



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

VIZUALIZACE VÝSTUPU OPTICKÉHO PŘEVODNÍKU

VISUALISATION OF THE OPTICAL CONVERTER OUTPUT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Vít Charvát

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Čala, Ph.D.

BRNO 2025

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Vít Charvát

ID: 240354

Ročník: 3

Akademický rok: 2024/25

NÁZEV TÉMATU:

Vizualizace výstupu optického převodníku

POKYNY PRO VYPRACOVÁNÍ:

V EMC laboratoři firmy Forvia (HELLA) se testují světlomety a zadní svítliny, kdy test se vyhodnocuje pomocí optických vláken a optického převodníku (převod světlo/napětí). Cílem bakalářské práce je vytvoření aplikací na PC a přenosném zařízení, které budou operátorovi ulehčovat fixování optických vláken důležité pro správné nastavení testu. Zadání lze shrnout do následujících bodů:

1. Popište současné vybavení EMC laboratoří ve firmě Forvia.
2. Proveďte rešerši HW pro bezdrátový přenos dat mezi PC a přenosným zařízením (tablet, mobilní telefon), zadavatel preferuje rozhraní Bluetooth (není podmínkou).
3. Navrhněte způsob přenosu dat mezi optickým převodníkem a zbylými částmi systému.
4. V prostředí LabVIEW implementujte náhradu za současnou ovládací aplikaci EMC32, která vyčítá hodnoty napětí z optického převodníku. Náhrada bude vizualizovat měřená data a dále umožní jejich vizualizaci i na přenosném zařízení.
5. Implementujte aplikaci na přenosném zařízení s operačním systémem Android. Snažte se o uživatelskou přívětivost.
6. Vytvořené řešení zdokumentujte, popište jeho silná a slabá místa a případné návrhy pro vylepšení.

Zadání je vypsáno firmou Forvia.

DOPORUČENÁ LITERATURA:

[1] BLUME, Peter A. The LabVIEW Style Book. Crawfordsville (Indiana): Pearson Education, 2007. ISBN 978-0131458352.

Termín zadání: 10.2.2025

Termín odevzdání: 28.5.2025

Vedoucí práce: Ing. Martin Čala, Ph.D.

Ing. Miroslav Jirgl, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

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

Abstrakt

Tato bakalářská práce se zabývá návrhem a realizací systému pro bezdrátový přenos dat z EMC komory do mobilního zařízení. Cílem je zjednodušení práce operátora během měření a zkrácení času potřebného k nastavení optických vláken. V rámci práce bylo analyzováno stávající vybavení EMC laboratoře a provedena rešerše dostupných technologií pro bezdrátovou komunikaci. Na základě testování byla zvolena technologie Bluetooth.

Byla navržena struktura přenosu dat mezi optickým převodníkem, aplikací vytvořenou v prostředí LabVIEW, mikrokontrolérem Arduino a mobilní Android aplikací. Systém byl navržen modulárně, s důrazem na přehledné uživatelské rozhraní, stabilitu přenosu a možnost dalšího rozšíření. V závěru jsou popsány silné i slabé stránky řešení a navržené možnosti vylepšení.

Klíčová slova

aplikace, vizualizace, bezdrátová technologie, Bluetooth, Android, LabVIEW

Abstract

This bachelor thesis deals with the design and implementation of a system for wireless data transfer from an EMC chamber to a mobile device. The aim is to simplify the operator's work during measurements and reduce the time required to position optical fibers. The current equipment of the EMC laboratory was analyzed, and a review of available wireless communication technologies was conducted. Based on practical testing, Bluetooth technology was selected.

A data transfer structure was designed between the optical converter, an application developed in LabVIEW, a microcontroller Arduino, and a mobile Android application. The system was designed modularly, with emphasis on a clear user interface, stable data transmission, and future scalability. The conclusion discusses the strengths and limitations of the solution and suggests possibilities for further improvement.

Keywords

application, visualization, wireless technology, Bluetooth, Android, LabVIEW

Bibliografická citace

CHARVÁT, Vít. *Vizualizace výstupu optického převodníku*. Online, bakalářská práce. Martin ČALA (vedoucí práce). Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2025. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/168213>. [cit. 2025-05-20].

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	<i>Vít Charvát</i>
VUT ID studenta:	240354
Typ práce:	<i>Bakalářská práce</i>
Akademický rok:	2024/25
Téma závěrečné práce:	<i>Vizualizace výstupu optického převodníku</i>

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne: 28.5.2025

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Martinu Čalovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: 28.5.2025

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. EMC KOMORA V AUTOMOTIVE.....	12
2. POPIS AKTUÁLNÍHO STAVU PRACOVIŠTĚ	13
2.1 ROZMĚRY MÍSTNOSTI	13
2.2 OMEZENÍ	14
2.3 POŽADAVKY NA REALIZACI.....	14
2.4 POPIS OPTICKÉHO PŘEVODNÍKU.....	15
2.5 NI MĚŘÍCÍ KARTA	15
3. SROVNÁNÍ BEZDRÁTOVÝCH TECHNOLOGIÍ	16
3.1 WI-FI.....	16
3.2 BLUETOOTH.....	16
4. VÝBĚR ŘEŠENÍ BLUETOOTH SPOJENÍ.....	18
4.1 POUŽITÝ SOFTWARE	18
4.2 MOŽNOSTI	18
4.2.1 BT dongle (USB pc)	19
4.2.2 BT modul pro Arduino	19
4.3 TESTOVÁNÍ.....	19
5. NÁVRH A IMPLEMENTACE APLIKACÍ.....	25
5.1 ZPŮSOB PŘENOSU DAT MEZI OPTICKÝM PŘEVODNÍKEM A ZBYLÝMI ČÁSTMI SYSTÉMU	25
5.2 POŽADAVKY NA APLIKACE	26
5.3 LABVIEW	27
5.3.1 Návrh aplikace.....	27
5.3.2 Uživatelské prostředí	28
5.3.3 Popis aplikace.....	29
5.4 ARDUINO	30
5.5 ANDROID	33
5.5.1 Návrh uživatelského rozhraní	33
5.5.2 Implementace uživatelského prostředí	34
5.5.3 Implementace hlavní logiky aplikace.....	35
5.5.4 Realizace Bluetooth spojení a čtení dat	37
5.5.5 Správa chyb.....	38
5.5.6 Testování aplikace	39
5.5.7 Doplnkové funkce.....	42
5.6 ODEZVA SYSTÉMU	43
6. POPIS VÝSLEDNÉHO PRODUKTU	44
6.1 FUNKČNOST APLIKACÍ.....	44
6.2 UŽIVATELSKÉ OVLÁDÁNÍ	45

6.3	SHRNUTÍ.....	46
7.	ZÁVĚR.....	47
	LITERATURA.....	48
	SEZNAM SYMBOLŮ A ZKRATEK	52
	PŘÍLOHA A – KRABÍČKA PRO ARDUINO.....	53
	PŘÍLOHA B – NÁVOD K ANDROID APLIKACI.....	55

SEZNAM OBRÁZKŮ

Obrázek 1 EMC komora [3]	12
Obrázek 2 Rozložení pracoviště	13
Obrázek 3 Blokové schéma předpokládaného systému	14
Obrázek 4 Blokové schéma optického převodníku.....	15
Obrázek 5 Možnosti použití bezdrátových technologií [5].....	16
Obrázek 6 Vývojový diagram testovací aplikace pro bezdrátové zařízení	21
Obrázek 7 Vývojový diagram testovací aplikace LabVIEW	22
Obrázek 8 Zapojení pro testování modulů s Arduinem	23
Obrázek 9 Zapojení Arduina s BT Modulem HC-05.....	23
Obrázek 10 Zapojení pro testování donglů	24
Obrázek 11 Blokové schéma celého systému s procházejícími daty a fyzickým spojením.....	25
Obrázek 12 Uživatelské prostředí LabVIEW aplikace	28
Obrázek 13 Stavový diagram LabVIEW aplikace	29
Obrázek 14 Schéma zapojení Arduina a modulu HC-05	31
Obrázek 15 Vývojový diagram Arduino aplikace	32
Obrázek 16 Návrh designu a propojení oken aplikace.....	34
Obrázek 17 Stavový diagram aplikace na Android.....	37
Obrázek 18 Umístění krabičky	46
Obrázek 19 3D model krabičky – pohled z levého horního rohu	53
Obrázek 20 Krabička	54
Obrázek 21 Snímek obrazovky - domovská stránka aplikace.....	55
Obrázek 22 Snímek obrazovky - seznam spárovaných zařízení	56
Obrázek 23 Snímek obrazovky – spuštěno čtení dat	56
Obrázek 24 Snímek obrazovky – zobrazení dat a hlavní menu	57
Obrázek 25 Snímek obrazovky – menu pro změnu jazyka nebo motivu	58

SEZNAM TABULEK

Tabulka 1 Použitý software pro testování Bluetooth modulů	18
Tabulka 2 Srovnání testovaných modulů/adaptérů	20
Tabulka 3 Požadavky na aplikace	26
Tabulka 4 Použitý software pro vývoj aplikací	26
Tabulka 5 Měření odezvy systému	43

ÚVOD

V EMC laboratoři firmy Forvia jsou testovány přední světlomety a zadní svítilny. Vzniká zde problém, kdy je přístup do komory vzdálený od operátora, který tak musí opětovně chodit mezi komoru a svým pracovištěm.

Úkolem této práce je urychlit a usnadnit operátorovi proces testování vyvinutím systému, který vizualizuje měřená data na bezdrátové zařízení. Tato data jsou klíčová pro správnou fixaci optických vláken, která jsou umístěna na reflektoru. Výsledek této práce bude řešit nadbytečné cesty do komory, a tím urychlí testování.

Aplikace pro optický převodník byla naprogramována pomocí prostředí LabVIEW. V tomto bodu byl navržen i způsob přenosu dat celým systémem. Při výběru technologie pro bezdrátový přenos dat bylo po testování vybráno Arduino s přídatným modulem, který umožňuje použití Bluetooth. Pomocí Arduina a Bluetooth modulu byla data posílána na bezdrátové zařízení, kde byla vyvinuta aplikace pro zobrazení hodnot.

V první kapitole je popsán aktuální stav pracoviště spolu se základními informacemi o již stávajících zařízeních, na které bude práce navázána. Dále je provedena rešerše bezdrátového spojení, kde jsou srovnány technologie, které dávají smysl, a na základě okolních podmínek je zvoleno finální řešení. Poté je popsáno testování při výběru samotné komponenty, díky které byla data bezdrátově posílána.

V šesté kapitole je popsán celý proces návrhu a implementace aplikací, které tvoří jádro celého systému. Dále je zde popsán způsob přenosu dat mezi všemi částmi systému s popisem, jaká data procházejí soustavou ve všech bodech. Nejprve je představena aplikace v prostředí LabVIEW, která zajišťuje sběr dat z měřicích karet, jejich vizualizaci a odesílání k dalšímu zpracování. Dále je popsán vývoj firmwaru pro mikrokontrolér Arduino, který slouží jako komunikační mezičlánek s Bluetooth rozhraním. Následně je rozebrána tvorba mobilní aplikace pro operační systém Android, která umožňuje uživateli bezdrátově přijímat a zobrazovat data v reálném čase.

Sedmá kapitola shrnuje výslednou funkcionalitu systému a jeho přínosy pro uživatele v praxi. Závěrečná osmá kapitola pak hodnotí dosažené výsledky, zhodnocuje naplnění cílů práce a navrhuje možnosti dalšího rozšíření systému.

1. EMC KOMORA V AUTOMOTIVE

V této kapitole je uvedeno, při kolika různých testech bude tato práce využívána a v jakém prostředí zlepšit podmínky pro operátory.

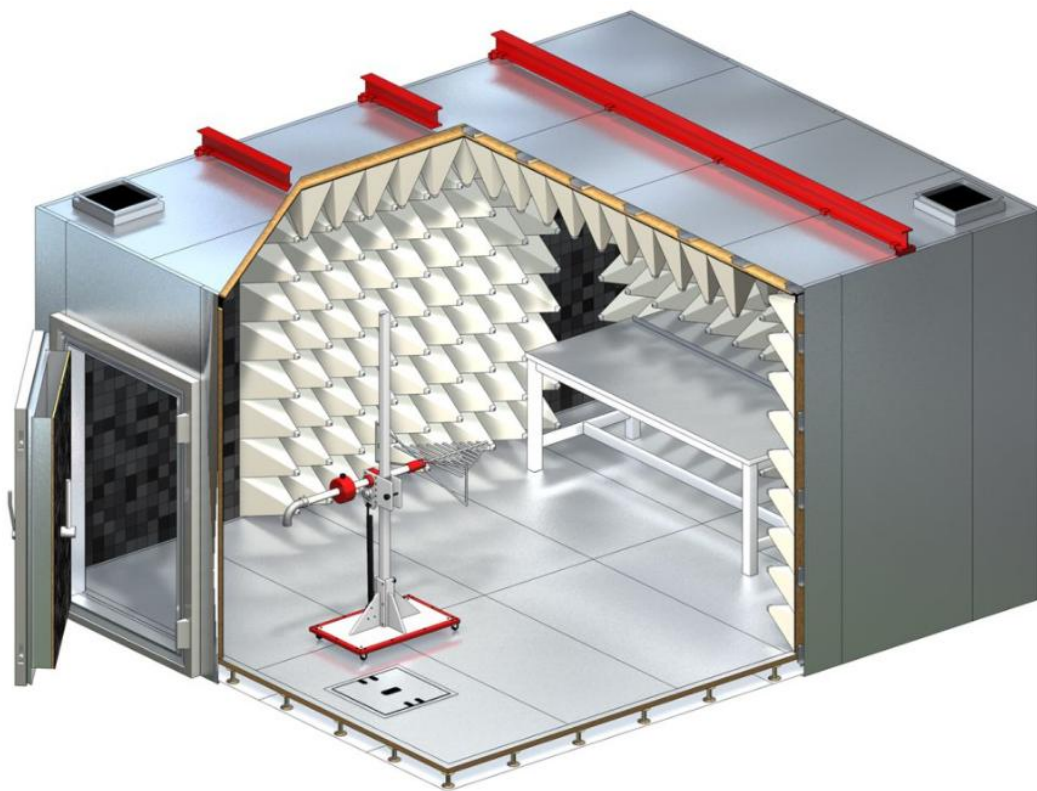
Zkratka EMC znamená v překladu elektromagnetická kompatibilita. Lze ji rozdělit na dvě hlavní složky, a to vyzařování rušení a odolnost vůči rušení. Jde o důležitý technický údaj, zda zařízení dokáže správně fungovat v určité soustavě. [1]

Komora je postavena tak, aby absorbovala vlny. Stěny jsou proto obloženy materiály, které vlnění pohlcují, a zároveň nedovolí, aby dovnitř pronikaly jakékoli vlny. Tím dosahuje velkého odstínění od okolí. [2]

Existuje několik typů komor pro různé druhy testování. Nejběžnější druhy jsou testování vyzařovaných emisí, testování vyzařované odolnosti, testování bezdrátového vysílače (RF), testování antén a testování specifické míry absorpce (SAR). [2]

EMC testy na komponentní úrovni:

- EMI – vyzařování emisí: Conducted emissions, Radiated emissions, Magnetic emissions, Interference emissions measurement. [2]
- EMS – odolnost vůči rušení: Radiated immunity, Bulk current injection, Transient pulses, Mobile transmitter test, Magnetic immunity, Stripline, Electrostatic discharge. [2]



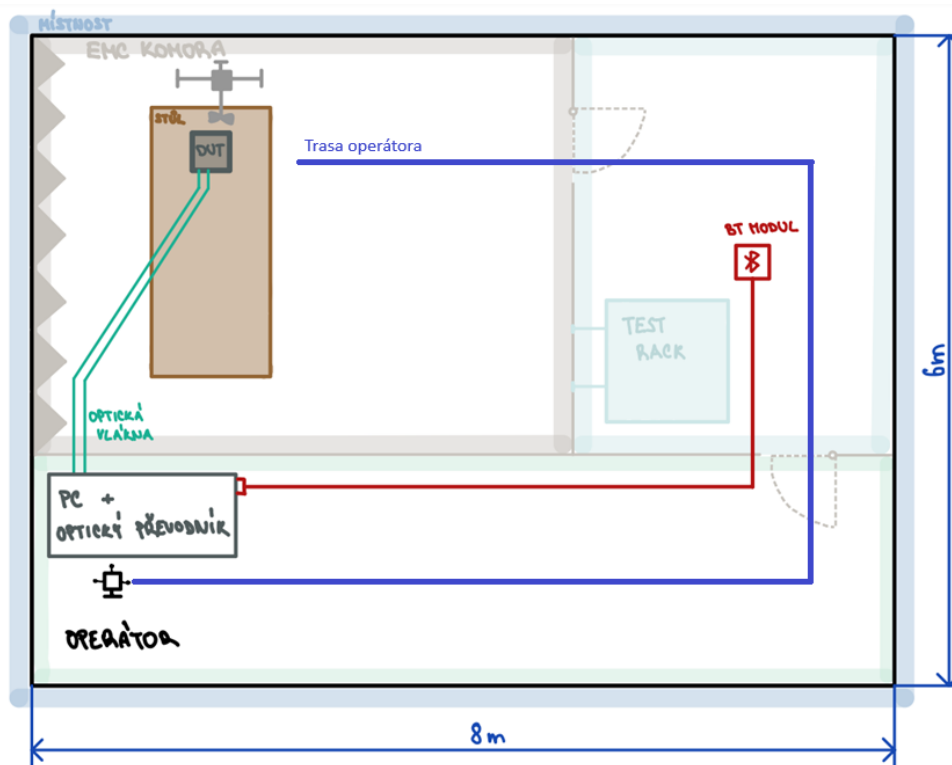
Obrázek 1 EMC komora [3]

2. POPIS AKTUÁLNÍHO STAVU PRACOVIŠTĚ

Stav pracoviště v době zahájení této práce je v této kapitole rozdělen do několika základních bodů.

2.1 Rozměry místnosti

Rozměry místnosti, ve které se kompletní pracoviště nachází, jsou cca 8 m na 6 m se stropem ve výšce 3,5 m (viz obrázek 2). Operátor má proto trasu do komory o délce zhruba 10 metrů. Po každé instalaci optických vláken musí jít zpět ke svému počítači a zkontrolovat referenční hodnoty napětí (po převodu světla na napětí – tuto konverzi zajišťuje optický převodník). Po dokončení své práce bude moci operátor na bezdrátovém zařízení jasně vidět, zda jsou optická vlákna nastavena správně či špatně. Nemusí tedy mezi komorou a PC několikrát přecházet, než bude seřízení správné, to vede k jasné úspoře času, a tudíž k efektivnějšímu testování a zrychlení procesu vývoje světlometů a zadních svítilen. Odhadovaný čas jedné cesty do komory a zpět je 14 sekund, přičemž operátor ji v průměru absolvuje minimálně třikrát. To znamená, že kdybychom brali průměrné hodnoty, ušetřilo by se přibližně 28 sekund před zahájením bloku měření. Zařízení je možno umístit na několik míst. K PC bude připojeno sériovou linkou, která povede do další místnosti skrz otvor ve zdi určený pro vedení kabelů. V této místnosti bude připevněno na zeď lepící gumou, hned vedle EMC komory, aby bylo dosaženo optimálního signálu.



Obrázek 2 Rozložení pracoviště

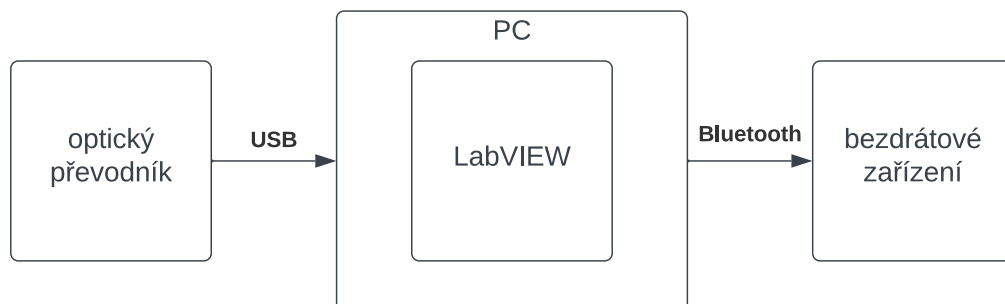
2.2 Omezení

- Hlavním omezením je stíněná EMC komora. Kvůli ní musí být zvoleno správné umístění zařízení, bylo otestováno nejlepší místo přímo vedle vstupu na zeď.
- Dalším je již zmíněné pracoviště operátora, které je od komory vzdáleno 10 metrů, tato vzdálenost bude vyřešena sériovou linkou, která povede přímo ke komoře.
- Dále je omezující IT struktura, která nedovoluje připojovat další zařízení do sítě na celém území firmy.
- Je třeba vzít v potaz i to, jak se kabel natáhne do další místnosti – průchod zdí již byl realizován pro další účely operátora.

2.3 Požadavky na realizaci

- Vytvoření aplikace v LabVIEW, prostředí bylo vybráno z důvodu laboratorního standardu a převodník je již vybaven měřicí kartou National Instruments.
- Dalším je zobrazení na bezdrátovém zařízení, aby operátor měl hodnoty převodníku k dispozici v komoře při fixaci optických vláken (do vlákna musí vyzařovat dostatečné množství světla, aby měření mělo správnou referenční hodnotu a proběhlo korektně).
- Zobrazení na zařízení s operačním systémem Android bylo vybráno z důvodu nižší pořizovací ceny.
- Zároveň byl požadavek o intuitivní ovládání vzniknutých aplikací s možností použít pouze jednu měřicí kartu. Dvě měřicí karty jsou současně používány pouze pro měření velkých předních světél.
- Rozlišení napěťových úrovní je požadováno na jedno desetinné místo a minimální obnovovací frekvenci 2 Hz (0,5s) a real-time zobrazení všech 8 kanálů, kterými převodník disponuje.

Předpokládaný systém, který má tato práce dokončit, byl zamýšlen na způsobu popsaném v obrázku 3.

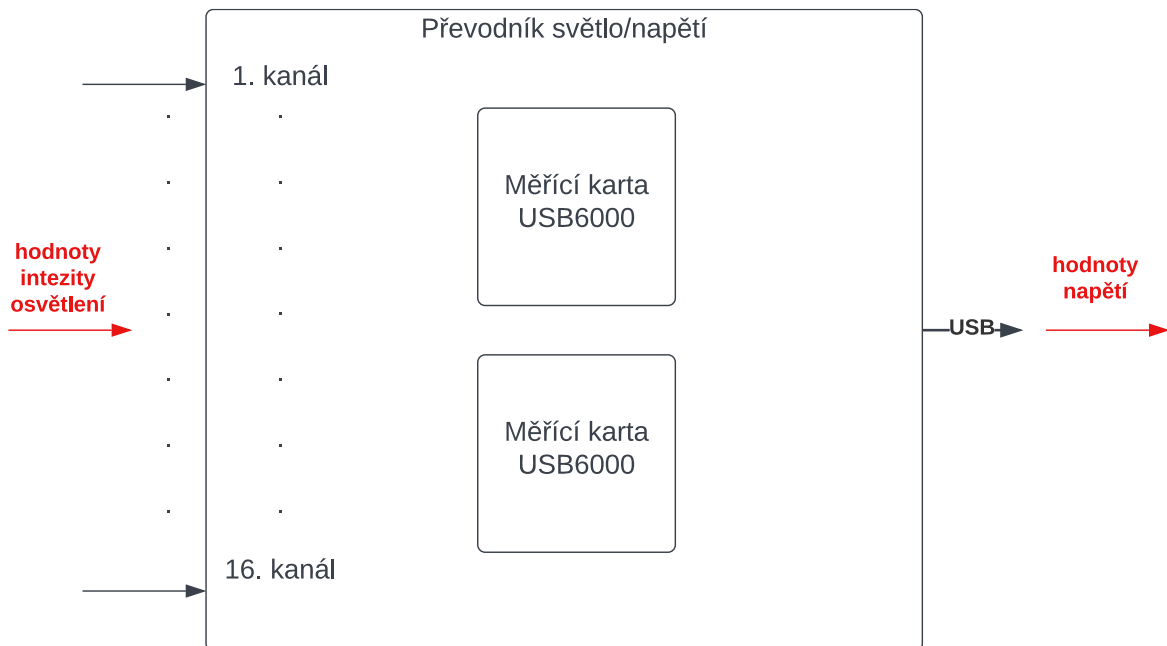


Obrázek 3 Blokové schéma předpokládaného systému

2.4 Popis optického převodníku

Jde o automatický převodník světla na napětí, vstupem je tedy světlo z optického vlákna a výstupem napěťový signál. Signál je měřen kartou National Instruments USB 6000 (viz kapitola 2.5) a převodník obsahuje hned dvě tyto karty. Převodník má 8 vstupů pro optická vlákna a lineární charakteristiku. Rozsah hodnot je od 0 V do 4 V. Vyhodnocení probíhá v měřicím ovládacím programu EMC32, který má nahradit moje aplikace v LabVIEW.

Posílaná data: měřené napětí maximálně 16 kanálů (1 měřicí karta = max. 8 kanálů). Data se po spuštění měření odesílají nepřetržitě.



Obrázek 4 Blokové schéma optického převodníku

Pro zbytek práce s ním bylo zacházeno jako s black-boxem (nebudou uvedena úplná vnitřní zapojení ani stěžejní body jeho funkce) z důvodu ochrany firemních dat.

2.5 NI měřicí karta

Přivedená intenzita osvětlení je převedena pomocí převodníku na napětí, které zpracovává měřicí karta USB-6000 od firmy National Instruments. Ta je připojena k počítači pomocí USB kabelu. Její maximální vzorkovací frekvence je 10 kS/s a disponuje 8 analogovými vstupními kanály. Do automatického převodníku byla osazena hlavně z důvodu nízké pořizovací ceny a kompaktních rozměrů. Karta je plně kompatibilní s prostředím LabVIEW a je možné ji ovládat a konfigurovat pomocí ovladače NI-DAQmx. [4]

3. SROVNÁNÍ BEZDRÁTOVÝCH TECHNOLOGIÍ

V této kapitole je uvedeno shrnutí známých a zároveň pro tuto práci vhodných bezdrátových technologií.

Obrázek 5 pomůže představit, kdy se které technologie typicky využívají v závislosti na typu dat, která systémem procházejí.

Spojení pomocí sériové linky (kabelem) bylo z výběru možných propojení vyřazeno, protože přímo do EMC komory nesmí být přidáno žádné přídavné zařízení, aby bylo zachováno odstínění nutné pro testování. Zároveň je bezdrátový přenos dat vhodnější, a to díky jednodušší manipulaci s malým zařízením, jako je mobilní telefon či tablet.

Short range wireless application areas

	Voice	Data	Audio	Video	State
Bluetooth ACL / HS	x	Y	Y	x	x
Bluetooth SCO/eSCO	Y	x	x	x	x
Bluetooth low energy	x	x	x	x	Y
Wi-Fi (VoIP)	Y	Y	Y	Y	x
Wi-Fi Direct	Y	Y	Y	x	x
ZigBee	x	x	x	x	Y
ANT	x	x	x	x	Y

Obrázek 5 Možnosti použití bezdrátových technologií [5]

3.1 Wi-Fi

Využívá dvě různé frekvence rádiových vln (2,4 GHz a 5 GHz) pro přenos dat mezi naším zařízením a routerem. [6]

Wi-fi nelze použít z důvodu ochrany dat firmy, a nebyla proto provedena rešerše její možné realizace. Do sítě je možný přístup pouze při schválení přístupu od vedení komplexu. Data, která se na síti odesílají a přijímají, jsou citlivá, a proto jsou zavedena všechna opatření pro maximální bezpečnost proti úniku těchto dat.

3.2 Bluetooth

BT je bezdrátová technologie, která umožňuje posílání dat mezi různými zařízeními v malém okruhu (typicky 10 metrů). Pro připojení není potřeba žádné síťové

infrastruktury (jako je např. router). Technologie operuje na frekvenci 2,4 GHz [7]

- **Bluetooth Classic**

- 79 kanálů na frekvenci 2.4 GHz
- Topologie pouze point-to-point
- Hlavní použití je pro bezdrátové streamování audia a nejčastěji propojovaná zařízení jsou mobilní telefony a různé bezdrátové periferie, například pro práci s počítačem (bezdrátová sluchátka, myš, klávesnice). [8]

- **Bluetooth Low Energy**

- Poprvé uveden ve verzi Bluetooth 4.0.
- 40 kanálů na frekvenci 2.4 GHz
- Topologie: point-to-point, broadcast, mesh
- Jednou z hlavních podmínek bylo vytvoření technologie s malou spotřebou energie. Zařízení mohou fungovat s pomocí jedné malé baterie dny, týdny nebo v krajních případech i měsíce či roky. [7]
- Vývoj probíhal se záměrem podpořit IoT (internet of things) prostřednictvím využívání nízkoenergetických zařízení s nízkou pořizovací cenou. [7]
- BLE používá široké spektrum snímačů, modulů a menších zařízení. Nově je pomocí této technologie možnost zjistit přítomnost, vzdálenost a směr, ve kterém se nachází jiné zařízení. [9]

- **Bluetooth profily**

Pro správný výběr Bluetooth rozšíření je potřeba vědět, pro jakou aplikaci bude používáno. Profil pracuje na jádru samotného BT a v některých případech přidává další komunikační protokoly. Profily tedy dávají možnost všem výrobcům, kteří používají tuto bezdrátovou síť, aby jejich zařízení mohla být využívána v dané oblasti se správnou funkčností. [10]

Informaci o tom, jakým profilem přístroj disponuje, se typicky nachází v manuálu od výrobce. Většina dostupných adaptérů a modulů dostačuje pro požadavky této práce, ale bylo zkontrolováno, že je podporován profil Device ID Profile. Ten umožňuje počítači, ke kterému je připojen, nainstalovat správné ovladače [11].

4. VÝBĚR ŘEŠENÍ BLUETOOTH SPOJENÍ

V této kapitole jsou rozebrány hlavní výhody a nevýhody jednotlivých komponent, vyzkoušena jejich funkčnost při zapojení do soustavy a ověřena vzájemná kompatibilita. Na závěr je provedeno finální srovnání výsledků a shrnutí, co bude použito a z jakých důvodů.

4.1 Použitý software

Pro ovládání převodníku byla vyvinuta aplikace v softwarovém prostředí LabVIEW, a to z důvodu laboratorního standardu a častého využívání tohoto programu firmou pro vytváření ovládacích aplikací na míru. Převodník obsahuje měřicí karty od stejné firmy jako tento software, tudíž je jasně daná kompatibilita a není možné zvolit vhodnější řešení.

Aplikace pro zařízení se systémem Android byla vyvinuta pomocí webového vývojového prostředí MIT App Inventor [12], a to z důvodu podstatně jednodušší realizace než například v programu Android Studio. Má velmi omezené funkce, ale pro fázi testování Bluetooth modulů maximálně dostačoval požadovanému účelu.

Vyvinutí finální androidové aplikace bude uskutečněno v programu Android Studio, které je nejrozšířenějším vývojovým prostředím pro aplikace na mobilní zařízení s tímto operačním systémem. Zároveň má oficiální podporu, bohatou zásobu různých příkladů a obrovskou komunitu téměř pro každé řešení všech problémů. [13]

Prostředí pro naprogramování Arduina bylo zvoleno Arduino IDE, což je oficiální program výrobce s rozsáhlými komunitními fóry. [14]

Tabulka 1 Použitý software pro testování Bluetooth modulů

Program	Verze	Přídavné knihovny
NI LabVIEW	2022 Q3 (32-bit)	VISA (23.0)
MIT App Inventor	-	-
Arduino IDE	1.18.19	-

4.2 Možnosti

Po srovnání bezdrátových technologií (v kapitole 3), byla zvolena možnost využití Bluetooth s tím, že budou otestovány verze Classic i Low Energy (LE) a na základě výsledků bude vybrána lepší varianta. Tuto variantu lze realizovat například pomocí USB Bluetooth donglu, použitím mikrokontroléru s BT modulem nebo samostatné vývojové desky se zabudovaným BT rozhraním.

4.2.1 BT dongle (USB pc)

Jedná se o zařízení, které je pomocí USB portu připojeno k počítači. Jde tedy o nejrychlejší způsob, jak doplnit Bluetooth rozhraní do soustavy, a to vzhledem k nízké pořizovací ceně a jednoduché instalaci (obvykle postačuje nainstalování ovladače, který výrobce poskytne).

Na trhu je mnoho druhů v různých cenových kategoriích. Testované dongly byly vybrány na základě toho, jaké BT profily podporují a do jaké cenové kategorie spadají.

- BLED112-V1 – disponuje Bluetooth LE s verzí 4.0.
- BT-8500 – obsahuje Bluetooth classic s verzí 5.0.
- USBBT1EDR2 – dongle s BT classic a jeho verzí 2.1.

4.2.2 BT modul pro Arduino

Přídavný modul pro vývojovou desku zajišťuje Bluetooth rozhraní, které je možné dále používat. Pro tuto platformu existuje několik typů.

- HC-05 – dokáže pracovat v režimu master i slave.
- HC-06 – pracuje pouze v režimu slave.
- XM-15B – vylepšený modul HC-05 v ochraně proti přepólování, rychlejšímu zapnutí a také watchdogu (resetování systému při zacyklení), nebyl testován z důvodu vyšší ceny než HC-05 a pouze malých výhod pro tuto aplikaci.
- AT-09 – oproti dalším modulům podporuje jak operační systém Android, tak i Apple a je dostupný pouze ve verzi BLE (z tohoto důvodu nebyl zařazen mezi testované moduly).

Druhy kontroléru byly vybrány od společnosti Arduino, a to z důvodu, že firma jej využívá v několika jiných případech, a také kvůli jeho jednoduché rozšiřitelnosti a vysoké úrovni podpory (ať už komunitní nebo oficiální obsah). Existují i další kontroléry, které mají podobné až identické chování a podporu, jako například ESP32.

4.3 Testování

V následující kapitole je popsán způsob testování jednotlivých komponent, a to v prostředí uzpůsobeném tak, aby co nejvíce simulovalo pozdější reálné používání operátorem.

Testování proběhlo za normálního provozu laboratoře. Tudíž všechny součásti v okruhu zařízení, co by mohly signál přerušovat, byly v provozu. Nebyl testován napěťový výstup z převodníku, ale pouze simulovaná data pro zjednodušení. Následné přidání převodníku na výsledku testování mělo velmi malý (zanedbatelný) dopad.

Hlavním účelem testování byl dosah signálu do EMC komory, tudíž bylo zařízení umístěno ve finální pozici (viz kapitola 2.1). Spolu s tím byla testována kompatibilita spojení jednotlivých částí systému, a to jak po hardwarové, tak softwarové stránce – šlo tedy o simulaci finálního stavu projektu. Zároveň bylo vyzkoušeno fyzické propojení

soustavy. Zpoždění v této fázi nebylo měřeno z důvodu nedokončení ostatních částí práce, konkrétně finální aplikace pro zařízení s operačním systémem Android.

Komponenty byly rozděleny do tří skupin a to:

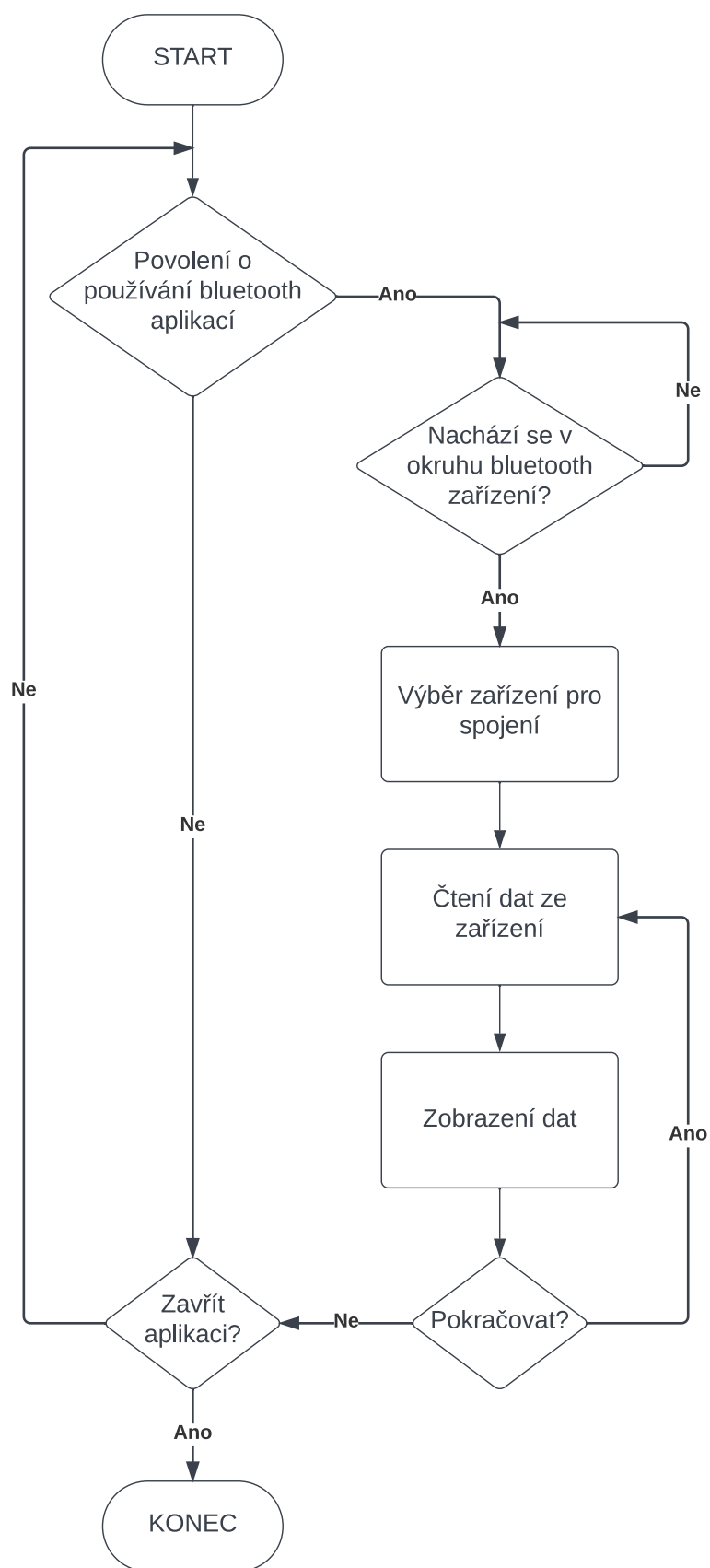
- Skupina **A** – Do této skupiny patří moduly pro Arduino HC-05 a HC-06, které disponují technologií BT Classic.
- Skupina **B** – Tato skupina obsahuje Arduino NANO 33 IoT a dongle BLED112-V1, které disponují technologií Bluetooth Low Energy.
- Skupina **C** – Obsahuje pouze USB/Bluetooth dongly, a to BT-8500 a USBBT1EDR2, oba disponují technologií Bluetooth Classic.

Tabulka 2 Srovnání testovaných modulů/adaptérů

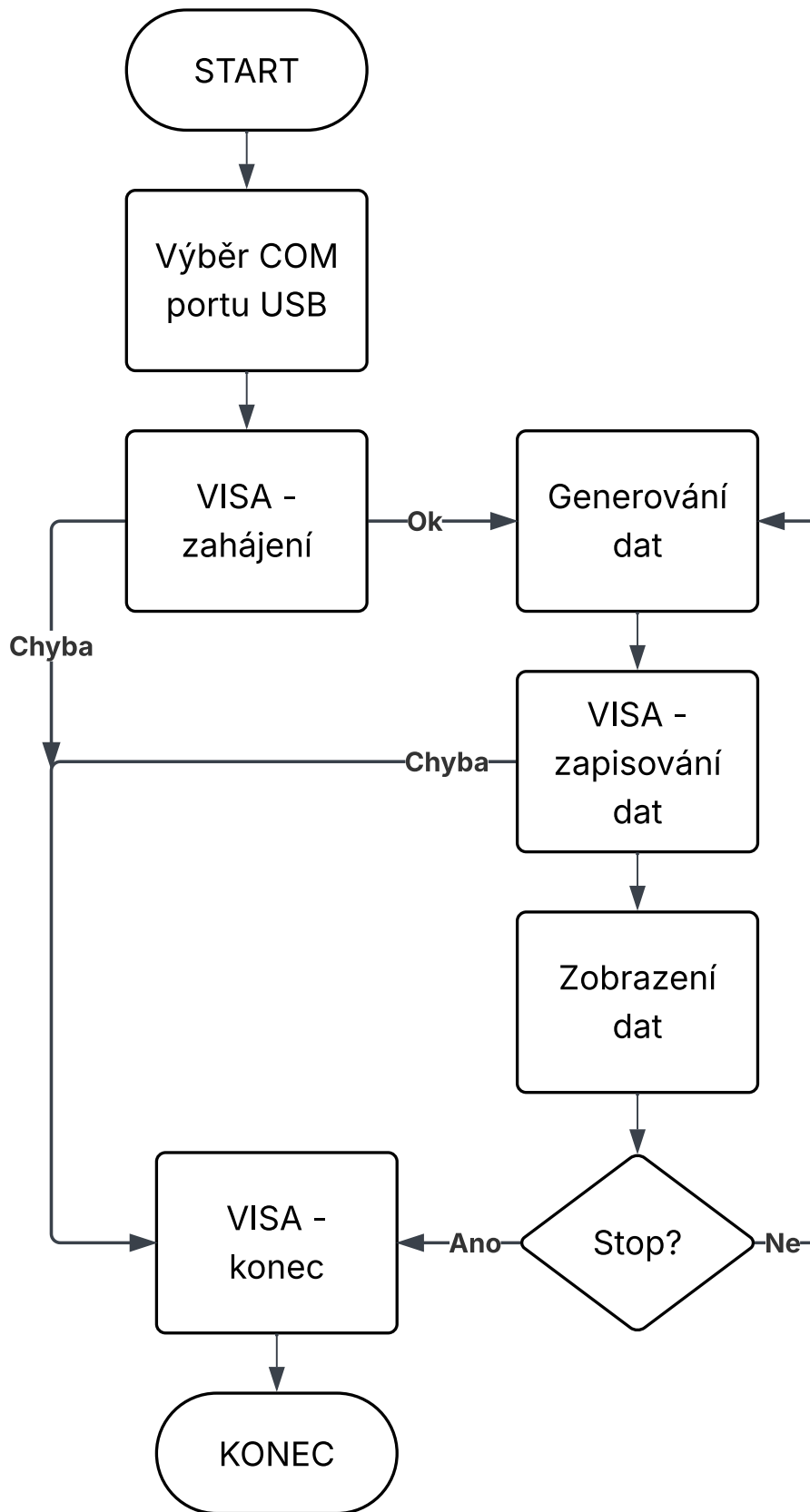
Název	Výrobce	Verze BT	BT classic/LE	Cena
HC-05	HWKITCHEN	V2.0	classic	128,00 Kč
HC-06	HWKITCHEN	V2.0	classic	148,00 Kč
Nano 33 IoT	Arduino	V2.1	LE	764,91 Kč
BLED112-V1	SiliconLabs	V4.0	LE	474,28 Kč
BT-8500	Edimax	V5.0	classic	488,16 Kč
USBBT1EDR2	StarTech	V2.1	classic	626,68 Kč

Pro spojení adaptéru s architekturou BLE a programem LabVIEW neexistuje oficiální řešení ze strany společnosti National Instruments [15]. Existují pouze dvě řešení. První z nich je toolkit vytvořený komunitou [16], který však nepodporuje všechny základní operace v programu, a není tedy zaručena spolehlivost přenosu dat. Druhou možností je využití knihovny systému Windows (Microsoft BluetoothAPIs.dll), která však ve většině případů nefunguje spolehlivě [17]. V případě programu Android Studio je oficiální podpora jak pro Classic, tak pro Low Energy variantu.

Popis funkce testovacích aplikací je znázorněn pomocí vývojových diagramů (viz obrázek 6 a obrázek 7). Vzhledem k tomu, že finální aplikace bude vytvořena v Android studiu, lze očekávat mírně odlišné chování systému. Hlavním účelem testování však byla ověřitelná kompatibilita propojení jednotlivých komponent, a z tohoto pohledu daný postup vyhovoval stanoveným požadavkům.



Obrázek 6 Vývojový diagram testovací aplikace pro bezdrátové zařízení

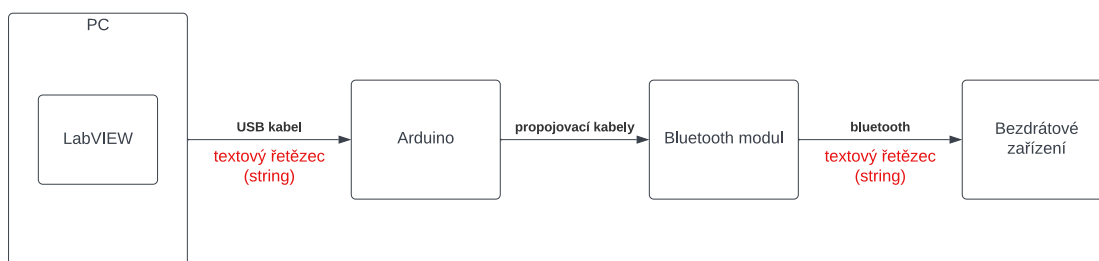


Obrázek 7 Vývojový diagram testovací aplikace LabVIEW

Generovaná data byla nejprve náhodná, a to v rozmezí očekávaných výstupních dat z převodníku, tento proces sloužil pro určení kompatibility. V dalším kroku byla posílána inkrementovaná data pro zjištění ztráty dat.

- Skupina A

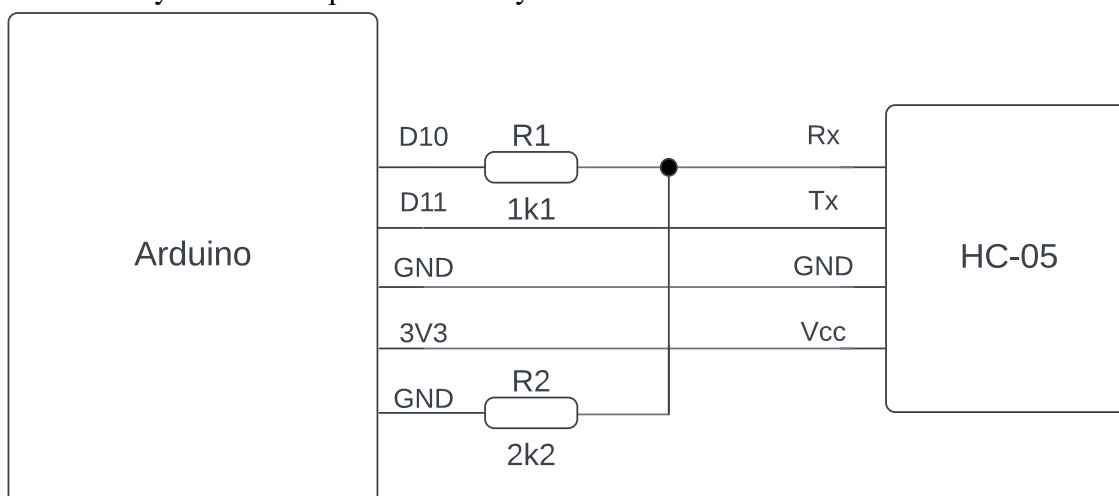
Moduly HC-05 a HC-06 byly připojeny k Arduino Nano (použito z důvodu přítomnosti na pracovišti a kompaktní velikosti, zapojení by bylo obdobné i za využití jiné podobné desky). Z programu LabVIEW byla generována data, která byla posílána přes USB do Arduina. Mikrokontrolér pouze zprostředkoval předání informací přidavnému Bluetooth modulu, který zajistil odesílání dat na mobilní telefon. Na koncovém zařízení byla data opět zobrazena. Proces je znázorněn na obrázku 8, kde jsou červenou barvou vyznačena přenášená data a černou barvou fyzické propojení.



Obrázek 8 Zapojení pro testování modulů s Arduinem

V případě použití modulu HC-05 bylo spojení bezproblémové, což se však nepotvrdilo u modulu HC-06. Připojení nebylo stabilní, docházelo k častým výpadkům a nepravidelné funkčnosti (tudíž nedocházelo k zobrazování dat na bezdrátovém zařízení). Komunikace s LabVIEW probíhala v obou případech bez problémů.

Na základě tohoto testování byl modul HC-05 vyhodnocen jako lepší – měl stabilní připojení a data systémem procházela bez nežádoucích stavů. Zapojení znázorněné na obrázku 9 bylo identické pro oba moduly.



Obrázek 9 Zapojení Arduina s BT Modulem HC-05

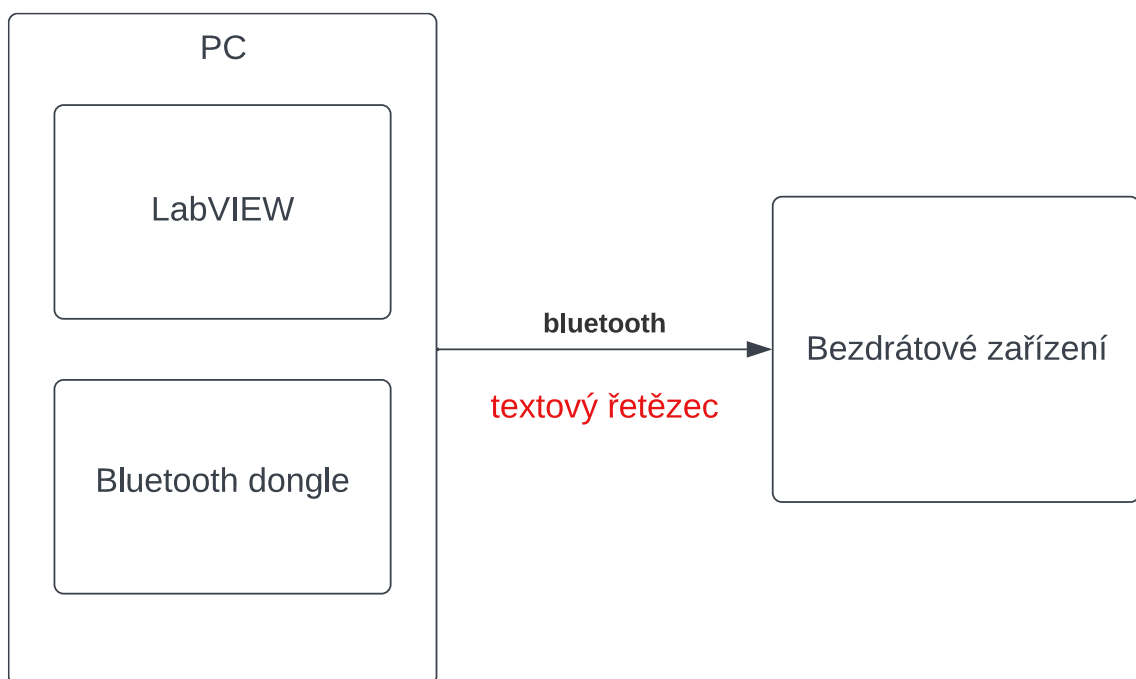
- Skupina B

Výše zmíněný problém s navázáním komunikace mezi LE zařízením a programem LabVIEW byl potvrzen. Architektura Bluetooth Low Energy není přímo určena pro prostředí LabVIEW a řešení pomocí dostupných knihoven nebylo spolehlivé – data nebyla přenášena ve správném formátu a nebylo možné zobrazit reálné hodnoty.

Při použití donglu nastalo více problémů, včetně toho, že komunikace mezi zařízeními nebyla navázána ani po instalaci ovladače doporučeného výrobcem.

- Skupina C

Při testování donglů s klasickým Bluetooth byla navázána komunikace, avšak dosah samotného signálu nebyl dostatečný. Docházelo ke ztrátě připojení při vstupu do komory. Tudíž tato varianta nebyla vhodná pro zadané použití.



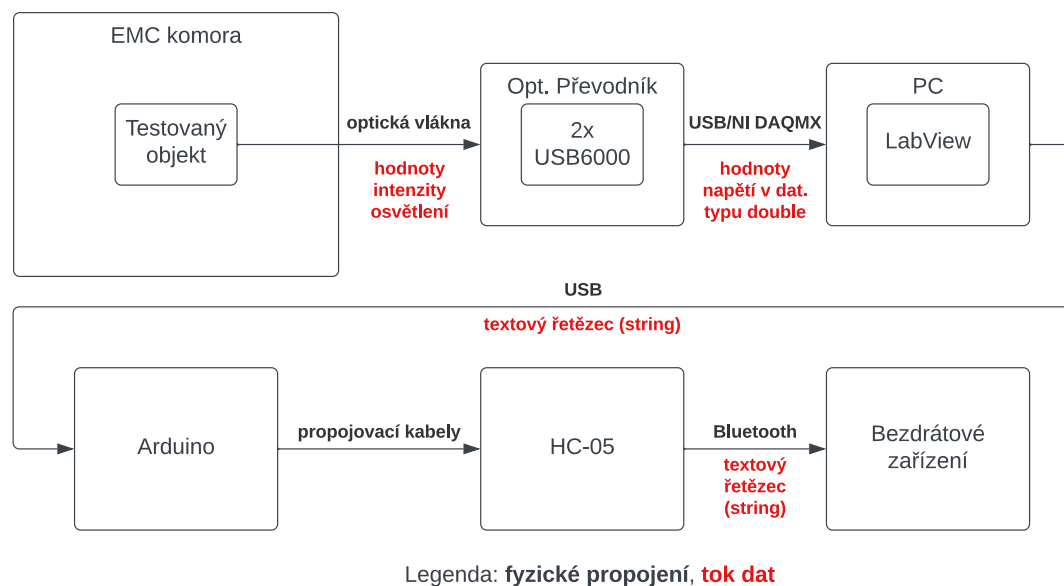
Obrázek 10 Zapojení pro testování donglů

Bylo vybráno Arduino s přídatným modulem HC-05, a to hned z několika důvodů: nejstabilnější signál, dostatečný dosah a celkově nejjednodušší realizace, která proběhla bez větších problémů. Další výhodou jeho využití je možné rozšíření, či vylepšení soustavy do budoucna, například o signalizaci stavu připojení Bluetooth nebo signalizaci zapnutého napájení. Současně firma o tuto možnost měla zájem. Cenové porovnání tohoto řešení s jedním z donglů je velmi podobné a liší se o zhruba 100 Kč (dle současné ceny komponent).

5. NÁVRH A IMPLEMENTACE APLIKACÍ

5.1 Způsob přenosu dat mezi optickým převodníkem a zbylými částmi systému

V této kapitole je popsána softwarová stránka spojení mezi jednotlivými komponentami soustavy. Jsou zde rozebrána posílaná, měřená a zobrazovaná data ve formátech, které byly použity pro konečnou realizaci.



Obrázek 11 Blokové schéma celého systému s procházejícími daty a fyzickým spojením

První částí systému je optický převodník, do kterého vstupuje světelný tok vedený optickým vláknem z EMC komory. Jeho výstupními daty jsou hodnoty napětí, která měří měřicí karta NI. Je vyčteno 1D pole hodnot z ovladače, které se dále používá v aplikaci. Vyčítání z karty je kontinuální a naměřená data jsou formátována na textový řetězec (string) a za použití funkce VISA [18] jsou data posílána dále do soustavy. Jednotlivé kanály jsou odděleny znakem středníku, aby bylo možné jejich rozlišení. Oddělení desetinné části bylo realizováno pomocí desetinné tečky. Kompletní zpráva je zakončena znakem nového řádku a má následující tvar v každé části systému:

```
Číslo1;Číslo2;...;Číslo15;Číslo16;\n
```

Data jsou dále posílána po USB kabelu do Arduina. Mikrokontrolér data zpracuje (viz kapitola 5.4) a následně odešle zpět textový řetězec ve stejném tvaru, v jakém jej přijal. Zpracování slouží ke kontrole počtu znaků a správného použití oddělovačů mezi jednotlivými hodnotami. Arduino prostřednictvím BT modulu posílá data na bezdrátové

zařízení. Android aplikace poté rozliší jednotlivé části dat pomocí oddělovače a rozdělí data na jednotlivé kanály. Ty jsou poté vizualizovány na obrazovce zařízení. Celý proces je opakován, dokud není ukončen operátorem nebo dokud nenastane chyba v některé z aplikací.

5.2 Požadavky na aplikace

Požadavky, které byly zadány firmou nebo vyplývají ze zadání bakalářské práce, lze shrnout v tabulce 3, kde požadavky 1 až 4 jsou pro aplikaci LabVIEW, požadavek 5 pro Arduino a požadavek 6 až 8 pro Android.

Tabulka 3 Požadavky na aplikace

Číslo	Požadavek	LabVIEW	Arduino	Android
1	Vyčítání dat z měřících karet převodníku a možnost výběru konfigurace měření.	✓	×	×
2	Zobrazení šestnácti kanálů pro operátora je realizováno s rozlišením na dvě desetinná místa a minimální obnovovací frekvencí 2 Hz.	✓	×	×
3	Komunikace s další komponentou systému – formátovat data pro přenos na další komponentu, kterou je Arduino.	✓	×	×
4	Preferovaná architektura – Stavový automat	✓	×	×
5	Zprostředkování bezdrátového přenosu dat z PC pomocí technologie BT.	×	✓	
6	Uživatelská přívětivost – cílem celého projektu je zrychlení testování, ovládání aplikace musí být intuitivní a nesmí operátora zdržovat při procesu upevňování optických vláken.	×	×	✓
7	Zobrazení 16 měřících kanálů – data musí být zobrazena jasně a čitelně s možností procházení mezi jednotlivými kanály či skupinou kanálů. Dále je požadavkem možnost oddělení 2 měřících karet čili zobrazení pouze osmi indikátorů na jednu obrazovku.	×	×	✓
8	Snaha o co nejmenší časovou prodlevu mezi vykreslením dat a zahájením Bluetooth komunikace – optimalizace aplikace pro co nejrychlejší zobrazení dat.	×	×	✓

Tabulka 4 Použitý software pro vývoj aplikací

Aplikace	Program	Verze	Přídavné knihovny
LabVIEW	NI LabVIEW	2024 Q1 (32-bit)	DAQmx (2024 Q1) [19], VISA (24.0)
Arduino	Arduino IDE	1.8.19	-
Android	Android Studio	2024.2.2 Ladybug	-

5.3 LabVIEW

Vizualizace dat je pro operátora důležitá především kvůli nastavení limitních hodnot pro testování v aplikaci EMC32.

Ovládací aplikace pro optický převodník byla navržena pomocí grafického vývojového prostředí LabVIEW, které umožňuje vytváření testovacích, řídicích nebo měřicích aplikací [20]. Tento software byl zvolen kvůli již předem osazené měřicí kartě převodníku, která je od stejného výrobce. Současně je program laboratorním standardem a jedním z požadavků firmy.

5.3.1 Návrh aplikace

Aplikace vytvořená v prostředí LabVIEW byla navržena na základě požadavků definovaných firmou i zadáním bakalářské práce, jak je uvedeno v tabulce 3. Hlavním cílem bylo vytvořit přehledné a stabilní uživatelské rozhraní pro operátora, které umožní načítání a zobrazení dat z měřicích karet, jejich základní vyhodnocení a následný přenos do další části systému. Návrh byl proveden tak, aby byla minimalizována možnost vzniku uživatelských chyb a aby bylo ovládání aplikace intuitivní i během opakovaného použití v provozu.

Pro řízení aplikace byl zvolen stavový automat (viz obrázek 13), který umožňuje přehledné rozvržení jednotlivých funkcí a snadnou rozšiřitelnost. Po spuštění je aplikace uvedena do výchozího stavu, ve kterém čeká na vstup uživatele. Na základě zvoleného ovládacího prvku je následně přecházeno do příslušného stavu – například spuštění měření, výpočet limitů, resetování limitních hodnot, zastavení měření nebo ukončení programu.

Byly navrženy následující funkční celky:

- možnost výběru měřicích karet a portu pro komunikaci s Arduinem,
- zobrazení až 16 kanálů s možností přepínání mezi dvěma měřicími kartami (každá se zobrazením osmi kanálů),
- výpočet a zobrazení limitních hodnot,
- resetování minimálních a maximálních hodnot,
- přenos dat po sériové lince do mikrokontroléru.

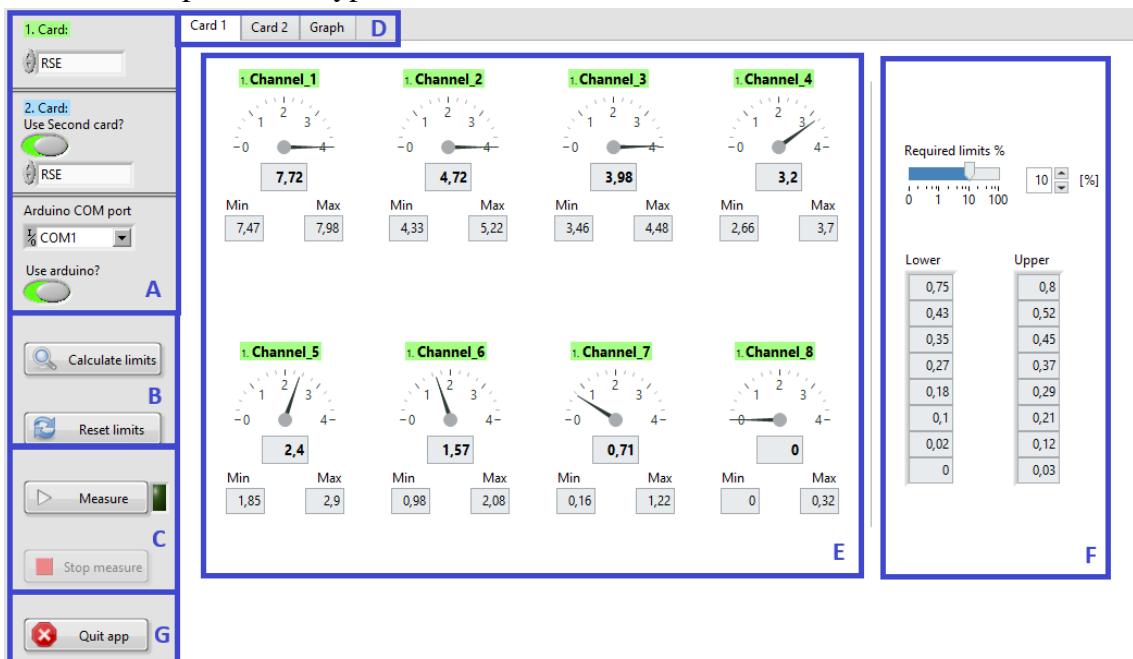
Jednotlivé ovládací prvky uživatelského rozhraní byly rozděleny do logických sekcí: nastavení zařízení, řízení měření, výpočet limitů a ukončení programu. Měření bylo implementováno tak, aby byla zajištěna stabilita a zároveň eliminována možnost nežádoucích stavů, například při opakovaném nebo nechtěném stisku ovládacích tlačítek.

Aplikace byla koncipována jako cyklický proces, ve kterém je po dokončení každého měření umožněno jeho opakování bez nutnosti restartu. Tím bylo dosaženo vyšší efektivity při praktickém použití na pracovišti.

5.3.2 Uživatelské prostředí

UI na obrázku 12, lze rozdělit do šesti hlavních částí a to:

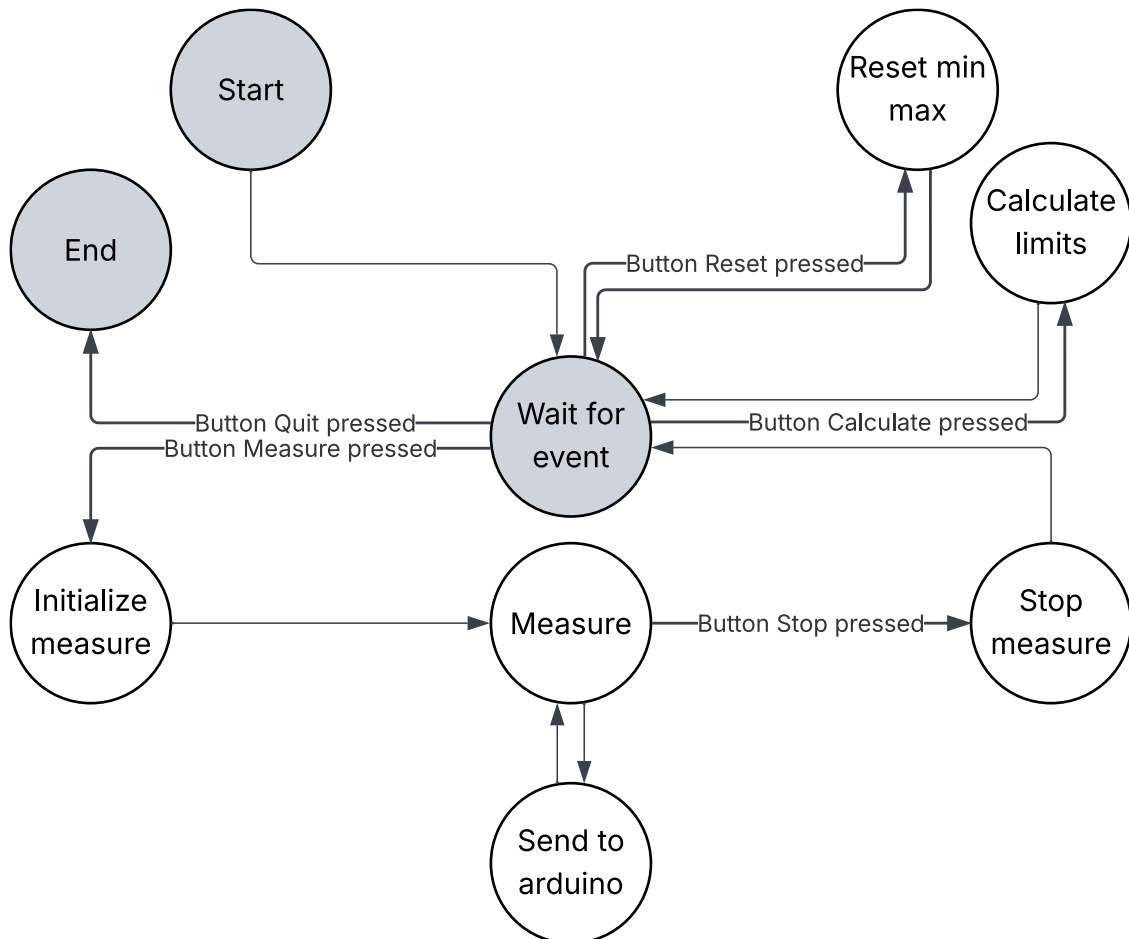
- Nastavení:** Operátor musí před zahájením měření vybrat konfiguraci měření (což je požadavek firmy) a pokud chce zobrazit data na bezdrátovém zařízení, musí vybrat port pro komunikaci s Arduinem.
- Ovládání výpočtů limitů:** Slouží k výpočtu limitů či resetování jejich hodnot. Limity jsou pro operátora důležité pro nastavení jiných měřicích programů. Počítají se jako procentuální podíl vzhledem k maximální nebo minimální hodnotě.
- Ovládání měření:** Po základním nastavení může být spuštěno měření, přičemž jeho stav je signalizován pomocí malého LED indikátoru. Během měření, až do doby stisknutí tlačítka *Stop measure*, operátor nemůže měnit nastavení, vypočítat limity ani vypnout aplikaci. Tyto akce jsou možné po zastavení měření.
- Navigační menu:** Umožňuje uživateli přepínat zobrazení naměřených dat mezi jednotlivými měřicími kartami a současně zkontrolovat měření ve společném grafu.
- Zobrazení naměřených hodnot:** Hodnoty jsou zobrazeny pomocí grafického ukazatele i číselné hodnoty s dvěma desetinnými místy. Zároveň jsou vyhodnoceny minimální a maximální hodnoty.
- Zobrazení limitů:** Přehledné zobrazení hodnot a nastavení procentuálního výpočtu.
- Ukončení aplikace:** Řádně ukončí veškeré vytvořené komunikační kanály a aplikace se vypne.



Obrázek 12 Uživatelské prostředí LabVIEW aplikace

Další nastavení, jako názvy měřících karet, vzorkovací frekvence a počet vzorků, jsou načítána z konfiguračního souboru INI [21], protože se předpokládá, že se nebudou často měnit. Tím pádem budou v souboru nastaveny pouze jednou.

5.3.3 Popis aplikace



Obrázek 13 Stavový diagram LabVIEW aplikace

Po spuštění aplikace musí operátor nejdříve nastavit základní parametry, jinak se mu neumožní provádět další akcí (například spuštění měření). Mezi tyto parametry patří výběr konfigurace měření a výběr portu COM pro komunikaci s Arduinem. Následně může být zvoleno tlačítko startu měření a program přejde do stavu Initialize Measure. V tomto stavu jsou načteny parametry z konfiguračního souboru. Po správném načtení potřebných údajů aplikace automaticky přejde do dalšího stavu Measure, kde bylo implementováno samotné měření (vyčítání dat z měřících karet) a také zobrazení hodnot do indikátorů a společného grafu. Po pěti iteracích měřicí smyčky (vybráno jako kompromis mezi rychlostí měření a odesílání dat) postoupí program do stavu Send to Arduino, kde bylo zhotoveno odesílání dat po sériové lince mikrokontroléru pomocí ovladače VISA. Formát posílaných dat je popsán v kapitole 5.1. Při měření může operátor

stisknout pouze tlačítko zastavení měření, tím jsou ošetřeny nežádoucí stavy jako dvakrát zmáčknuté nebo nechtěně zmáčknuté tlačítko. Po stisku tlačítka pro ukončení měření se aplikace přesune do stavu *Stop measure*, kde jsou řádně zpracovány zdroje a měření i odesílání dat je ukončeno. Následně přechází aplikace opět do stavu *Wait for event*, kde čeká na akci uživatele. Operátor může například vypočítat a zobrazit požadované limity, které se mohou mírně upravit. Celý proces se může libovolně opakovat a záleží plně na uživateli. Současně aplikace celou dobu kontroluje možné chyby a při vyskytnutí operátora informuje jaký problém nastal a program ukončí.

Pokud operátor vybere měření pouze jednou kartou, jsou odeslány hodnoty z první karty (8 hodnot), zatímco pro druhou kartu jsou odeslány pouze nuly.

Data jsou odesílána do Arduina přibližně každých 430 ms. Tato hodnota vyplývá z naměřeného celkového času pro pět iterací měřicí smyčky, po jejichž dokončení dochází k přechodu do stavu odesílání dat. Změření hodnoty bylo provedeno pomocí funkce *Tick count* [22].

Nastal problém s nespolehlivou komunikací mezi aplikací v LabVIEW a mikrokontrolérem Arduino. Ačkoliv bylo Arduino připojeno k napájení přes USB již delší dobu, po inicializaci sériového portu pomocí funkce *VISA Open* zařízení občas nereagovalo na příchozí data. Tento jev se projevoval nekonzistentně – komunikace někdy probíhala správně, jindy však Arduino neobdrželo žádná data, aniž by aplikace vyhodila chybu. Dodatečnou analýzou bylo zjištěno, že otevření sériového portu způsobuje na Arduino deskách automatický reset mikrokontroléru. Pokud začne LabVIEW odesílat data okamžitě po otevření spojení, je vysoká pravděpodobnost, že Arduino ještě není připraveno komunikaci přijmout. Tento jev je známý a potvrzen komunitou Arduino [23].

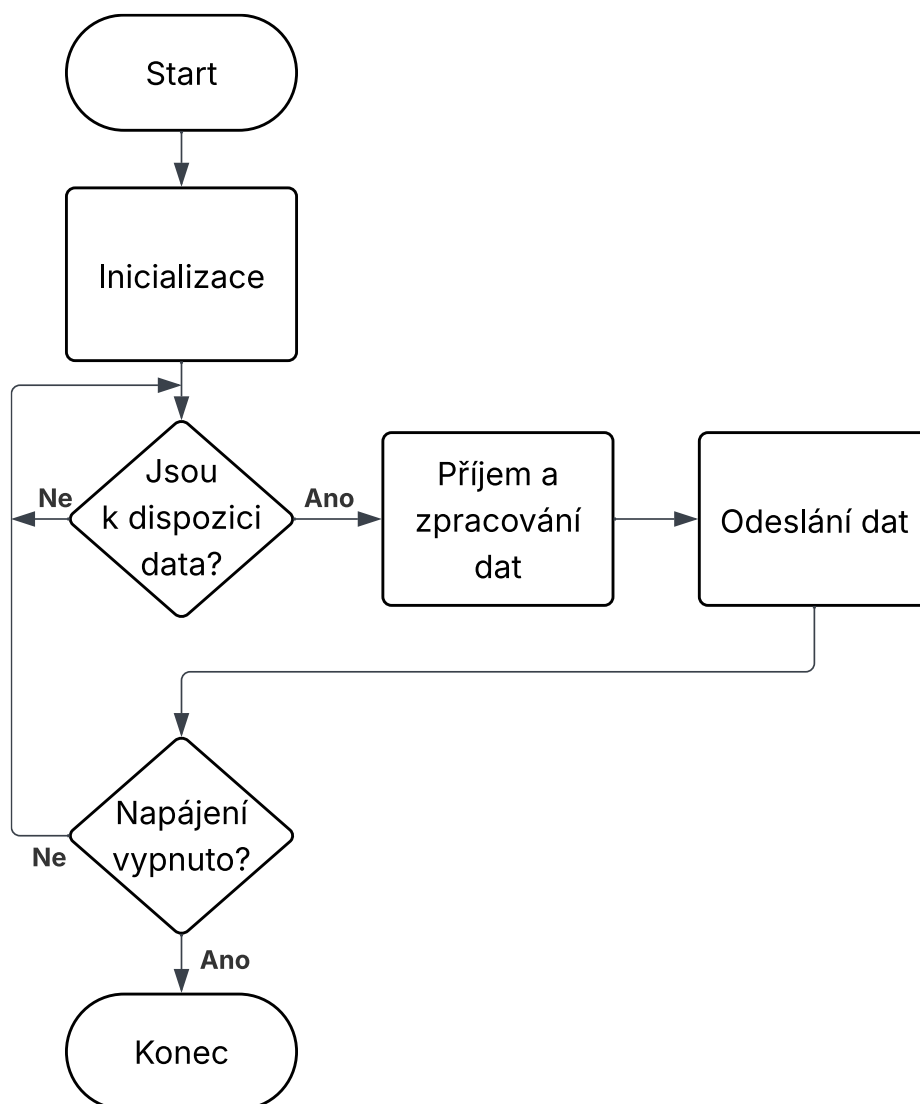
5.4 Arduino

Pro úspěšné naprogramování mikrokontroléru je potřebné, aby se v samotném programu nacházely dvě funkce – *void setup()* a *void loop()*. Funkce *void setup()* se vykoná po spuštění desky pouze jednou a typicky se v ní nastavují parametry jako přenosová rychlost sériové linky, režimy pinů (vstupní nebo výstupní) nebo probíhá inicializace knihoven. Funkce *void loop()* je smyčka, která se neustále opakuje a obsahuje hlavní část kódu [24]. V případě této práce zde byl implementován příjem dat ze sériové linky a následné odesílání dat přes Bluetooth.

Arduino slouží v systému pouze jako prostředník pro bezdrátovou komunikaci – vytváří spojení mezi aplikací na PC a bezdrátovým zařízením. Na tuto aplikaci se vztahuje požadavek 5 z tabulky 3. Na obrázku 14 je znázorněno zapojení mikrokontroléru Arduino MEGA spolu s modulem HC-05, určeným pro bezdrátovou komunikaci přes Bluetooth.

Dále je ve schématu znázorněno propojení komunikačních pinů. V rámci této práce byl konkrétně využit pouze pin *Tx* (vysílací), protože Android aplikace neodesílá žádná data zpět do Arduina, tudíž nemusí být pin *Rx* (příjmací) zapojen. Schéma rovněž uvádí

desetinného oddělovače je realizována jak v prostředí LabVIEW, tak i v kódu na straně Arduina, protože během testování docházelo ke záměně formátu v závislosti na lokalizačním nastavení systému. Poté je řetězec rozdělen na jednotlivé hodnoty podle oddělovače (;). Tyto hodnoty jsou postupně převedeny na číselné hodnoty datového typu *float* a uloženy do pole *ch[]*, které obsahuje celkem 16 měřicích kanálů. Pokud bylo úspěšně načteno všech 16 hodnot, je vytvořen nový řetězec, ve kterém jsou jednotlivé hodnoty odděleny středníkem. Tento řetězec je následně odeslán přes sériový port Serial1 do Bluetooth modulu. Přenos je ukončen znakem nového řádku. Po odeslání se program vrací zpět do hlavní smyčky a čeká na další vstup.



Obrázek 15 Vývojový diagram Arduino aplikace

5.5 Android

Aplikace pro bezdrátové zařízení byla vytvořena ve vývojovém prostředí Android Studio (verze je zobrazena v tabulce 4), to je oficiálním programem pro vývoj softwaru určeného pro zařízení s operačním systémem Android. Struktura projektu v tomto programu se skládá ze tří hlavních modulů [27] a to:

- Manifest – soubor, který obsahuje požadavky a důležitá data Android aplikace [28].
- Java – obsahuje zdrojový kód, je možné si vybrat ze dvou programovacích jazyků, a to Java nebo Kotlin.
- Res – obsahuje všechny nekódové soubory jako například rozložení prvků na obrazovce („layout“), obrázky nebo textové řetězce uživatelského rozhraní (UI).

Dále se rozlišuje zdrojový kód (s příponou „.kt“) a uspořádání funkčních prvků na obrazovce (s příponou „.xml“), které tento kód ovládá. Pod prvky patří například tlačítka, textová pole, přepínače nebo nástroje („widgety“). Tato práce byla zhotovena v jazyce Kotlin.

5.5.1 Návrh uživatelského rozhraní

Byl vytvořen design tak, aby bylo zachováno co nejjednodušší ovládání aplikace. Aplikace je tedy rozdělena do několika oken, kterými se uživatel pohybuje pomocí navigačního menu, které lze otevřít stisknutím tlačítka v pravém horním rohu. Tento prvek je dostupný v každém okně, aby bylo možné jednoduše procházet jednotlivé stránky.

Měřicí kanály byly rozděleny do dvou samostatných oken, a to v rozsahu 1 až 8 a 9 až 16. Na hlavní obrazovce, která je nastavena jako domovská, je realizováno nastavení a spojení přes Bluetooth – konkrétně zobrazení seznamu spárovaných zařízení, výběr zařízení pro připojení, odpojení od zařízení a zapnutí Bluetooth v zařízení. K indikátorům zobrazujícím měřené napětí je přiřazena aktuální hodnota pro přesné vyobrazení.

Okna a interakce (akce po stisknutí aktivního prvku) jsou znázorněny na obrázku 16. Interakce jsou popsány modrými šipkami – začátek je označen bílým kroužkem, konečný stav po stisknutí pak koncem šipky.

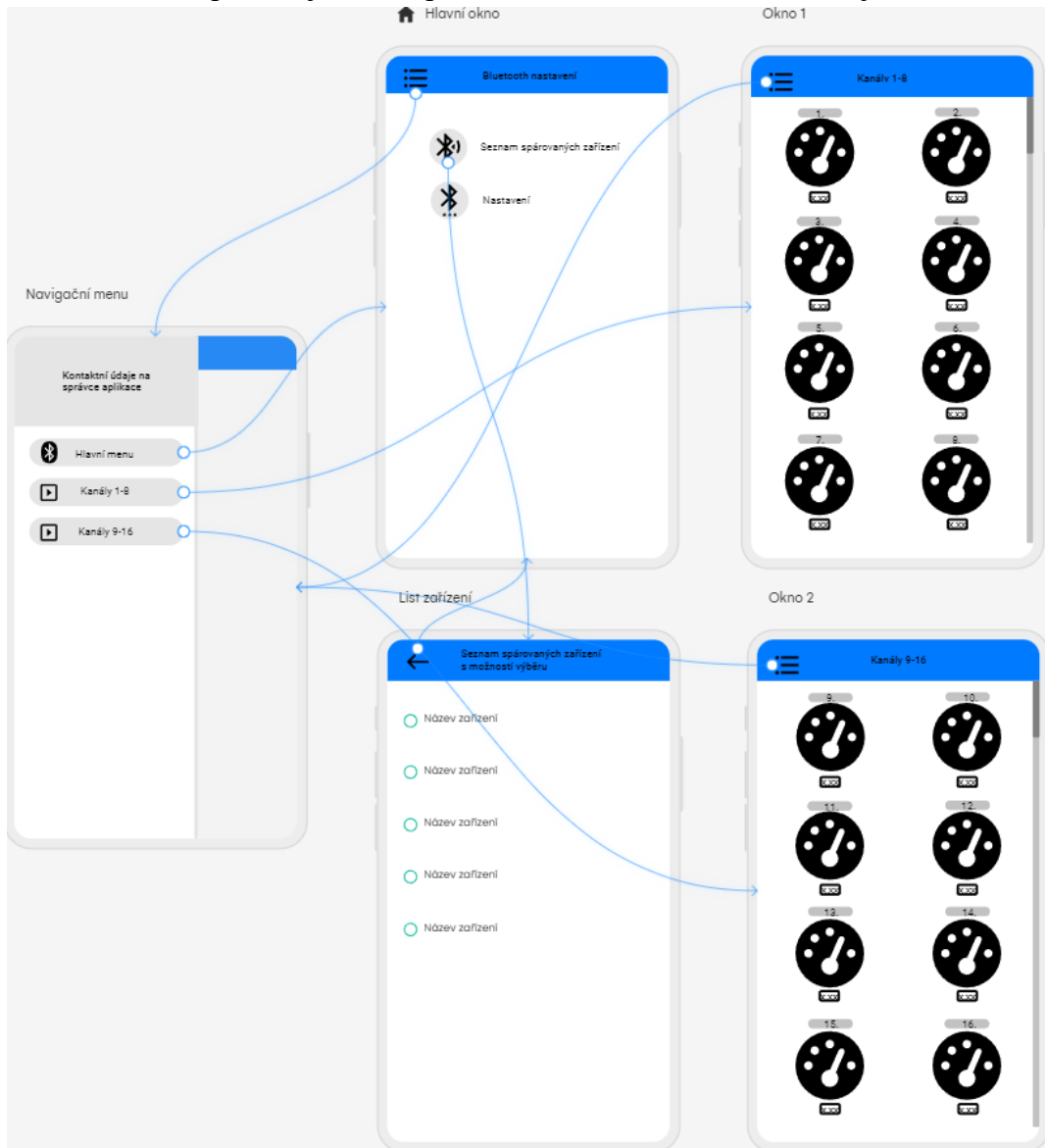
Každý prvek má kolem sebe určitou oblast, ve které při kliknutí dojde k aktivaci dané funkce. Tím je pro uživatele zajištěna snazší navigace v aplikaci, aniž by bylo nutné přesně trefit pozici prvku.

Barevná paleta byla zvolena na základě firemního loga, které se skládá primárně z modré a sekundárně z bílé barvy. Méně důležité prvky, například zvýraznění oblastí, jsou zhotoveny ve světle šedé barvě pro lepší přehlednost.

Byly použity základní prvky s vektorovou grafikou. Ikony jsou nativní pro systém Android (každé zařízení s tímto operačním systémem je bude bez problémů zobrazovat). Díky tomu bude aplikace na všech zařízeních fungovat správně a nedojde ke zmatení uživatele ani k omezení či narušení funkčnosti. Rozmístění jednotlivých komponent bylo

navrženo pomocí pružných vazeb, aby hodnoty mezi prvky nebyly pevně zakódovány a nedocházelo tak na zařízeních s různými rozměry displeje k překrývání. Finální vzhled aplikace je uveden v Příloha B – Návod k Android aplikaci.

Veškeré části aplikace jsou kompatibilní s verzí Android 12 a novějšími.



Obrázek 16 Návrh designu a propojení oken aplikace

Při implementaci bylo dbáno na kompatibilitu s různými displeji. Aplikace proto musí být optimalizována pro různé velikosti a rozlišení obrazovky. Způsob jakým toho bylo dosaženo, je popsán v kapitole 5.5.2. Dalším cílem je, aby aplikace při výskytu jakékoli chyby upozornila uživatele vyskakovacím oknem. V tomto malém okně bude vypsána krátká hláška popisující chybu, která v programu nastala.

5.5.2 Implementace uživatelského prostředí

Při implementaci uživatelského prostředí bylo využito XML slovníku, který je součástí

Android Studio a umožňuje implementovat prvky a měnit jejich základní parametry jako jsou například názvy, umístění na obrazovce nebo barva pozadí [29]. Při návrhu UI (viz kapitola 5.5.1) bylo cílem vytvořit tři obrazovky, kterými je možné libovolně procházet pomocí hlavního menu.

První obrazovka, označena jako hlavní a zobrazena při spuštění, obsahuje prvky pro správu Bluetooth. Těmito prvky jsou tlačítka *Bluetooth Enable* (zapnutí funkce Bluetooth na zařízení), *Show paired devices* (zobrazení spárovaných zařízení), *Read data* (start čtení dat přes Bluetooth) a *Disconnect* (odpojení připojeného zařízení). Příklad kódu pro implementaci tlačítka zapnutí Bluetooth:

```
<Button
    android:id="@+id/button_enable_bluetooth"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Enable Bluetooth"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toTopOf="@+id/button_show_devices"
    app:layout_constraintVertical_chainStyle="packed" />
```

V tomto kódu je nastaveno několik parametrů. Prvním z nich je *android:id*, které slouží k pojmenování prvku a toto jméno je možné využít například pro spuštění aktivity. Dalším klíčovým parametrem je sada *app:layout_constraint* [30], která umožňuje definovat pozici tlačítka na obrazovce pomocí vymezení vazeb. Tyto vazby lze umístit na jiný prvek nebo na okraje obrazovky. Díky tomu jsou jednotlivé komponenty vzájemně propojeny, což umožňuje jejich správné rozmístění i při spuštění aplikace na zařízeních s odlišnými rozměry displeje.

Další dvě strany slouží pro zobrazení čtených dat z Arduina. Dle požadavku jsou rozděleny na první až osmý kanál (první měřící karta) a devátý až šestnáctý kanál (druhá měřící karta). Je na každé z nich tedy realizováno 8 indikátorů přečtených dat pomocí *TextView* [31].

Ovládání aplikace bylo realizováno pomocí navigačního menu. Procházení jednotlivými stránkami bylo implementováno pomocí fragmentů [32], které umožňují opakované použití částí (stránek) uživatelského rozhraní. Tím jsou stránky rozděleny na bloky a vytvořen kontejner, který je obsahuje. Díky tomu je možné uživatele po kliknutí na prvek v navigačním menu přesměrovat na požadovanou stránku.

5.5.3 Implementace hlavní logiky aplikace

Logika aplikace je rozčleněna do jednotlivých fragmentů, které zajišťují specifickou funkcionalitu – od správy připojení přes Bluetooth až po zobrazení přijatých dat v uživatelském rozhraní. Komunikace mezi fragmenty a jádrem aplikace je řešena pomocí sdílené instance třídy *BluetoothService* (viz kapitola 5.5.4), která je inicializována

při spuštění aplikace v rámci *MainActivity* [33]. Tímto způsobem je zajištěna konzistentní správa připojení a datového toku napříč celou aplikací.

Aplikace byla navržena jako multi-fragmentová s využitím *Navigation Component* [34]. V hlavní aktivitě (*MainActivity*) je zajištěn přístup ke sdíleným komponentám jako je třída pro ovládání Bluetooth a inicializace navigace včetně šuplíkového menu (*DrawerLayout* [35]). Struktura fragmentů byla použita následovně:

- *HomeFragment* – Hlavní stránka aplikace, kde uživatel může ovládat Bluetooth spojení a čtení dat.
- *ChannelsFirstFragment* – Zobrazuje prvních 8 měřících kanálů. Hodnoty jsou aktualizovány na základě přijatých dat přes Bluetooth.
- *ChannelsSecondFragment* – Obdobná funkce jako u fragmentu *ChannelsFirst*, ale zobrazuje zbylých 8 měřících kanálů.

Ve fragmentu *HomeFragment*, který slouží jako úvodní obrazovka aplikace, je implementován *ViewModel* [36] pro uchování stavu tlačítek a textového popisku připojení. *ViewModel* zde umožňuje přetrvávání UI stavu i při změnách rozložení nebo přechodu mezi fragmenty. Tlačítka v tomto fragmentu umožňují následující akce:

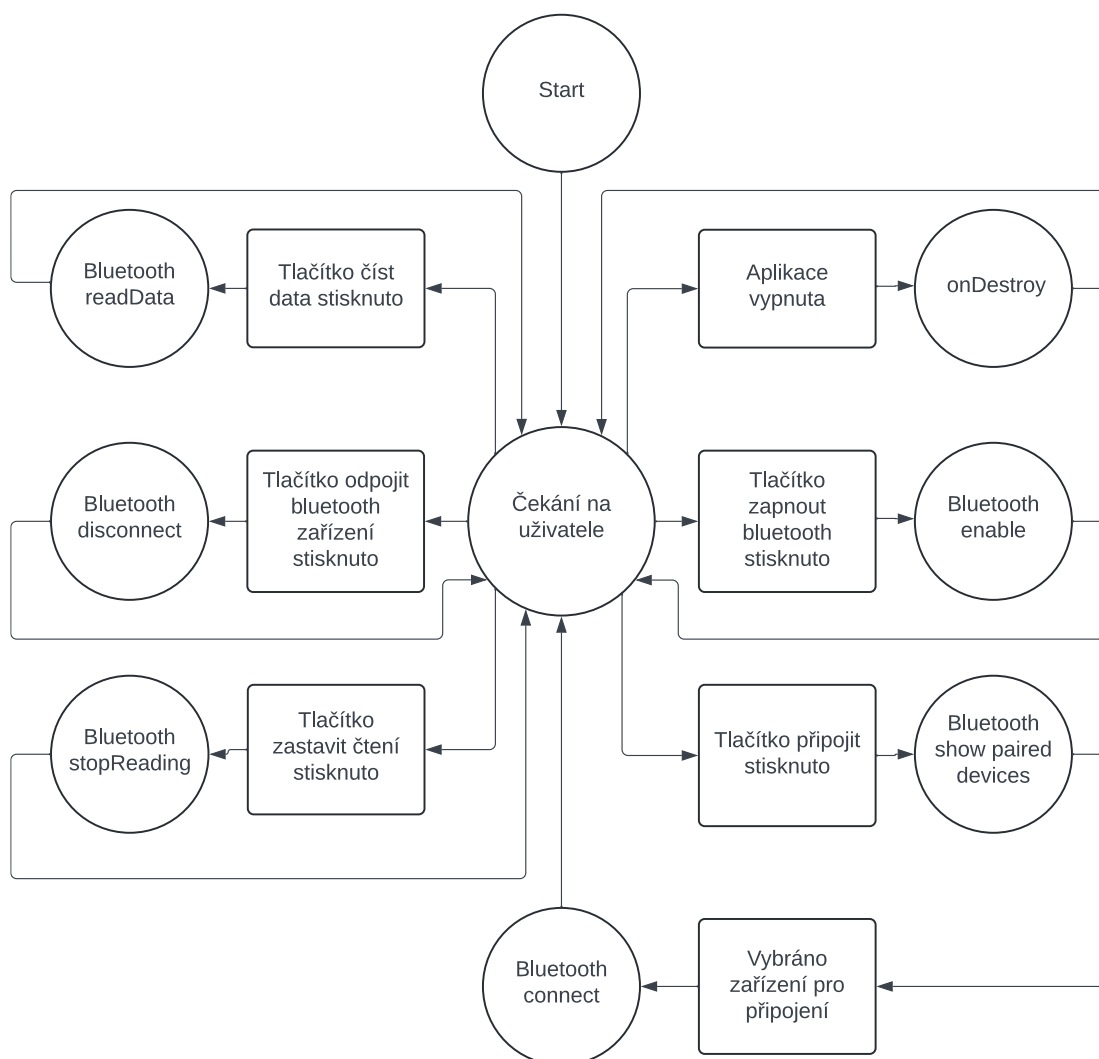
- Zobrazit seznam spárovaných zařízení a vybrat jedno z nich k připojení.
- Spustit nebo zastavit čtení dat z připojeného zařízení.
- Ručně odpojit aktuálně připojené bezdrátové zařízení.

Na domovské obrazovce byla také implementována kontrola a požadavky spojené s užíváním Bluetooth v aplikaci. Aplikace požádá uživatele o povolení Bluetooth používat a ověří, zda zařízení funkcionalitu bezdrátové technologie podporuje.

U fragmentů *ChannelsFirstFragment* a *ChannelsSecondFragment*, které slouží pro zobrazení přijatých dat, není *ViewModel* využíván. Tyto fragmenty přímo reagují na změny dat přijatých z Bluetooth pomocí *LiveData* [37], které je vystaveno třídou *BluetoothService*. Data se aktualizují hned po přijetí celé nové zprávy, tudíž nebylo nutné uchovávat stav fragmentu pomocí *ViewModelu*. Pokud jsou přijatá data v neočekávaném formátu, fragment zůstane nezměněn a zachová poslední platná data.

V případě úspěšného/neúspěšného spojení, či odmítnutí povolení aplikace vykreslí uživateli na obrazovku oznamovací zprávu o stavu. Současně stav připojení je zobrazen v *HomeFragmentu*, kde se dynamicky mění text na základě aktuálního stavu (například „Připojeno“). Textové pole se aktualizuje pomocí příslušné volané metody (*connect()*).

Stavový diagram znázorňuje přechody mezi jednotlivými stavy aplikace během připojování a čtení dat přes Bluetooth.



Obrázek 17 Stavový diagram aplikace na Android

Navržená architektura umožňuje snadné rozšíření aplikace o další fragmenty či datové kanály bez nutnosti zásahu do stávajícího jádra aplikace.

5.5.4 Realizace Bluetooth spojení a čtení dat

Bluetooth komunikace v rámci Android aplikace byla realizována prostřednictvím vlastní třídy *BluetoothService*, která zajišťuje kompletní obsluhu spojení se zařízením HC-05.

V prvním kroku realizace bylo potřeba uvést do *Manifest* souboru příslušná povolení, která jsou důležitá při použití Bluetooth rozhraní [38]. Povolení jsou:

```

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
  
```

```
android:usesPermissionFlags="neverForLocation"/>
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT"
/>
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE"
/>
```

V *MainActivity* je během *onCreate()* procesu (spuštění aplikace) vytvořena jediná instance třídy *BluetoothService*, která je následně sdílena napříč jednotlivými fragmenty. Uživateli umožňuje pomocí tlačítka vybrat spárované Bluetooth zařízení, ke kterému se aplikace následně pokusí připojit. Zároveň může uživatel spustit nebo zastavit čtení dat či odpojit se od připojeného zařízení. Stisknutím příslušného tlačítka je volána metoda pro uskutečnění požadované operace. Například pro připojení k bezdrátovému zařízení je volána metoda *bluetoothService.connect()* s argumentem *device*. Připojení probíhá asynchronně pomocí stavových proměnných a uživatel je informován o stavu připojení prostřednictvím UI. Samotné spojení je navázáno pomocí RFCOMM soketu, který je vytvořen metodou *createRfcommSocketToServiceRecord(uuid)*, přičemž proměnná *uuid* je definována jako *UUID.fromString("00001101-0000-1000-8000-00805F9B34FB")*, což odpovídá standardnímu sériovému profilu Bluetooth (SPP – Serial Port Profile) [39].

Pro čtení dat byla implementována metoda *readData()*. Metoda spouští *Coroutine* na vlákne *Dispatchers.IO* [40], která čeká na data přicházející po *InputStream* [41]. Přijátá data jsou zpracovávána jako textová zpráva ukončená znakem nového řádku (*\n*). Zpráva je rozdělena pomocí středníku na jednotlivé hodnoty a následně rozdělena na dvě poloviny, první pro fragment *ChannelsFirstFragment* a druhá pro *ChannelsSecondFragment*.

Během vývoje se objevila chyba, při které se hodnotám posílaným z LabVIEW odřezávala desetinná část. Tento problém byl způsoben rozdílným formátem desetinných čísel – zatímco LabVIEW používá desetinnou čárku, Android očekává desetinnou tečku. Řešení spočívalo v nastavení funkce, která v aplikaci LabVIEW převádí číslo na textový řetězec – konkrétně v nastavení konstanty na jeden ze vstupních pinů daného bloku.

Odpojení od bezdrátového zařízení je při *onDestroy()* (zavření aplikace) automaticky voláno. Uživatel má ale možnost stiskem tlačítka odpojit manuálně. V tomto případě je volána metoda *disconnect()*.

Tímto způsobem bylo zajištěno spolehlivé navázání Bluetooth spojení, příjem a rozdělení dat mezi UI komponenty. Celý proces probíhá asynchronně, čímž je zajištěna plynulost uživatelského rozhraní bez blokace hlavního vlákna aplikace.

5.5.5 Správa chyb

Správa chyb a výjimek byla implementována ve všech klíčových částech aplikace, které pracují s Bluetooth spojením, vstupně-výstupními operacemi a čtením dat. Výjimky jsou zachytávány pomocí *try-catch* [42] bloků a uživatel je informován prostřednictvím *Toast* zpráv přímo v uživatelském rozhraní [43].

Například při pokusu o připojení bezdrátového zařízení může dojít k výjimkám *IOException* (zařízení není dostupné) nebo *SecurityException* (aplikace nemá oprávnění) [42].

Ještě před jakoukoliv Bluetooth operací aplikace ověřuje, zda má všechna potřebná oprávnění (viz kapitola 5.5.3). Pokud oprávnění chybí, uživatel je vyzván k jejich udělení pomocí systémového dialogu.

Pokud při čtení dat dojde k výpadku spojení nebo uzavření datového toku, je vyvolána výjimka *IOException*. Uživatel je informován o výpadku a má možnost navázat spojení znovu stisknutím příslušného tlačítka.

Také při ručním odpojení může dojít k výjimce – např. při pokusu uzavřít již neplatný socket. I tato situace je ošetřena pomocí *try-catch* bloků.

5.5.6 Testování aplikace

Za účelem ověření správné funkčnosti aplikace byly ručně testovány následující stavy, které by mohly nastat při běžném používání. Aplikace byla již od počátku vývoje vybavena správou výjimek, proto u několika testovaných scénářů nebyl nutný zásah do funkcionality. Následující testy popisují jednotlivé scénáře včetně případných problémů a jejich řešení:

1. Ztráta připojení při čtení dat

Problém: Po manuálním vypnutí zařízení HC-05 během procesu čtení dat došlo ke ztrátě spojení. Aplikace na tuto událost nereagovala žádnou chybovou hláškou. Stav připojení zůstal nezměněn a pro opětovné navázání spojení bylo nutné nejprve ručně provést odpojení pomocí tlačítka „Odpojit“. Poté bylo možné zařízení znovu připojit.

Řešení: V metodě *readData()* ve třídě *BluetoothService* byl rozšířen *try-catch* blok o detekci výjimky *IOException*, která značí ztrátu spojení. V případě výskytu této chyby je pomocí *LiveData* proměnné *_connectionLost* změněn stav, který sleduje *HomeFragment*. Ten následně zobrazí uživateli oznamovací zprávu (*Toast*) informující o přerušení spojení. Tím je zajištěno, že je uživatel okamžitě upozorněn, aniž by musel ručně diagnostikovat stav spojení.

Závěr: Aplikace nyní automaticky detekuje ztrátu spojení s Bluetooth zařízením a informuje uživatele pomocí viditelné zprávy. Uživatel tak může ihned reagovat a provést opětovné připojení bez nutnosti ručního odpojování, což zlepšuje použitelnost aplikace v reálném provozu.

2. Návrat z minimalizace aplikace

Závěr: Po navrácení z minimalizace aplikace se spojení i čtení dat obnovilo bez problému. Stavové proměnné zůstaly zachovány, není nutný zásah.

3. Zátěžové chování při dlouhodobém běhu

Závěr: Aplikace byla spuštěna po dobu několika hodin s aktivním přenosem dat. Nebylo zaznamenáno žádné chybové chování ani únik paměti. Tím byla potvrzena stabilita i při dlouhém provozu.

4. Změna orientace obrazovky při běhu aplikace

Problém: Při otočení obrazovky docházelo ke ztrátě aktivního Bluetooth spojení a přerušení čtení dat. Aktivita *MainActivity* byla při změně orientace znovu vytvořena, čímž došlo k obnovení všech komponent a ztrátě stavu spojení i stavu spuštěného přenosu.

Řešení: Do deklarace aktivity *MainActivity* v souboru *AndroidManifest.xml* byl přidán parametr `android:configChanges="orientation|screenSize"`. Tím bylo dosaženo toho, že při změně orientace se aktivita nerestartuje, ale pouze zavolá metoda `onConfigurationChanged()`, čímž zůstane zachován stav aplikace včetně připojení k Bluetooth zařízení.

Závěr: Po úpravě manifestu zůstává aplikace funkční i při otočení obrazovky. Bluetooth spojení zůstává aktivní a čtení dat pokračuje bez přerušení. Uživatel nemusí provádět žádnou akci – aplikace se přizpůsobí novému rozložení automaticky.

5. Neplatný vstup

Závěr: Aplikace kontroluje formát přijatých dat. Každá zpráva musí obsahovat přesně 16 hodnot oddělených středníkem, přičemž každá z těchto hodnot musí být platné reálné číslo ve formátu s desetinnou tečkou (například „4.12“). Pokud je zpráva poškozená, neúplná nebo obsahuje nečíselné hodnoty, je označena jako neplatná, není dále zpracována a uživatelské rozhraní zůstává beze změny. Uživatel je o tomto stavu informován vyskakovací zprávou. Tím je zajištěna stabilita a odolnost aplikace vůči chybným přenosům.

6. Neúspěšné navázání spojení

Závěr: V případě neúspěšného připojení bylo správně zobrazeno chybové hlášení formou Toast zprávy „Failed to connect to device!“.

7. Ztráta připojení mimo čtení dat

Problém: Pokud došlo k manuálnímu odpojení zařízení HC-05 v době, kdy nebylo aktivní čtení dat, aplikace tuto událost nezachytala. Stav připojení zůstal na první pohled nezměněn a uživatel nebyl informován, že spojení bylo přerušeno.

Řešení: Do aplikace bylo přidáno tzv. *watchdog* vlákno, které pravidelně (každé 2 sekundy) kontroluje funkčnost spojení. Ověřuje se, zda je možné odeslat testovací bajt pomocí `socket?.outputStream?.write(0)`. V případě chyby (*IOException*) je uživatel upozorněn pomocí zprávy a aplikace aktualizuje stav připojení. Frekvence kontroly byla

nastavena na každé dvě sekundy, aby nedocházelo ke zbytečnému zatěžování systému, ale zároveň byla zaručena včasná reakce.

Závěr: Uživatel byl nově informován o ztrátě připojení i mimo aktivní čtení dat. Tím byla zvýšena stabilita aplikace a její schopnost reagovat na neočekávané výpadky spojení, což přispívá k lepší uživatelské zkušenosti.

8. Vícenásobné nebo náhodné klikání na tlačítka

Problém: Při opakovaném stisknutí tlačítka „Připojit“ během navazování spojení docházelo k nestandardnímu chování. Například po druhém stisku tlačítka se připojené zařízení odpojilo, případně se pokus o připojení provedl vícekrát souběžně. Stejný problém se mohl vyskytnout i u tlačítek „Číst data“ a „Odpojit“, pokud byly stisknuty opakovaně během probíhající akce.

Řešení: Pro všechna kritická tlačítka („Připojit“, „Číst data“, „Odpojit“) byla implementována logika blokace uživatelského vstupu během asynchronních operací. Tlačítka jsou při spuštění akce dočasně deaktivována (*isEnabled = false*) a po dokončení operace opět povolena (*isEnabled = true*). Pro zajištění konzistence i při změně orientace nebo návratu do fragmentu byla do třídy *BluetoothService* přidána proměnná *isReading*, která uchovává informaci o aktivním čtení, a stavová proměnná *isConnected*, která informuje o aktivním připojení. Fragment *HomeFragment* tyto informace sleduje a podle nich dynamicky upravuje dostupnost tlačítek.

Tlačítko „Číst data“ se po stisknutí deaktivuje, aby se zabránilo jeho opakovanému stisknutí. Po dokončení čtení se tlačítko opět povolí. Podobný mechanismus byl následně aplikován i na ostatní ovládací prvky (tlačítko „Připojit“ a „Odpojit“).

Závěr: Aplikace nyní správně reaguje i při rychlém nebo opakovaném stisknutí ovládacích prvků. Uživatelské rozhraní vždy reflektuje aktuální stav (například probíhající připojování nebo čtení dat) a nedochází k souběžnému spouštění stejných operací. Tím je zajištěna vyšší robustnost a spolehlivost aplikace.

9. Změna jazyka během běhu aplikace

Problém: Při přepnutí jazyka došlo ke standardnímu restartu aktivity. Pokud bylo v daný moment navázáno Bluetooth spojení nebo probíhalo čtení dat, aplikace o tom po restartu „nevěděla“. Zařízení zůstalo fyzicky připojeno, ale aplikace nadále neprováděla čtení a uživatel nebyl upozorněn, že musí operaci obnovit ručně.

Řešení: Byla přidána kontrola stavu připojení a čtení dat přímo před změnou jazyka. Pokud je zařízení připojeno nebo je aktivní čtení, zobrazí se uživateli upozornění a změna jazyka se neprovede. Tím se předejde nečekanému přerušení spojení.

Závěr: Při pokusu o změnu jazyka během připojení je uživatel informován a změna se neprovede.

10. Změna motivu během běhu aplikace

Problém: Přepnutí motivu (světlý/tmavý), stejně jako u jazyka, způsobovalo restart celé aktivity. Tím došlo ke ztrátě aktuálního stavu spojení a čtení dat bez upozornění uživatele.

Řešení: Před změnou motivu byla rovněž implementována kontrola stavu. Pokud je zařízení připojeno nebo probíhá čtení dat, změna motivu se neprovede a aplikace uživatele informuje o nutnosti nejprve spojení ukončit.

Závěr: Uživatel je nyní upozorněn na nemožnost změnit motiv během aktivního spojení. Funkce tak respektuje aktuální stav připojení a zabraňuje neočekávaným výpadkům.

Na základě provedeného testování bylo zjištěno, že se aplikace ve všech simulovaných situacích chovala stabilně a předvídatelně. Výjimky byly korektně ošetřeny, stavové proměnné zajišťovaly ochranu před nežádoucím chováním a uživatel byl ve všech případech informován o aktuálním stavu aplikace. Implementovaná opatření přispěla k vyšší robustnosti a celkové spolehlivosti systému v běžném provozu.

5.5.7 Doplnkové funkce

Byly implementovány dvě funkce, které nebyly požadovány zadavatelem, ale přispívají ke zlepšení uživatelského komfortu. Jedná se o:

- Možnost přepínání světlého/tmavého motivu – umožňuje uživateli zvolit si vzhled aplikace dle osobních preferencí nebo aktuálních světelných podmínek.
- Možnost přepínání mezi českým/anglickým jazykem – přizpůsobuje rozhraní potřebám zahraničních uživatelů a eliminuje případnou jazykovou bariéru, jelikož je zadavatelem mezinárodní firma. Byla zvážena i možnost čistě anglického jazyka, ale zadavatel práce preferoval implementaci dvoujazyčného rozhraní.

Obě tyto funkce byly implementovány pomocí třídy „*PreferenceManager*“ [44], která zajišťuje správu nastavení v rámci aplikace. Nastavení se ukládají do „*SharedPreferences*“ [45], což je úložiště dostupné v rámci Android API [46]. Vybraný motiv i jazyk je trvale uložen a načítá se při každém spuštění aplikace.

Možnost změny je dostupná po kliknutí na rozbalovací menu v pravém horním rohu aplikace, která je dostupná v každé stránce aplikace. Uživatel přepnutí dokáže zmáčknutím příslušného tlačítka ve vyskakovacím menu. Stiskem jednoho z tlačítek je zavolaná příslušná metoda.

a) Přepínání **motiv**

Po kliknutí na tlačítko „*Světlý motiv*“ je volána metoda, která nejprve ověří aktuálně uložené nastavení motivu. Pokud je jiný než zvolený, dojde k jeho uložení pomocí „*PreferenceManager.setTheme(...)*“ „*AppCompatDelegate.setDefaultNightMode(...)*“ je následně volána s parametrem „*MODE_NIGHT_NO*“, čímž se přepne aplikace do

světlého režimu. Nakonec se aktivita znovu načte pomocí *recreate()*, aby se změny okamžitě projeví. Stejný princip platí i pro tmavý režim („*MODE_NIGHT_YES*“).

b) Přepínání jazyka

Po kliknutí na tlačítko „*Čeština*“ nebo „*Angličtina*“ se zavolá metoda „*PreferenceManager.setLanguage(...)*“, která uloží jazyk do „*SharedPreferences*“. Následně je volána metoda „*recreateWithNewLocale(...)*“, která nastaví nové „*Locale*“ (objekt obsahující oba jazyky), upraví konfiguraci aplikace a provede opětovné vytvoření aktivity tak, aby se jazyková změna projevila ihned bez nutnosti restartu celé aplikace.

5.6 Odezva systému

Pro orientační zhodnocení celkové odezvy systému mezi odesláním dat z prostředí LabVIEW a jejich zobrazením na Android zařízení byl pořízen videozáznam, na kterém byla synchronně snímána obrazovka PC (LabVIEW) a tablet s běžící Android aplikací. Na záznamu bylo následně ručně odečteno několik časových okamžiků, ve kterých došlo k vizuální změně zobrazovaných dat na obou zařízeních.

Tabulka 5 Měření odezvy systému

Číslo měření	Čas na PC [s]	Čas na tabletu [s]	Odezva [ms]
1	5,19	5,38	190
2	13,13	13,37	240
3	18,66	18,85	190
4	21,66	21,87	210
5	27,14	27,39	250

Na základě těchto měření lze odhadnout průměrnou odezvu systému přibližně na 216 ms. Tato hodnota zahrnuje nejen samotný přenos dat z LabVIEW přes sériovou linku do Arduina, ale také jejich zpracování a přenos přes Bluetooth do Android zařízení a následné zobrazení na uživatelském rozhraní.

Je však třeba upozornit na možné chyby měření, které plynou zejména z:

- Nepřesnosti ručního odečtu z videozáznamu (časový krok záznamu nebo subjektivní odhad okamžiku změny).
- Zpoždění způsobené rozdílnou obnovovací frekvencí displejů.
- Případné prodlevy v samotném Android systému při zpracování přijatých dat (zanepřázdněnost).
- Nastavení měřících parametrů v LabVIEW, konkrétně vzorkovací frekvence 1000 Hz a počtu vzorků 100.

Vzhledem k těmto skutečnostem je třeba výsledný čas odezvy považovat za přibližný, nicméně dostačující pro posouzení celkové odezvy systému v rámci tohoto řešení.

6. POPIS VÝSLEDNÉHO PRODUKTU

Výsledkem práce je kompletně navržený a realizovaný systém, který umožňuje bezdrátový přenos dat mezi počítačem, mikrokontrolerem a mobilní aplikací. Hlavním cílem bylo zprostředkování přenosu dat z optického převodníku do prostředí Android prostřednictvím Bluetooth technologie. Systém se skládá ze tří hlavních komponent – desktopové aplikace v prostředí LabVIEW, mikrokontroleru Arduino MEGA s Bluetooth modulem HC-05 a Android aplikace vytvořené v Android Studiu. V této kapitole jsou jednotlivé části systému detailně popsány z pohledu výsledné implementace, včetně způsobu komunikace a interakce mezi nimi.

Výsledný systém je tvořen třemi hlavními komponentami:

- Android aplikace – vyvinuta v jazyce Kotlin v prostředí Android Studio. Zajišťuje uživatelské rozhraní, připojení k Bluetooth zařízení a zobrazení přijatých dat.
- Arduino MEGA s modulem HC-05 – přijímá data z PC přes sériovou linku a dále je odesílá pomocí Bluetooth do mobilní aplikace.
- LabVIEW aplikace – snímá měřená data a odesílá je do Arduina přes USB kabel.

Komunikace probíhá v následujícím směru: LabVIEW → Arduino → HC-05 → Android aplikace.

6.1 Funkčnost aplikací

LabVIEW aplikace slouží pro sběr, zpracování a vizualizaci měřených dat z DAQ karet a umožňuje:

- Vizualizaci dat prostřednictvím analogových ukazatelů a grafu pro každý z měřených kanálů.
- Výpočet limitních hodnot, a to na základě zvoleného procentuálního rozsahu, který je definován uživatelem.
- Konfiguraci měřicích karet, včetně výběru vstupního zařízení, typu zapojení (např. RSE) a připojení přes specifikovaný COM port.
- Přepnutí do režimu přenosu přes Arduino, včetně aktivace sériového výstupu.
- Spuštění a zastavení měření pomocí tlačítek, která jsou dynamicky aktivována dle aktuálního stavu aplikace.

Android aplikace, vytvořená v prostředí Android Studio, nabízí uživateli následující funkce:

- Výběr a připojení ke spárovanému Bluetooth zařízení typu HC-05 prostřednictvím seznamu dostupných zařízení.

- Manuální spuštění a zastavení čtení dat, které jsou přijímány přes sériové Bluetooth spojení.
- Zobrazení aktuálních hodnot ve dvou oddělených blocích po osmi kanálech.
- Možnost přizpůsobení vzhledu aplikace, včetně přepnutí mezi světlým a tmavým režimem a volby jazyka (čeština / angličtina).
- Zobrazení oznamovacích zpráv formou Toast hlášek při událostech jako je ztráta spojení, neplatný vstup nebo neúspěšné připojení.

Android aplikace přijme zprávu jako jeden řetězec, rozdělí ji na jednotlivé hodnoty a aktualizuje zobrazení v reálném čase. Proces čtení je vždy řízen uživatelem a je chráněn proti vícenásobnému spouštění.

Obě aplikace zároveň implementují ochranu proti vícenásobnému stisku ovládacích tlačítek během probíhajících asynchronních operací, čímž je minimalizováno riziko chybného chování. Veškeré chyby během připojení, přenosu nebo výpočtu jsou zpracovány pomocí výjimek, které jsou uživatelsky srozumitelně ohlášeny.

Celý systém je navržen tak, aby data proudila kontinuálně a bez výrazného zpoždění. LabVIEW zajišťuje pravidelný výstup naměřených hodnot do sériové linky, kterou Arduino MEGA okamžitě přijímá. Po zformátování dat do jednotného řetězce jsou tyto informace přes Bluetooth modul HC-05 odeslány do mobilní aplikace. Android aplikace následně zprávu rozdělí a zobrazí jednotlivé hodnoty téměř okamžitě po jejich přijetí.

Zpoždění celého řetězce od změření hodnot v LabVIEW až po jejich zobrazení v Android aplikaci je přibližně 220 ms (viz kapitola 5.6). Přesná hodnota závisí na použitém zařízení, vytížení systému a síle Bluetooth signálu. Zpoždění nebylo cíleně minimalizováno, ale v kontextu použití je přijatelné a nezpůsobuje funkční problémy.

6.2 Uživatelské ovládání

Uživatelské ovládání celého systému bylo navrženo s důrazem na jednoduchost, přehlednost a minimalizaci chyb způsobených nesprávným použitím. Veškeré funkce byly zpřístupněny přímo z hlavních obrazovek obou aplikací, čímž bylo dosaženo intuitivního ovládání bez nutnosti hlubších technických znalostí.

V případě desktopové aplikace LabVIEW bylo rozhraní rozděleno do několika logických sekcí. V levé části se nachází konfigurace měřicích karet a volba režimu přenosu dat (přímý přenos vs. přenos přes Arduino), zatímco pravá část je vyhrazena pro vizualizaci měřených hodnot a ovládání samotného měření. Interaktivní prvky (např. tlačítka „Start“, „Stop“) byly dynamicky aktivovány nebo deaktivovány v závislosti na aktuálním stavu aplikace, čímž bylo zajištěno konzistentní a bezpečné ovládání.

Mobilní aplikace pro Android byla vytvořena jako jednoduchý a přehledný nástroj pro zobrazení přijatých dat. Uživatel má možnost vybrat zařízení ze seznamu spárovaných Bluetooth modulů, ručně spustit a zastavit čtení dat a sledovat aktuální

hodnoty rozdělené do dvou bloků. Ovládací prvky byly doplněny o vizuální a textovou odezvu, včetně oznamovacích zpráv (např. o ztrátě spojení nebo chybném vstupu). Dále byla zpřístupněna možnost přepnutí motivu (světlý/tmavý) a změna jazyka, přičemž všechny volby byly snadno dostupné přímo z vyskakovacího z menu.

Pro použití Android aplikace je nutné stáhnout instalační soubor ve formátu „.apk“. Tento soubor byl vygenerován v prostředí Android Studio pomocí nástroje Build APK, který převádí zdrojový kód do podoby vhodné pro instalaci na zařízení s operačním systémem Android.

6.3 Shrnutí

System byl navržen tak, aby splňoval požadavek bezdrátového přenosu měřených dat do mobilního zařízení a zároveň poskytoval přehledné a bezpečné ovládání. Všechny komponenty spolupracují spolehlivě, chyby jsou zachytávány a uživatel je o nich vhodně informován. Celková architektura je dostatečně flexibilní pro budoucí rozšíření – například o ukládání dat, přenos zpětné vazby nebo nasazení do jiného typu měřicí aplikace.

Původně měl USB kabel délku 10 metrů, avšak po přemístění zařízení do přední části komory postačoval kabel o délce 5 metrů. Ke změně umístění došlo z důvodu silného Bluetooth signálu, a proto nebylo nutné zařízení instalovat v bezprostřední blízkosti komory. Uchycení krabičky k nástěnné tabuli bylo realizováno pomocí lepicí plastelíny, ale je možné ji připevnit také pomocí malých hřebíků – pro tento účel má krabička otvory na zadní straně.



Obrázek 18 Umístění krabičky

7. ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a realizovat systém, který umožní bezdrátový přenos dat z EMC komory do mobilního zařízení obsluhy. Hlavní motivací byla snaha zefektivnit práci operátora, který jinak musí opakovaně přecházet mezi komorou a ovládacím místem. Výsledné řešení přináší nejen zvýšení komfortu, ale také úsporu času během měřicích postupů.

S ohledem na prostředí a požadavky firmy Forvia byla po testování vybrána technologie Bluetooth, konkrétně modul HC-05. Ten byl použit v kombinaci s mikrokontrolérem Arduino, který zajišťuje přenos dat přijatých ze systému LabVIEW do Android aplikace. Každá část systému byla navržena samostatně, ale navzájem propojeně a s cílem zajistit kompatibilitu.

Výsledný systém byl podroben testování v reálných podmínkách. Během testování se vyskytlo několik problémů, například ztráta Bluetooth spojení při přenosu dat, neočekávané chování při změně orientace obrazovky nebo problémy se zahájením sériové komunikace Arduina. Všechny tyto situace byly ošetřeny vhodnými softwarovými úpravami, včetně systémového zachování stavu aplikace, ověření oprávnění nebo blokace tlačítek během probíhajících operací a jsou popsány v textu práce. Zároveň byl systém implementován do laboratoře ve firmě.

Za přínos řešení lze považovat zejména jeho modularitu, jednoduchost nasazení bez zásahu do stávající infrastruktury, nízké náklady a intuitivní ovládání. Nevýhodou je částečně omezený dosah Bluetooth přenosu, který může být ovlivněn specifickými podmínkami na pracovišti, a vyšší nároky na ošetření nestandardních situací na straně Android aplikace.

Do budoucna je možné doplnit další funkcionalitu, například archivaci a analýzu dat, rozšíření Android aplikace o grafické zobrazení dat nebo zpětnou komunikaci směrem k systému LabVIEW. Díky zvolenému řešení lze celý systém přizpůsobit i jiným měřicím úlohám nebo podobným typům pracovišť.

Tímto byly naplněny všechny cíle bakalářské práce.

LITERATURA

- [1] *Electromagnetic Compatibility (EMC) Testing*. Online. Dostupné z: <https://www.tuvsud.com/en/services/testing/electromagnetic-compatibility-testing>. [cit. 2023-11-14].
- [2] *What is an EMC or Anechoic Chamber?* Online. 2021. Dostupné z: <https://www.etssolution-asia.com/blog/what-is-an-emc-or-anechoic-chamber-and-how-is-working>. [cit. 2023-11-14].
- [3] *ACTC-CISPR 25 Automotive Component Testing Chamber*. Online. 2023. Dostupné z: <https://frankonia-solutions.com/portfolio/actc/>. [cit. 2023-11-14].
- [4] *USB-6000*. Online. Dostupné z: https://www.ni.com/cs-cz/shop/model/usb-6000.html?srsId=AfmBOoqQ8COQQOvPjI2IuNUArBE_4_FyJheD5-C5IH2ZyA9F-OC8uKxY. [cit. 2023-11-14].
- [5] BIDEQUIR, Joe. *Bluetooth 4.0: Low Energy*. Online. In: Intel.com. Dostupné z: https://www.intel.com/wp-content/uploads/2015/09/20160330-IEEE_BLE.pdf. [cit. 2023-11-14].
- [6] KAUSHIK, S. An overview of Technical aspect for WiFi Networks Technology. Online. International Journal of Electronics and Computer Science Engineering. 2012, roč. 1, č. 1, s. 28-32. Dostupné z: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1c012c9f2b67a3cb2bf299e48cfab6c0be6cbf50>. [cit. 2023-11-14].
- [7] INTEL CORPORATION. What Is Bluetooth® Technology? Online. Intel.com. 2023. Dostupné z: <https://www.intel.com/content/www/us/en/products/docs/wireless/what-is-bluetooth.html>. [cit. 2023-11-14].
- [8] Bluetooth Wireless Technology. Online. In: Bluetooth.com. 2023. Dostupné z: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>. [cit. 2023-11-14].
- [9] Internet of Things (IoT): A Literature Review. Online. Journal of Computer and Communications. 2015, roč. 3, č. 5, s. 1-3. Dostupné z: https://www.scirp.org/html/56616_56616.htm. [cit. 2023-11-14].
- [10] GRATTON, Dean A. Bluetooth profiles The Definitive Guide. Online. PENTICE HALL, 2003. Dostupné z: https://books.google.cz/books?hl=cs&lr=&id=08eByqhzJ3wC&oi=fnd&pg=PR11&dq=bluetooth++profiles&ots=v_bQJVxBOe&sig=Qlxzf3m-A9-ke_TbGFgA5zMUdGs&redir_esc=y#v=onepage&q&f=false. [cit. 2023-12-10].
- [11] Device Identification Profile. Online. Bluetooth.com. C2023. Dostupné z: <https://www.bluetooth.com/specifications/specs/device-id-profile-1-3/>. [cit. 2023-12-10].
- [12] *Get Started with MIT App Inventor*. Online. Dostupné z: <https://appinventor.mit.edu/explore/get-started>. [cit. 2023-12-10].

- [13] *Android Studio*. Online. Dostupné z: <https://developer.android.com/studio>. [cit. 2023-12-10].
- [14] *Downloads*. Online. Dostupné z: <https://www.arduino.cc/en/software/>. [cit. 2023-12-10].
- [15] Communicate with Bluetooth Low Energy (BLE) Devices in LabVIEW. Online. Knowledge.ni.com. C2023. Dostupné z: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019LuHSAU&l=cs-CZ>. [cit. 2023-12-10].
- [16] *LabVIEW BLE (Bluetooth Low Energy) toolkit*. Online. In: NATIONAL INSTRUMENTS CORP. Community Documents. 07-28-2016, 11-29-2021. Dostupné z: <https://forums.ni.com/t5/Community-Documents/LabVIEW-BLE-Bluetooth-Low-Energy-toolkit/ta-p/3538612>. [cit. 2023-12-14].
- [17] HLAVÁČ, M. BEZDRÁTOVÝ SBĚR DAT Z BOSCH XDK V LABVIEW. Online, Bakalářská práce. Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2021. Dostupné z: <https://dspace.vut.cz/server/api/core/bitstreams/2597c8e3-bdc6-4b2a-8bb9-a82faa0b9f79/content>. [cit. 2023-12-14].
- [18] NI-VISA Overview. Online. NATIONAL INSTRUMENTS CORP.Ni.com. C2023. Dostupné z: <https://www.ni.com/en/support/documentation/supplemental/06/ni-visa-overview.html>. [cit. 2023-12-20].
- [19] *NI-DAQmx*. Online. Dostupné z: <https://www.ni.com/en/support/downloads/drivers/download.ni-daq-mx.html#494676>. [cit. 2023-12-20].
- [20] NATIONAL INSTRUMENTS CORP. What is NI LabVIEW? Online. C2023. Dostupné z: <https://www.ni.com/en/shop/labview.html>. [cit. 2023-12-20].
- [21] *Configuration Files*. Online. 2025-04-01. Dostupné z: https://www.ni.com/docs/en-US/bundle/labview/page/configuration-files.html?srsId=AfmBOorFc4LlrkB00by2RqwggtB1LVw57XByxxOl-5_fcrPJdxNuosx8. [cit. 2025-05-14].
- [22] *Tick Count (ms) Function*. Online. Dostupné z: <https://www.ni.com/docs/en-US/bundle/labview-api-ref/page/functions/tick-count-ms.html?srsId=AfmBOop6xJnK-PbFSUA9yjNIP4j-e4jb3ioDuWsC11khZm6g5THwcc5o>. [cit. 2025-05-14].
- [23] *Required serial delay after reset?* Online. 2016. Dostupné z: <https://forum.arduino.cc/t/required-serial-delay-after-reset/371457>. [cit. 2024-03-15].
- [24] *Getting Started with Arduino*. Online. 2024. Dostupné z: <https://docs.arduino.cc/learn/starting-guide/getting-started-arduino/>. [cit. 2024-04-11].
- [25] ECLIPSE S.R.O., CZECH IMPORTER. Bluetooth modul HC-05. Online. Dratek.cz. 2021. Dostupné z: <https://dratek.cz/docs/produkty/1/1401/1427822941.pdf>. [cit. 2024-04-11].

- [26] *Digital Pins*. Online. 2024. Dostupné
z: <https://docs.arduino.cc/learn/microcontrollers/digital-pins/>. [cit. 2024-04-11].
- [27] *Projects overview*. Online. 2024. Dostupné
z: <https://developer.android.com/studio/projects>. [cit. 2024-04-11].
- [28] *App manifest overview*. Online. 2024. Dostupné
z: <https://developer.android.com/guide/topics/manifest/manifest-intro>. [cit. 2024-04-11].
- [29] *Layouts in views*. Online. 2024. Dostupné
z: <https://developer.android.com/develop/ui/views/layout/declaring-layout>. [2024-04-11].
- [30] *ConstraintLayout*. Online. 2024. Dostupné
z: <https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>. [cit. 2024-04-11].
- [31] *TextView*. Online. 2024. Dostupné
z: <https://developer.android.com/reference/android/widget/TextView>. [cit. 2024-04-11].
- [32] *Fragments*. Online. 2024. Dostupné
z: <https://developer.android.com/guide/fragments>. [cit. 2024-04-11].
- [33] *Introduction to activities*. Online. 2024. Dostupné
z: <https://developer.android.com/guide/components/activities/intro-activities>. [cit. 2024-04-11].
- [34] *Navigation*. Online. 2024. Dostupné
z: <https://developer.android.com/guide/navigation>. [cit. 2024-04-11].
- [35] *DrawerLayout*. Online. 2024. Dostupné
z: <https://developer.android.com/reference/androidx/drawerlayout/widget/DrawerLayout>. [cit. 2024-04-11].
- [36] *ViewModel overview*. Online. 2024. Dostupné
z: <https://developer.android.com/topic/libraries/architecture/viewmodel>. [cit. 2024-12-15].
- [37] *LiveData overview*. Online. 2024. Dostupné
z: <https://developer.android.com/topic/libraries/architecture/livedata>. [cit. 2024-12-15].
- [38] *Bluetooth permissions*. Online. 2024. Dostupné
z: <https://developer.android.com/develop/connectivity/bluetooth/bt-permissions>. [cit. 2024-12-15].
- [39] *Connect Bluetooth devices*. Online. 2024. Dostupné
z: <https://developer.android.com/develop/connectivity/bluetooth/connect-bluetooth-devices>. [cit. 2024-12-15].
- [40] *Kotlin coroutines on Android*. Online. 2024. Dostupné
z: <https://developer.android.com/kotlin/coroutines>. [cit. 2024-12-15].
- [41] *InputStream*. Online. 2024. Dostupné
z: <https://developer.android.com/reference/java/io/InputStream>. [cit. 2024-12-15].

- [42] *Catch and handle exceptions*. Online. 2025. Dostupné
z: <https://developer.android.com/health-and-fitness/guides/health-connect/develop/exceptions>. [cit. 2025-02-08].
- [43] *Toast*. Online. 2025. Dostupné
z: <https://developer.android.com/reference/android/widget/Toast>. [cit. 2025-02-08].
- [44] *PreferenceManager*. Online. 2025. Dostupné
z: <https://developer.android.com/reference/android/preference/PreferenceManager>
. [cit. 2025-02-08].
- [45] *SharedPreferences*. Online. 2025. Dostupné
z: <https://developer.android.com/reference/android/content/SharedPreferences>.
[cit. 2025-02-08].
- [46] *Android API reference*. Online. 2025. Dostupné
z: <https://developer.android.com/reference>. [cit. 2025-02-08].

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

BT	Bluetooth
LE	Low energy
PC	Počítač
UI	Uživatelské prostředí
NI	National Instruments

PŘÍLOHA A – KRABÍČKA PRO ARDUINO

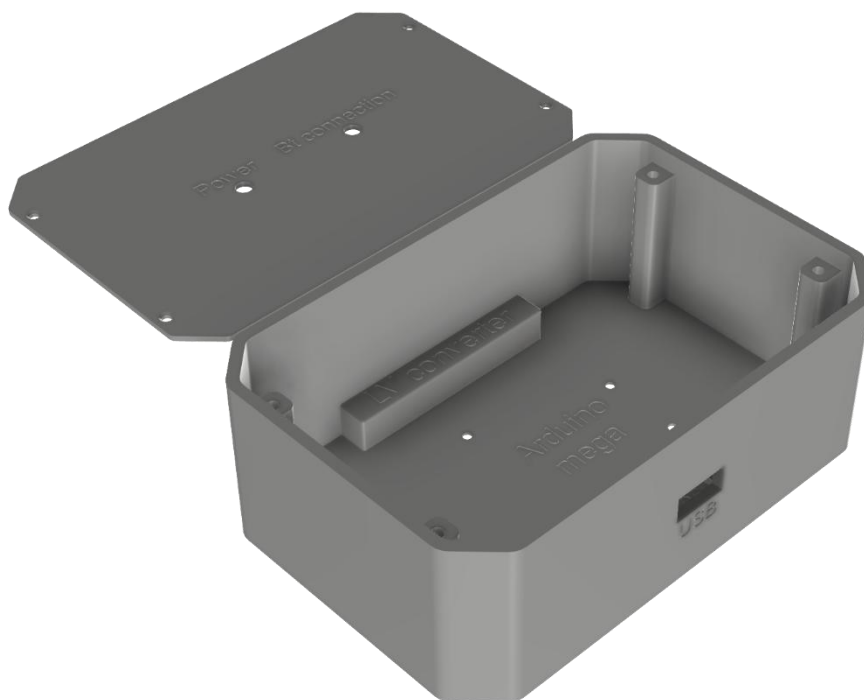
Pro praktickou část práce byla navržena a vytvořena vlastní krabička pro uložení elektroniky. Návrh byl vytvořen v programu Fusion 360. Při nákresu byl kladen důraz na rozměrovou přesnost s ohledem na rozmístění vnitřních komponent a snadný přístup k ovládacím prvkům a konektorům.

Model byl následně exportován ve formátu „3mf“ a zpracován v programu pro přípravu tisku Bambu Studio, kde byly nastaveny parametry tisku, jako je výška vrstvy, hustota výplně nebo využití podpor. Krabička poté byla vytisknuta pomocí technologie FDM (Fused Deposition Modeling) z materiálu PETG na 3D tiskárně Bambu Lab X1C Carbon.

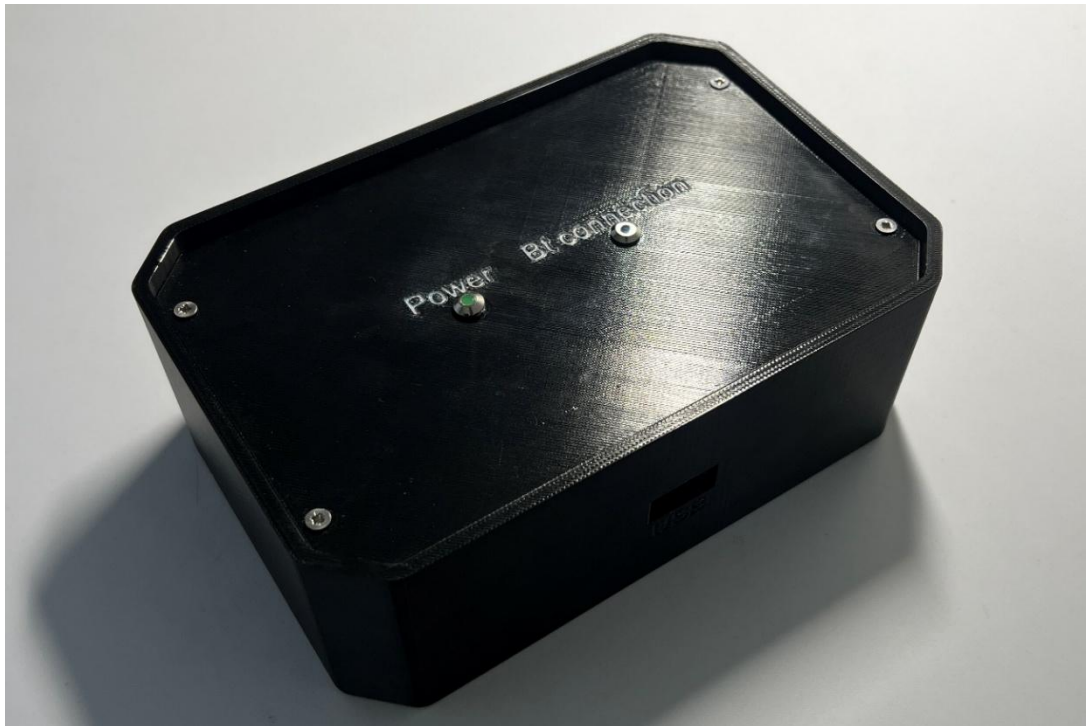
Výsledný produkt byl mechanicky očištěn od zbytků plastu (například z podpěr) a osazen elektronikou. Pro upevnění víka byly využity závitové vložky do plastu, které byly zataveny do předem připravených otvorů. Víko bylo upevněno pomocí čtyř šroubků o velikosti M3. Na víku jsou dále připevněny dvě LED diody, zajištěné maticemi.

Arduino je uchyceno pomocí plastových distančních sloupků, které jsou přišroubovány ze spodní strany krabičky. BT modul byl zasunut do zasouvacího doku, bez mechanického spojení. Pro připojení USB kabelu, kterým je mikrokontroler napájen a přes který přijímá data, byl ve spodní části krabičky vytvořen otvor.

Na zadní straně zařízení se nacházejí dva otvory určené pro zavěšení na hřebíky, což umožňuje jednoduchou fixaci na zeď. V době zpracování práce však bylo zařízení uchyceno pomocí lepicí plastelíny.



Obrázek 19 3D model krabičky – pohled z levého horního rohu



Obrázek 20 Krabička

PŘÍLOHA B – NÁVOD K ANDROID APLIKACI

Android aplikace byla navržena jako nástroj pro bezdrátový příjem dat z optického převodníku prostřednictvím mikrokontroleru Arduino MEGA s modulem HC-05. Umožňuje vizualizaci přijatých dat, základní interakci s Bluetooth zařízením a přizpůsobení uživatelského rozhraní.

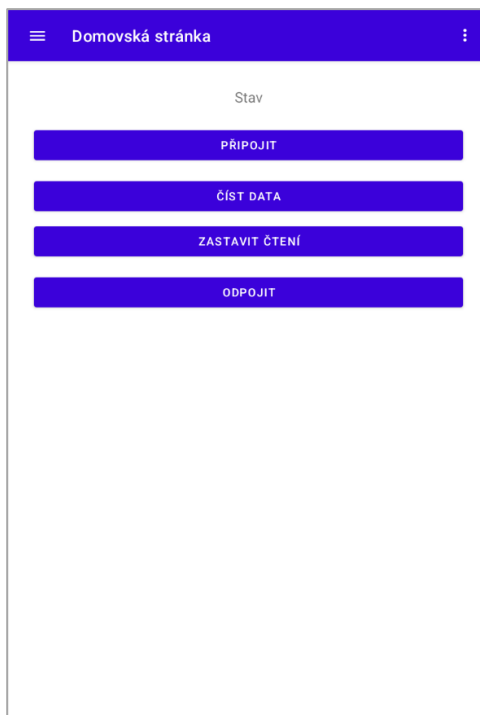
Systémové požadavky: Zařízení s operačním systémem Android 12.0 nebo vyšším a podporou Bluetooth rozhraní. Požadavky na připojení: Aplikace vyžaduje, aby bylo zařízení HC-05 již spárováno s mobilním zařízením, na kterém je aplikace spuštěna.

Instalace aplikace je možná následujícím způsobem:

1. Zkopírujte soubor *app-debug.apk* (vygenerovaný v prostředí Android Studio) do cílového zařízení s operačním systémem Android – například pomocí USB kabelu, e-mailu nebo cloudového úložiště.
2. Na zařízení otevřete přenesený soubor „.apk“.
Pokud se zobrazí upozornění, že instalace z neznámých zdrojů není povolena, přejděte do: Nastavení → Zabezpečení → Instalace neznámých aplikací a povolte instalaci.
3. Po otevření souboru „.apk“ potvrďte instalaci podle pokynů na obrazovce.
4. Po instalaci spusťte aplikaci, která bude připravena k použití.

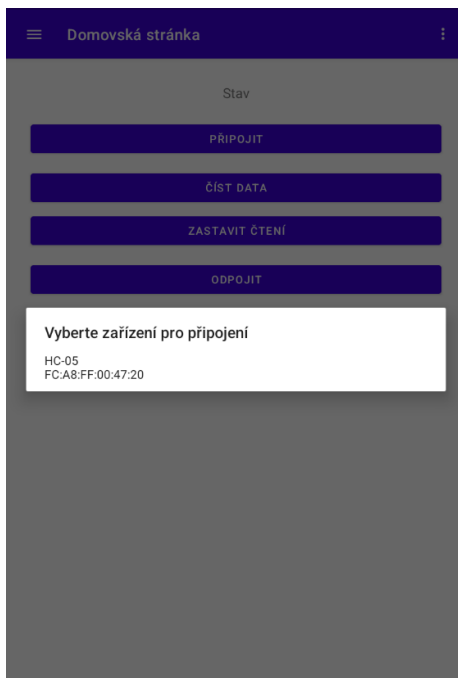
Základní ovládání aplikace:

1. Spuštění aplikace – Po otevření se zobrazí domovská obrazovka s hlavními ovládacími tlačítky.



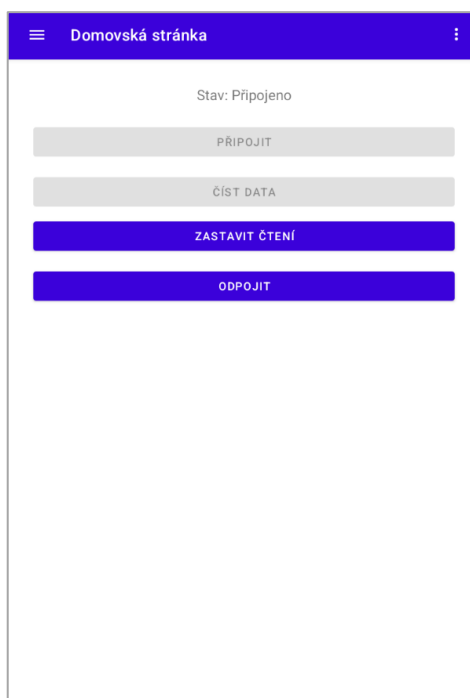
Obrázek 21 Snímek obrazovky - domovská stránka aplikace

2. Výběr Bluetooth zařízení – Stiskněte tlačítko „Připojit“. Zobrazí se seznam dostupných spárovaných zařízení. Po výběru zařízení HC-05 dojde k pokusu o připojení.



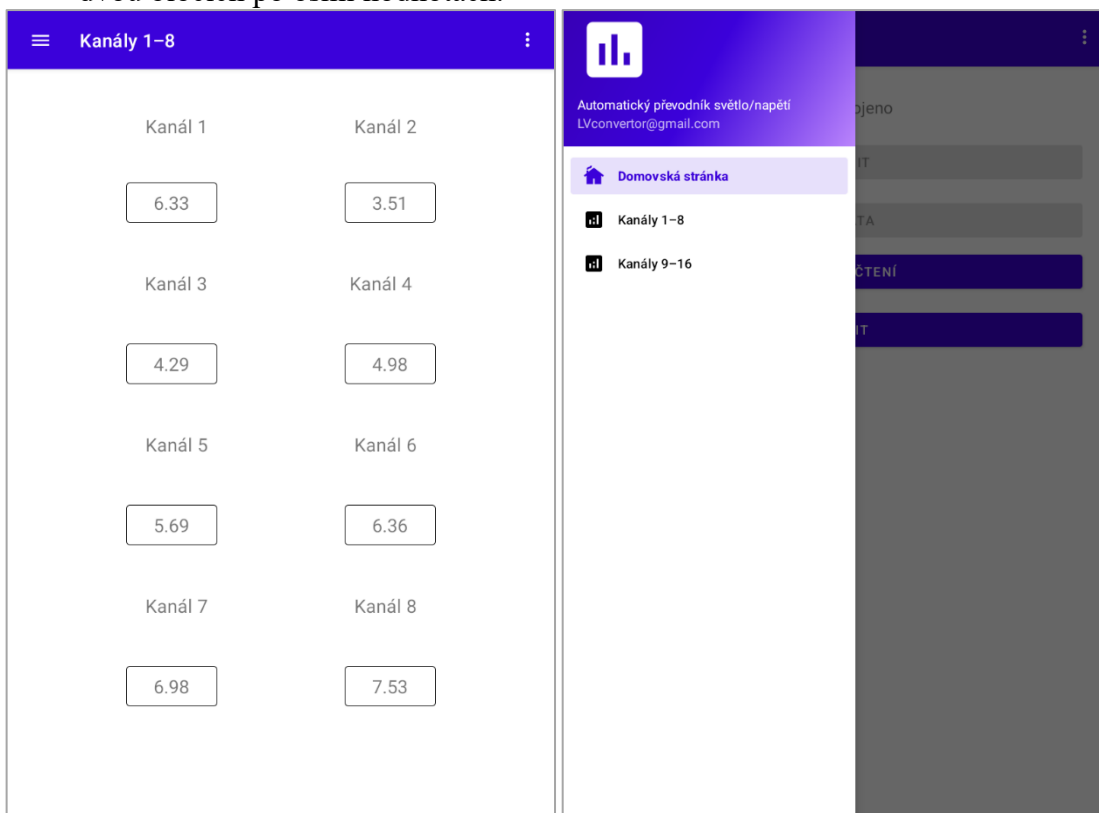
Obrázek 22 Snímek obrazovky - seznam spárovaných zařízení

3. Zahájení přenosu dat – Po úspěšném připojení je aktivováno tlačítko „Číst data“, kterým spustíte příjem dat. Příjem lze kdykoli ručně zastavit opětovným stiskem.



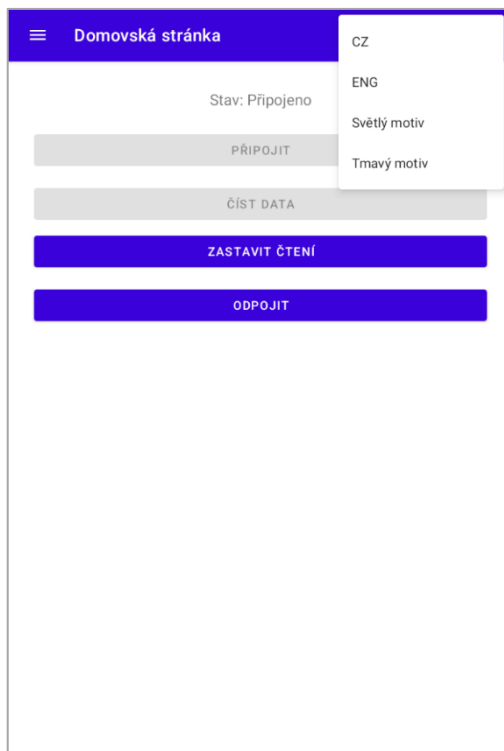
Obrázek 23 Snímek obrazovky – spuštěno čtení dat

4. Zobrazení přijatých dat – Data jsou automaticky zobrazována v reálném čase ve dvou blocích po osmi hodnotách.



Obrázek 24 Snímek obrazovky – zobrazení dat a hlavní menu

5. Odpojení zařízení – Pomocí tlačítka „Odpojit“ lze ukončit spojení.
6. Změna jazyka a motivu – V pravém horním rohu lze prostřednictvím nabídky přepnout jazyk aplikace (CZ/ENG) a vzhled (tmavý/světlý režim). Upozornění: změna těchto nastavení způsobí restart aktivity a přerušení spojení.



Obrázek 25 Snímek obrazovky – menu pro změnu jazyka nebo motivu