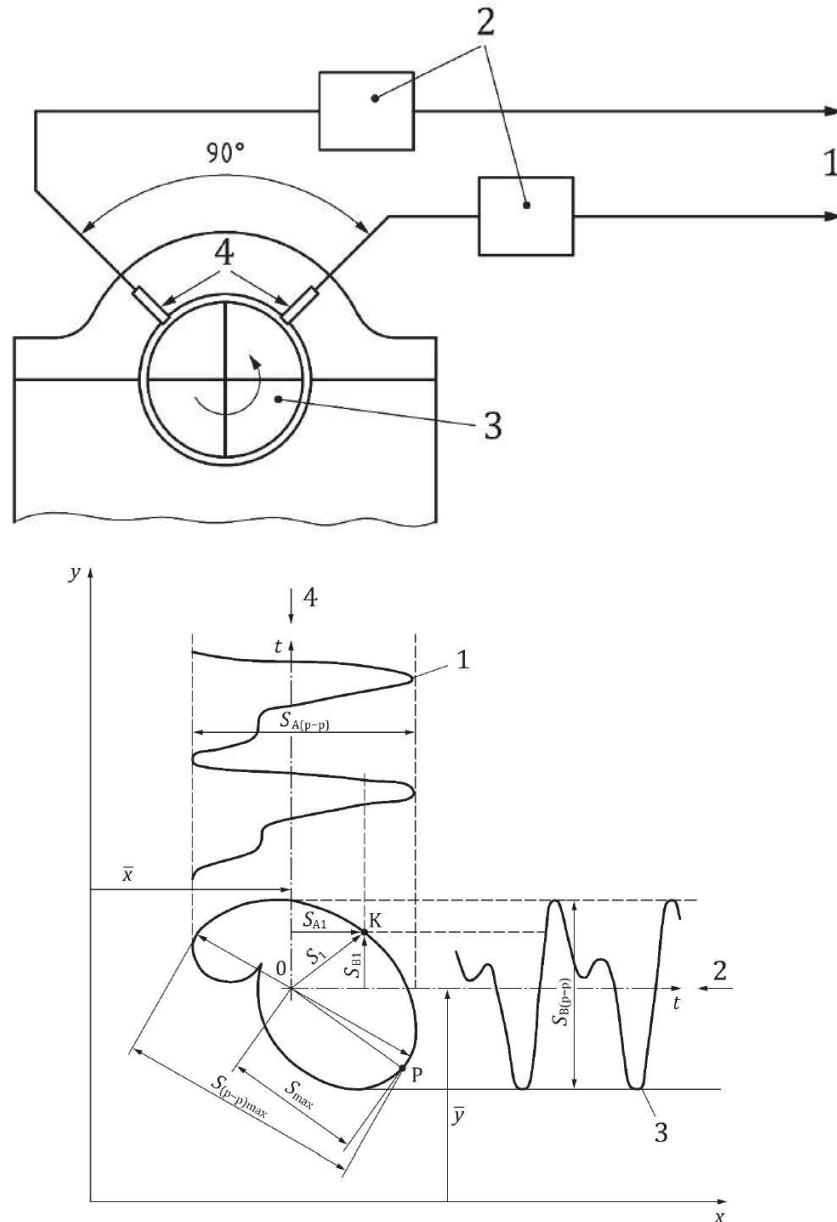
Z obrázku jednoznačně plyne, že zatímco neinvazivní metody vynikají svou jednoduchostí montáže a samozřejmě svou podstatou, že jsou neinvazivní. Naproti tomu invazivní metody poskytují významně širší použitelný frekvenční rozsah.

1.5 Metody analýzy dat a stanovení diagnózy a prognózy ve vibrodiagnostice

Tyto metody do určité míry specifikuje norma ČSN ISO 20816-1 a věnuje se jim rovněž článek [1]. Z těchto zdrojů plyne, že je směřodlatné vyhodnocování vibrací jak v časové, tak ve frekvenční oblasti. Na základě takto vyhodnocených dat lze provést příslušnou diagnózu a prognózu.

1.5.1 Vyhodnocení v časové oblasti

Tuto metodu vyhodnocení využíváme, chceme-li měřit pohyb hřídele stroje. Měříme výchylku a to ve dvou kolmých směrech, jak je ukázáno na následujících obrázcích:



Obr. 8 Měření výchylky a vyhodnocení v časové oblasti dle ČSN ISO 20816-1

Výsledkem této metody je tzv. kinematická dráha hřídele. Analýza takto získaných dat pak vychází z předpisů a postupů uvedených v normě ISO 20816.

1.5.2 Vyhodnocení ve frekvenční oblasti

K provedení analýzy dat ve frekvenční oblasti je nutné nad naměřenými daty provést matematickou operaci známou jako Fourierova transformace. Protože ale máme data z akcelerometru převedená do digitální podoby pomocí A/D převodníku a to s určitou vzorkovací frekvencí, máme tedy nikoliv spojitý, ale diskrétní signál. Vstupní signál je vzorkován v pravidelných intervalech. Pokud je vzorkovací frekvence příliš nízká, vstupní signál bude vzorkován nesprávně, čímž vznikají chybné výsledky vzorkování. Tento jev se

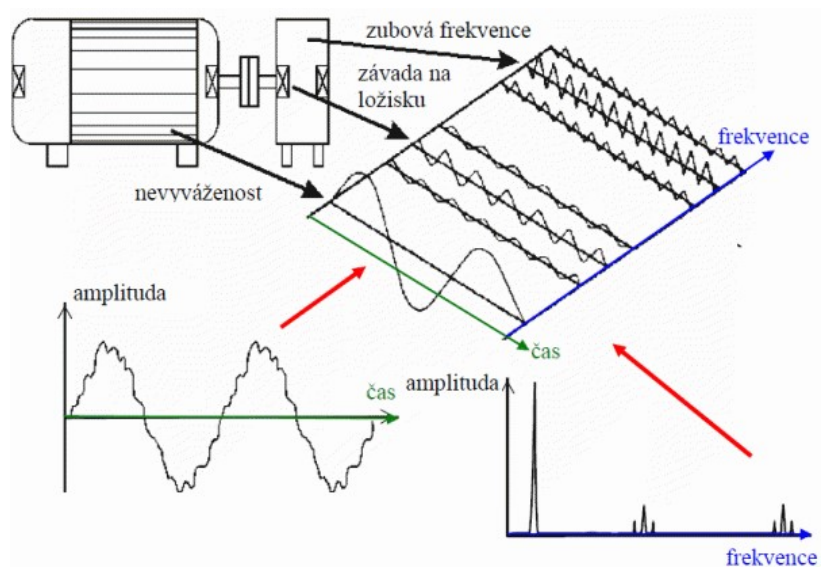
nazývá aliasing efekt. [12] Ten se projeví i pokud s daty provedeme diskrétní Fourierovu transformaci. Na jejím vstupu je N vzorků měřeného signálu a na jejím výstupu je N harmonických funkcí, jejichž součet by přibližně měl odpovídat měřenému signálu. Amplitudy těchto harmonických funkcí se následně zobrazí jako frekvenční amplitudové spektrum s maximální frekvencí odpovídající vzorkovací frekvenci f_s . Platí ale Shannon-Kotelnikovův teorém, který je popsán v citaci [12]. Podle něho je toto spektrum zrcadleno okolo $f_s/2$. Frekvenční spektrum, které nás zajímá je tedy jen od nuly do $f_s/2$, z čehož plyne, že vzorkovací frekvence musí být alespoň dvojnásobkem požadovaného frekvenčního rozsahu.

Vyhodnocení ve frekvenční oblasti má tedy za výsledek především amplitudové spektrum. Možné poruchy lze odhalit analýzou tohoto spektra. Především si všímáme násobků mechanické frekvence stroje.

Rozbor je zpravidla možné rozdělit do tří hlavních oblastí, kterými jsou:

- *pásmo nízkých frekvencí, sahající od frekvencí pod nejpomalejší rychlostí otáčení hřídelí až po frekvence nižších násobků (čtvrtou až šestou harmonickou) hřídelí s největšími otáčkami: získají se informace o závadách způsobených nevyvážeností, ohybem hřídelí, přesazením hřídelí, nestabilitou radiálních ložisek a uvolněním mechanických vazeb,*
- *pásmo středních frekvencí, nacházející se nad oblastí nízkých frekvencí až po frekvence od asi 1 po 5 kHz (podle otáček a typu převodů): obsahuje informace o závadách ozubených převodů a převodovek,*
- *pásmo vysokých frekvencí od horní hranice pásma středních frekvencí nahoru: obsahují informace o počínajících závadách valivých ložisek (např. poškození kroužků, trhlinky apod. se projevují úzkými a ostrými impulzy v oblasti vysokých kmitočtů a odpovídající složky signálu zde nejsou překryty intenzivnějšími vlivy převládajícími na nízkých a středních frekvencích). [1]*

Informace uvedené v této citaci velmi dobře shrnuje následující obrázek:



Obr. 9 Vyhodnocení poruch ve frekvenční oblasti. [14]

2 Návrh

2.1 Koncepce systému

Základní myšlenkou pro tuto diplomovou práci bylo vytvoření vibrodiagnostického systému, který by sice pravděpodobně nebyl tolik přesný a spolehlivý jako jiná komerčně dostupná řešení pro průmyslové aplikace, ale na druhou stranu by byl levný a dobře dostupný a bylo jej možné snadno instalovat na rozsáhlých objektech určených ke sledování. Přitom by si měl stále zachovat svoji schopnost odhalit blížící se poruchy testovaných zařízení a v případě potřeby by mělo být poměrně snadné jej doplnit o další případně jiné senzory.

Samotná myšlenka vytvořit takovýto systém vznikla na základě zkušeností autora se systémy pro ukládání a zpracování dat akcelerometrů, gyroskopů a jiných senzorů používaných v letových řídicích jednotkách jednoduchých fixed-wing dronů a multikoptér, kde tyto systémy slouží jako jednoduchá letová černá skříňka. Nejznámějším z těchto systémů je Betaflight Blackbox a k němu přidružená aplikace Betaflight Blackbox explorer, která slouží k prohlížení a vyhodnocování dat. Další velmi hojně využívaná aplikace pro tyto účely je PIDtoolbox. Jedná se o aplikaci v prostředí MATLAB.

Tyto systémy pracují na jednoduchém principu, kdy data z MEMS IMU jednotky jsou přímo ukládána v reálném čase na paměťovou SD kartu nebo na integrovaný paměťový čip a následně jsou čtena a zpracována výše uvedeným softwarem v PC.

Tento přístup ukazuje, že běžně dostupné MEMS akcelerometry poskytují vysokou přesnost a kvalitu zpracovaného signálu. Z toho důvodu je možné, aby byly nasazeny i pro účely monitoringu a vyhodnocení vibrací.

Nejen díky těmto akcelerometrům, ale i zvolenému paměťovému médiu tyto systémy poskytují vysokou úroveň integrace, nízkou hmotnost a cenu. Toto vše je předurčuje právě jako velmi výhodné pro využití právě v bezpilotních systémech. Problém zde ale nastává kvůli relativně malé paměti a značně omezeným možnostem data zpracovávat průběžně v reálném čase. Právě tyto vlastnosti naopak požadujeme u vibrodiagnostiky stacionárně umístěných točivých strojů jako jsou turboalternátory, hydrogenerátory, motory pohánějící výrobní technologii apod.

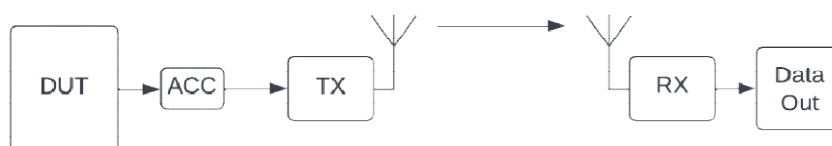
Z těchto důvodů musí navržený systém fungovat mírně odlišným způsobem. Jednak na točivých strojích je více významných míst, kde je nutné vibrace měřit a proto bude nutné mít čidel více. Dále kvůli nutnosti provádění měření za chodu zařízení je nutné zajistit

bezpečnost obsluhy. To lze zařídit například vhodně provedeným oddělením snímací části systému od výstupního a zpracovacího zařízení, se kterým uživatel během měření pracuje. To lze realizovat typicky tak, že tyto dvě části mezi sebou budou komunikovat bezdrátově. Což nejen výrazně zvyšuje bezpečnost obsluhy ale také pohodlnost práce s tímto zařízením a značně přispívá k variabilitě nasazení systému v různých podmínkách. U bezdrátového přenosu je však nutné mít neustále na paměti problematiku jejich zabezpečení aby nedocházelo například k jejich odposlechnutí a zneužití.

Co se zpracování dat týká, je zde potřeba dostatečný výpočetní výkon, neboť data budou přicházet v reálném čase a bude nutné je převádět z časové oblasti do frekvenční. K tomu nám slouží typicky Fourierova transformace. Ta, jak vychází ze své matematické definice je vysoce výpočetně náročná, avšak existují i způsoby značného zjednodušení těchto výpočtů tak, aby bylo lépe možné je realizovat v procesorech. Takto zpracovaná data je dále nutné porovnávat s přípustnými limity, které stanovují normy. Tento proces je vhodné v maximální míře automatizovat a získané výsledky předat jednoduchou a srozumitelnou formou obsluze tak, aby na základě naměřených hodnot obsluha ihned věděla, zda je zařízení v pořádku, nebo zda je třeba naplánovat odstávku a údržbu, případně zda okamžitě ukončit provoz.

2.2 Návrh distribuovaného CMD systému

2.2.1 Obecné schéma systému



Obr. 10 Obecné blokové schéma systému

Toto obecné schéma vychází z předpokladů uvedených v kapitole koncepcí systému v teoretickém rozboru. DUT (Z anglického Device Under Test) je testované zařízení, ACC (Z anglického Accelerometer) je MEMS akcelerometr, TX je obecně vysílač dat, RX je přijímač a blok označený jako Data Out představuje výstupní zařízení. Ze schématu již částečně vyplývá, jaký hardware budeme k realizaci systému potřebovat.

2.2.2 Volba komponent

Akcelerometry

Volbu komponent začneme od snímačů. Problematice měření vibrací se věnuje kapitola 1.3. Ze senzorů, které popisuje tato kapitola, byly zvoleny MEMS senzory kvůli jejich velmi dobré dostupnosti a ceně. MEMS senzorů je na trhu k dispozici celá řada.

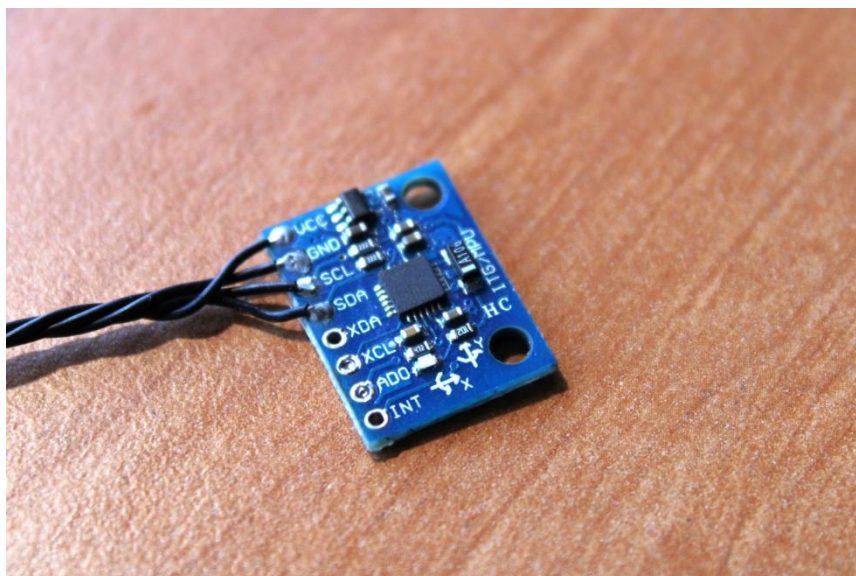
Výběr konkrétního typu záleží na aplikaci senzoru. Při výběru nás pak zajímají především tyto faktory:

- Cena
- Dostupnost
- Měřicí rozsah
- Citlivost
- Komunikační rozhraní

Na základě předpokladu, že potřebujeme navrhnout velmi levné a univerzální zařízení volíme akcelerometry typu MPU-6050.

MPU-6050 mají tu výhodu, že používají komunikační protokol I²C, který umožňuje připojení více zařízení na jednu sběrnici. Navíc v blízkosti točivých strojů velkých výkonů bude nutné eliminovat vlivy točivého magnetického pole. Proto je velmi žádoucí komunikaci řešit digitálně. Dále jsou tyto senzory velmi levné a hojně vyžívané a právě díky tomu je možné o nich zjistit mnoho informací i z jiných zdrojů než jen od výrobce a dodavatele, což je pro vývoj podobných zařízení poměrně značná výhoda. Kromě toho, jakmile je již komunikace po této sběrnici implementována, není problém později přidat další, klidně jakékoliv jiné senzory, pokud budou také umožňovat komunikaci po této sběrnici. To samozřejmě za předpokladu doplnění příslušného kódu do softwaru všech částí.

U těchto senzorů však existuje i jistá nevýhoda a to konkrétně nastavování adres, které nelze realizovat softwarově na libovolnou adresu. Výběr adresy je zde možný hardwarově pomocí padu označeného jako AD0. Podle toho, zda je tento pin ponechán volně, nebo je na něj přivedeno napájení, mění se adresa mezi 0x68 0x69. Jiné adresy bohužel nejsou k dispozici.



Obr. 11 MPU-6050

Na obrázku výše můžeme vidět zvolený senzor připojený na DPS, která obsahuje pár dodatečných součástek jako stabilizátor napětí, pull-up rezistory pro I²C sběrnici, blokovací a filtrační kondenzátory a LED, která indikující přítomnost napájení.

Vysílač a přijímač

Další nezbytnou součástí systému je vysílač a přijímač. Zde je potřeba říci, že sice existují zařízení schopná tvořit jakýsi bezdrátový most na I²C sběrnici, z velké většiny se ale jedná o celkově dražší a hůře dostupná speciální zařízení, která ale i tak bývají založena dvou mikrokontrolérech ESP8xxx nebo ESP32 od výrobce Espressif.



Obr. 12 Vývojový kit ESP32

Tyto čipy mají integrovaný WiFi / Bluetooth transceiver, jehož funkce je uživatelsky libovolně konfigurovatelná. Kromě toho je framework pro programování těchto čipů podporován mnoha vývojovými prostředími. Díky tomu lze tyto mikrokontroléry programovat například i ve velmi rozšířeném vývojovém prostředí Arduino IDE a Arduino IDE 2.0, přičemž není problém využívat příkazy určené původně pro AVR mikrokontroléry. Navíc výrobce Espressif zde má již implementovaný vlastní protokol ESP-NOW sloužící k jednoduchému zabezpečenému přenosu paketů mezi více stanicemi. Není tak nutné mít na každé straně samostatný mikrokontrolér s externím modulem pro bezdrátovou komunikaci, což přispívá k jednoduchosti a robustnosti zařízení.

Tato kombinace vlastností dělá z čipů výrobce Espressif ideální kandidáty. Samotná volba mezi ESP8xxx a ESP32 je poměrně jednoduchá. Navzdory tomu, že oba procesory (ačkoliv jejich název může vyvolávat dojem, že to tak není) jsou 32-bitové, ESP32 je dvoujádrové a cena je takřka shodná, vedle toho s ESP32 mám o něco lepší zkušenosti, proto volím ESP32. Opět ESP32 samotné je jen mikrokontrolér. Pro vývoj zařízení volím raději vývojový kit, tedy čip osazený na DPS obsahující také USB/UART převodník, stabilizátor napětí a dvě indikační LED plus několik součástek náležejícím k těmto obvodům. To značně usnadňuje vývojovou práci. Například k programování mikrokontroléru není potřeba speciální programátor. K naprogramování stačí vývojový kit propojit s PC pomocí USB kabelu a pro ověření funkčnosti není nutné ani připojovat externí LED, jedna uživatelsky ovladatelná LED je již součástí vývojového kitu. Kromě toho je kit má k propojení připravené kolíky s roztečí 2,54 mm. K nim je možno zakoupit dutinky, které lze přímo zapájet do DPS, kterou procesor obsluhuje. Díky tomu je možné v případě potřeby vývojový kit z DPS vyjmout, což může být velmi užitečné, potřebujeme-li

například ověřit jeho správnou funkci a vyloučit jiné chyby v průběhu vývoje a testování nebo i během nasazení provozu.

Zařízení pro zpracování a zobrazení výsledků měření

Jako zobrazovací zařízení by zde mohl sloužit například jednoduchý LCD displej, který lze pomocí ESP32 ovládat s využitím rozhraní SPI. Tyto displeje se vyrábí i v dotykovém provedení, což z nich činí zajímavou volbu, zvláště jde-li o to, aby byl systém kompaktní. Problém však nastává ve chvíli, kdy na tomto displeji potřebujeme zobrazit velké množství dat. A protože výstupem bude většinou graf frekvenčního spektra, mohli bychom zde snadno narazit na problémy související s relativně malým rozlišením. Kromě toho ESP32 není zrovna nejvhodnější procesor k zobrazování složitějších grafických výstupů. Přesto by tato alternativa jistě stála za vyzkoušení. Je možné, že tento směr by mohl přinést cenné poznatky.

Jelikož vím, že budu potřebovat zpracovávat větší množství dat a bude potřeba u těchto dat provádět Fourierovu transformaci a další zpracování, zvolil jsem jiný přístup. Je totiž potřeba vytvořit uživatelsky přívětivou aplikaci, která by spolehlivě dokázala data načítat ze sériové linky, provádět nad nimi složitější matematické operace a zároveň předávala vhodně graficky zpracované výsledky uživateli. K tomuto účelu je vhodné použít běžný počítač, v tomto případě laptop. Avšak bude k tomu potřeba vhodný software. Proto musím zvolit vhodné prostředí pro tvorbu takového softwaru.

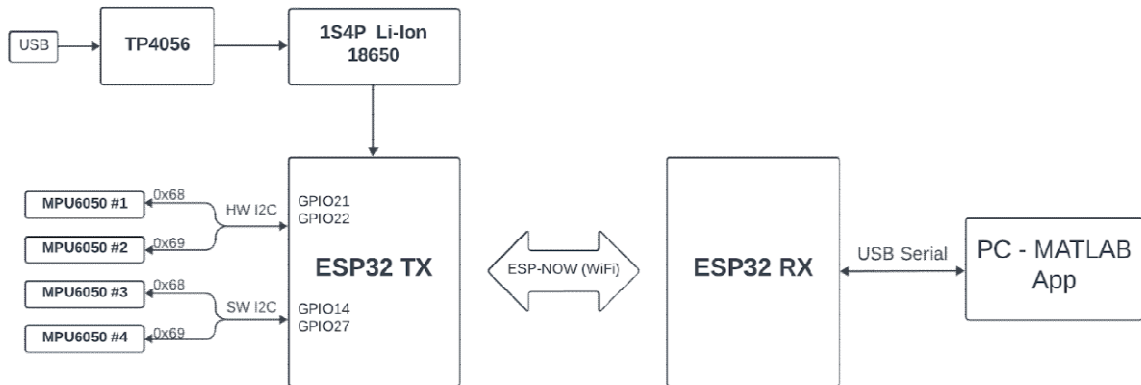
Aplikace pro Windows je možno tvořit v mnoha různých uživatelských rozhraních. Spousta z nich také již má předpřipravenou implementaci různých grafických prvků a matematických operací. Díky tomu je pak tvorba samotné aplikace značně zjednodušena. K vytvoření aplikace pro analýzu, zpracování a zobrazení dat v reálném čase existují různé alternativy. První z nich je LabVIEW vyvíjený firmou National Instruments. Další, již ne tolik známou alternativou je MATLAB App Designer.

Vyzkoušel jsem obě tyto alternativy a nakonec jsem zvolil MATLAB vzhledem k jeho možnosti psát kód přehledněji v textovém editoru a s tím souvisejícím lepším vlastnostem při řešení detailů, které budou popsány dále v této práci.

3 Realizace a ladění

3.1 Hardware

Na základě zvoleného hardwaru, jehož volbou se zabývám předchozí kapitole, lze sestavit blokové schéma systému.



Obr. 13 Konkrétní blokové schéma

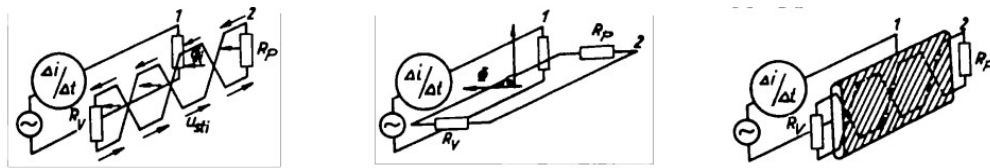
Porovnáme-li toto schéma se schématem uvedeným v kapitole 2.2.1, zjistíme, že toto schéma z něho sice vychází, avšak do značné míry se od něho odlišuje. Největší patrná změna je v tom, že datové toky zde nejsou jen jedním směrem. Je to z toho důvodu, že během vývoje jsem dospěl ke zjištění, které vedly k potřebám zde tyto datové cesty vytvořit. Těmto změnám se budu věnovat dále v této práci především pak v kapitole 3.1.3. Software. Další, co je zde navíc je nabíječka lithiových článků TP4056 a akumulátorová baterie. Tato část zde vznikla z důvodu, že vysílací část potřebuje vlastní napájení a právě takto lze zajistit, že vibrodiagnostika bude schopna pracovat určitou dobu i bez napájení. Díky tomu je zde zaručena určitá odolnost proti výpadkům napájení. Na straně přijímače je tato odolnost zajištěna buď akumulátorem laptopu, nebo v případě připojení ke stolnímu PC by bylo nutné zajistit jeho napájení z UPS zdroje.

3.1.1 Vysílač

Z hlediska hardwaru je část vysílače nejkomplexnější. Tato část totiž zahrnuje DPS, která musí splnit následující požadavky:

- Připojení všech snímačů pomocí vzájemně nezávislých konektorů splňujících princip poka-yoke. Musí tedy být dostatečně robustní a zároveň nesmí být možné do nich připojit snímače obráceně, což by jinak vedlo k nežádoucímu otočení polarity napájecího napětí snímačů.

- Jednotlivé konektory musejí být na dvou samostatných sběrnicích I²C z toho důvodu, že počet adres v této sběrnici je u akcelerometrů stejného typu omezen, jak popisuje kapitola 2.2.2. Jedna z těchto sběrnic musí být připojena k pinům 21 a 22 na ESP32. Zde se jedná o hardwarově realizovaný I²C port. Druhá sběrnice musí být připojena k pinům 26 a 27, které jsem zvolil pro softwarově emulovaný port.
- DPS musí zahrnovat nezávislý napájecí zdroj pro ESP32. Tento zdroj musí být možné libovolně vypnout.
- Samotné akcelerometry musí být možné umístit na příslušná místa stroje, ve kterých provádíme měření. Proto je vhodné je k vysílací DPS připojit pomocí dostatečně dlouhých kabelů. Ty musejí mít měkkou izolaci schopnou odolat jednak vibracím tam kde dochází k jejich dotyku s měřeným strojem a jednak musí vydržet tepelné namáhání. Je proto vhodnější zde použít kabely se silikonovou izolací. Ke každému akcelerometru budeme potřebovat čtveřici vodičů. Napájení, zem, datový signál a hodinový signál. Tyto vodiče musí být kroucené za účelem omezení nežádoucích vlivů magnetického pole.



Obr. 14 Omezení nežádoucích vlivů elektromagnetického pole [5]

V případě, že se kroucení vodičů ukáže jako nedostatečné, je možné dosáhnout dalšího zlepšení přidáním stínění například v podobě měděného knotu nebo použitím průmyslového kabelu se stíněním.

Pro připojení akcelerometrů volíme konektory typu JST-XH. Jsou velmi robustní a mají výše zmíněnou poka-yoke konstrukci zamesující nesprávnému zapojení. Tyto konektory jsou díky těmto vlastnostem často používány pro připojení baterií k systémům BMS, kde jsou velmi vysoké požadavky na bezpečnost. Existují i jiné alternativy, často se pak ale jedná o konektory s SMD montáží, která je oproti THT značně náchylnější na mechanické poškození.

Aby byla zajištěna vyšší variabilita a robustnost systému, vznikla myšlenka napájet vysílač z baterií, které by během normálního provozu byly udržovány v nabitěm stavu a jakmile by došlo ke ztrátě vnějšího napájení, zařízení by zůstalo napájené z baterií.

Návrh akumulátoru začneme od jmenovitého napětí. Zde rozhoduje, jaké zařízení bude z akumulátoru napájeno. V našem případě je to vývojová deska ESP32, která má na vstupu lineární stabilizátor (LDO) AMS1117 s pevně nastaveným výstupním napětím 3,3 V. Tento stabilizátor má výrobcem udávané maximální napětí 15 V. Díky tomu by při použití lithiových článků bylo možné využít konfiguraci až tří článků zapojených do série, které by při plném nabití vykazovalo napětí 12,6 V. Zde je nutné zvážit, že použití takového akumulátoru by ale vedlo ke značnému úbytku napětí na LDO a tudíž k velkým tepelným ztrátám. Další značný problém, který se k tomu váže, je potřeba obvodu, který by zajistil balancování článků. Vzhledem k těmto předpokladům bude snaha pracovní napětí dostat co nejnižší a to ideálně tak, aby stačilo uspořádání jednoho článku v „sérii“ (Prakticky označeno jako 1S), což by vedlo k tomu, že nebude potřeba žádného balančního obvodu.

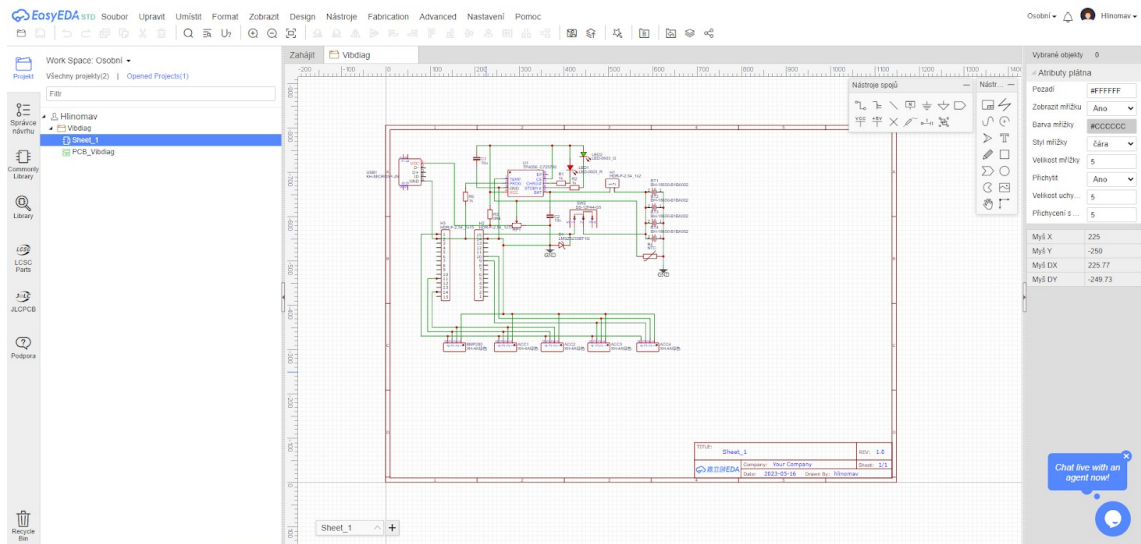
Za účelem zjištění možností napájení byl proveden pokus napájení ESP32 právě pouze jedním takovým článkem, přičemž bylo měřeno napětí a současně bylo kontrolováno, při jakém napětí spolehlivé vysílání přestane fungovat.

Pokus prokázal, že krátkodobý provoz na baterie v uspořádání 1S je možný. Díky tomu skutečně nebude potřeba balančního obvodu, čímž dosáhneme značného zjednodušení, snížení ceny a zvýšení robustnosti zařízení.

Kapacitu akumulátoru a tedy i jeho výdrž lze dále zlepšit propojením více článků paralelně. DPS jsem však navrhnul až na kombinaci čtyř článků paralelně, což na jednu stranu není nijak omezující, pokud chceme například kvůli ceně systém provozovat na méně článků, avšak stále je tu možnost zcela libovolného navýšení kapacity.

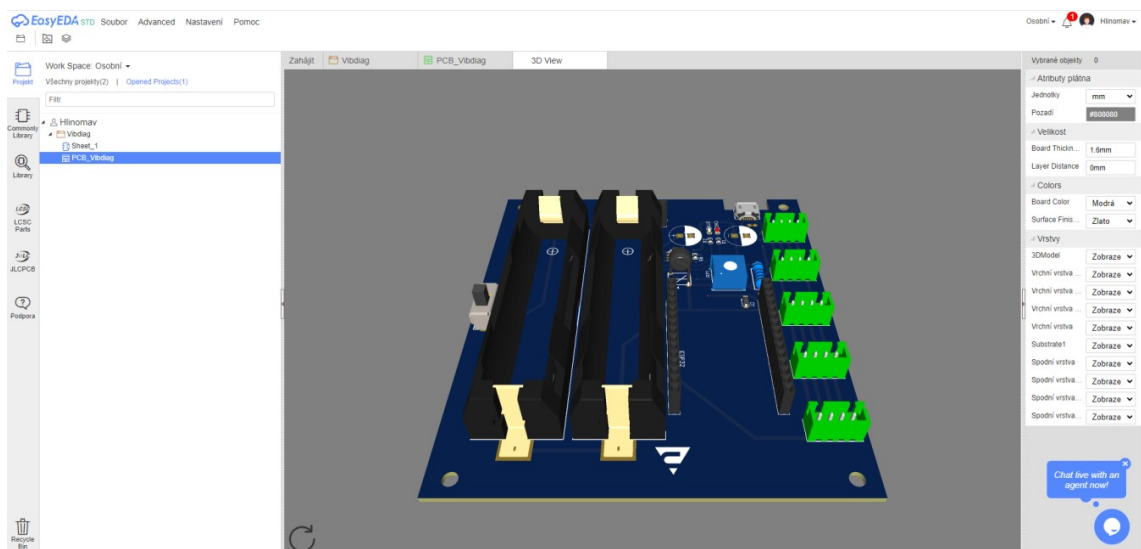
Články je však potřeba také nabíjet. Díky zvolené konfiguraci k tomu postačí jednoduchý integrovaný obvod TP4056, který je značně rozšířen v podobných zařízeních napájených z 1S lithiových článků jako jsou powerbanky, přenosné bezdrátové reproduktory apod. Zapojení nabíjecího obvodu odpovídá doporučenému schématu „Typical applications“ v datasheetu.

Návrh samotné DPS byl proveden v online prostředí EasyEDA Designer tak, aby splnila výše uvedené požadavky. Technická dokumentace jako přesné schéma a výkresy DPS jsou uvedeny v přílohách 1 až 3. Prostředí EasyEDA Designer jsem zvolil především proto, že je pro nekomerční účely zdarma, má velmi rozsáhlou knihovnu a vedle toho je přímo napojené na výrobce DPS JLCPCB, což značně usnadňuje celý proces objednávky navržených DPS.



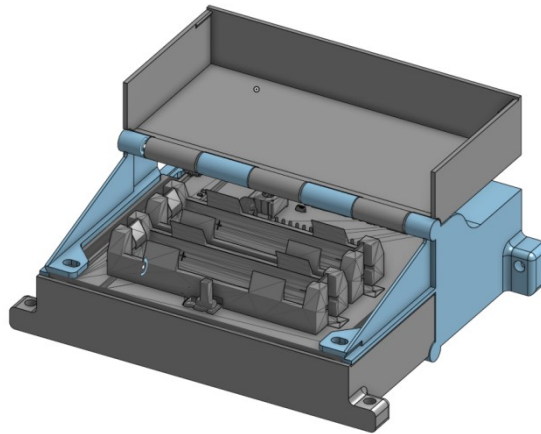
Obr. 15 Prostředí CAD EasyEDA Designer

Dále toto CAD prostředí umožňuje rendering navržené DPS jako 3D modelu a jeho export.



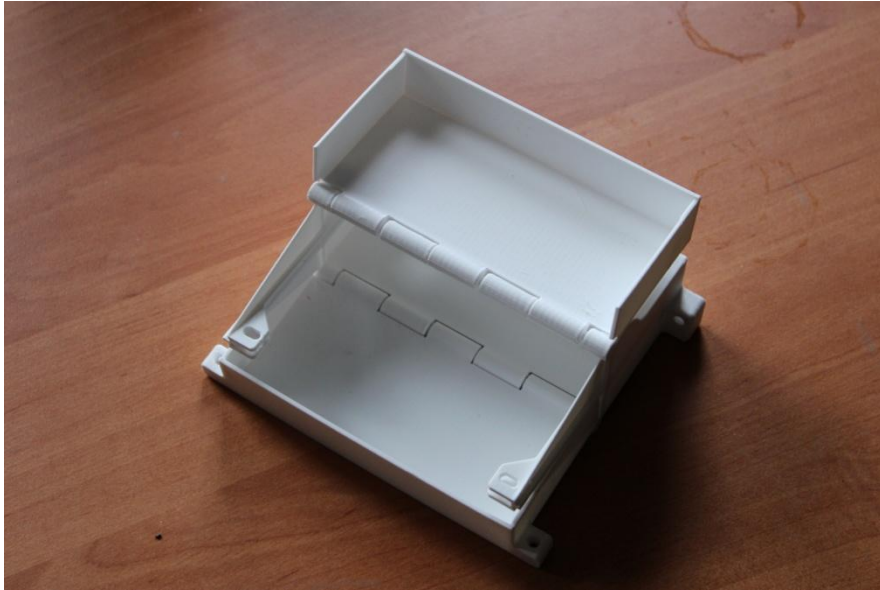
Obr. 16 3D rendering v prostředí EasyEDA

Tato funkce programu značně usnadňuje možnosti návrhu krabičky pro toto zařízení. Tato krabička byla navržena a metodou 3D tisku vytvořena v rámci předmětu KEP/PED v online CAD prostředí OnShape. Návrh krabičky byl proveden tak, aby bylo možné ji snadno tisknout v jednom kuse a bez podpor a aby byla bezmontážní, což znamená, že po vytištění má být možné ji okamžitě použít bez nutnosti dodávat další mechanické součásti. Zároveň zde byl požadavek, aby krabička měla jakási dvířka, kterými bude DPS dostupná, aby bylo možné vysílač vypnout nebo vyměnit baterie.



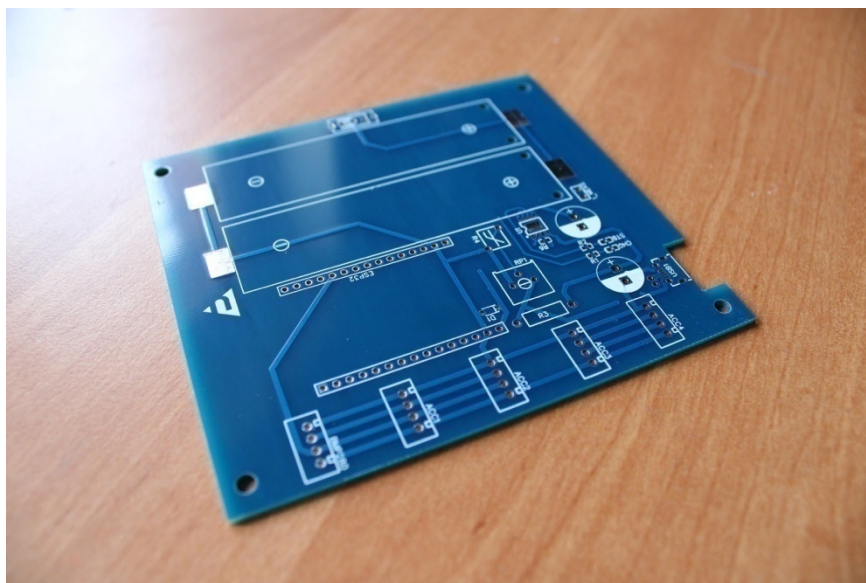
Obr. 17 Sestava krabičky s otevřenými horními dvířky a osazené DPS

Tisk krabičky probíhal s otevřenými dvířky, k tomu je uzpůsobený pant jejich dvířek, který umožňuje jejich otevření až do úhlu 180°. Tím odpadá problematika tisku vodorovné horní stěny krabičky. Zároveň ale musí být pant tištěný, čímž u FDM/FFF metody tisku vzniká problematika tisku na více různých tiskárnách, kdy každá poskytuje jinou přesnost tisku, což může vést k tomu, že pant bude buď příliš volný, nebo naopak půjde příliš ztuhá nebo dokonce nebude možné s ním pohnout. Z toho důvodu byla krabička modelována parametricky a jedním z parametrů je přímo volitelná tolerance pantu. Tím lze dosáhnout požadované tolerance přesně pro libovolnou 3D tiskárnu.

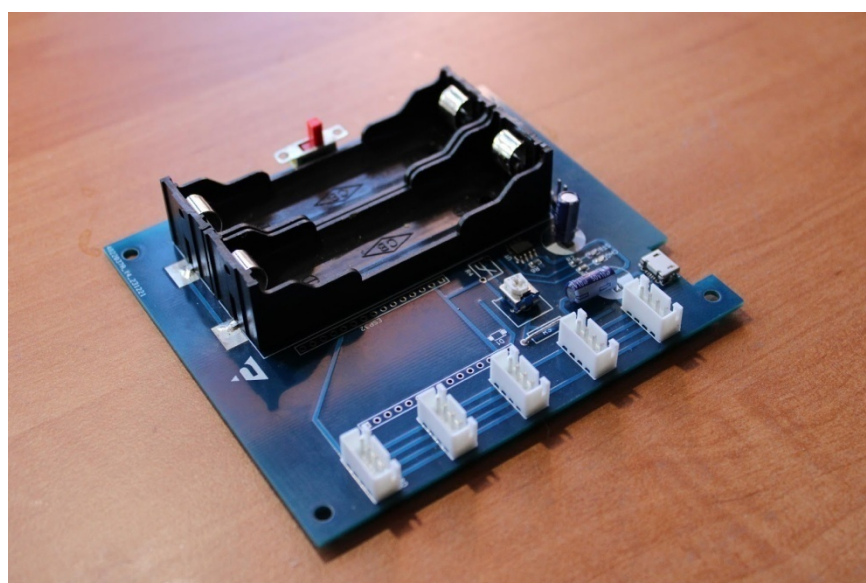


Obr. 18 Vytiskovaná krabička

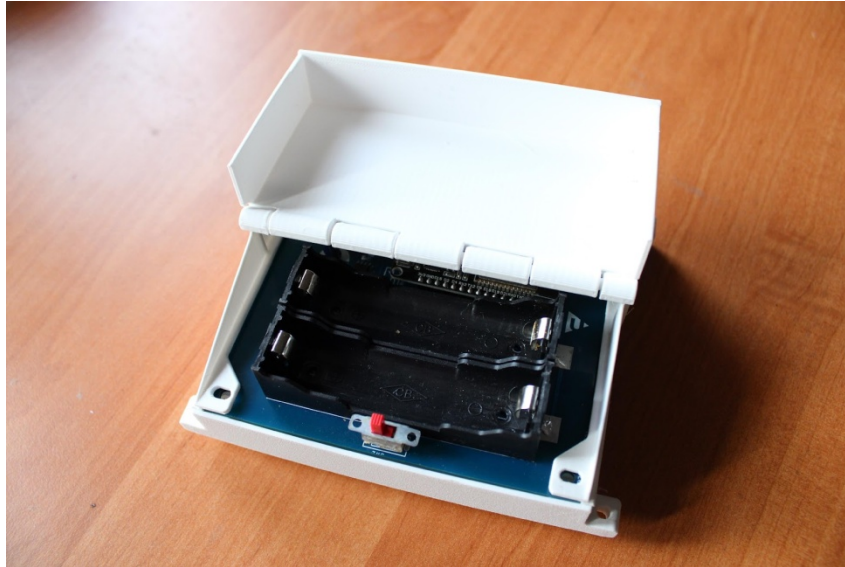
Výroba DPS byla zadána čínskému výrobcí JLCPCB o němž byla zmínka výše. Tento výrobce poskytuje v daném segmentu výroby prototypů pravděpodobně nejlepší cenu, nevýhodou je zde oproti tuzemským výrobcům delší doba dodání.



Obr. 19 Vyrobená DPS



Obr. 20 Osazená DPS bez ESP32



Obr. 21 DPS v krabičce

V této fázi již zbývá pouze vložit lithiové akumulátory velikosti 18650. Tím je hardware vysílače kompletní.

Po dokončení této části hardwaru se vyskytly drobné problémy. Konkrétně nesprávně vymodelovaný otvor v krabičce pro připojení USB kabelu sloužícímu k napájení zařízení, dále nedostatečný prostor pro ESP32, pokud by bylo na DPS umístěno pomocí dutinových konektorů, jak bylo původně zamýšleno. Tyto problémy lze dále řešit úpravou 3D modelu krabičky a opětovným 3D tiskem. Dalším drobným problémem je přehřívání integrovaného obvodu TP4056, který je pomocí odporu $R6 = 1K2$ nastaven na proud až 1A. To lze řešit montáží chladiče na tento obvod nebo vhodnou změnou $R6$ tak, aby se proud snížil. Je třeba však dbát na to, že proud musí být stále dostatečný k tomu, aby pokryl i spotřebu ESP32 a připojených senzorů.

3.1.2 Příjímač

Hardware přijímače je v porovnání s vysílačem značně jednodušší. Rozdíl tkví především v tom, že přijímač nepotřebuje vlastní napájení. Jeho napájení je zajištěno pomocí USB portu, kterým je připojen k PC a který vedle funkce napájení zajišťuje také sériovou komunikaci.

Fakt, že na přijímací straně použijeme druhé ESP32 také přináší tu výhodu, že kalibrace dat bude řešena právě zde a EEPROM paměť přijímacího ESP32 zde poslouží jako úložiště pro kalibrační koeficienty. To přinese výhodu, že mimo záměrně vyvolanou kalibraci nebude možné tyto koeficienty změnit. Tím bude zajištěna vyšší bezpečnost a jistota, že nedojde k nechtěnému nebo neoprávněnému přepisu kalibračních koeficientů, což by jinak

vedlo ke znehodnocení výsledků měření. Toto samozřejmě platí, zanedbáme-li jiné okolní vlivy jako je například kosmické záření, které by mohlo mít na integritu uložených dat nepříznivé vlivy.

3.2 Software

Software celého systému vychází z předpokladů, které jsem popsal v předchozích kapitolách, ačkoliv jeho vývoj ve skutečnosti probíhal souběžně s vývojem hardwaru. Během práce na celém projektu se totiž vyskytlo hned několik problémů, které bylo nutné řešit a které vedly do jisté míry k určitým změnám v hardwarové koncepci systému. Detaily a řešením těchto problémů se budu věnovat v následujících kapitolách. Rozdělení softwaru je provedeno tak, aby vhodně odpovídalo samotnému hardwaru a aby tím bylo dosaženo co možná největší jednoduchosti při zachování spolehlivosti.

Principiálně software musí ve směru od senzorů k uživateli zajišťovat tyto funkce:

1. Konfigurace a inicializace komunikačních sběrnic
2. Identifikace připojených senzorů
3. Čtení dat z připojených senzorů
4. Odesílání načtených dat v reálném čase
5. Příjem dat v reálném čase a jejich korekce kalibračními konstantami
6. Odesílání dat po sériovém portu
7. Příjem dat ze sériového portu
8. Přepočítání zrychlení na rychlost
9. Převod z časové oblasti do frekvenční oblasti pomocí Fourierovy transformace
10. Vyhodnocení výsledků
11. Zobrazení výsledků uživateli

Z důvodů, které popíšu dále, bylo ale potřeba zavést i opačný směr komunikace, aby bylo možné ovládat vybrané prvky systému

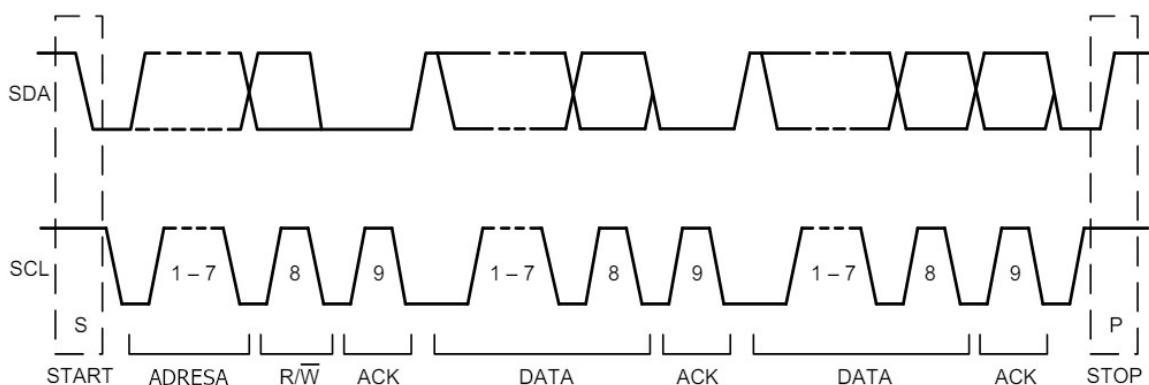
1. Výběr akcelerometru, se kterým chce uživatel pracovat a zobrazit data, která měří
2. Výběr zobrazení dat v časové či ve frekvenční oblasti
3. Zahájení kalibračního procesu vybraného čidla
4. Provedení uživatele celým kalibračním procesem
5. Ukládání a přepis kalibračních konstant v EEPROM paměti
6. Zpětná vazba o stavu systému pro uživatele

Komunikační rozhraní I²C

Jedná se o komunikační rozhraní využívající dva vodiče. Z tohoto důvodu se můžeme v dokumentaci setkat i s označením TWI (Two Wire Serial Interface), které používají například někteří výrobci mikrořadičů (Atmel).[8] Standardně jsou využívány dva vodiče SDA a SCL, kdy první slouží k přenosu dat a druhý jako hodinový signál. Vedle nich však obě zařízení musí být propojeny společnou zemí, čímž je nutné započítat ještě jeden vodič. Komunikující zařízení také často mají společné napájení, takže ještě jeden vodič navíc. Komunikující zařízení tedy bývá velmi často nutné propojit hned čtyřmi vodiči. Při tom ovšem existuje možnost komunikace jen jedním směrem v jednu chvíli. Prakticky se ale jedná o obousměrnou komunikaci, kdy je ale nutné zajistit, aby sběrnice byla před vysíláním z každého zařízení uvolněna.

Běžně bývá sběrnice provozována v režimu, kdy je přítomné jedno “master“ zařízení a jedno nebo více “slave“ zařízení, přičemž každé zařízení má přidělenou svoji adresu. Pouze master má na starosti řízení provozu po sběrnici a každé slave zařízení musí být o odesílání dat nejprve master zařízením požádáno. Jinak slave zařízení nesmí data samo odesílat. Funkce je pak taková, že master se všech slave zařízení neustále dotazuje na data, což je právě případ využitý v této práci.

Existují i jiné způsoby provozu této sběrnice, kdy například slave zařízení odesílají žádost o vysílání dat po jednom dalším vodiči, který zajišťuje přerušení procesoru, v němž proběhne komunikace výše popsaným způsobem.



Obr. 22 Časové schéma komunikace po sběrnici I²C (Datasheet MPU-6050)

Každou komunikační relaci zahajuje master zařízení tzv. start bitem – SDA je nastaveno na logickou nulu, přičemž SCL zůstává v logické jedničce. Poté proběhne odvyšlání adresy, se kterou má být zahájena komunikace a hned za ní je odvyšlán bit, který označuje, zda má dané zařízení číst data nebo je odeslat. Komunikace vždy probíhá po

osmi bitech, mezi kterými probíhá potvrzení ACK, které je dáno logickou nulou na SDA na deváté logické jedničce na SCL.

Komunikační protokol ESP-NOW

ESP-NOW is a kind of connectionless Wi-Fi communication protocol that is defined by Espressif. In ESP-NOW, application data is encapsulated in a vendor-specific action frame and then transmitted from one Wi-Fi device to another without connection. [7]

Výše uvedená citace dokumentace výrobce říká, že ESP-NOW je komunikační protokol založený na Wi-Fi vyvinutý společností Espressif. Tento protokol sdružuje data do paketů a odesílá je z jednoho zařízení na druhé bez nutnosti nastavit připojení jako je tomu v případě Wi-Fi.

Každý packet má vždy hned na začátku hlavičku obsahující MAC adresy vysílajícího ESP32 a ESP32, kterému je packet určen. Toto je velmi podobné běžnému packetu, jaký využívá Wi-Fi podle normy IEEE 802.11. Dále packet obsahuje kód kategorie (1 byte), dle výrobce nastaven na hodnotu 127 a identifikátor organizace (3 byty), výrobcem nastaven na hodnotu v hexadecimální soustavě vyjádřenou jako 0x18fe34. Poté jsou v packetu čtyři byty náhodných čísel. Ty slouží k zabezpečení přenášených dat. Poté následuje “vendor-specific content“. Ten obsahuje prvních sedm bitů zahrnující náležitosti jako ID Elementu, délka identifikátoru organizace, typu, verze a těla packetu, samotný identifikátor organizace, typ a verzi. Po nich následují již vlastní přenášená data (tělo), jimž je v každém packetu přiděleno 250 bytů.

Element ID	Length	Organization Identifier	Type	Version	Body
1 byte	1 byte	3 bytes	1 byte	1 byte	0-250 bytes

Obr. 23 Organizace části packetu “Vendor Specific content” nesoucí data [7]

Na konci každého packetu se ještě nachází 4 byty FCS, tedy Frame check sequence. Ta slouží k protichybovému zabezpečení přenosu.

Standardně tento protokol přenáší data rychlostí 1 Mb/s a díky své jednoduché implementaci a ceně příslušného hardwaru nachází velmi často uplatnění v IoT aplikacích, jako je chytrá domácnost, dálkové ovládání, či právě k přenosu dat z různých senzorů, což je právě tento případ využití.

3.2.1 Vysílač

Jak již bylo naznačeno výše, software vysílače bude mít dvě základní funkce, konkrétně čtení dat z akcelerometrů a jejich odesílání do přijímače.

Vše je nutné nejprve inicializovat. K tomu, jak již předchozí kapitola napovídá, slouží funkce *setup()*. Ještě než se k ní dostanu, bych věnoval další kapitolu definicím, deklaracím a knihovnám.

Knihovny, definice a deklarace

Na samotném začátku programu je nutné nejprve deklarovat knihovny, které bude program využívat.

V případě vysílače to jsou tyto:

- `esp_now.h` – Slouží ke konfiguraci zařízení a komunikaci s využitím protokolu ESP-NOW
- `WiFi.h` – Je potřeba ke konfiguraci transceiveru ESP32
- `Adafruit_MPU6050.h` – Slouží ke konfiguraci a čtení dat z MPU-6050
- `Adafruit_SHT31.h` – Slouží ke konfiguraci a čtení dat ze senzoru vlhkosti SHT31
- `Adafruit_Sensor.h` – Podpůrná knihovna pro dvě předchozí knihovny
- `Wire.h` – Knihovna sloužící pro komunikaci po sběrnici I²C.

Dále definujeme konstanty *pocet_mpu* a *wait_it_cnt*, kromě nich definujeme i GPIO pro softwarovou I²C sběrnici.

```
#define SDA2 26
#define SCL2 27
```

Obr. 24 Definice pinů pro softwarové I²C rozhraní

Tyto piny jsem zvolil na základě předchozích zkušeností získaných z pokusů zprovoznit co nejvíce softwarových I²C sběrnic. V případě požadavku na jiné hardwarové uspořádání lze zvolit i jiné. MAC adresu přijímacího ESP32 deklaruje jako pole čísel v šestnáctkové soustavě. V další části deklaruje počáteční hodnoty pro proměnné *icnt*, kterou používáme pro odpočet iterací, a proměnných *tm_l* a *tm_now*, které využíváme pro výpočet časového kroku. Poté definujeme datové struktury *prij_hodnoty* pro přijímaná a *odes_hodnoty* pro odesílaná data. Dále voláme konstruktory objektů `Adafruit_MPU6050` a `Adafruit_SHT31`. Tyto objekty slouží pro zjednodušení práce s příslušnými senzory.

Nakonec deklaruujeme strukturu *vlastnostiTX*, do které budou později uložena data přijímače.

Funkce setup()

Zde je nejprve nastavena frekvence procesoru na 240 MHz, pracovní režim Wi-Fi vysílače, dále je inicializována sériová linka a nakonec je inicializován protokol ESP-NOW.

V další části funkce *setup()* je definováno základní nastavení vysílače. Jmenovitě jde o MAC adresu přijímače, dále kanál (frekvence) a povolení šifrování přenosu. Po uložení těchto atributů je deklarována callback funkce *OnDataRecv*, která bude automaticky volána vždy, jakmile budou přijata data. Navzdory tomu, že se jedná o vysílač, je i zde nutné tuto funkci mít, aby bylo možné přijímat požadavek na čtení dat z konkrétního akcelerometru.

V poslední části funkce *setup()* jsou volány uživatelské funkce *i2c_init(int i)*. Tyto funkce, jak již jejich názvy napovídají, slouží k inicializaci I²C sběrnic, přičemž argument *i* udává, o jakou sběrnici se jedná.

Funkce i2c_init(int i)

Podle vstupní proměnné *i* se zvolí sběrnice, která se má inicializovat. K tomu je zavolána příslušná metoda *Wire.begin()* pro HW I²C sběrnici, nebo *Wire1.begin(SDA2,SCL2,100000U)* pro softwarovou sběrnici. Deklarace SDA2 a SCL2 byla popsána výše.

Toto mapování pinů vychází z pokusů zprovoznit více nezávislých softwarových sběrnic I²C za účelem čtení dat z více než čtyř MPU-6050. Nikdy se mi však spolehlivě nepodařilo zprovoznit více než dvě I²C sběrnice bez ohledu na to, zda byla jedna z nich hardwarová, či nikoliv.

Podle toho jakou hodnotu vrátí metoda *Wire.begin()* nebo *Wire1.begin()* vypíše funkce *i2c_init* po sériové lince výsledek operace.

Hlavní smyčka programu

Viz výše, běží ve funkci *loop()*. Hned první funkce, která je v této smyčce volána je *mpu_handler(int i)*, kde vstupní proměnná *i* označuje akcelerometr, jehož data mají být načtena. Do argumentu rovnou vkládám proměnnou *acc_w* ze struktury *prij_hodnoty*. Tato

struktura vždy uchovává naposledy přijatá data a je přepsána vždy až v okamžiku, kdy jsou přijata nová data.

Dále je v hlavní smyčce algoritmus určení časového kroku *dt*. Časový krok *dt* je z důvodu interpretace dat nutné odesílat ke zpracování. Výsledek této operace je tedy uložen do struktury *odes_hodnoty*.

Poté jsou v hlavní smyčce příkazy pro sériový výpis dat udávajících stav připojených akcelerometrů a přijatého požadavku na čtení dat z konkrétního akcelerometru. Tato data slouží pouze pro ověření správné funkčnosti a pro samotnou funkci není tento výpis nutný.

Nakonec je zavolána funkce *esp_now_send(RXMAC, (uint8_t *) &odes_hodnoty, sizeof(odes_hodnoty))*. První vstupní proměnnou je zde MAC adresa přijímacího ESP32, *(uint8_t *) &odes_hodnoty* udává adresu struktury odesílaných dat, přetypovanou na ukazatel na osmibitové celé kladné číslo. Zápis musí být bohužel proveden takto složitě neboť uvedná funkce má jako vstupní proměnnou *const uint8_t *data*, tedy ukazatel na proměnnou data.

Funkce *mpu_handler(int i)*

Jak již název funkce naznačuje je to nadřazená funkce pro administrativu nad akcelerometry. Vždy jednou za 100 iterací voláním funkce *i2c_test(int i)* zjišťuje, které akcelerometry jsou zrovna připojené a pamatuje si, které byly připojeny v předchozí iteraci. Je zde i část kódu, která provádí stejnou operaci pro snímač vlhkosti vzduchu a je-li připojen, pak je načtena vlhkost a je uložena do odesílané struktury.

Pokud byl připojen nový akcelerometr, je zavolána funkce *sens_init(int i)* pro příslušný akcelerometr, která zajistí jeho inicializaci a zahájí komunikaci s ním.

Dále pokud je uživatelem požadovaný akcelerometr připojen, zavolá se funkce *cti_acc(int i)*, která načte měřená zrychlení. Pokud požadovaný akcelerometr není připojen, budou v odesílané struktuře na místo zrychlení ve všech osách zapsány nuly.

Funkce `i2c_test(int i)`

Podle toho, který akcelerometr testuje, jsou zavolány příslušné metody pro zahájení a ukončení komunikace s konkrétním čidlem. Je-li na komunikaci odezva, pak je čidlo evidentně připojeno. Tyto metody vypadají pro akcelerometr na hardwarové sběrnici s adresou 0x68 takto:

```
Wire.beginTransmission(0x68);
odes_hodnoty.mpu_status[0] = !Wire.endTransmission();
```

Obr. 25 Algoritmus testující, zda je připojeno zařízení s adresou 0x68

Funkce `sens_init(int i)`

V případě, že je čidlo nově nebo opětovně připojeno, provede tato funkce jeho inicializaci a konfiguraci. Pro příklad uvedu část funkce, která proběhne, je-li argument roven nule – inicializujeme první akcelerometr:

```
if(MPU1.begin(0x68))
{
  MPU1.setAccelerometerRange(MPU6050_RANGE_8_G);
  MPU1.setFilterBandwidth(MPU6050_BAND_260_HZ);
  okm = 1;
}
```

Obr. 26 Inicializace a konfigurace akcelerometru

Metoda `MPU1.begin(0x68)` inicializuje komunikaci s MPU-6050 na adrese 0x68. Je-li inicializace v pořádku, jsou zavolány ostatní metody, které nastaví rozsah akcelerometru a vestavěného dolnoproputného filtru. Proměnná `okm` zde pouze slouží u výpisu, který je na konci funkce. Je-li `okm = 1`, znamená to, že inicializace proběhla v pořádku.

Funkce `cti_acc(int i)`

Tato funkce opět, stejně jako předchozí funkce vybírá akcelerometr k načtení dat podle své vstupní proměnné. Protože se mi z důvodu nižší rychlosti neosvědčilo číst data způsobem, který umožňuje knihovna `Adafruit_MPU6050`, využil jsem alternativní přístup, který používá pouze metody knihovny `Wire` sloužící k zajištění jednoduché komunikace po I²C sběrnici:

```
Wire.beginTransmission(0x68);
Wire.write(0x3B);
Wire.endTransmission(false);
Wire.requestFrom(0x68, 6, true);
odes_hodnoty.acc.x = (Wire.read() << 8) | Wire.read();
odes_hodnoty.acc.y = (Wire.read() << 8) | Wire.read();
odes_hodnoty.acc.z = (Wire.read() << 8) | Wire.read();
```

Obr. 27 Hlavní kód funkce `cti_acc(int i)`

Metoda `beginTransmission(0x68)` zahájí komunikaci se zařízením na adrese `0x68`, což je zde konkrétní akcelerometr na HW sběrnici a metoda `write(0x3B)` připraví do fronty požadavek na data z registru `0x3B` v akcelerometru. Až metoda `endTransmission(false)` jej zapíše na sběrnici a pozice čtení je tak nastavena na registr `0x3B`, který uchovává načtená data snímaného zrychlení. Argument `false` však stále udržuje sběrnici v režimu komunikace, čímž zamezuje nežádoucí komunikaci od jiných zařízení. Metoda `requestFrom(0x68, 6, true)` požádá akcelerometr na adrese `0x68` o 6 bytů z vybraného registru, přičemž argumentem `true` je nastavena tak, aby po dokončení čtení dat uvolnila sběrnici.

Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT_H	R	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT_L	R	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT_H	R	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT_L	R	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT_H	R	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT_L	R	ACCEL_ZOUT[7:0]							

Obr. 28 Tabulka dat zrychlení v registrech MPU-6050 (Datasheet MPU-6050)

Metoda `read()` pak vždy načte právě jeden byte, tedy osm bitů. Protože načtené hodnoty z MEMS akcelerometru jsou ale převáděny A/D převodníkem s rozlišením 16 bitů jsou data v paměti vždy podle tabulky uložena jako osm více významných bitů, které jsou v tabulce označeny jako `ACCEL_XOUT_H` a osm méně významných bitů, označených jako `ACCEL_XOUT_L`. Do výsledku tedy načteme nejprve osm významnějších bitů a provedeme jejich bitový posun vlevo o osm, čímž je dostaneme v šestnáctibitovém čísle na

své místo. Poté načteme zbylých osm méně významných bitů a provedeme mezi nimi a již načteným číslem bitovou operaci OR. Tím budou „zprava natažené“ nuly nahrazeny právě těmito méně významnými bity. Tím je výsledné číslo složeno zpět a uloženo do odesílané struktury jako *int16_t*. Tento proces se opakuje ještě pro zbylé dvě osy.

3.2.2 Příjímač

Software přijímače zajišťuje funkce uvedené v bodech 5 až 7 seznamu funkcí v kapitole 2.1.4 Software, tedy bezdrátový příjem dat pomocí protokolu ESP-NOW, korekce těchto dat kalibračními konstantami a jejich odesílání po sériové lince v reálném čase.

Aby bylo možné provádět korekci kalibračními koeficienty, musejí být tyto koeficienty nejprve získány vhodnou metodou. Pokud již byla provedena kalibrace, jsou korekční konstanty načteny z EEPROM paměti čipu, aby nebylo nutné senzory kalibrovat vždy po opětovném spuštění znovu. Pro provedení kalibrace má software přijímače vlastní funkci, jejíž chod je řízen uživatelem.

Program, opět jako v předchozí kapitole, rozeberu postupně od deklarací a definic až po jednotlivé uživatelské funkce.

Knihovny, definice a deklarace

Program přijímače využívá tyto knihovny:

- *esp_now.h* – funkce viz kapitola 3.2.1 vysílač
- *WiFi.h* – funkce rovněž ve výše uvedené kapitole
- *string.h* – definuje proměnnou *String* a funkce pro operace s ní.
- *EEPROM.h* – zahrnuje funkce pro operace s pamětí EEPROM.

Definice a makra zahrnují konstantu *A_G*, která představuje gravitační zrychlení Země. Konstanta *pocet_mpu_na_esp* definuje kolik celkem je k vysílací straně maximálně připojeno zařízení na I²C sběrnici. Makro *EEPROM_SIZE* je vypočteno jako *pocet_mpu_na_esp*6*4*, kde *6* představuje fakt, že u každého akcelerometru musí být pro každou ze tří os uložen absolutní člen a lineární člen a *4* představuje 4 byty, jelikož právě tolik zabírá proměnná typu *float*.

V další části jsou deklarace proměnných:

TXMAC je adresa vysílacího ESP32, ačkoliv zde se v podstatě jedná o adresu přijímače, protože zde konfigurujeme protokol ESP-NOW sice velmi podobně jako v případě vysílače, zde se ale bude jednat o vysílání opačným směrem. Pojem „Vysílací ESP32“ zde

tedy užívám pouze ve vztahu k běžnému pracovnímu stavu systému, kdy komunikace opačným směrem neprobíhá.

Globální proměnná *prikaz* bude sloužit k řízení jednotlivých funkcí, kdy pokud přijde příkaz po sériové lince, bude tato proměnná změněna z nuly na číslo příslušného příkazu po vykonání daného příkazu se změní zpět na nulu.

Dále jsou deklarovány pole korekčních proměnných o délce odpovídající počtu připojených senzorů minus jeden, protože poslední senzor je pouze snímač vlhkosti.

Potom jsou stejně jako v případě vysílače deklarovány struktury obsahující bezdrátově přenášená data.

Funkce setup()

Nejprve zde proběhne nastavení frekvence procesoru, režimu WiFi, inicializace komunikace po sériové lince a protokolu ESP-NOW. Hned poté jsou do proměnné *vlastnostiTX* uloženy konfigurační parametry vysílače ESP, ačkoliv zde platí, že se v daném případě jedná vlastně o přijímač. Tato inicializace tak má stejné náležitosti jako v případě vysílače, jen názvy jsou zde pozměněny.

Funkce *setup()* se však od této funkce vysílače dále již liší. Je zde inicializována komunikace s EEPROM pamětí a hned potom je ve *for* cyklu volána funkce *nacti_kal_data(int i)*. Tato funkce má za úkol načíst korekční konstanty z EEPROM paměti. Ve stejném *for* cyklu s ní jsou provedeny výpisy kalibračních konstant po sériové lince. Tato část kódu slouží pouze ke kontrolním účelům.

Nakonec je deklarována callback funkce *OnDataRecv*, která je volána protokolem ESP-NOW, kdykoliv jsou přijata data.

Funkce loop()

Jde o hlavní smyčku, ve které program vysílače běží.

Pomocí metody *Serial.available()* kontroluje, zda nepřišel příkaz po sériové lince. V případě, že ano, je zavolána funkce *command_handler(int prikaz)*, kde vstupní proměnná vychází z funkce *q_prikaz*, která čte příkazy ze sériové linky, porovnává je se svou databází příkazů a na základě toho vrátí číslo příchozího příkazu. Funkce *command_handler* na základě čísla příkazu provede příslušné kroky. Detailní rozbor této funkce bude následovat dále.

V každé iteraci hlavní smyčky je volána funkce *S_vypis(int i)*, které se věnuje následující část textu.

Funkce `S_vypis(int i)`

Jejím argumentem je číslo požadovaného akcelerometru. Tato funkce slouží nejen k výpisu, ale ještě před výpisem dat na sériovou linku provede jejich korekci kalibračními koeficienty. Kromě toho provádí přepočítání časového kroku na frekvenci, což jednak zmenšuje náročnost na přenášená data a také tím získáme rovnou vzorkovací frekvenci, kterou budeme později potřebovat pro správnou interpretaci dat ve frekvenční oblasti. Tato funkce také provádí převod „binárního čísla“ uloženého v poli `mpu_status` na číslo dekadické. Tím je opět zjednodušen výpis po sériové lince.

Funkce `command_handler(int prikaz)`

Nejprve testuje, zda je příkaz číslo mezi 101 a 104 – tedy příkaz požadující výpis dat z určitého akcelerometru. Pokud se jedná o tento příkaz, vypíše se na sériové lince `ack` (acknowledge – potvrzení příjmu), odešle se vysílači požadavek na čtení z jiného akcelerometru, přičemž tento požadavek zůstává stále uložen v příslušné proměnné a je dále přijímačem používán jako argument funkce `S_vypis`, kde slouží k výběru správné korekční konstanty.

Pokud je příkaz roven číslo jedna, pak to znamená, že má být zahájena kalibrace aktuálně vybraného akcelerometru. Aby nemohlo dojít ke kalibraci omylem, je po sériové lince odeslána odpověď `ack` a program čeká, až bude odpověď potvrzena příjmem `ack` od řídicího systému, který odpovídá příkazu číslo 2. Jakmile se toto stane, zavolá se kalibrační funkce `calib(int i)`, jejíž funkci popíšu později. Tato funkce má logickou výstupní hodnotu, která udává, zda funkce proběhla v pořádku, či nikoliv. Podle toho, zda tedy kalibrace proběhne v pořádku, je buď vypsáno na sériovou linku `cal_err` nebo `cal_comp`. V obou případech se čeká na potvrzení `ack` od řídicího systému.

V případě, že číslo příkazu není ani mezi 101 a 104 ani 1, pak je na sériové lince vypsáno `n_ack`, tedy not acknowledge (nerozumím).

Na konci celé funkce je proměnná `prikaz` přepsána na nulu, aby nemohlo dojít k tomu, že bude jinde v programu nesprávně interpretována.

Funkce `calib(int i)`

Tato funkce slouží k provedení kalibrace vybraného čidla, které je dáno vstupní proměnnou.

Princip kalibrace spočívá v tom, že pro každou osu je načtena hodnota gravitačního zrychlení, která udává, je-li daná osa svisele k zemskému povrchu, čímž na ni působí přesně definované gravitační zrychlení Země. Toto měření se provede pro každou osu jak ve směru nahoru, tak dolů. Vychází se z předpokladu, že je-li osa čidla kolmo k zemi čidlo a je v klidu, pak musí daná osa vykazovat právě gravitační zrychlení Země. Během procesu kalibrace jsou načtena gravitační zrychlení pro každou osu nahoru a dolů, tedy dohromady šest hodnot zrychlení. Tím získáme hodnoty pro výpočet tří soustav dvou rovnic pro dvě neznámé. Tyto neznámé jsou právě hledané kalibrační konstanty.

Funkce nejprve přepíše kalibrační konstanty tak, že lineární členy jsou přepsány na jedničku a absolutní členy na nulu. Tím je zajištěno, že zaznamenané hodnoty nebudou původními kalibračními konstantami nijak ovlivněny.

Funkce si dále deklaruje lokální proměnné, do kterých budou ukládány hodnoty načtené během kalibrace.

Pole d a proměnná sum_done slouží v kalibraci pouze pro kontrolu, zda již byly načteny všechny hodnoty, které jsou potřeba.

Hlavní částí kalibrační funkce je cyklus $while(sum_done < 6)$. Ten běží, dokud nejsou načtena všechna požadovaná zrychlení. Tento cyklus vždy čeká na příkaz, který je vyhodnocen vždy výše popsanou funkcí q_prikaz . Příkazy vždy udávají, pro kterou polohu akcelerometru má být načtena která hodnota. Vždy po jejím načtení je načtená hodnota odeslána po sériové lince a čeká se na příjem ack od řídicího systému, který tak potvrdí příjem dané hodnoty za účelem jejího zobrazení uživateli. Jakmile jsou načteny všechny požadované hodnoty, jsou kalibrační konstanty vypočteny z výše zmíněných soustav rovnic. Tyto rovnice vycházejí z dobře známého předpisu lineární funkce:

$$y = kx + q \quad (3.1)$$

Kde y je výsledná (zobrazovaná hodnota), x je surová hodnota (uint16) na výstupu A/D převodníku senzoru a k a q jsou korekční konstanty.

Program už obsahuje tyto rovnice přepsané do tvaru, z něhož již přímo vycházejí hledané konstanty:

```
kx[i] = 2*A_G/(x_u-x_d);
qx[i] = A_G-(kx[i]*x_u);
```

Obr. 29 Výpočet kalibračních konstant

Před samotným výpočtem je ještě kontrola zda rozdíl proměnných x_u a x_d není nulový, aby nedošlo k dělení nulou.

Po výpočtu této soustavy rovnic jsou vypočtené konstanty uloženy do paměti EEPROM. K tomu slouží metody *EEPROM.writeFloat(int adresa, float hodnota)* a *EEPROM.commit()*;

Na konci funkce jsou uložené proměnné načteny voláním funkce *nacti_kal_data(int i)*.

Funkce *nacti_kal_data(int i)*

Načítá data uložená v EEPROM s využitím metody *EEPROM.readFloat(int adresa)*. Po jejich načtení tyto data vypíše po sériové lince. Tento výpis nemá funkční účel, slouží pouze pro kontrolu.

3.2.3 PC – MATLAB aplikace

Pro zobrazení výsledků celého systému bude sloužit samostatná aplikace vytvořená v prostředí MATLAB. Ano, nejedná se o pouhý skript v MATLABu, ale o celou aplikaci s uživatelským rozhraním. K jejímu vytvoření lze využít nástroj známý jako MATLAB App Designer.

App Designer lets you create professional apps without having to be a professional software developer. Drag and drop visual components to lay out the design of your graphical user interface (GUI) and use the integrated editor to quickly program its behavior. [6]

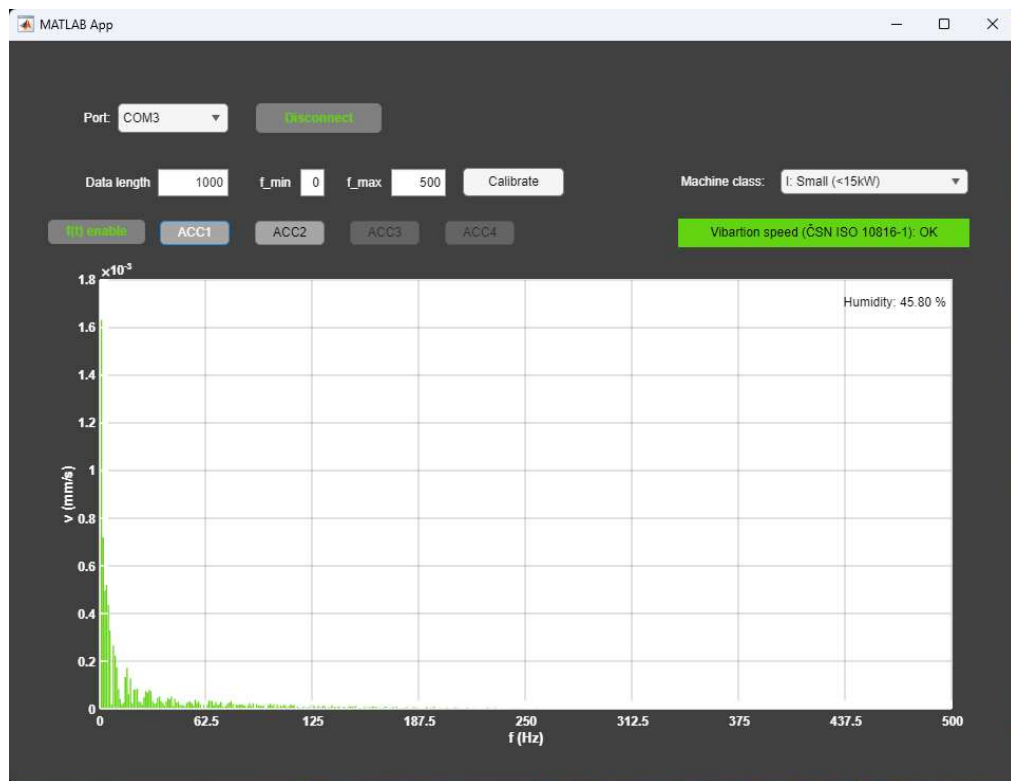
Takto je tento vývojový nástroj popsán přímo na webových stránkách společnosti MathWorks. Je zde také uveden fakt, že vývojové prostředí využívá nástroje jako MATLAB Compiler a Simulink Compiler k vytvoření samostatných aplikací s uživatelským rozhraním navrženém v tomto nástroji a naprogramovaných v jazyce MATLAB. Tento vývojový nástroj tak podporuje příkazy, které jsou v tomto jazyce využívány a výrazně tak usnadňuje implementaci algoritmů pro provádění matematické analýzy dat.

Již prvotní pokusy s tímto rozhraním prokázaly, že aplikace vytvořená touto cestou bez problému dokáže fungovat v reálném čase. Jak jsem již dříve v této práci uvedl, vyzkoušel jsem podobnou aplikaci vytvořit i v prostředí NI LabVIEW. Toto prostředí se sice z počátku jeví jako stabilnější, netrpí drobnými záseky a podobně. Bohužel, jakmile vzniknou složitější požadavky na matematickou analýzu získaných dat, začne zde být implementace podstatně složitější než v prostředí MATLAB a nakonec nedosahuje takové spolehlivosti.

Uživatelské rozhraní

Uživatelské rozhraní se v MATLAB App Designeru navrhuje metodou „Drag and drop“. Podobný způsob návrhu je již dobře známý například z tvorby jednoduchých aplikací pro Windows v jazyce Visual Basic.

Uživatelské rozhraní své aplikace jsem navrhnul tak, aby bylo co možná nejpřehlednější.



Obr. 30 Uživatelské rozhraní MATLAB aplikace

Je vidět, že hned první řádek odkazuje na výběr sériového portu, na který je připojen přijímač ESP32. Jakmile je vybrán správný port, zpřístupní se tlačítko “Connect“ pro připojení. Po kliknutí na toto tlačítko je zahájena komunikace se zařízením a nápis na tlačítku se změní na “Disconnect“. Pokud bylo vybráno správné zařízení, aplikace již nyní zobrazí přijatá data ve frekvenční oblasti. Pokud byl systém čerstvě spuštěn a nebyl mu dosud zadán příkaz, pak tato data jsou vždy z akcelerometru číslo 1. Není-li tento akcelerometr připojen, pak jsou výsledky Fourierovy transformace velmi blízké nule – jedná se o zaokrouhlovací chyby procesoru. Uživateli jsou rovněž zpřístupněny volby akcelerometrů, které jsou aktuálně připojeny. Po kliknutí na vybraný akcelerometr začnou být načítána jím měřená data, což je zároveň signalizováno tím, že text tohoto akcelerometru změní svoji barvu z černé na bílou.

Uživatel si může přepsáním hodnot v okénku “Data length“ zvolit délku okna pro vyhodnocení dat Fourierovou transformací, případně pro zobrazení v časové oblasti. Pokud uživatel chce zobrazit nějakou konkrétní část spektra ve frekvenční oblasti, slouží k tomu okénka f_{\min} a f_{\max} . Jak jsem již naznačil výše, data je možné zobrazit jak v časové, tak ve frekvenční oblasti. K tomu slouží tlačítko “f(t) enable“. Posledním tlačítkem, které se v této části nachází je tlačítko “Calibrate“. Toto tlačítko je odemčeno vždy až po výběru akcelerometru a slouží pro zahájení jeho kalibrační procedury.

V pravé části aplikace se nachází část sloužící k vyhodnocení vibrací. Zde si uživatel v sekci “Machine class” zvolí velikost testovaného stroje podle normy ČSN ISO 10816-1. Pod touto sekcí je zobrazen výsledek odpovídající tabulce, která je uvedena v této normě. Světle zeleně je zobrazen výsledek, kdy jde o malé kmitání a stroj lze bez problému provozovat. Tmavě zeleně je zobrazen stav, kdy je měřeno ještě přípustné kmitání, stroj lze stále normálně provozovat, ale lze očekávat, že by se vibrace mohly začít zhoršovat. Žlutě je zobrazen stav, kdy stroj již kmitá na mezi přípustnosti, což znamená, že je třeba neprodleně naplánovat odstávku a příslušnou opravu. Červeně je nakonec zobrazen stav nepřípustného kmitání, kdy je potřeba stroj okamžitě odstavit. Je třeba mít na paměti, že tyto hodnoty jsou pouze dány normou ČSN ISO 20816, ale existují i stroje se specifickými předpoklady a požadavky, které se mohou lišit, proto nelze ve všech případech považovat tyto výsledky za zcela směrodatné a je vždy třeba se řídit informacemi dostupnými ke konkrétnímu diagnostikovanému stroji.

Kód aplikace

App Designer podle prvků vytvořených v editoru uživatelského rozhraní tvoří některé části kódu automaticky a uživatel je nemůže měnit. V této kapitole se tak budu zabývat jen kódem, který jsem vytvořil tak, aby aplikace fungovala podle dříve zmíněných požadavků.

Na začátku jsou definovány globální proměnné programu v sekci *properties* (*Access = public*) a v sekci *methods* (*Access = private*) jsou definovány uživatelské funkce. Nachází se zde vždy jak tělo funkce, tak její hlavička.

```

properties (Access = public)
    serial_ins;
    pripojeno;
    Mat;
    sloupcu = 6;
    ActiveACC;
    a_abs;
    v_vekt;
    fmax;
    VmaxISO = [0.71,1.12,1.8, 2.8;
               1.8, 2.8 ,4.5, 7.1;
               4.5, 7.1,11.2,18 ];
    machine = 1;
    cal_proc_stg = 0;
end

methods (Access = private)

    function nacti_porty(app)
    function connected(app)
    function quali(app,rmsv)
    function vykresli_fft(app)
    function ft_lims(app)
    function routine(app)
    function nacteno(app,text)
end

```

Obr. 31 První část programu zahrnující globální proměnné a uživatelské funkce

Uživatelská funkce *nacti_porty(app)* využívá MATLAB funkci “*serialportlist*” k načtení všech dostupných sériových portů připojených k PC a takto získaná data ukládá do seznamu portů, které uživatel zobrazí ve formě drop-down menu. Je-li v tomto seznamu alespoň jedna položka, je zpřístupněno tlačítko “Connect” sloužící k připojení k vybranému portu. Samotné připojení potom provede callback funkce *app.ConnectButtonPushed(app, event)*, která je volaná v okamžiku, kdy uživatel klikne na tlačítko “Connect“. Hlavní roli zde hraje funkce *serialport(COM port, Baud rate)*, která jednak provede připojení k portu a jeho vlastnosti nahraje do proměnné *serial_ins*, která dále bude sloužit jako argument funkcí pracujících s tímto portem. Na konci této callback funkce je volána uživatelská funkce *connected(app)*.

Funkce *connected(app)* zahrnuje hlavní pracovní smyčku programu, která pomocí funkce *readline(app.serial_ins)* zajišťuje čtení dat ze sériového portu a pomocí funkce *sscanf* tato data ukládá do matice *Mat*, přičemž je vždy kontrolován formát těchto dat. Před každým zápisem řádku je volána funkce *circshift(app.Mat, [1 0])*, která posune všechny řádky v této matici o jeden dolů a hodnoty posledního řádku jsou zapsány na první řádek této matice. Tento řádek je však okamžitě přepsán nově načtenými hodnotami. Tento cyklus načítání dat běží 70x a poté je zavolána funkce *routine(app)*, která provádí další operace, které jsou náročnější, ale stačí je provádět méně často. Tím je zajištěna menší náročnost a větší plynulost programu.

Funkce *routine(app)* nejprve prověří stav tlačítka *f(t) enable* a podle toho buď rovnou vykreslí načtená data v časové oblasti a nebo zavolá funkci *vykresli_fft(app)*, která podle uživatelem nastavených parametrů vykreslí vektor rychlosti ve frekvenční oblasti. Vektor rychlosti je vždy vypočten z absolutní hodnoty zrychlení pomocí funkce *vyp_v(app)*. Dále funkce *routine(app)* volá funkci *quali(app.rmsv)*, která pouze porovnává naměřená data s hodnotami danými normou a na základě toho vyhodnocuje stav měřeného stroje. Ve funkci *routine(app)* je poté zjištěn stav připojených akcelerometrů a načtena měřená relativní vlhkost.

Uživatelská funkce *ft_lims* je pouze podpůrná funkce pro výpis dat do grafu, kde nastavuje parametry osy x.

Funkce *nacteno(app, text)* slouží k provádění výpisů během kalibrace. A kromě toho také zajišťuje odesílání ACK, čímž řeší komunikaci s přijímacím ESP32.

Callback funkce

Dalšími funkcemi, které zajišťují správný chod programu jsou tzv. callback funkce. Tyto funkce jsou automaticky volány při interakci uživatele s konkrétními prvky programu. Mezi ně patří také funkce *startupFcn(app)*, která je volána vždy při spuštění aplikace. Zde pouze volá funkci *nacti_porty(app)* a nastavuje barvu popisků grafu. Funkci *nacti_porty(app)* taktéž volá callback *PortDropDownOpening(app, event)*, která je vždy volána, jakmile uživatel otevře rozbalovací menu portů.

Callback funkci *ConnectButtonPushed(app, event)* jsem již popsal výše.

Jakmile uživatel změní délku okna, je zavolána funkce *DatalengthEditField.Value(app, event)*. Ta zjistí opětovnou alokaci paměti pro proměnné jako je matice *Mat* a vektory *a_abs* a *v_vekt*. Zároveň nové délce okna přizpůsobí osu x grafu voláním funkce *ft_lims()*;

Protože řízení výpisu grafu je již dáno hodnotou, která se mění při stisknutí tlačítka “f(t) enable“, v callback funkci tohoto tlačítka jsou již jen přenastaveny popisky grafu a rozsah jeho osy x.

Dále jsou zde Callback funkce pro stisknutí tlačítek ACC1 až ACC4. Každá z nich nejprve po sériové lince pošle požadavek na čtení dat z tohoto čidla, nastaví dané čidlo jako aktivní a nastaví barvu fontu tohoto čidla na bílou, aby bylo zřejmé, že je toto čidlo vybráno.

Callback funkce pro změnu vybrané položky v dropdown menu velikosti stroje obsahuje switch-case podmínku, podle které vybere uživatelem zvolenou třídu stroje, což

je nutné pro funkci $quali(app, rmsv)$, která na základě toho vybírá s jakým sloupcem v matici povolených rychlostí má porovnávat naměřené hodnoty.

Nejsložitější ze všech callback funkcí je bezpochyby $CalibrateButtonPushed(app, event)$. Aby byl celý proces kalibrace co možná nejjednodušší, je navržen tak, aby jej bylo možné ovládat právě jen tímto jedním tlačítkem, přičemž text tohoto tlačítka se v průběhu kalibrace mění. Tato callback funkce je řízena proměnnou cal_proc_stg , která obsahuje informaci o tom, ve kterém kroku se kalibrace nachází. Podle toho je vždy odeslán příslušný požadavek na načtení konkrétní hodnoty a pomocí funkce $nacteno(app, text)$ je pak zajištěna odezva na tento požadavek včetně konkrétní načtené hodnoty. Po dokončení těchto procesů je nastaven další kalibrační krok a příslušně tomu je nastaven text kalibračního tlačítka.

3.3 První validační měření

Validační měření proběhla nejprve generováním sinusového signálu pomocí samostatné funkce ve vysílacím ESP32. Tato funkce funguje na principu integrace úhlu θ , a to vždy s krokem $\Delta\theta$. Ten je určen z požadované frekvence a vzorkovací frekvence, resp. počtu vzorků v okně. Dále je v každém vzorku k vypočítána hodnota funkce sinus takto:

$$x(k) = A \cdot \sin(\theta(k)) \quad (3.2)$$

kde $x(k)$ je hodnota k -tého vzorku a A je amplituda. Hodnoty získané touto funkcí byly dosazeny místo naměřených hodnot. Již tato metoda ovšem naznačila závažné problémy v celém původním konceptu. Další pokusy odhalily, že problémy pramení především z nedostatečné rychlosti protokolu ESP-NOW, který evidentně není určen pro tak časté odesílání dat. Další věc je, že dle normy ČSN ISO 10816-6 nestačí vzorkovací frekvence 1 kHz, ale vyhodnocení ve frekvenční oblasti musí být do 1 kHz. Z toho důvodu bude dle Shannon-Kotělnikovova teorému nutné zajistit vzorkovací frekvenci alespoň 2 kHz. Z těchto důvodů bylo nutno systém do značné míry přepracovat.

Pozn. Kompletní kód původního softwaru je uveden v přílohách 4 až 6.

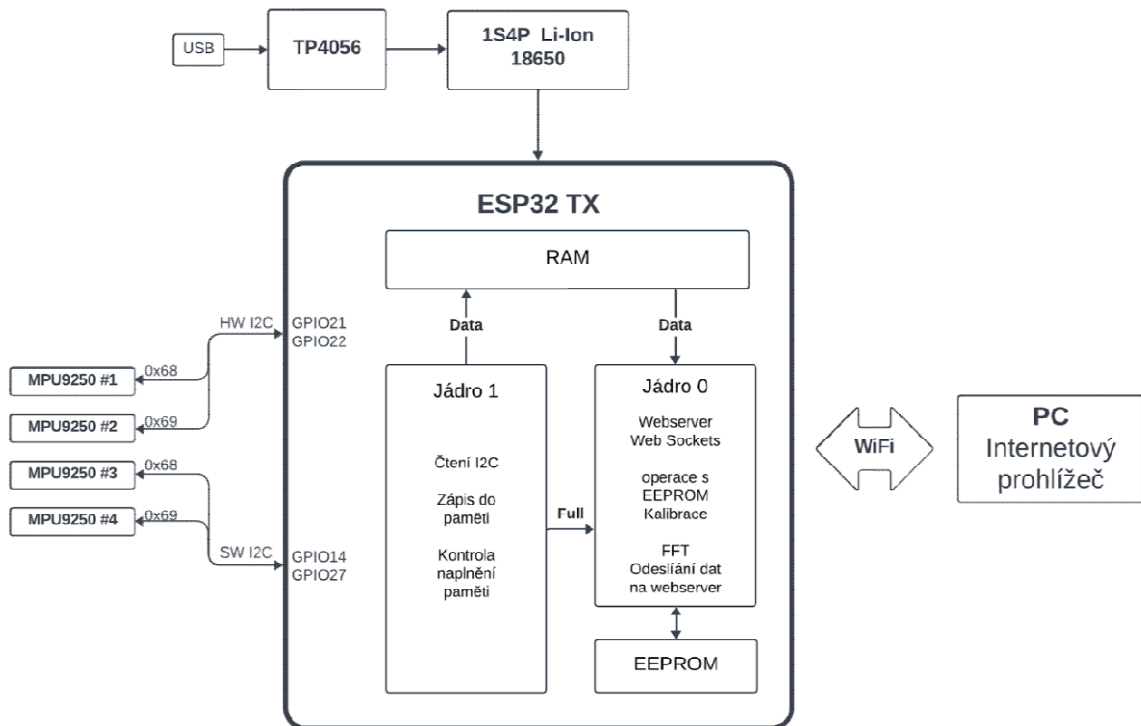
3.4 Upravená koncepce systému

Na základě toho, co je uvedeno v předchozí kapitole, byla příslušně upravena koncepce systému. Ta je nyní zaměřená primárně na plnění požadavků norem, zejména pak ČSN-ISO 20816. Myšlenkou této nové koncepce je vyřazení protokolu ESP-NOW. To si vyžádalo významné změny systému. Místo ESP-NOW bude využita běžná Wi-Fi. Vysílací ESP32 zde poslouží jako Access Point na kterém poběží Webový server. Značná část

matematických výpočtů pro zpracování dat se tak bude muset přenést na samotné vysílací ESP32. Aby bylo možné zároveň zajistit dostatečnou rychlost čtení dat po I²C sběrnici a k tomu spolehlivý provoz serveru a výpočty FFT, bude muset být na tomto ESP32 zaveden vícejádrový provoz, přičemž jedno jádro bude pracovat se sběrnici I²C a získaná data bude ukládat do paměti a druhé jádro vždy po naplnění paměťového bufferu naměřenými daty s nimi provede příslušné operace a výsledky odešle do rozhraní webového serveru.

Rovněž bude nutné nahradit stávající MEMS snímače typu MPU-6050 za lepší snímače typu MPU-9250, které mají vzorkovací frekvenci až 4 kHz. Zároveň bylo rozhodnuto o omezení měření ze tří os pouze na jednu z toho důvodu, že upevnění pomocí magnetických čoček nemá ve všech třech osách stejné parametry.

Základ pro upravenou koncepci je přenesen z původního řešení, ale významné změny plynoucí z výše uvedených informací budou shrnuty v následujících kapitolách.



Obr. 32 Nová koncepce vibrodiagnostického systému

Z obrázku plyne, že po hardwarové stránce došlo ke změně snímačů a k odstranění přijímacího ESP32. Další změny už se poté týkají softwaru ESP32.

3.5 Významné části softwaru upraveného systému

Snímače MPU-9250

Protože pro MPU-9250 neexistuje žádná vyhovující knihovna, bylo využito znalostí získaných při implementaci čtení dat přímo z registrů MPU-6050 k vytvoření vlastní funkce pro inicializaci jednotlivých snímačů přímým zápisem do jejich registrů:

```
uint8_t WriteReg(TwoWire &sbe, uint8_t addr, uint8_t reg, uint8_t hod)
{
    sbe.beginTransaction(addr);
    sbe.write(reg);
    sbe.write(hod);
    sbe.endTransmission();

    sbe.beginTransaction(addr);
    sbe.write(reg);
    sbe.endTransmission(false);
    sbe.requestFrom(addr,1U);
    if(sbe.read() == hod){return(1);}
    else{return(0);}
}
```

Obr. 33 Zápis do registrů pomocí uživatelské funkce *WriteReg*

Vstupní proměnná *sbe* odkazuje na konkrétní sběrnici, na které se obsluhovaný senzor nachází, *addr* poté definuje, jakou adresu má dané zařízení na sběrnici. Vstupní proměnná *reg* udává, do jakého registru má být zapsáno a nakonec proměnná *hod* udává, jaké hodnoty mají být do registru zapsány. Funkce je navržena tak, aby zapsané hodnoty byly znovu přečteny a porovnáním s požadovanou zapisovanou hodnotou je ověřeno, že zápis proběhl správně.

Konkrétně zápis do registrů MPU9250 vypadá takto:

```
WriteReg(Wire, 0x68, 0x6B, 0b10000000); //Reset
delayMicroseconds(30);
if(!WriteReg(Wire, 0x68, 0x1A, 0b00000000)) {okm = 0;} //CONFIG reistr.
if(!WriteReg(Wire, 0x68, 0x19, 0b00000000)) {okm = 0;} //SMPLRT_DIV reistr.
if(!WriteReg(Wire, 0x68, 0x1C, 0b00011000)) {okm = 0;} //ACCEL_CONFIG_1 reistr.
if(!WriteReg(Wire, 0x68, 0x1D, 0b00001100)) {okm = 0;} //ACCEL_CONFIG_2 reistr.
if(!WriteReg(Wire, 0x68, 0x6B, 0b00000000)) {okm = 0;} //PWR_MGMT_1 reistr.
```

Obr. 34 Hodnoty zapsané do registrů MPU-9250

První řádek, tedy zápis do registru 0x6B nastaví bit s indexem 7 na logickou jedničku. Tím je vyvolán reset všech registrů snímače a po resetu je tento bit opět vynulován. Funkce proto čeká 30 mikrosekund, aby proběhly všechny změny.

Po resetu je vynulován registr 0x1A s názvem CONFIG. To znamená, že žádná data ve FIFO paměti nečekají na přečtení, ale jsou neustále přepisována nejnovějšími hodnotami. Dále je zakázáno externí spouštění vzorkování dat a je vypnuta digitální dolní propust.

Registr 0x19 je rovněž vynulován. Jedná se o registr který by za jiných okolností sloužil k omezení vzorkovací frekvence. Omezení vzorkovací frekvence se zde totiž řídí tímto vztahem:

$$f_{vz} = \frac{f_{vzint}}{1 + SMPLRT\ DIV} \quad (3.3)$$

kde f_{vz} je konečná vzorkovací frekvence, f_{vzint} je základní vzorkovací frekvence snímače a SMPLRT DIV je hodnota nastavená v registru.

V registru 0x1C ACCEL_CONFIG bylo nutno nastavit třetí a čtvrtý bit na hodnotu logická 1. Jedná se o nastavení rozsahu akcelerometru. Stejným způsobem se nastavuje i registr 0x1D ACCEL_CONFIG2, jen se jedná o druhý a třetí bit a jejich nastavení na logickou jedničku volí maximální základní vzorkovací frekvenci.

Nakonec je pro jistotu vynulován registr 0x6B. Jde především o vypnutí sleep režimu a volbě vnitřního oscilátoru s frekvencí 20 MHz.

Multithreading

Práce s více jádry procesoru ESP32 lze zařídit například tak, že víme-li, že funkce `loop()` běží na jádře 1, pak jádro 0 procesoru inicializujeme tímto způsobem:

```
TaskHandle_t Core0task;

xTaskCreatePinnedToCore(Core0, "Core0task", 65535, NULL, 0, &Core0task, 0);

void Core0(void* pvParameters)
{
    WiFi.softAP("VibDiag", "heslo123");
    ServerConf();
    while(1) ...
}
```

Obr. 35 Deklarace proměnných a funkce jádra 0

Objekt typu `TaskHandle_t` deklarujeme jako globální proměnnou. Poté ve funkci `Setup()` zavoláme funkci `xTaskCreatePinnedToCore()`. Tato funkce slouží přímo k přiřazení úlohy pro specifikované jádro. Její argumenty jsou zleva: Obslužná funkce jádra, Název obsluhy jádra, velikost zásobníku v bytech, ukazatel na parametr obsluhy, priorita, ukazatel na funkci obsluhy a číslo jádra.

Samotná funkce `Core0()` je již samotná obslužná funkce obsahující kód pro jádro 0. Příkazy před smyčkou `while(1)` se zde vykonají jen jednou, ty co jsou v nekonečné smyčce `while(1)` pak běží stále dokola. Zacyklení ze zde plně žádoucí.

Webový server

Obsluha webového serveru je řešená v uživatelské funkci *ServerConf()*.

```
void ServerConf()
{
  SPIFFS.begin(true);
  Serial.println(WiFi.softAPIP());
  webSocket.begin();
  // webSocket.onEvent(webSocketEvent);

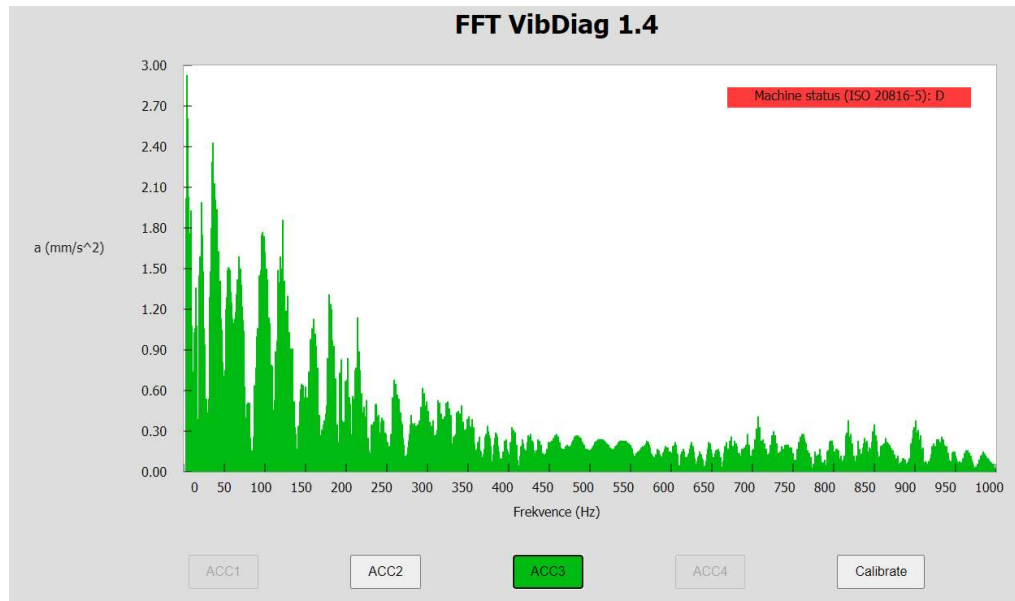
  server.on("/", HTTP_GET, [](AsyncWebServerRequest* request) {
    request->send(SPIFFS, "/index.html", "text/html");
  });
}
```

Obr. 36 Začátek funkce *ServerConf()*

Jak je vidět na obrázku, funkce nejprve zavolá příslušné metody pro inicializaci SPIFFS, což speciální funkce ESP32, která slouží k vytvoření souborového systému ve flash paměti mikroprocesoru. Díky tomu je možno pomocí ESP32 načítat složitější stránky webového serveru aniž by hrozilo přehlcení operační paměti mikroprocesoru. Dále je v této funkci inicializována Wi-Fi v režimu AP a inicializován webSocket server. Poté následují deklarace obslužných funkcí asynchronního webového serveru. První z nich (uvedená na obrázku) slouží například k načtení samotného rozhraní. V závěru funkce *ServerConf()* je pak již jen zavolána metoda *server.begin()*, která inicializuje asynchronní server. Kompletní kód dle nové koncepce je uveden v příloze 7.

3.6 Uživatelské rozhraní Webového Serveru

Vzhledem k nutnosti vytvořit interaktivní webové rozhraní a současné autorově nedostatečné znalosti jazyků HTML a JavaScript bylo rozhraní postupně vytvořeno především s pomocí umělé inteligence GPT-4 Copilot firmy Microsoft. Kompletní kód je uveden v příloze 8.



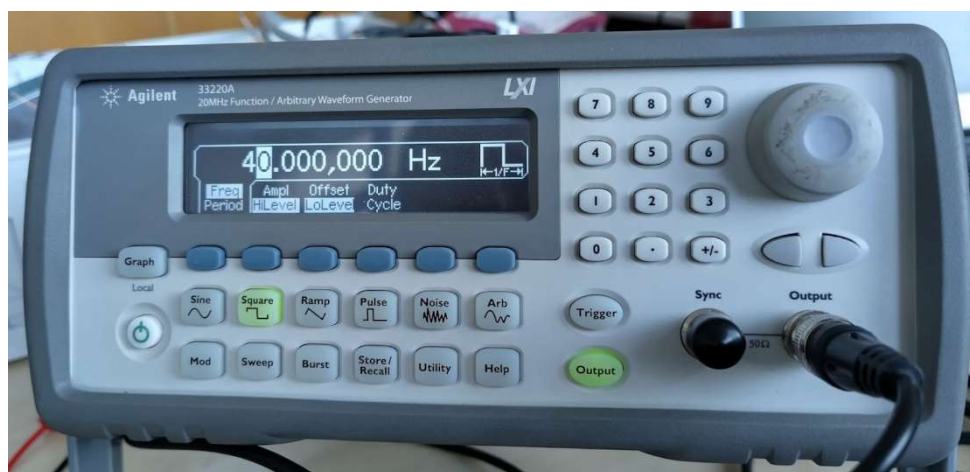
Obr. 37 Uživatelské rozhraní Webového serveru

Rozhraní obsahuje plátno (canvas), na němž jsou pomocí funkce *updateGraph* v JavaScriptu vykreslovány obdélníky které tak tvoří sloupcový graf zobrazující jednotlivé frekvenční složky spektra. Tato funkce je volána vždy příslušnou událostí WebSocketu, tedy kdykoliv jsou vypočtená data k dispozici. Stejně jako je tomu v případě předchozí verze, vždy je čten právě jen jeden akcelerometr. Součástí rozhraní jsou tedy samozřejmě tlačítka odkazující na jednotlivé akcelerometry. Tato tlačítka předávají reakci na kliknutí vždy jako callback funkci deklarovanou ve funkci *ServerConf()*. Jinak tomu není ani v případě tlačítka pro kalibraci. K němu je však ještě v JavaScriptu implementován příslušný kód, aby se jeho popisek měnil podle toho, v jaké poloze má být uživatelem umístěn kalibrovaný akcelerometr, čímž je uživateli poskytována základní zpětná vazba během procesu kalibrace. V principu tato stránka vychází z původního konceptu, ačkoliv některé prvky zde chybí. Konkrétně jde o přepínač zobrazení v časové a frekvenční oblasti a výběr stroje k vyhodnocení výsledků. Je zde pevně pouze vyhodnocení pro soustrojí s horizontálně uloženou Francisovou turbínou GE-DE na vibrace jeho ložiskové skříně.

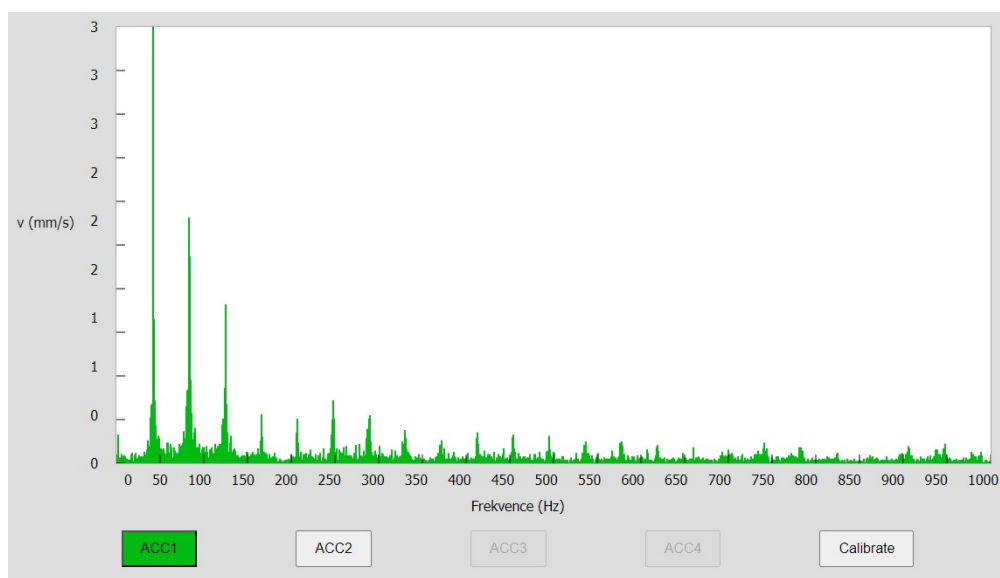
4 Testování

4.1 Druhé validační měření

Toto měření již využilo skutečného buzení akcelerometru za pomoci elektromagnetického aktuátoru. Měření bylo provedeno v poměrně rané fázi vývoje softwaru pro změněnou koncepci systému, kdy ještě nebylo zavedeno použití SPIFFS, jak je uvedeno výše. Z toho důvodu se stránka webového serveru chovala značně nestabilně, avšak i přesto měření prokázalo základní funkčnost systému. System v době tohoto měření ještě nedisponoval žádnou funkcí pro přímé ukládání dat. Avšak příkladný výsledek měření pro 40 Hz je zachycen na následujících obrázcích.



Obr. 38 Nastavení funkčního generátoru pro buzení aktuátoru při druhém validačním měření



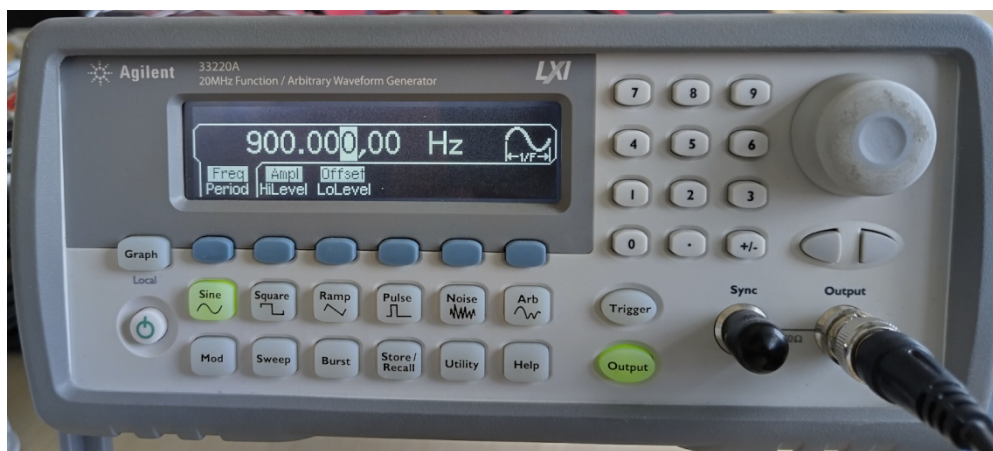
Obr. 39 Výsledky frekvenční analýzy druhého měření

4.2 Třetí validační měření

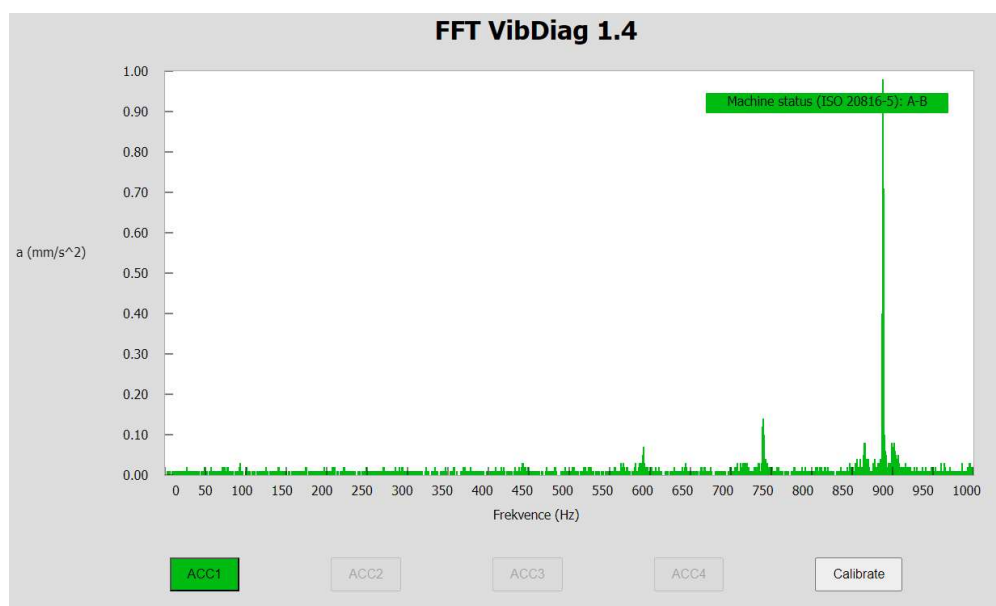
Na začátku třetího měření se ukázalo, že dochází k zrcadlení spektra okolo 500Hz. Dle teorie tohoto jevu, který se nazývá Nyquistův limit bylo zřejmé, že skutečná vzorkovací frekvence není požadované 2 kHz, ale pouze 1 kHz. Výpisem čtených hodnot bylo ověřeno, že k chybě dochází ve snímači MPU-9250. To bylo opraveno změnou zápisu do registrů snímače tak, jak je uvedeno v kapitole 3.5. Po tomto zásahu již systém fungoval dle očekávání až do frekvence 1 kHz.



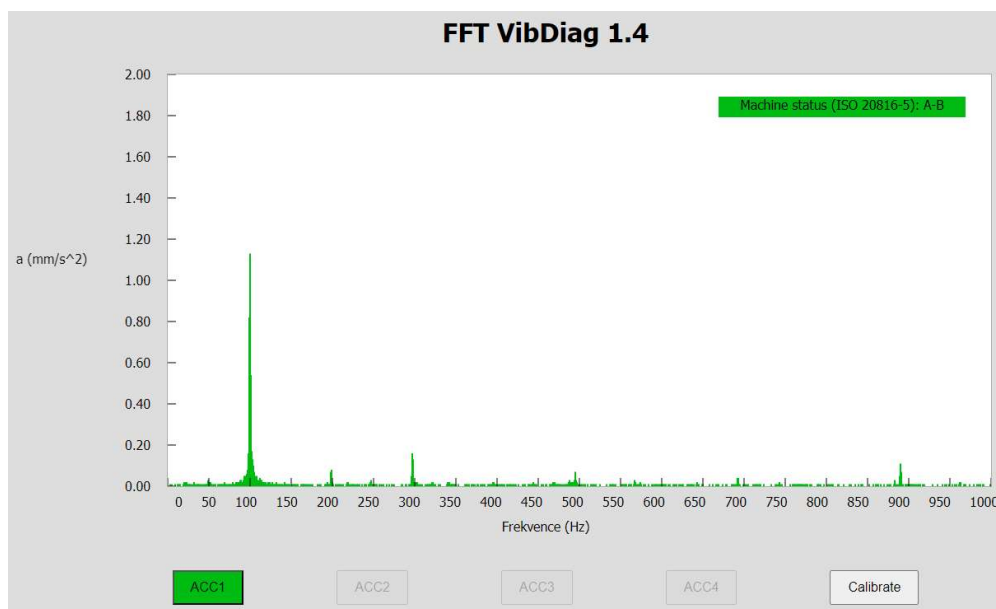
Obr. 40 Umístění akcelerometru na aktuátor



Obr. 41 Nastavení funkčního generátoru pro vybuzení aktuátoru na 900 Hz



Obr. 42 Výsledek měření se snímačem vybuzeným na 900 Hz



Obr. 43 Výsledek měření se snímačem vybuzeným na 100 Hz

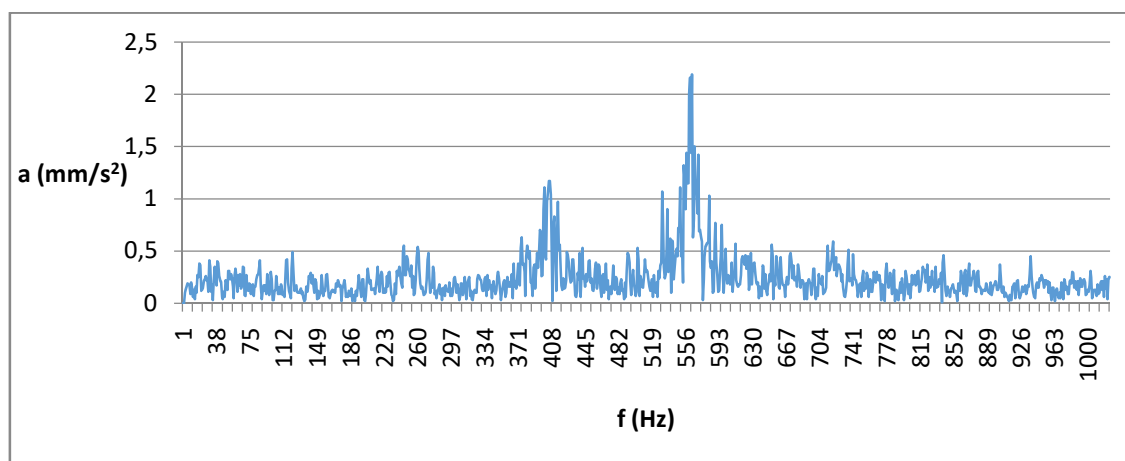
Z výsledků je patrné, že se zvyšující se frekvencí dochází k nepřesnostem v měření. To je dáno ne zcela přesně nastaveným časovým parametrem v programu, při jehož dosažení dochází k přečtení dat z akcelerometru. Tento parametr lze upravit a dosáhnout tak optimální vzorkovací frekvence.

4.3 Čtvrté validační měření

Toto měření proběhlo již s použitím vibrační stolice, na kterou byl kromě MEMS snímače připevněn i referenční snímač výrobce Brüel & Kjaer typ 4507 B 004 sloužící k ověření správnosti naměřených dat.

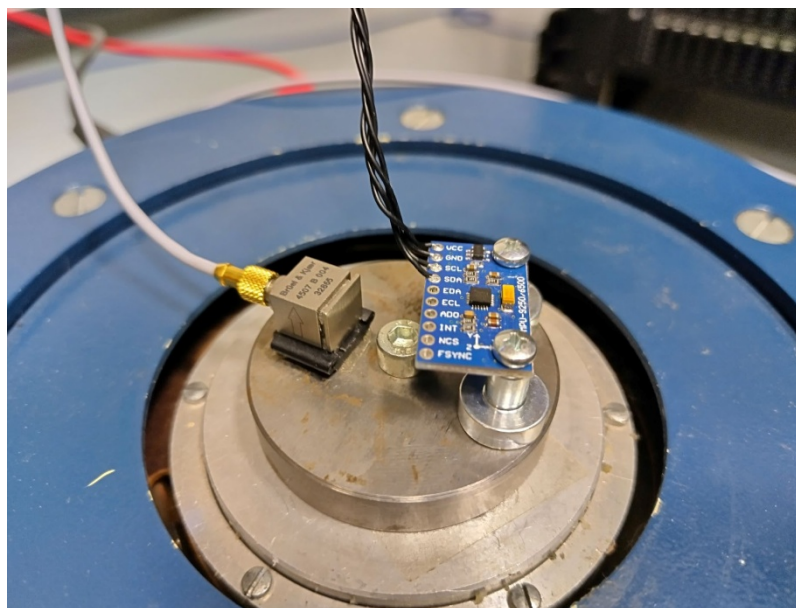
Měření zahrnovalo vybuzení vibrační stolice na konkrétní frekvence čistým sinusovým signálem a také vybuzení bílým šumem.

Výsledkem tohoto měření bylo, že systém na nízkých frekvencích, řádově desítky Hz pracuje správně a hlavně poměrně přesně. Relativní chyby v amplitudě se zde dostaly na minima 3,7 % při frekvenci 10 Hz a 3,97 % při frekvenci 60 Hz. Na vyšších frekvencích, přibližně do 800 Hz se však začaly projevovat jisté nepřesnosti vznikající s vysokou pravděpodobností vlivem vzorkování, neboť symetricky v okolí vyšetřované frekvence se objevovaly zrcadlené obrazy. Tento jev se již projevil dříve při druhém a třetím validačním měření. Protože však u třetího měření nebyl k dispozici referenční senzor, nebylo možno vyloučit, že by se mohlo jednat o mechanické problémy samotného aktuátoru. Přesto se zde relativní chyba měření pohybovala pod 19 %. Na frekvenci 1 kHz už však byla chyba bohužel značná a to konkrétně 42,04 %. Bohužel tyto chyby v amplitudě se již nepodařilo odstranit. Dále byl proveden pokus s upevněním snímače za pomoci pouze jedné magnetické čochky namísto dvou. Tím bylo ověřeno, že tento způsob uchycení rozhodně není vhodný, protože se okamžitě začnou projevovat nepříznivé vlivy pramenící z příliš nízké tuhosti. Tyto nedostatky se projevují zkreslením spektra viz obrázek 44.



Obr. 44 Spektrum naměřeného signálu při uvolnění jednoho montážního bodu

Tabulky a grafy ostatních naměřených výsledků jsou k dispozici v přílohách 9 až 11.



Obr. 45 Umístění akcelerometru MPU9250 (vpravo) a referenčního snímače (vlevo) na vibrační stolici

4.4 Testování funkční integrity

Testování funkční integrity systému proběhlo s využitím dvou nezávislých akcelerometrů umístěných na soustrojí o třech strojích. Toto soustrojí zahrnovalo dva prototypy synchronních strojů s permanentními magnety, kdy jeden byl navinut jako standardní třífázový a druhý byl navinut jako devítifázový. Právě na těchto dvou strojích byly umístěny akcelerometry viz obrázky níže. Posledním strojem v tomto soustrojí byl asynchronní motor, který byl umístěn na hřídeli uprostřed mezi oběma synchronními motory a sloužil při tomto experimentu jako pohon celého soustrojí. Právě tento asynchronní stroj v soustrojí byl napájen měničem, aby bylo možné zajistit jeho kompletní řízení a jednotlivé PMSM byly zatěžovány reostaty a byly tak provozovány v generátorickém režimu.



Obr. 46 Umístění prvního akcelerometru na 9f. PMSM.



Obr. 47 Umístění druhého akcelerometru na 3f. PMSM.



Obr. 48 Soustrojí s nasazenou vibrodiagnostikou

První část testování byla zaměřena na postupné odpojování jednotlivých fází devítifázového motoru, což vede k prudkým změnám momentu během jedné každé otáčky stroje a tudíž k vibracím. Výsledky této části jsou uvedeny v příloze 12. Druhá část byla obdobná první části s tím rozdílem, že byly odpojovány jednotlivé fáze třífázového motoru. Výsledky této části jsou uvedeny v příloze 13. Obě části zároveň obsahují měření stroje naprázdno, tj. kdy na svou hřídel nepůsobí stroj samotný žádným momentem.

Testování prokázalo slibnou použitelnost systému v praxi. Zároveň ale poukázalo i na výrazné nedostatky, které se během laboratorních měření neprokázaly. Jako problematická se zde ukázala zejména komunikace po sběrnici I²C. Například pokud vodiče k sensorům nebyly “smotány”, docházelo na nich k takovému rušení, že buď jen nebylo možné číst data ze sensorů, nebo se senzory nebylo ani možné navázat komunikaci. Vývoj zařízení by tedy logicky pokračoval příslušnými úpravami zejména vedení k akcelerometrům a také DPS a případně i krabičky. Tyto úpravy by dále bylo vhodné ověřit příslušnými testovacími metodami dle IEC 61000-6-4 a IEC 61000-6-2.

Zhodnocení a závěr

V teoretické části práce byly stručně probrány skutečnosti a jejich souvislosti důležité pro porozumění problematice, jejímž řešením se dále zabývá praktická část.

První iterace praktické části ukázala značné možnosti využití čidel MPU-6050 v případech, kdy není nutné měřit frekvence nad 500 Hz. Dle normy ISO 20816 je však nutné vyhodnocení alespoň do 1 kHz pro nízkofrekvenční přístup. Z toho důvodu bylo nutné nahrazení MPU-6050 za typ MPU-9250, který umožňuje dosažení vyšších vzorkovacích frekvencí. Dále však tato iterace poukázala zejména na nedostatky týkající se rychlosti protokolu pro bezdrátový přenos ESP-NOW. Tento protokol je evidentně schopen realizovat přenos dat maximálně přibližně 200x za vteřinu a proto není dostačující odesílání vždy po načtení jednoho každého vzorku. V rámci vývoje proběhly snahy toto řešit organizací vždy více dat do jednoho packetu, což se však velmi rychle bohužel ukázalo jako nevhodné kvůli nedostatečnému protichybovému zabezpečení tohoto protokolu a vzhledem k maximální velikosti packetu neexistovala možnost, jak přenést celé načtené datové okno naráz.

Z těchto důvodů bylo rozhodnuto nahradit protokol ESP-NOW běžnou WiFi standardu IEEE 802.11. Tato změna vyřešila výše uvedené problémy a přinesla zjednodušení hardwaru, kdy již není nutné mít k PC připojený přijímač a místo něho postačí samostatná síťová karta počítače podporující bezdrátovou síť. Avšak toto řešení přineslo svojí problematiku, týkající se zvýšených nároků na vysílací ESP32. Konkrétně jde o to, že ESP32 musí nejen číst data v reálném čase, ale musí také fungovat jako webový server. K takové úloze je však tento procesor poměrně dobře připraven, a proto se tyto změny podařilo úspěšně realizovat.

Po provedení výše uvedených změn byl již kompletní systém testován. Nejprve na jednoduchém aktuátoru, poté na vibrační stoličce a nakonec byl systém otestován v laboratoři na soustrojí s prototypem devítifázového motoru s permanentními magnety.

Během těchto testů byly postupně odhaleny a následně odstraněny některé chyby systému (viz kapitoly 3.3 až 4.2). Ve čtvrtém validačním měření byla ověřena přesnost měření v porovnání s profesionálním vibrodiagnostickým systémem. Toto měření prokázalo, že systém navzdory své jednoduchosti a nízké ceně dokáže zejména na nízkých frekvencích (přibližně do 100 Hz) měřit s poměrně vysokou přesností. Na vyšších frekvencích bohužel není již přesnost v amplitudě tak vysoká. Příčinu této chyby se bohužel v současné době v rámci této práce nepodařilo jednoznačně identifikovat ani

odstranit. Naproti tomu přesnost vyhodnocované frekvence si systém zachovává bez problému až do 1 kHz.

V závěrečném testu se projevíly závažné nedostatky související evidentně s elektromagnetickou kompatibilitou vedení k akcelerometrům a to i navzdory tomu, že již ve fázi návrhu byla přijata jistá opatření proti této problematice (viz kapitola 3.1.1). Tato opatření se však ukázala jako nedostatečná, nicméně v případě dalšího vývoje by bylo dalším krokem opatření vedení stíněním a testování na odolnost systému proti elektromagnetickému rušení. Je však nutno podotknout, že problém s EMC se projevil v blízkosti nestíněného kabelu frekvenčního měniče, který je významným zdrojem EMC rušení.

Pokud by se podařilo vyřešit i tyto problémy, pravděpodobně by byl tento systém již poměrně dobře použitelný v praxi a to i navzdory své velmi nízké ceně. Neuvažujeme-li náklady na vývoj, který zabral přibližně jeden a půl roku (nesoustavné práce), náklady na pořízení materiálu k sestavení tohoto systému činí pouze nižší jednotky tisíc Kč, což při porovnání s cenou profesionálních vibrodiagnostických systémů, jejichž cena bývá řádově desítky až stovky tisíc Kč, je řádově nižší částka. Jedná se tedy o skutečně levnou alternativu pro krátkodobý a střednědobý onsite monitoring a diagnostiku. Systém sice nemá garantovanou provozní spolehlivost, avšak vzhledem ke své pořizovací ceně může mít v porovnání s profesionální vibrodiagnostikou značnou výhodu v podobě investičních nákladů a mohl by být nasazován například jako doplňkový systém pro monitoring nebo na levnější aktiva.

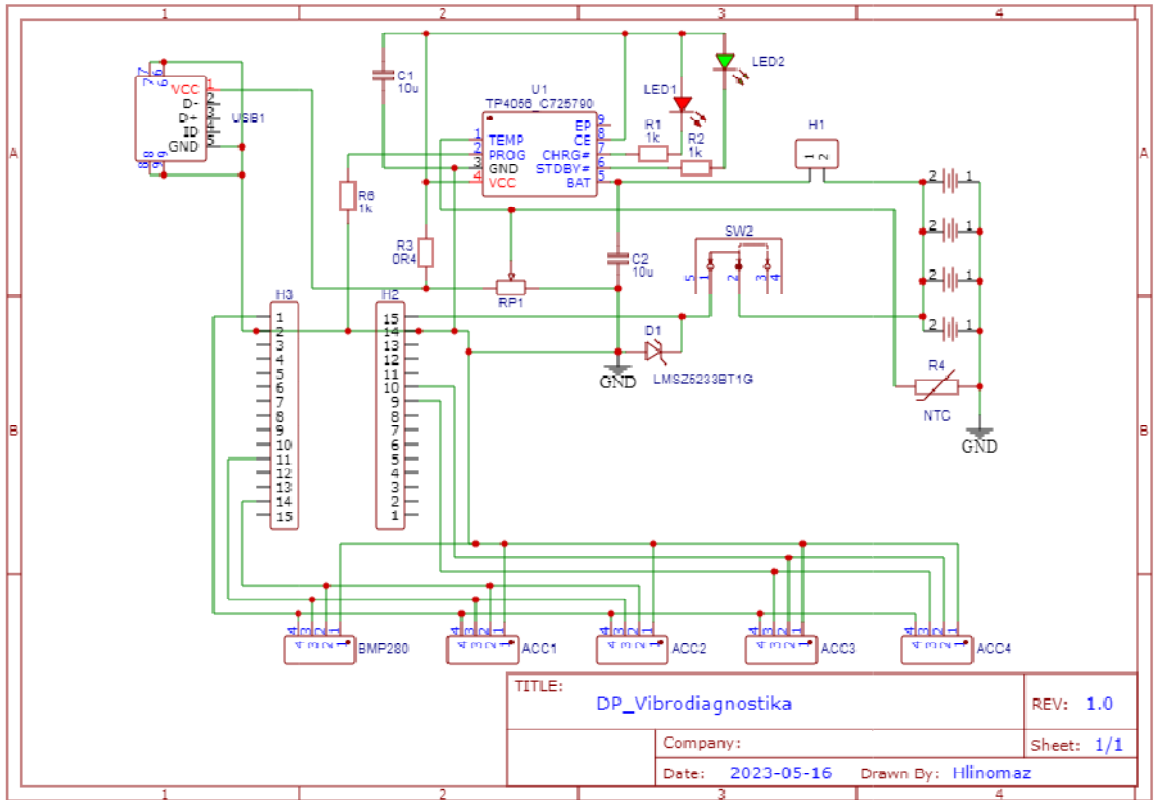
Literatura

- [1] ZUTH, Daniel; VDOLEČEK, František. Měření vibrací ve vibrodiagnostice. Online. Automa, 1/2010. Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/40375.pdf>. [citováno 2024-01-24].
- [2] VOJÁČEK, Antonín. Principy akcelerometrů - 1. Díl - Piezoelektrické. Online. In: automatizace.hw, 14.1.2007. Dostupné z: <https://automatizace.hw.cz/clanek/2007011401>. [citováno 2024-01-24].
- [3] VOJÁČEK, Antonín. Jak pracují nové 3D MEMS akcelerometry Freescale?. Online. In: vyvoj.hw.cz, 30.4.2007. Dostupné z: <https://vyvoj.hw.cz/soucastky/jak-pracuji-nove-3d-mems-akcelerometry-freescale.html>. [citováno 2023-01-24].
- [4] HUSÁK, Miroslav. Užití MEMS v průmyslu. Online. Automa, 12/2008. Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/38277.pdf>. [citováno 2024-01-24].
- [5] KUBÍK, Zdeněk. Interferenční zdroje a jejich charakteristika. Přednáška. Plzeň: Západočeská univerzita, 2. 11. 2023. [citováno 2024-02-06].
- [6] MathWorks (bez data). MATLAB App Designer. Online. Dostupné z: <https://www.mathworks.com/products/matlab/app-designer.html>. [citováno 2024-02-05].
- [7] Espressif Systems Co., Ltd. ESP-NOW, https://docs.espressif.com/projects/espidf/en/latest/esp32/api-reference/network/esp_now.html. [citováno 2024-02-06].
- [8] TIŠNOVSKÝ, Pavel. Komunikace po sériové sběrnici I²C. Online. root.cz, 2009. Dostupné z: <https://www.root.cz/clanky/komunikace-po-seriove-sbornici-isup2supc/>. [citováno 2024-02-06].
- [9] MENTLÍK, Václav; PIHERA, Josef; POLANSKÝ, Radek; PROSR, Pavel; TRNKA, Pavel. Diagnostika elektrických zařízení, 1. vydání, Praha: Technická literatura BEN, 2008. ISBN 978-80-7300-232-9. [citováno 2024-02-26].
- [10] STONE, Greg C.; CULBERT, Ian; BOULTER, Edward A.; DHIRANI, Hussein. Electrical Insulation for Rotating Machines, 2. vydání, Piscataway, New York: IEEE Press, 2014. ISBN 978-1-118-05706-3. [citováno 2024-02-26].
- [11] KREIDL, M. – ŠMÍD, R.: Technická diagnostika. BEN – technická literatura, Praha, 2006, 408 s., ISBN 80-7300-158-6.

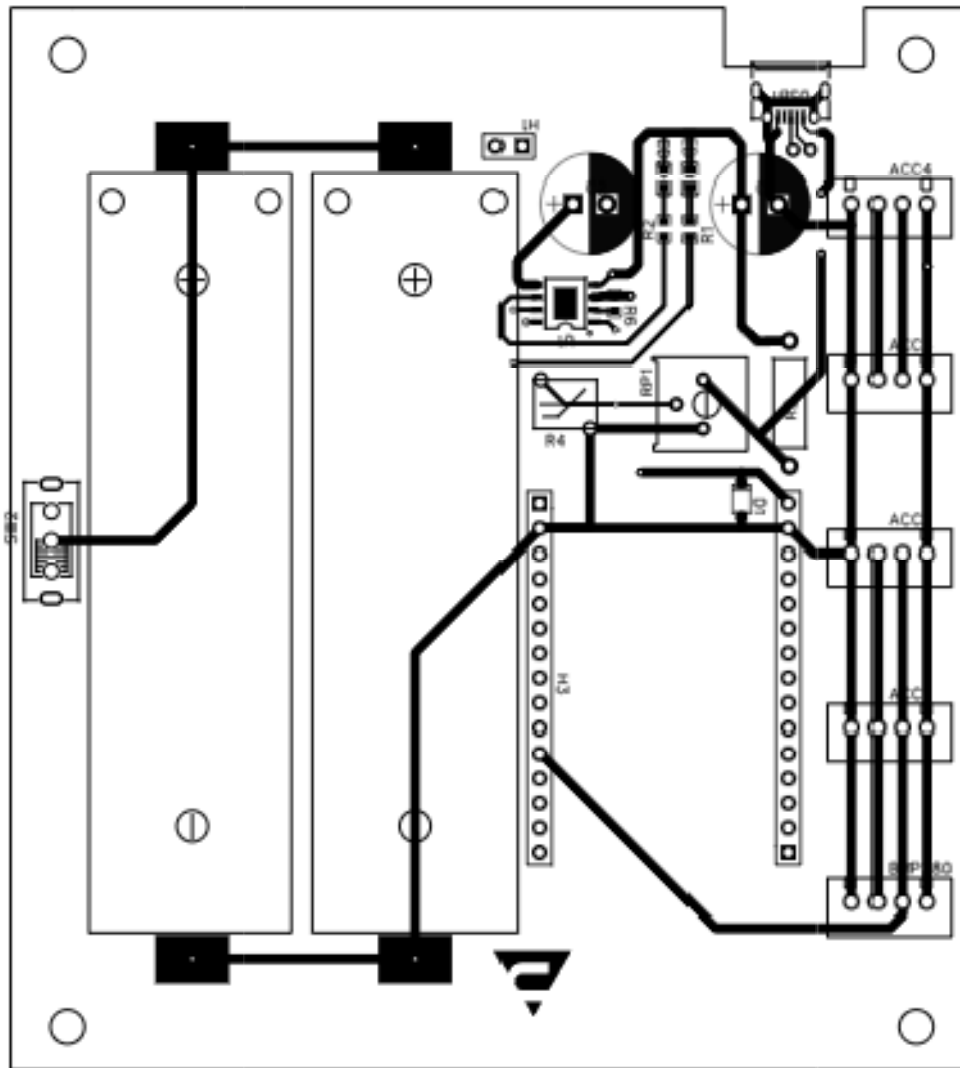
-
- [12] Condition monitoring. B&R Industrial Automation GmbH, 2021. Uživatelský manuál a datasheet.
- [13] LEFFLER, Jan; TRNKA, Pavel. Failures of Electrical Machines - Review. Online. IEEE. 2022. 8th International Youth Conference on Energy (IYCE), Hungary. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/9857519/authors#authors>
- [14] GONÇALVES, Aparecido Carlos; LAGO, Daniel Fabiano; ALBUQUERQUE, Maria da Consolação Fonseca de. Maintenance of Reducers with an Unbalanced Load Through Vibration and Oil Analysis Predictive Techniques. Online. IntechOpen. 2011. Dostupné z: <https://www.intechopen.com/chapters/18982>
- [15] MENTLÍK, Václav. Spolehlivostní aspekty elektrotechnologie. Praha: BEN – technická literatura, 2011. ISBN 978-80-7300-412-5.

Přílohy

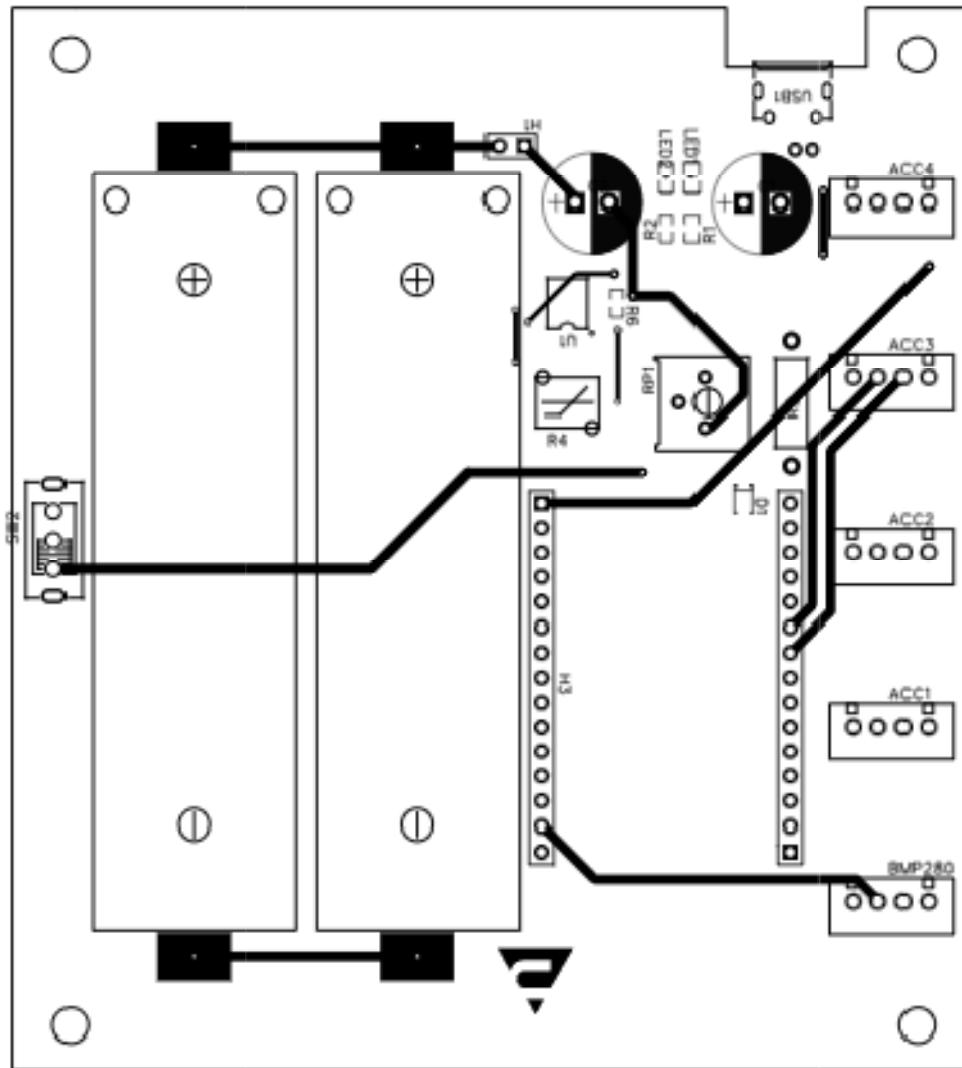
Příloha 1: Schéma DPS



Příloha 2: Výkres horní vrstvy DPS včetně potisku



Příloha 3: Výkres spodní vrstvy DPS včetně potisku



Příloha 4: Zdrojový kód pro vysílač dle první koncepce (Vysilac_0.5.ino)

Příloha 5: Zdrojový kód pro přijímač dle první koncepce (Prijimac_0.5.ino)

Příloha 6: Zdrojový kód pro MATLAB aplikaci (VibDiag1_05.mlapp)

Příloha 7: Zdrojový kód pro ESP32 ve finální verzi (main.cpp)

Příloha 8: Zdrojový kód stánky webového serveru (index.html)

Příloha 9: Výsledky měření na vibrační stolici sinus.xlsx

Příloha 10: Grafy měření na vibrační stolici sinus.xlsx

Příloha 11: Měření na vibrační stolici bílý šum.xlsx

Příloha 12: Vyhodnocení validačního měření 9f.xlsx

Příloha 13: Vyhodnocení validačního měření 3f.xlsx

Pozn.: Přílohy 4 až 13 jsou dostupné na webu portal.zcu.cz a dále k 24. 5. 2024 na odkazu:

<https://drive.google.com/drive/folders/1Q5kElat9M7cbZzUIO2Ps5GO7Azlg6vv1?usp=sharing>