



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

Katedra biomedicínské techniky

Mobilní jednotka pro sběr biomedicínských dat v experimentálním prostředí

Mobile unit for collecting biomedical data in experimental environment

Diplomová práce

Studijní program: Biomedicínská a klinická technika

Studijní obor: Biomedicínský inženýr

Vedoucí práce: Ing. Leoš Tejkl

Bc. Daniela Kolíková

Kladno 2020



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kolíková** Jméno: **Daniela** Osobní číslo: **456597**
Fakulta: **Fakulta biomedicínského inženýrství**
Garantující katedra: **Katedra biomedicínské techniky**
Studijní program: **Biomedicínská a klinická technika**
Studijní obor: **Biomedicínský inženýr**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Mobilní jednotka pro sběr biomedicínských dat v experimentálním prostředí

Název diplomové práce anglicky:

Mobile unit for collecting biomedical data in experimental environment

Pokyny pro vypracování:

Navrhněte a zkonstruujte mobilní systém pro snímání analogového EKG, tlaku krve a signálů z dvouosého inklinometru v experimentálním a klinickém prostředí. Systém musí umožňovat sběr dat alespoň ze 4 analogových vstupů, využívat dostatečnou vzorkovací frekvenci pro biosignály, musí mít snadné a robustní ovládání a velikost interní baterie a paměťového média pro několikadenní záznamy. Vytvořený systém porovnejte s dílčími běžně používanými metodami záznamu a zhodnoťte kvalitu a spolehlivost zaznamenaných signálů.

Seznam doporučené literatury:

- [1] John G. Webster, Encyclopedia of Medical Devices and Instrumentation, ed. 6, Wiley, 2006, ISBN 978-0-471-26358-6
- [2] Walter Boron, Emile L. Boulpaep, Textbook of Medical Physiology, ed. 2nd, Elsevier, 2009, ISBN 978-1-4160-3115-4

Jméno a příjmení vedoucí(ho) diplomové práce:

Ing. Leoš Tejkl

Jméno a příjmení konzultanta(ky) diplomové práce:

MUDr. Mikuláš Mlček, Ph.D.

Datum zadání diplomové práce: **10.02.2020**

Platnost zadání diplomové práce: **19.09.2021**


prof. Ing. Peter Kneppo, DrSc., dr.h.c.
podpis vedoucí(ho) katedry


prof. MUDr. Ivan Dylevský, DrSc.
podpis oškana(ky)

PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci s názvem „Mobilní jednotka pro sběr biomedicínských dat v experimentálním prostředí“ vypracovala samostatně a použila k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně 14.5.2020

.....

Bc. Daniela Kolíková

PODĚKOVÁNÍ

Ráda bych touto cestou vyjádřila poděkování inženýrovi Leošovi Tejklovi za jeho cenné rady, doporučení a trpělivost při vedení mé diplomové práce. Taktéž bych chtěla poděkovat Mariánovi Grygovi za vstřícnost, ochotu a pomoc při tvorbě softwarové části práce.

ABSTRAKT

Mobilní jednotka pro sběr biomedicínských dat v experimentálním prostředí

Téma diplomové práce je zaměřeno na konstrukci systému pro sběr dat v experimentálním prostředí, který bude disponovat především kompaktností, robustním ovládním a mobilitou. První část práce je věnována rozboru trhu, který uvádí stávající zařízení pro sběr dat předními výrobci a dostupných komponent pro konstrukci mobilní jednotky. Z tohoto přehledu vyplývá potřeba konstrukce nového systému, který je oproti stávajícím systémům mobilní a disponuje vlastním softwarem pro ukládání dat. Pro samostatný sběr dat pomocí zkonstruované mobilní jednotky není potřeba dalšího hardwarového ani softwarového příslušenství. V druhé části je řešena realizace hardwaru, která umožňuje sběr dat ze čtyř analogových vstupů, disponuje vhodným AD převodníkem, interní baterií a paměťovým médiem. Základním prvkem hardwarové konstrukce je jednodeskový počítač Raspberry Pi, kdy ostatní komponenty jsou s tímto systémem plně kompatibilní. Robustní ovládním je zajištěno pomocí dotykového displeje v kombinaci s vyvinutou webovou aplikací. Vývojem softwarového řešení pomocí webové aplikace, která umožňuje robustní ovládním a ukládání dat, se zabývá třetí část diplomové práce. Poslední část práce je věnována zhodnocení kvality záznamu naměřeného zkonstruovanou mobilní jednotkou. Pro ověření funkčnosti a kvality zaznamenaných dat bylo provedeno nejdříve měření pomocí generátoru signálu, kdy byla ověřována funkčnost a přesnost zkonstruované mobilní jednotky. Druhá část měření se věnovala ověření sběru biomedicínských dat se stávajícím využívaným systémem pro sběr dat PowerLab. V laboratoři 1.LF UK byl měřen signál EKG, arteriální tlak a náklon lůžka pomocí dvouosého inklinometru. Pro vyhodnocení kvality byl vypočten pro každý signál koeficient determinace, který dosahoval hodnot pro měření EKG, arteriálního tlaku a náklonu lůžka hodnot od 0,74 do 0,98. Hlavním výsledkem této práce je funkční mobilní jednotka pro sběr dat disponující vlastním softwarovým systémem a robustní konstrukcí pomocí dostupných hardwarových komponent. Přínos této práce je především mobilita zkonstruované jednotky, kompaktnost a vlastní software pro ukládání dat.

Klíčová slova

Sběr dat, Mobilní Jednotka, Biosignál, EKG, ABP

ABSTRACT

Mobile unit for collecting biomedical data in experimental environment

The topic of the master's thesis is focused on the construction of a system for data collection in an experimental environment, which will have mainly compactness, robust control and mobility. The first part of the work is devoted to the analysis of the market, which lists the existing equipment for data collection by leading manufacturers and available components for the construction of a mobile unit. This overview shows the need to design a new system, which is mobile compared to existing systems and has its own software for data storage. No additional hardware or software accessories are required for separate data collection using the designed mobile unit. The second part deals with the implementation of hardware, which allows data collection from four analog inputs, has a suitable AD converter, internal battery and storage media. The basic element of the hardware design is a single-board computer Raspberry Pi, where the other components are fully compatible with this system. Robust control is ensured by means of a touch screen in combination with a developed web application. The third part of the diploma thesis deals with the development of a software solution using a web application that enables robust control and storage of data. The last part of the work is devoted to the evaluation of the quality of the record measured by the designed mobile unit. To verify the functionality and quality of the recorded data, measurements were first performed using a signal generator, which verified the functionality and accuracy of the designed mobile unit. The second part of the measurement was devoted to verifying the collection of biomedical data with the existing PowerLab data collection system used. The ECG signal, arterial pressure and bed tilt were measured in the laboratory of the 1st. Faculty of Medicine, Charles University using a biaxial inclinometer. To evaluate the quality, a coefficient of determination was calculated for each signal, which reached the values for measuring ECG, arterial pressure and bed tilt values from 0.74 to 0.98. The main result of this work is a functional mobile unit for data collection with its own software system and robust construction using available hardware components. The benefit of this work is mainly the mobility of the designed unit, compactness and its own software for data storage.

Keywords

Data Acquisition, Mobil Unit, Biosignal, ECG, ABP

Obsah

Seznam symbolů a zkratk	10
1 Úvod	11
2 Přehled současného stavu	12
2.1 Průzkum trhu	13
2.2 Hlavní komponenty pro konstrukci systému.....	15
3 Cíle práce	18
4 Návrh systému	19
4.1 Vybrané komponenty pro realizaci	20
5 Realizace hardwarové části	22
5.1 Propojení s přístrojem	22
5.1.1 Převedení signálu do rozmezí vstupních hodnot AD převodníku	22
5.2 Raspberry Pi 4B	24
5.2.1 GPIO.....	24
5.3 Konstrukce mobilní jednotky	28
6 Realizace softwarové části	31
6.1 Analýza softwarového provedení.....	31
6.2 Operační systém	31
6.3 Použité programovací jazyky	31
6.3.1 Python.....	32
6.3.2 VueJS.....	32
6.4 Vlastní implementace	32
6.4.1 Použitý framework	32
6.4.2 API.....	33
6.4.3 Databáze	33
6.4.4 MQTT.....	34
6.5 Architektura systému.....	36
6.5.1 Konfigurace	39
6.5.2 Databáze	39
6.5.3 Start měření	40
6.5.4 Stop měření.....	40

6.6	Procesy běžící na pozadí	41
7	Návrh ověření systému	43
7.1	Metody ukládání dat.....	43
7.1.1	Použité přístroje.....	43
7.1.2	PowerLab ADInstruments	44
7.1.3	Mobilní jednotka pro sběr dat.....	44
7.2	Metody ověření funkčnosti mobilní jednotky	44
7.2.1	Metody měření EKG	45
7.2.2	Metody měření arteriálního tlaku	46
7.2.3	Metody měření náklonu lůžka	46
7.2.4	Vyhodnocení podobnosti naměřených dat	47
8	Výsledky.....	48
8.1	Ověření pomocí generátoru signálu	48
8.2	Měření fyziologických dat	50
8.2.1	Měření EKG	50
8.2.2	Měření dvouosého inklinometru.....	51
8.2.3	Měření arteriálního tlaku	53
9	Diskuze.....	54
9.1	Hardwarová část	54
9.2	Softwarová část	55
9.3	Experimentální část	56
9.3.1	Ověření pomocí generátoru signálu.....	56
9.3.2	Měření fyziologických dat.....	57
9.3.3	Limitace měření z důvodu pandemie Covid-19	58
10	Závěr	59
11	Seznam použité literatury	60
Příloha A:	Ovládání jednotky.....	65
	Spuštění ukládání dat	66
	Výběr kanálů pro uložení	66
	Přehled aktivních kanálů AD převodníku	67
	Start měření	68
	Informace o měření	70

Zastavení měření	71
Získání ID měření.....	71
Stažení měření ve formátu .csv	72
Ukončení provozu.....	75
Příloha B: Realizace mobilní jednotky	76
Příloha C: Obsah přiloženého CD.....	78

Seznam symbolů a zkratek

Seznam zkratek

Zkratka	Význam
AD	Analogově digitální
DAQ	Sběr dat (<i>Data Acquisition</i>)
NI	National Instruments
GPIO	Univerzální vstupně/výstupní piny (<i>General-purpose input/output</i>)
USB	Univerzální sériová sběrnice (<i>Universal Serial Bus</i>)
SPS	Vzorky za sekundu
LCD	Displej z tekutých krystalů
TFT	Tenkofilmový tranzistorový displej (<i>Thin Film Transistor</i>)
VGA	Video Graphics Array
HDMI	High Definition Multimedia Interface
SPI	Sériové periferní rozhraní (<i>Serial Peripheral Interface</i>)
I2C	Sběrnice (<i>Inter-integrated Circuit</i>)
RAM	Random Access Memory
PGA	Zesilovač s nastavitelným ziskem (<i>Programmable Gain Amplification</i>)
HLO	Analogový výstup (<i>High Level Output</i>)
API	Aplikační programovací rozhraní (<i>Application Programming Interface</i>)
csv	Hodnoty oddělené čárkami (<i>Comma-separated values</i>)
ABP	Arteriální tlak (<i>Arterial Blood Pressure</i>)
ECG	Elektrokardiogram

1 Úvod

Každý živý organismus vysílá signály, které je možno měřit. Biologickým signálem může být teplota, napětí na membráně, nebo elektrická aktivita srdce. Pro měření biologických signálů jsou k dispozici měřicí přístroje, které tento signál pouze snímají. Pro uložení těchto dat je potřeba k danému přístroji připojit další zařízení, které data uloží, nebo se přístroj připojí do nemocniční sítě. Po připojení měřicího přístroje do nemocniční sítě se data uloží do nemocniční databáze do složky příslušného pacienta.

Zařízení pro sběr dat jsou neustále řešeným tématem. V praxi se dnes využívají systémy pro sběr dat, které data uloží, ale pro zobrazení a extrakci naměřených dat je potřeba další příslušenství, například počítač s vhodným softwarem. Dále často tyto systémy nelze odpojit od zdroje a manipulace se systémem během měření je obtížná.

V případě, že příslušný přístroj měřená data zasílá okamžitě do databáze příslušného pacienta (pomocí vodiče ethernet), tak sběrač dat není potřebný. Nemocniční síť představuje Nemocniční informační systém, kterým je soustava vzájemně provázaných informačních systémů a podsystémů umožňující podporovat, sledovat, dokumentovat a řídit procesy lůžkového zdravotnického zařízení [1]. Nemocniční informační systém je program, který reprezentuje databázi. Tato databáze obsahuje data o pacientech příslušících danému zdravotnickému zařízení. Do této databáze má přístup pouze zdravotnický personál. Extrakce dat z těchto databází je složitá.

Výzkum se aktuálně zabývá bezdrátovým sběrem dat, kdy se data nasnímaná senzorem uloží pomocí mobilní brány, nebo radiového vlnění přímo do příslušné databáze. Zmíněné studie zabývající se bezdrátovým sběrem dat jsou uvedeny v následující kapitole.

Tato práce zabývá návrhem a realizací mobilní jednotky pro sběr dat, která bude disponovat kompaktností, pohodlnou extrakcí naměřených dat do vlastního externího úložiště (USB) pomocí sestaveného softwaru a dostupných hardwarových komponent.

2 Přehled současného stavu

Optimalizací sběru dat ve zdravotnictví se zabývá především oblast telemedicíny. Studie se často zabývají sběrem dat ve zdravotnictví bezdrátovým zasíláním dat přímo z přístroje, snímajícího data do nemocniční databáze.

V roce 2002 bylo na odborné konferenci EMBS (Engineering in Medicine and Biology Society) představena metoda pro bezdrátový sběr dat z Holterovského EKG monitorování. V této studii byly použity bezdrátové senzory, které nasnímaná data zasílaly na jednotku pro sběr dat, která byla umístěna na zápěstí subjektu o rozměrech hodiněk. Jednotka pro sběr dat měla dostatečně velké interní úložiště a data byla okamžitě ukládána pomocí mobilní bezdrátové brány. V případě, že by jednotka nemohla v určitém momentě zaslat data na příslušný server, byla data uložena v lokálním úložišti jednotky. Kromě Holterovského snímání EKG byla sledována i hladina SpO₂. Tato studie byla provedena na skupině subjektů procházejícím vojenským výcvikem. [2]

V roce 2009 Hána a Mužík ve své studii představili software pro uložení, okamžité zpracování a zobrazení snímaných dat z multikanálového EKG [3]. Tento systém je založen na frameworku LiveMap, který byl využíván pro okamžité mapování EEG. Software využívá tzv. multithreadingu, který zajišťuje zpracovávání několika operací najednou (zobrazení, uložení, zpracování).

V roce 2012 byla na Technické Univerzitě v Ostravě představena jednotka pro vícekanálové získávání biomedicínských signálů. Toto zařízení bylo ve studii [4] využito pro získávání dat z vícekanálového EKG a zároveň byla měřena pletysmografická křivka. Zařízení formátovalo data do paketů a tyto pakety dál zasílalo do počítače, kde byla nasnímaná data zobrazena a zhodnocena. Představená jednotka pro sběr dat se lišila od nabízených systémů pro sběr dat na trhu především svými rozměry a okamžitým přenosem dat do média pro uložení a zpracování. [4]

Jsou však studie, které sledují vývoj smart technologie, jako jsou chytré náramky, hodinky nebo smartphony. Tato zařízení jsou schopna sbírat některá fyziologická data. Jedna ze studií se zabývala právě přehledem technologií využitých v těchto zařízeních [5]. Zkoumáno bylo, která ze smart technologií by se dala využít pro dohled pacientů ve zdravotnickém zařízení. Jedná se o bezdrátový sběr dat, tzn. nasbírané informace se automaticky zasílají na cloudové úložiště.

Další studie vedou k bezdrátovému získávání dat pomocí inteligentních senzorů, nebo přídatných modulů k danému zařízení pro monitorování pacienta. Tyto moduly nebo senzory komunikují se serverem pomocí mobilních bran a naměřená data zasílají na určitý server [6]. Vyvíjejí se nové komunikační protokoly, které umožňují zobrazení a případné zpracování dat na smartphonu [7]. Bezdrátový způsob sběru dat, na který je aktuálně výzkum zaměřen, není v této práci využit, a proto tato problematika není blíže

rozvedena. Z tohoto důvodu je dále zahrnuta kapitola zabývající se aktuální nabídkou trhu, kde je shrnuta nabídka dostupných systémů pro sběr dat.

2.1 Průzkum trhu

Mobilní systém pro využití v klinickém a experimentálním prostředí musí splňovat určité požadavky, kdy se zohledňuje povaha sledovaného signálu. Z povahy sledovaného signálu vyplývají požadavky na systém, a to například dostatečná vzorkovací frekvence, typ vstupního signálu a dále počet aktivních kanálů. Tato kapitola je zařazena pro přehled nabízených produktů na trhu využívaných v experimentálním, případně klinickém prostředí od předních výrobců laboratorního vybavení.

Ambulantně měřené signály jsou jednorozměrné, kde je sledována závislost dané veličiny na čase. Biologické signály obsahují malé amplitudy napětí v řádech μV až mV a frekvenčně mají rozsah setin až stovek Hz [8].

Několik předních výrobců zaměřujících se na vybavení laboratoří pro klinická a experimentální měření nabízejí mezi mnoha jinými produkty i právě systémy pro sběr dat. Problém těchto produktů je často kompaktnost, složité ovládání a nutnost využití externího hardwaru a softwaru. Níže uvedená tabulka 2.1 představuje přehled systémů pro sběr dat na trhu od předních výrobců laboratorního vybavení.

Tabulka 2.1: Srovnání systémů pro sběr dat s přihlédnutím na požadavky pro sběr biosignálů

Výrobce	Produkt	Počet kanálů	Typ vstupních dat	Vzorkovací frekvence	A/D převodník
Pico Technology	Picolog 1216	16	Analog	>1kS/s	12 bit
ADInstruments	PowerLab	16	Analog	400kS/s	16 bit
NI	DAQ 6002	8 A/13 D	Analog/Digital	50kS/s	16 bit

Prvním zástupcem je systém pro sběr experimentálních dat Picolog 1216 (Pico Technology, UK), který spolehlivě dokáže sbírat hodnoty napětí z 16 kanálů. Hardwarový systém Picolog znázorněn na obrázku 2.1 je potřeba pro zobrazení, extrakci měřených dat, nebo další zpracování signálu připojit k počítači, který zobrazí data pomocí programovacích jazyků využívaných v prostředí pro práci se signály jako je programovací jazyk C nebo LabView (požadavky na systém pro využití LabView 4.0: procesor Pentium 4G1 nebo Intel i5, Windows 7,8.1 nebo 10 a 4GB RAM a 4 GB volného místa na disku). Součástí tohoto produktu je software pro načtení dat (Picolog data logging package) a osciloskop (PicoScope). Pro využití tohoto softwaru však počítač uživatele musí splňovat určité systémové nároky, a to je operační systém Windows 7,8 nebo 10, macOS, nebo Linux a podmínka přítomnosti USB portu 1.1. [9]



Obrázek 2.1: Zařízení Picolog 1216 s připojením k počítači pomocí USB kabelu. Převzato z [9].

Dalším produktem uvedeným v tabulce 2.1 je systém pro sběr dat PowerLab (ADInstrument, USA). Hardwarové provedení představuje robustnější systém sběru dat a je vytvořen přímo pro laboratorní měření (znázorněno na obrázku 2.2). Vyznačuje se vysokou vzorkovací frekvencí (až 400 Hz) převodníku při 16 bitech. Každý kanál obsahuje vlastní filtry a obvody potlačující šum. Při koupi je v ceně zahrnut i software LabChart přímo od výrobce [10]. Nevýhodou softwaru LabChart je maximální počet vstupních kanálů 32, což jsou dva systémy sběru dat PowerLab. Pro zobrazení naměřených hodnot a manipulaci s daty je potřeba vysoce výkonný externí počítač s výše uvedeným softwarem. Další nevýhody představují velké rozměry systému a také nutnost neustálého napájení ze zdroje.



Obrázek 2.2: Systém pro sběr dat PowerLab. Převzato z [10].

Posledním uvedeným zařízením v tabulce 2.1 je systém pro sběr dat NI DAQ (obrázek 2.3), který je využíván při výuce a aplikován v experimentálních studiích [11]. Jedná se o zařízení s analogovými a digitálními vstupy. K zařízení je možno připojit senzory a napájení je zajištěno pomocí sběrnic. Pro zobrazení a manipulaci s naměřenými daty je potřeba počítač, kdy výrobce poskytuje ovladače a konfigurační nástroje pro správu uložených dat. [12]



Obrázek 2.3: Zařízení pro sběr dat z dílny National Instruments. Převzato z [12].

2.2 Hlavní komponenty pro konstrukci systému

Z důvodu velmi obtížné manipulace, nebo potřeby dalšího příslušenství ke stávajícím systémům pro sběr dat uvedených v předchozí kapitole, je tato práce zaměřena na návrh a konstrukci kompaktnějšího systému, se kterým lze manipulovat dle potřeby uživatele. Výsledná zkonstruovaná mobilní jednotka, bude obsahovat jednodeskový počítač malých rozměrů, který bude spravovat nasbíraná data, AD převodník, displej pro obsluhu aplikace a interní baterii, která při odpojení zařízení od zdroje dokáže toto zařízení napájet. V této kapitole jsou představeny vhodné komponenty pro realizaci mobilní jednotky.

Tabulka 2.2: Jednodeskové počítače

Výrobce	Produkt	RAM	Přídavné piny
Raspberry	Raspberry Pi 4 Model B	4 GB	ANO
LattePanda	V 1.0	4 GB	NE
Arduino	Arduino UNO	32KB	ANO
Banana Pi	Banana Pi M3	2 GB	ANO

Tabulka 2.2 představuje přehled nabízených jednodeskových počítačů a mikrokontrolerů na trhu, které jsou vhodné pro zaměření této práce. Jsou uvedeny pouze

ty nejvhodnější řady od výrobce, které by bylo možno použít (nejsou uvedeny další druhy daného produktu). Hlavní atribut pro výběr představuje paměť (RAM) pro čtení a zápis dat a přídavné piny, kterými lze jednodeskový počítač rozšířit. Jako ideální řešení se jeví jednodeskový počítač Raspberry Pi, proto další komponenty uvedeny v následujících tabulkách jsou kompatibilní především s tímto zařízením.

Uvedená tabulka 2.3 níže shrnuje všechny kompatibilní displeje, které lze připojit k vybranému počítači Raspberry Pi. Jsou představeny konkrétní typy displejů, které by vyhovovaly požadavkům na systém především svými rozměry

Tabulka 2.3: Displeje kompatibilní s Raspberry Pi

Název	Úhlopříčka	Propojení	Rozlišení	Panel	Typ
PiTFT Plus	3,5"	GPIO	480x320	LCD	Dotykový rezistivní
Waveshare	5"	USB	800x480,	LCD	Kapacitní dotykový
Waveshare	3,5"	GPIO	320×480	LCD	Dotykový rezistivní
Joy-iT Touch HDMI display V2	5"	GPIO	800x480p	LCD	Dotykový rezistivní
PiTFT Plus displej	2,8"	GPIO	320x240	TFT LCD	rezistivně dotykový

Uvedená tabulka 2.4 představuje souhrn nabízených AD převodníků, které by bylo možné v konstrukci mobilní jednotky pro sběr dat použít. Všechny uvedené převodníky jsou typu sigma-delta. Převodníky typu sigma-delta jsou pomalejší, ale přesné, což je při měření biosignálů potřebné [13].

Tabulka 2.4: AD převodníky kompatibilní s Raspberry Pi

Název	Počet kanálů	Vzorkovací frekvence [SPS]	Velikost převodníku [bit]	Maximální napájecí napětí [V]	Podporovaná sběrnice
MCP3208-BI/P	8	100k	12	5	SPI
MCP3304-BI/SL	4	100k	13	5.5	SPI
ADC Differential Pi Kit	8	240	12	5	I2C
ADC-DAC Pi Zero	2	100k	12	3.3	SPI

Poslední tabulka 2.5 obsahuje zdroje baterií, které lze v této práci využít. Jsou uvedeny i baterie, které nejsou odpojitelné a byla by potřeba je k zařízení pevně připájet.

Tabulka 2.5: Externí baterie kompatibilní s Raspberry Pi

Název	Kapacita	Další možnosti
PiJuice HAT	1820 mAh	MCU
PiSupply	12 000 mAh	-
Waveshare Li-polymer HAT	3000 mAh	Integrovaný čip pro rychlé nabíjení

Z komponent uvedených v této kapitole byly vybrány ty, které splňovaly požadavky na systém sběru biosignálů a následně byla provedena realizace hardwaru. Výběrem komponent se zabývá kapitola 4. Návrh systému.

3 Cíle práce

Cílem této diplomové práce je navrhnout, realizovat a otestovat mobilní jednotku pro sběr biologických signálů prostřednictvím patientských monitorů v klinickém a experimentálním prostředí. Hlavním cílem této práce je zhotovení mobilní jednotky pro sběr dat disponujícími kompaktními rozměry, robustním ovládním a zajištěním funkce bez nutnosti připojování externích zařízení. Výstupem této práce bude mobilní jednotka pro sběr dat malých rozměrů disponující zobrazovacím médiem, vlastní interní baterií a vlastním softwarem pro ukládání dat.

Přínos této práce je především v klinicko-experimentálním prostředí pro sběr dat. Mobilní jednotku pro sběr dat by v případě pozitivních výsledků bylo možno aplikovat aktuálně i do nemocnic, kde probíhá léčba pacientů postižených virem SARS-CoV-2. Aktuální výzkum probíhající na 1.LF UK se zaměřuje na vliv náklonu lůžka na pacienty postihnutými virem SARS-CoV-2 připojenými na mechanickou ventilaci [14]. V tomto výzkumu by měla mobilní jednotka patřičně zastoupení, jelikož by díky malým rozměrům a interní baterii mohla sbírat data i přes manipulaci s pacientem.

4 Návrh systému

Jak již bylo uvedeno v kapitole 2.2, je tato práce zaměřena na konstrukci systému pro sběr dat. Jeho výhodou budou především malé rozměry a robustní ovládání. Díky využití jednodeskového počítače, který bude řídit chod celého systému, nebude využíváno externího příslušenství, jako je například počítač.

Z důvodu malé amplitudy biologického signálu lze implementovat i zesilovač. Je nutné podotknout, že některé přístroje již mají výstup zesílený, nebo jinak upravený. Proto je potřeba aplikovat zesilovač individuálně na základě detailnější analýzy výstupů měřících přístrojů.

Zesilovač biologických signálů musí mít vysoký vstupní odpor ($M\Omega$), diskriminační činitel neboli činitele potlačení soufázové složky tzv. CMRR minimálně 80 dB, zesílení ideálně 1000x, nízký výstupní odpor (jednotky Ω), zotavovací doba při přebuzení do 10 s a ochranu proti poškození napětím (5000 V). Souhrn těchto požadavků na zesilovač biologického signálu představuje tabulka níže 4.1.

Tabulka 4.1: Požadavky na zesilovač biosignálu

Vstupní odpor	V řádech $M\Omega$
CMRR	>80 dB
Zesílení	1000x
Vstupní odpor	Jednotky Ω
Zotavovací doba při přebuzení	<10 s
Ochrana proti poškození napětí	5000 V

Další komponenta, která musí splnit požadavky pro operaci se signály je AD převodník. Aby nedošlo při převodu z analogové podoby do digitální ke zkreslení a tím ke ztrátě důležité informace, je potřeba, aby vzorkovací frekvence byla alespoň 240 vzorků za sekundu a výše. Výstupní signály z měřících přístrojů obsahují frekvence do 150 Hz a je potřeba splnění Nyquistova teoremu [8]. Pro dostatečné rozlišení signálu je podmínka alespoň 12bitového převodníku. Tyto požadavky shrnuje tabulka 4.2. [15]

Tabulka 4.2: Požadavky na AD převodník.

Vzorkovací frekvence	>240 Hz
Velikost převodníku	>12 bit
Počet kanálů	>4
Komunikace s Raspberry	I2C

Zbylé komponenty pro konstrukci mobilní jednotky se odvíjejí ze zadání diplomové práce. Je potřeba obstarat ideální baterii a displej o vyhovujících rozměrech, aby mobilní

jednotka splňovala požadavek robustního ovládání. Vybrané komponenty z kapitoly 2.2. jsou obsaženy v následující kapitole 5, kde je jejich výběr zdůvodněn.

4.1 Vybrané komponenty pro realizaci

Konkrétní komponenty vybrané pro realizaci mobilní jednotky splňující požadavky pro sběr biologických signálů (uvedeny v předchozí kapitole 4) jsou uvedeny v tabulce 4.1 níže. Zvolené komponenty jsou vybrány ze souhrnu dostupných komponent, který představuje kapitola 2.2.

Tabulka 4.1: Vybrané komponenty pro sestavení systému

Součástka	Typ
Zesilovač	LM 324
Jednodeskový počítač	Raspberry Pi 4B
AD převodník	ADC Differential Kit
Baterie	PiJuice HAT
Displej	Joy-it Touch V2
Paměťové médium	SD card 64 GB

Z důvodu měření biosignálů, které jsou ve většině případů v řádu milivoltů, je potřeba tento signál zesílit, případně převést do rozsahu vstupních hodnot AD převodníku. Proto je jako zesilovač zvolena součástka LM 324, která disponuje čtyřmi zesilovači, které se nacházejí na jednotlivých kanálech. Je využito diferenčního zapojení zesilovače, jelikož vybraný AD převodník má parametry vstupního signálu v rozsahu $\pm 2,048\text{V}$. Dále z důvodu měření arteriálního tlaku pomocí konkrétního modulu, je potřeba od vstupního signálu odečíst určitou hodnotu, aby AD převodník byl schopný data zaznamenat. Toto řešení je uvedeno v kapitole 5.1.1.

Pro řízení dat a další zpracování (uživatelské rozhraní, ukládání dat do databáze, zobrazení atp.) je vybrán jednodeskový počítač Raspberry Pi, který disponuje přídatnými piny a lze jej tak rozšířit o potřebné komponenty jako je AD převodník, displej, nebo baterie. Modelů Raspberry Pi je na trhu již několik řad. Model 4B je zatím jedním z mála dostupných jednodeskových počítačů disponujících dostatečnou RAM pamětí a to 4 GB a zároveň přídatnými piny.

Převodník ADC Differential Kit je zařazen z důvodu kompatibility s počítačem Raspberry Pi a přítomnosti dvou mikročipů představující AD převodníky o možnosti čtyř vstupů. Lze tedy dále mobilní jednotku rozšířit o další čtyři analogové vstupy. Tento převodník disponuje nastavitelnou vzorkovací frekvencí a počtem bitů převodu. Komunikace s počítačem probíhá pomocí sběrnice I2C. Jediný omezující parametr je vstupní napětí $\pm 2,048\text{ V}$, ale tento problém lze regulovat výše uvedeným zapojením zesilovače.

Baterie PiJuice HAT je vybrána z důvodu přítomnosti jednočipového počítače (MCU). Zařízení Raspberry Pi je náchylné na nesprávné odpojení od zdroje, kdy hrozí ztráta dat. Tento modul díky MCU dokáže napájet zařízení Raspberry Pi 4 až 6 hodin a poté zařízení automaticky vypnout a předejít tak ztrátě nebo poškození dat.

Displej Joy-it Touch V2 je vybrán především díky svým rozměrům a dotykovému povrchu, který bude umožňovat obsluhu aplikace.

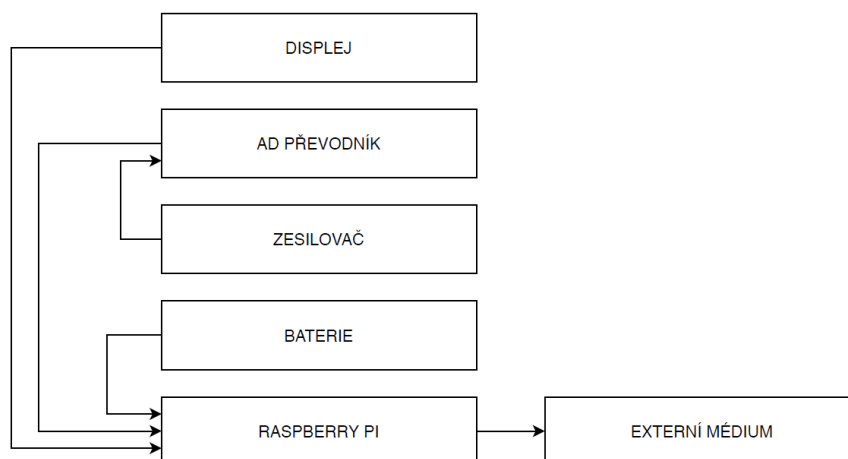
Jako paměťové médium s operačním systémem a aplikací je zvolena SD karta, která je doporučována výrobcem. Soubor, který obsahuje naměřená data je automaticky ukládán do složky stažené, kdy bude zabírat místo na tomto paměťovém médiu. Aby bylo dosaženo co nejmenší velikosti souboru obsahujícího naměřená data je zvolen typ `.csv`.

Mobilní jednotka sbírá především biosignály z pacienta, a to signál EKG (elektrokardiogram), arteriální krevní tlak a analogový výstup z dvouosého inklinometru (měření náklonu lůžka). Vstupy mobilní jednotky bude mimo jiné možno využít pro jakýkoliv analogový signál. Pro tuto práci je využito čtyř vstupů a pro další rozšíření lze využít další čtyři vstupy, které poskytuje druhý mikročip na desce vybraného AD převodníku.

Podrobnějším popisem těchto konkrétních hardwarových komponent se zabývá kapitola 5.

5 Realizace hardwarové části

Tato kapitola blíže představuje využití zvolených komponent pro realizaci sy uvedených v kapitole 4.1. Je zde uvedena detailnější implementace komponent a jejich využití v rámci této práce. Obrázek 5.1 představuje zjednodušené schéma zapojení hardwarových komponent.



Obrázek 5.1: Zjednodušené schéma zapojení hardwarových komponent, kdy šipky znázorňují komunikaci mezi jednotlivými moduly.

5.1 Propojení s přístrojem

Pro propojení s přístrojem, ze kterého jsou ukládána data, bylo využito koaxiálních kabelů. Toto propojení bylo závislé na konektorech měřícího přístroje. Pro využití propojení zařízení s jednotkou lze využít i jiných vodičů, například silikonových dvoužilových. Problematiku koaxiálních kabelů řeší literatura [16].

5.1.1 Převedení signálu do rozmezí vstupních hodnot AD převodníku

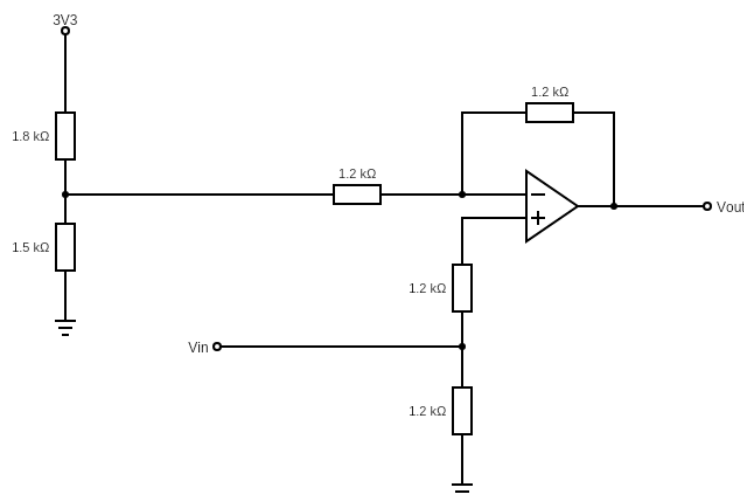
Hodnoty výstupného napětí z modulu snímající tlak (kapitola 7.3) se pohybují v rozmezí od $-0,5$ V do $3,5$ V, což nesplňovalo podmínku vstupních parametrů AD převodníku, proto bylo potřeba tento signál posunout do rozmezí $\pm 2,048$ V. Jeden kanál součástky LM324 byl zapojen jako diferenční zesilovač, kdy se od vstupní hodnoty tlaku odečítala konstantní hodnota $1,5$ V. Pro záporný vstup zesilovače bylo využito konstantní hodnoty z GPIO pinů Raspberry Pi 3V3 a pomocí odporového děliče se získalo požadované napětí $1,5$ V. Zesílení nebylo nutné, proto všechny odpory v obvodu mají shodné hodnoty a to 1200Ω což odpovídá značení odporové řady E12 součástce 1K2.

Pro výpočet adekvátních hodnot odporového děliče bylo využito vztahu pro výpočet výstupního napětí [17]. Po dosazení do tohoto vztahu byl získán poměr odporů. Vstupní napětí U_0 je 3,3 V, které bylo získáno z GPIO pinů na Raspberry Pi. Pro výstupní napětí U_2 byla zvolena hodnota 1,5 V, která byla odečítána od vstupního signálu tlaku. Tímto řešením bylo docíleno vstupního rozsahu AD převodníku.

$$3,3 = 1,5 \frac{R_2}{R_1 + R_2}$$

$$1,5R_1 = 1,8R_2$$

Pro dodržení výše uvedeného poměru, byly zvoleny hodnoty odporů $R_1 = 1800 \Omega$ a $R_2 = 1500 \Omega$, což odpovídá součástkám z řady E12 1K5 a 1K8. Tímto poměrem byl zajištěn konstantní výstup děliče o hodnotě 1,5 V, který byl odečítán od vstupního signálu. Zapojení implementovaného zesilovače je znázorněno na obrázku 5.3.



Obrázek 5.3: Schéma zapojení diferenčního zesilovače, kdy od vstupního signálu tlaku V_{in} byla odečítána hodnota 1,5V. Tímto odečtem bylo dosaženo vstupních hodnot AD převodníku. Výstup V_{out} je připojen na AD převodník kanál č.2.

V případě měření náklonu lůžka pomocí inklinometru využívaného v experimentální laboratoři 1.LF UK, bylo potřeba vstupní signál posunout do rozmezí $\pm 2,048$ V. Hodnoty naměřené pomocí inklinometru uvedeném v kapitole 7.4 dosahovaly hodnot od 0,5 V do 4,5 V. Bylo aplikováno stejné řešení jako pro posunutí vstupních hodnot měřeného tlaku. Od vstupního signálu byla odečítána konstantní hodnota 2,5 V, kdy tímto odečtem bylo docíleno posunu signálu z inklinometru do rozmezí ± 2 V. Zesilovač byl napájen jako v předchozím případě konstantním napětím 3V3 z GPIO pinů Raspberry. Dle vzorce pro dělič napětí byl vypočten poměr pro zvolení odporů děliče napětí. Hodnoty těchto odporů dle řady E12 jsou pro $R_1 = 1K5$ a pro $R_2 = 2K2$. Zapojení tohoto kanálu bylo

stejně jako na obrázku 5.3, které se liší se pouze v hodnotách odporu děliče napětí R1 a R2. Takto zapojeny byly dva kanály (konkrétně kanál 3 a 4), na kterých byly připojeny výstupy z inklinometru.

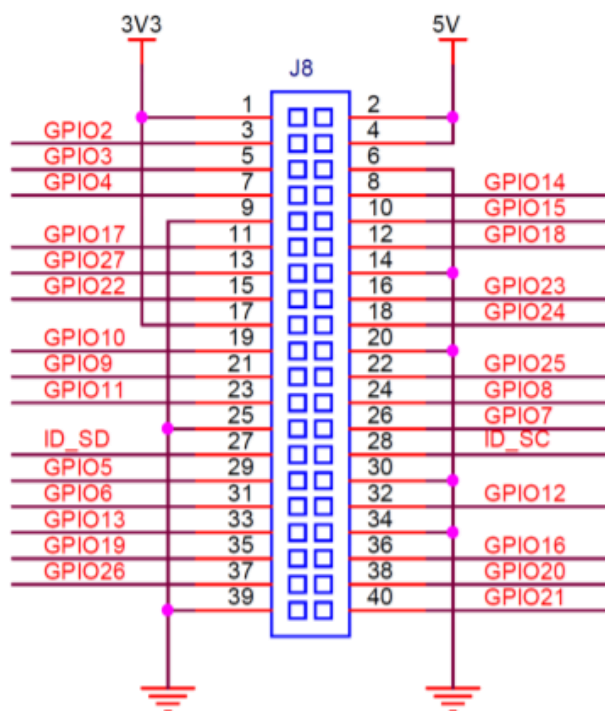
Ze součástky LM324 byly využity tři vstupy. Zapojení byla realizována na pájivém poli, kdy výstup zesilovače byl připojen k příslušnému kanálu AD převodníku.

5.2 Raspberry Pi 4B

Jak je v kapitole 4.1 zmíněno, jako řídicí jednotka mobilního systému byl zvolen jednodeskový počítač Raspberry Pi model 4B. Tento model disponuje dostatečnou RAM pamětí (4 GB) a obsahuje rozšiřující GPIO (Univerzální vstupně/výstupní piny) piny. Zařízení Raspberry Pi bylo v této práci využito především jako nosič softwaru pro uložení dat, kdy AD převodník zasílal data s frekvencí 240 hodnot za sekundu. Takovýto objem dat musela řídicí jednotka správně zpracovat (uložení do databáze, převod do souboru .csv atd). Více o jednodeskových počítačích v dostupné literatuře [18, 19].

5.2.1 GPIO

Pro využití dalších rozšíření jednodeskového počítače jako je AD převodník, baterie nebo displej, bylo potřeba využít vzájemného propojení pomocí pinů GPIO. Tyto piny jsou znázorněny na obrázku 5.4 a jejich kompletní přehled představuje tabulka 5.1. Pomocí těchto pinů lze rozšiřující komponenty napájet a řídit komunikaci mezi řídicí jednotkou a danou rozšiřující hardwarovou komponentou. Přídavné komponenty s Raspberry Pi komunikovaly pomocí sběrnice I2C. Obsazené adresy přídavných modulů jsou zobrazeny na obrázku 5.5.



Obrázek 5.4: Rozložení GPIO pro Raspberry 4B. Převzato a upraveno z [18].

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:																
10:					14											
20:																
30:																
40:																
50:																
60:									68		6a	6b				
70:																

Obrázek 5.5: Adresy modulů využitých v této práci. Na adrese 14 je umístěn modul PiJuice pro baterii a adresa 68 6a a 6b jsou adresy AD převodníku.

Tabulka 5.1: Označení GPIO výstupů Raspberry Pi. Převzato a upraveno z [18].

Označení výstupu z procesoru	Číslo vývodů GPIO		Označení výstupu z procesoru
3V3	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3V3	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
ID_SD	27	28	ID_SC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

AD Převodník

AD převodník (Analog-Digital) je elektronické zařízení, které má na starosti převod analogového signálu (nebo jiné analogové veličiny) na diskretní řadu. Tento převod se využívá kvůli dalšímu zpracování dat na digitálním zařízení. Přesnost a rychlost práce převodníku jsou hlavní faktory určující kvalitu a účinnost celého systému. Vzorkování signálu a detailnějším rozboru AD převodníků se věnuje řada literatury jako je například [21, 22, 23, 24].

Zařízení Raspberry Pi nedisponuje vlastním AD převodníkem, a proto bylo nutno jej implementovat pomocí rozšiřující komponenty. Tato podkapitola se věnuje vybranému AD převodníku.

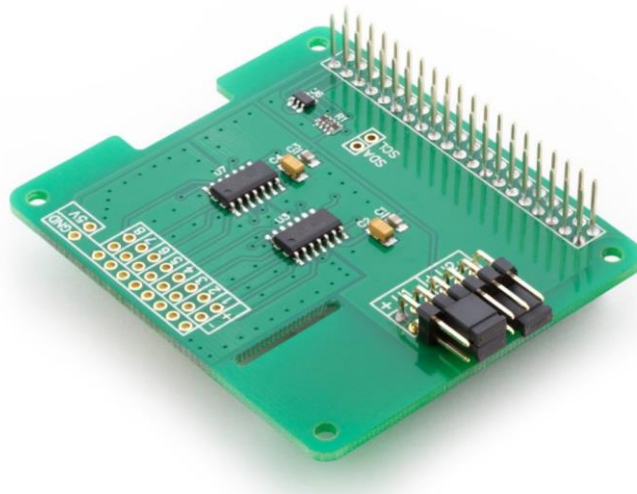
Vybraný AD převodník z dílny Abeelectronics ADC Differential Pi Kit se zakládá na dvou mikročipech MCP3424, kdy každý obsahuje čtyři analogové vstupy. MCP3424 je delta-sigma AD převodník s nízkošumovými diferenčními vstupy. Napájení tohoto modulu bylo řešeno napájením přímo z počítače Raspberry pomocí portu GPIO. [19]

Pomocí tohoto převodníku lze měřit pouze nízká napětí, tedy pro měření biologických signálů byl tento převodník označen jako postačující. Rozšiřující GPIO piny na vrchní straně ADC Differential Pi umožňují jeho nasazení a použití společně s jinými rozšiřujícími HAT deskami (moduly, které lze připojit k Raspberry Pi pomocí GPIO pinů). [19]

I2C adresové bity jsou volitelné. MCP3424 podporuje až osm rozdílných I2C adres, tedy pomocí dvou AD převodníků na každém ADC Differential Pi Kitu lze na jeden počítač Raspberry Pi nad sebe zařadit až čtyři moduly ADC Differential Kit. Takto lze získat až 32 analogových vstupů. [19]

Modul ADC Differential Kit komunikuje s počítačem pomocí sběrnice I2C. Tento modul disponuje vlastním přístrojovým zesilovačem PGA s programovatelným ziskem, a to se zesílením dvakrát, čtyřikrát či osmkrát. Zesílení je provedeno před konverzí analogového signálu na digitální. V této práci možnost zesílení přítomným PGA zesilovačem není využita z důvodu nevyhovující velikosti zesílení. [19]

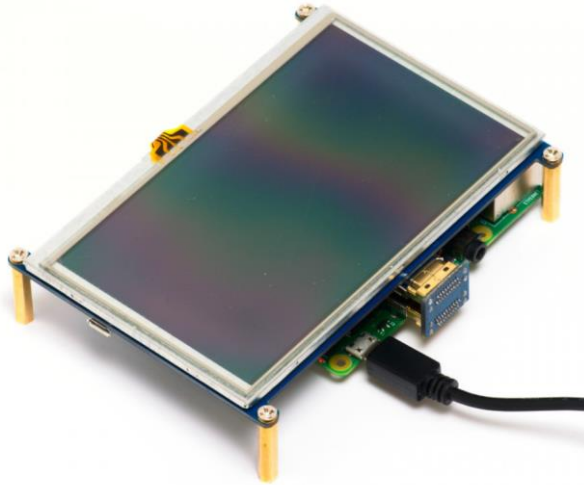
Pro správnou funkčnost byla naprogramována rychlost přenosu dat. Převodník ADC Differential Kit disponuje více přenosovými rychlostmi uvedených v příslušném manuálu [19]. Výběr rychlosti se provádí pomocí výběru I2C adresy. Pro tuto práci nejvíce vyhovuje: 240 vzorků za sekundu při 12 bitech, čímž je splněn Nyquistův teorém pro sběr signálů EKG, arteriálního tlaku a výstupního signálu z dvouosého inklinometru. Je však nutno zmínit, že vstupní signál musí být v rozsahu $\pm 2,048$ V. [19]



Obrázek 5.6: Součástka ADC Differential Pi obsahující dva čtyřkanálové čipy MCP3424 a GPIO piny, pomocí kterých lze připojit další rozšiřující hardwarové komponenty. Převzato z [13].

Displej

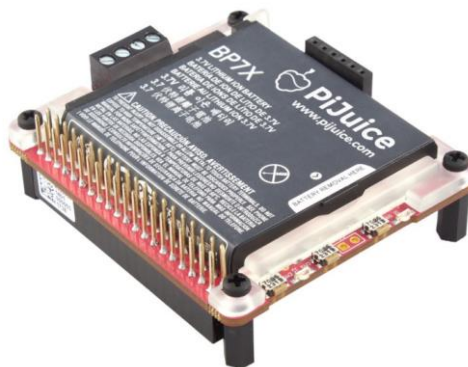
Displej byl v této práci začleněn pro ovládání a kontrolu mobilní jednotky. Pro zachování kompaktnosti byl zvolen displej s pětipalcovou úhlopříčkou. Vybraný displej Joy-it Touch V2 (obrázek 5.7) disponuje dotykovou vrstvou, která uživateli usnadní ovládání systému v kombinaci s uživatelským rozhraním. Pro přiblížení problematiky týkající se mobilních zobrazovacích médií se zabývá dostupná literatura [20].



Obrázek 5.7: Displej Joy-iT Touch V2. Převzato z [14].

Baterie

Vybraná baterie PiJuice disponuje vlastním mikročítačem. Tato baterie byla napájena pomocí pinů GPIO a při odpojení od zdroje zařízení bylo dále nepřerušovaně napájeno. Pomocí mikročítače dokáže tato baterie fungovat zcela autonomně. Navíc tato baterie obsahuje i modul reálného času, tedy zařízení Raspberry Pi může získávat informace o reálném čase přímo z tohoto zařízení. Modul s baterií je znázorněn na obrázku 5.8



Obrázek 5.8: Modul PiJuice s baterií a MCU. Převzato z [21].

5.3 Konstrukce mobilní jednotky

Po zapojení využitých hardwarových komponent do jednoho celku, byl tento celek uložen do konstrukční krabičky o vhodných rozměrech, které jsou uvedeny v tabulce 5.4. Uvedené obrázky v této kapitole představují zrealizovaný hardwarový systém,

kdy obrázek 5.9 znázorňuje aktivní mobilní jednotku bez připojení do zdroje napětí. Provoz jednotky je realizován pomocí napájení z baterie. Další obrázky 5.9 - 5.10 znázorňují vzhled, vstupní prvky jednotky pro měřicí přístroje porty pro připojení USB. Podrobnější vzhled jednotky je uveden v příloze B.

Tabulka 5.2 Rozměry konstrukční krabičky

Výška [mm]	Délka [mm]	Šířka [mm]
60	160	140



Obrázek 5.9: Přední strana mobilní jednotky a provoz mobilní jednotky bez napájení.



Obrázek 5.10: Boční část s koaxiálními konektory pro propojení mobilní jednotky s měřicími přístroji.

Všechny využívané komponenty pro konstrukci mobilní jednotky byly nakonfigurovány dle vlastních dokumentací a komunikace probíhala pomocí sběrnice I2C, kdy konkrétní obsazené adresy jsou uvedeny na obrázku 5.5. Pro správnou funkčnost mobilní jednotky a využití všech vybraných komponentů bylo potřeba sestavit aplikaci, která ukládala měřená data AD převodníkem.

6 Realizace softwarové části

Pro správnou funkčnost mobilní jednotky bylo potřeba navrhnout a implementovat software, který se staral o interakci s uživatelem, dále pak o vlastní zpracování vstupních signálů a ukládání naměřených hodnot. Tento software bude možno aplikovat na jakýkoliv hardware pro ukládání dat. Výstupní soubor naměřených dat je ve formátu .csv (Comma-separated values). Tento typ výsledného souboru byl zvolen z důvodu rozšířené kompatibility s programy pro zpracování dat, jako je například MATLAB.

6.1 Analýza softwarového provedení

Pro získávání a ukládání hodnot naměřených AD převodníkem je potřeba vytvořit přehlednou a robustní aplikaci s vhodným uživatelským rozhraním. Tato aplikace bude obsahovat jednotlivé moduly, které budou vykonávat potřebné úkony. Je potřeba tyto moduly sestavit tak, aby byly schopné pracovat nezávisle, aby se při poruše nebo výskytu chyby nezastavil celý systém (například modul pro zobrazení vyhodnotí nezobrazitelnou část, avšak ostatní moduly pro ukládání a snímání budou data nadále zpracovávat). Bude potřeba modul pro získávání dat z převodníku, dále modul pro ukládání dat do databáze, modul uživatelského rozhraní a další moduly zajišťující komunikaci mezi sebou. Modul pro komunikaci mezi producentem a konzumentem dat (blíže vysvětleno v kapitole 7.4.4 MQTT) bude zajišťovat modul broker.

Zobrazení informací o ukládání dat uživateli bylo řešeno pomocí dotykového displeje, který je popsán v kapitole 5.2.1.

6.2 Operační systém

Zařízení Raspberry disponuje operačním systémem Rasbian s linuxovým jádrem, kdy je tento operační nejpoužívanější a doporučován výrobcem., lze však využít i dalších operačních systémů. Pro potřeby sběru dat byl jako nejvhodnější operační systém zvolen Linux z důvodu mnoha uživatelských možností a malé velikosti. [22]

Linux představuje open source operační systém, jsou tedy k dispozici zdrojové kódy. Licence umožňují systém využívat zdarma, upravovat i distribuovat. Z hlediska bezpečnosti nejsou zaznamenány problémy s viry nebo jiným malware. Další výhodou je takzvaný balíčkovací systém, který umožňuje práci s aplikacemi. I z těchto důvodů byl tento operační systém zvolen. [23]

6.3 Použité programovací jazyky

Mezi nejvyžívanější jazyky v linuxovém prostředí patří jazyk C, C++ nebo Java, ale Python je součástí většiny distribucí a je vyhlášen mezi vývojáři jako nejideálnější

programovací jazyk. Tento jazyk byl zvolen pro psaní kódu z důvodu přehlednosti a rozsáhlé dokumentace včetně mnoha doplňujících knihoven umožňujících efektivnější vývoj aplikace.

Uživatelské rozhraní bylo vytvořeno pomocí frameworku VueJS, který je založen na programovacím jazyku JavaScript. Tento framework byl zvolen z důvodu široké dokumentace, mnoha doplňujících knihoven pro rozšíření, a především pro přehlednost kódu.

Kombinace těchto dvou jazyků byla zvolena především z důvodu velmi dobré přehlednosti kódu a vzájemné kompatibility. Pro budoucí vývoj, nebo rozšíření (například přidání více AD převodníků, zřízení více mobilních jednotek) aplikace bude přehledná a z důvodu nezávislosti jednotlivých modulů bude možno aplikaci dále upravovat a rozšiřovat.

6.3.1 Python

Jedná se o vysokoúrovňový skriptovací programovací jazyk. Podporuje různá programovací paradigmaty především objektově orientované, dále imperativní, procedurální nebo funkcionální. Python je vyvíjen jako open source, kdy zdarma nabízí instalační balíčky pro většinu běžných platform. Python mimo jiné umožňuje tvorbu rozsáhlých plnohodnotných aplikací včetně grafických uživatelských rozhraní, i proto byl pro tuto práci zvolen jako ideální. [24]

6.3.2 VueJS

VueJS představuje framework využívající jazyk JavaScript pro vytváření uživatelských rozhraní. Tento framework pracuje pomocí webových prohlížečů, kdy usnadňuje čitelnost kódu. VueJS umožňuje webovou aplikaci rozdělit na několik částí, které lze vyvíjet nezávisle na sobě. Model je založen na komponentách, který usnadňuje použití často opakovaných částí aplikace. Pomocí knihoven třetích stran lze využít různých animací komponent a usnadnění přechodu mezi nimi. Dostupnost hotových řešení napomáhá snížení množství kódu a udržuje jeho přehlednost. [25]

6.4 Vlastní implementace

Software sestavený v této práci bude možno aplikovat na jakýkoliv hardware pro ukládání dat. Výstupem této práce je kód, který obsahuje moduly, které zajišťují ukládání měřených dat a robustní ovládání pomocí dotykového displeje.

6.4.1 Použitý framework

Pro přehlednost kódu aplikace je potřeba zvolit vhodný framework (Bližší definice pojmu framework je v dostupné literatuře [26]). Framework zajistí především

znovupoužitelnost (jedna metoda může být volána v různých částech kódu mnoho krát) a předchází duplicitě kódu. Nejpoužívanější frameworky pro Python a vývoj webových aplikací umožňující efektivní práci s databázemi jsou Django a Flask. Django je rozsáhlý framework a pro použití v této práci se nehodí z důvodu velikosti (má implementováno mnoho knihoven, nebo vyžaduje speciální nástroje), a disponuje vlastní složitou strukturou. Flask z důvodu malé velikosti a možnosti implementace pouze potřebných knihoven se jeví jako ideální řešení pro tuto práci.

Flask

Flask je microframework napsaný v jazyce Python. Jako microframework je klasifikován z důvodu absence požadavků na speciální nástroje nebo knihovny. Využívá flexibility jazyku Python pro poskytnutí základní šablony pro vývoj webových aplikací. Flask umožňuje aplikace psát i škálovat. Flask disponuje rozsáhlou dokumentací a má možnost řady rozšíření, kdy každé z nich má vlastní dokumentaci. [32, 33]

6.4.2 API

API neboli Application Programming Interface (rozhraní pro aplikace a jejich programování) označuje sbírku funkcí tříd a protokolů určité knihovny. API udává způsob volání funkcí ze zdrojového kódu programu. API lze jednoduše označit jako abstrakci, která definuje a popisuje rozhraní pro interakce s funkcemi, které používají součásti softwarového systému. [26]

Flask REST API Plus

Flask REST API plus je rozšíření frameworku Flask, který představuje podporu pro vytváření REST API (Bližší definice pojmu REST v dostupné literatuře [26]). Disponuje osvědčenými postupy, kdy nejsou potřeba další složitá nastavení pro správné fungování frameworku. [27]

6.4.3 Databáze

Databázový systém je program, kterému se naměřená data předají do úschovy. Program vytvoří soubor, kam data uloží. Představuje prostředníka při práci s těmito daty. Uložení dat do souboru představuje rychlejší a snadnější vyhledávání. Databázové systémy se využívají pro rychlost práce s daty, jsou optimalizované pro práci s velkým množstvím dat a představují co nejnižší zatížení systému. [20, 35, 36]

Detailnější rozbor databází je k dispozici v dostupné literatuře [28].

Aplikace pro mobilní jednotku potřebuje vhodnou databázi z důvodu práce s obrovským množstvím dat (240 hodnot za sekundu).

Flask-SQLAlchemy

Flask-SQLAlchemy je dalším rozšířením frameworku Flask, který přidává podporu SQLAlchemy do aplikace. SQLAlchemy je sada nástrojů Python a Object Relational Mapper, která poskytuje přístup k databázi SQL. Pomocí tohoto frameworku byly implementovány databáze SQLite a PostgreSQL.

Detailnější rozbor Flask-SQLAlchemy je k dispozici v příslušné dokumentaci [29] a v dostupné literatuře [30].

SQLite

SQLite představuje relační databázový systém, který je obsažen v malé knihovně (velikost je do 700 KB) napsán pomocí programovacího jazyka C. Na rozdíl od ostatních databází, které pracují na principu klient – server a jsou spouštěny jako samostatný proces, SQLite je pouze knihovna disponující jednoduchým rozhraním. Každá databáze je uložena v samostatném souboru *.dbm* (Database Manager), kde se data ukládají pomocí jednoduchého primárního klíče do bloků stejných rozměrů a pro rychlý přístup k datům využívá hašovací techniky (při vyhledávání dle klíče). [20, 39]

Tuto databázi lze v této práci využít pro ukládání dat z modulů aplikace jako je například broker, user management, GUI atd.

PostgreSQL

PostgreSQL je objektově-relační databázový systém. Tento systém má za sebou kontinuálně přes 20 let aktivního vývoje, a proto je dobře konfigurovatelný a nastavitelný. Výhodou tohoto databázového systému je schopnost zpracovávat velké množství dat.

Databázi PostgreSQL detailněji popisuje dostupná literatura [31] a je k dispozici rozsáhlá dokumentace [32].

Tuto databázi jsem se rozhodla v této práci implementovat pro uložení nasnímaných dat z AD převodníku, kdy se jedná právě o velké objemy dat. Z AD převodníku se bude zasílat 240 hodnot za vteřinu, tedy 14 400 hodnot za minutu, a za hodinové měření to představuje 864 000 hodnot pro jeden kanál. Takové množství informací pro systém je velmi náročné a databáze PostgreSQL by měla takovou frekvenci zapisování zvládnout.

6.4.4 MQTT

Message Queuing Telemetry Transport (neboli přenos vzdálených měření pomocí front zpráv) je uznávaný a hojně používaný standard pro obecný přenos dat. Jedná se o protokol, který určuje postupy přenosu dat. Jeho jednoduchost spočívá ve využívání sítě TCP/IP (běžná počítačová síť) a od koncových zařízení nevyžaduje velký výkon, proto jej lze využít i pro jednoduché moduly, jako je například Arduino. V praxi MQTT frontu využívá například Facebook Messenger. [41, 42]

Zdroj dat je označován jako producent a cíl dat jako konzument. Přiřazení dat od producenta ke správnému konzumentu zařizuje takzvaný broker. Producent a konzument zprostředkovává typ odeslaných a přijatých dat, nikoli cestu. Správnou cestu dat zařizuje takzvaný broker. [41, 42]

Pro zajištění komunikace jsou potřeba prvky producent, broker a konzument. Implementací MQTT je celá řada pro všechna běžně používaná prostředí. Je dostupná jako knihovna pro řadu programovacích jazyků jako je například jazyk C, Java, Python atd. Pro zasílání dat (tzv. publish) v nejzákladnějším režimu je potřeba spojení v první řadě s brokerem a v druhé řadě publikovat zprávu (jedna přenášená jednotka dat obsahující hlavičku – potřeba pro protokol, a data) do tématu (proud dat, která se předávají). Pro příjem dat je potřeba se opět spojit s brokerem a poté se přihlásit k odběru tématu (tzv. subscribe). Více dostupné v oficiální dokumentaci [33].

Pro tuto práci byl zvolen tento protokol díky své kompaktnosti. Aplikace pro mobilní jednotku bude přenášet velké množství dat mezi uživatelským rozhraním a REST API. Další prvky, které by byly potřeba pro zajištění komunikace a přenosu dat mezi GUI a jednotlivými moduly. pomocí jiných metod by zbytečně tento přenos dat komplikovaly a zpomalovaly chod aplikace.

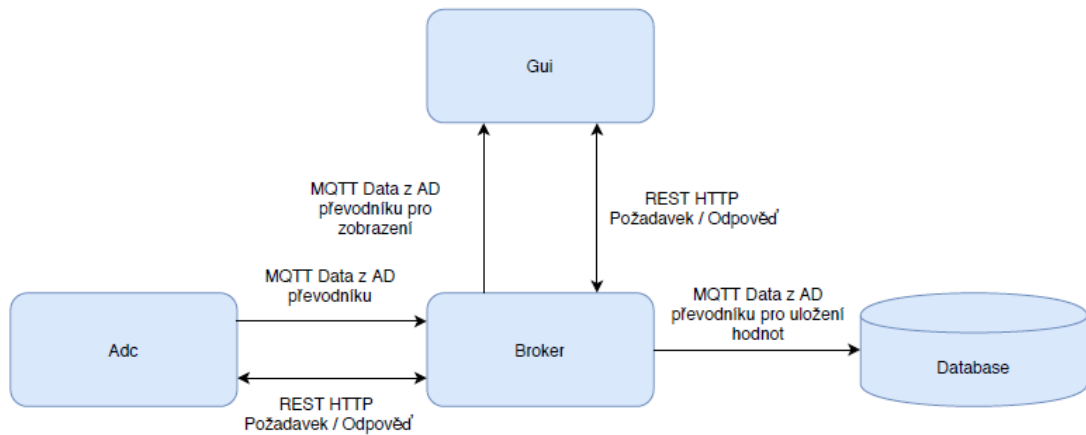
Flask MQTT

Flask MQTT představuje další rozšíření frameworku Flask, která slouží k usnadnění integrace MQTT do aplikace. Toto rozšíření bylo zvoleno z důvodu implementace výše uvedeného protokolu MQTT do aplikace pro uložení. Více dostupné v oficiální dokumentaci na [27].

HBMQTT

HBMQTT představuje open source implementaci MQTT klienta a brokera založena na frameworku asyncio (standartní asynchronní Python I/O framework). Tato implementace poskytuje přímé API založené na corountiens, kdy představuje jednoduchý návrh daných aplikací. [41, 43]

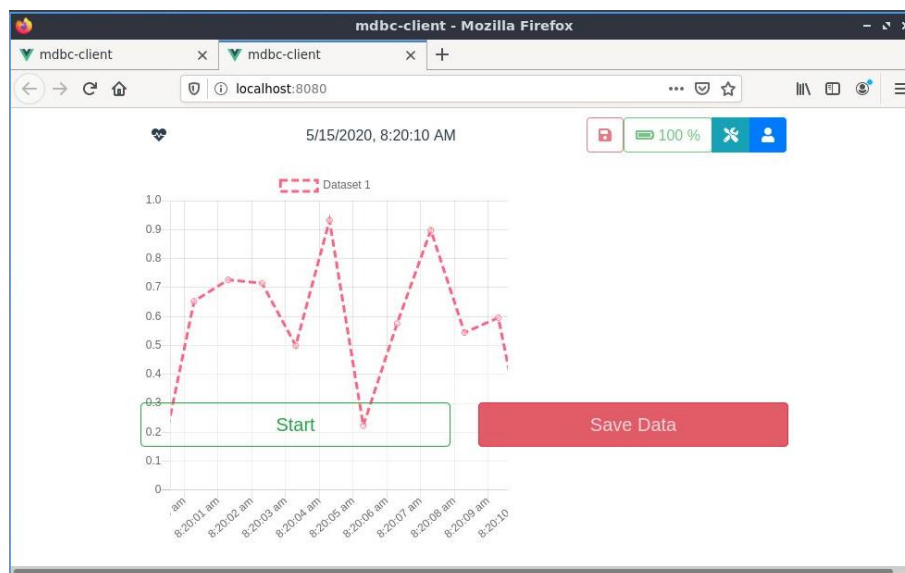
6.5 Architektura systému



Obrázek 6.1: Schéma aplikace s jednotlivými moduly, kdy šipky znázorňují komunikaci příslušných modulů s popisem .

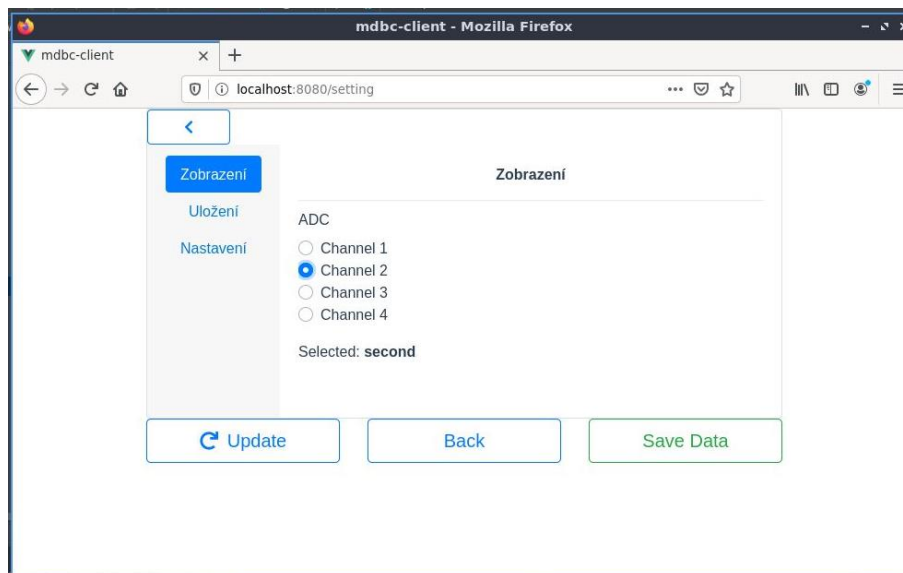
1. Modul GUI

Jedná se o modul sestavený pomocí VueJS, který řídí provoz systému a řeší uživatelské rozhraní. Popis GUI je znázorněn pomocí vytvořených záznamů obrazovky zkonstruovaného systému.



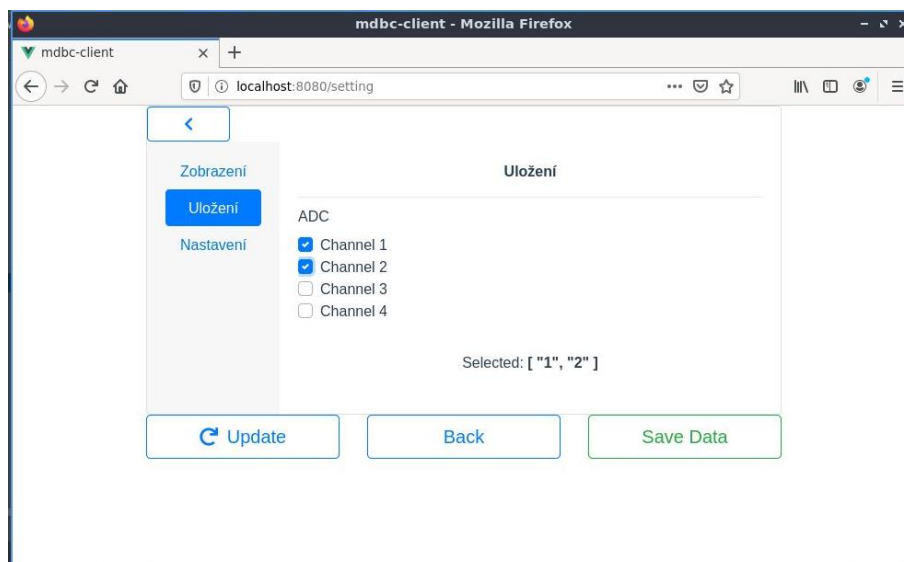
Obrázek 6.2: Domovská stránka aplikace pro uložení a zobrazení dat.

Obrázek 6.2 reprezentuje domovskou stránku systému, Aktivní graf je posunut vlevo z důvodu možnosti rozbalení ikon v pravém horním rohu. Pro uložení souboru hodnot naměřených zpětně, slouží tlačítko „Save data“. Tento příkaz uživateli zobrazí soubory uložené ve složce stažené, odkud si je uživatel může extrahovat na vlastní USB.



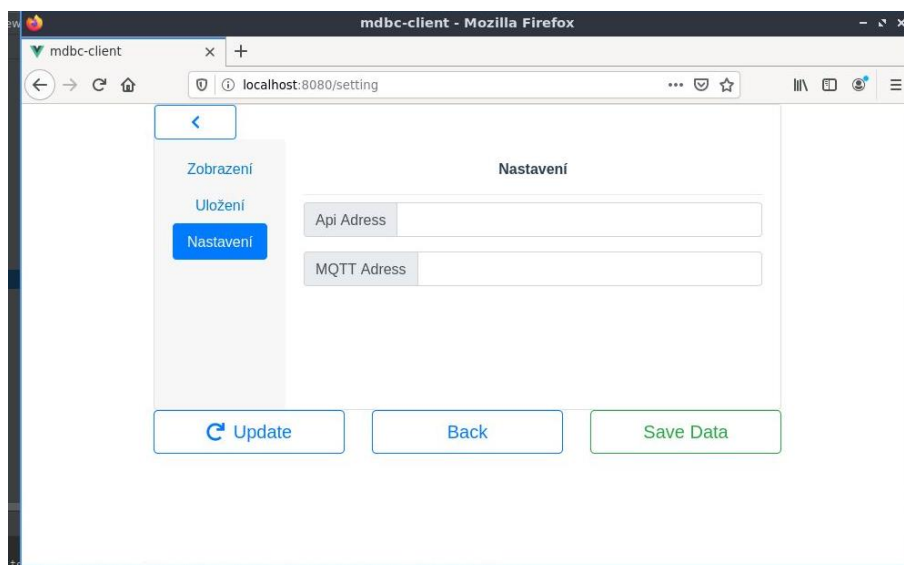
Obrázek 6.3: Nastavení zobrazení jednoho z kanálů AD převodníku. Sbíraná data budou zobrazována pomocí dynamického grafu na domovské stránce.

Po kliknutí na domovské stránce na ikonu nástroje se uživateli zobrazí možnosti nastavení jako je zobrazeno na obrázku 6.3. Možnost nastavit zobrazení požadovaného kanálu lze měnit i během měření. Lze zobrazit vždy jen jeden kanál, a to z důvodu malých rozměrů displeje. Pro volbu zpět na domovskou stránku je k dispozici tlačítko „Back“, nebo šipka v levém horním rohu. Tlačítko „Save data“ potvrzuje zvolené nastavení a tlačítko „Update“ provede aktualizaci softwaru.



Obrázek 6.4: Pro uložení dat má možnost si uživatel vybrat kanál, ze kterého bylo zaznamenáváno měření.

Pro ukládání dat z konkrétních kanálů AD převodníku slouží nastavení Uložení (znázorněno na obrázku 6.4). Uživatel má možnost zvolit 0 až 4 aktivní kanály, ze kterých budou data ukládána do výsledného souboru.



Obrázek 6.5: Rozšířené nastavení.

Lišta nastavení slouží pro obsluhu systému, kdy se nastaví adresa API, která zpracovává procesy na pozadí a fronta MQTT, která posílá data z pozadí na uživatelské rozhraní a naopak. Tato možnost nastavení je vhodná pouze pro správce systému.

2. ADC

Modul ADC zprostředkovává získávání dat z AD převodníku. Řeší především konfiguraci AD převodníku, kdy se při každém měření nastavuje automaticky vzorkovací frekvence, adresa I2C a počet bitů.

3. Broker

Message broker zajišťuje cestu proudících dat mezi konzumentem a producentem. Data ukládá do databáze skrz frontu MQTT SQLite.

4. Databáze

Tento modul zajišťuje ukládání dat naměřených AD převodníkem do databáze PostgreSQL, odkud se po skončení měření exportují do .csv souboru

Následující podkapitoly představují ty nejdůležitější metody pro funkčnost softwaru. Celý kód je k dispozici v příloze.

6.5.1 Konfigurace

Konfigurace obsahuje informace o nastavení databáze. Níže je příklad vytvoření databázové konfigurace pro AD převodník. Tímto způsobem byly vytvořeny konfigurace pro vybrané kanály, status měření (probíhající, ukončen), konkrétní měření a jeho hodnoty.

```
class AdcConfig(db.Model):
    __tablename__ = 'adc_config'

    id = db.Column(db.Integer, primary_key=True)
    primary_address = db.Column(db.DECIMAL, default=0x6a, nullable=False, unique=True)
    secondary_address = db.Column(db.DECIMAL, default=0x6b, nullable=False, unique=True)
    raw = db.Column(db.Boolean, default=True, nullable=False, unique=True)
    gain = db.Column(db.Enum(Gain), default=Gain.ONE, nullable=False, unique=True)
    rate = db.Column(db.Enum(BitRate), default=BitRate.TWELVE, nullable=False, unique=True)
    conversion_mode = db.Column(db.Enum(Mode), default=Mode.ONE_SHOT, nullable=False, unique=True)
    channels = db.relationship('AdcChannelConfig', lazy=True)
    default = db.Column(db.Boolean, default=True, nullable=False)
```

Obrázek 6.6: Příklad struktury kódu pro nakonfigurování databáze, konkrétně konfigurace AD převodníku.

6.5.2 Databáze

Pro vytvoření databáze bylo potřeba importovat knihovnu *db*, která zajišťuje práci s databázemi. Po spuštění měření byla interně volána metoda *find*, která vytvoří defaultně první databázi, kam jsou data ukládána (obrázek 6.7). Příklad vytvořené databáze je v kapitole 6.6.

```

@staticmethod
def find() -> AdcConfig:
    adc: AdcConfig = AdcConfig.query.filter_by(default=True).first()
    return AdcConfigService._create_default(adc)

@staticmethod
def _create_default(adc) -> AdcConfig:
    if not adc:
        adc: AdcConfig = AdcConfig()
        AdcConfigService.save(adc)
        adc.channels = AdcConfigService._default_channels(adc.id)
        db.session.commit()
    return adc

```

Obrázek 6.7: Příklad struktury metody pro vytvoření databáze pro konfiguraci AD převodníku.

6.5.3 Start měření

Pro start měření byla vytvořena metoda run, která spouští měření a sbírá hodnoty z AD převodníku a vytváří pole těchto hodnot a přiřazuje hodnoty příslušnému kanálu. Jedná se o samostatný nezávislý thread [34]. Příklad kódu níže na obrázku 6.8.

```

def run(self) -> None:
    measure = AdcMeasure()
    AdcMeasureService.save(measure)
    adc_measure_values: List[AdcMeasureValue] = []
    while self.__running:
        if self.__cache_counter < self.__max_value_per_minute:
            timestamp = datetime.utcnow()
            channel_values: Dict[int, float] = {}
            for channel in self.__channels:
                channel_values[channel] = self.__adc_differential_pi.read_voltage(channel)
            measure_channel_values: AdcMeasureValue = AdcMeasureValue(adc_id=measure.id, timestamp=timestamp, values=channel_values)
            adc_measure_values.append(measure_channel_values)
            self.__cache_counter += 1
        else:
            update_process = Process(target=update_values, args=(measure.id, adc_measure_values))
            update_process.start()
            self.__cache_counter = 0
            adc_measure_values.clear()
            time.sleep(self.__sleep)

```

Obrázek 6.8: Příklad struktury kódu pro start měření.

6.5.4 Stop měření

Pro zastavení měření byla vytvořena metoda stop, která detekuje v databázi měření, které má status označen IN_PROGRESS. Poté toto měření zastaví. Příklad kódu níže.


```

def stop(self):
    measure = AdcMeasureService.find_by_status(AdcMeasureStatus.IN_PROGRESS)
    adc_measure_worker: AdcMeasureWorker = self.__measure_processes.get(_uuid)
    if measure:
        measure.end_date = datetime.utcnow()
        measure.status = AdcMeasureStatus.FINISHED
        AdcMeasureService.save(measure)
    if adc_measure_worker:
        adc_measure_worker.terminate()
        self.__measure_processes.clear()

```

Obrázek 6.9: Příklad kódu pro zastavení měření.

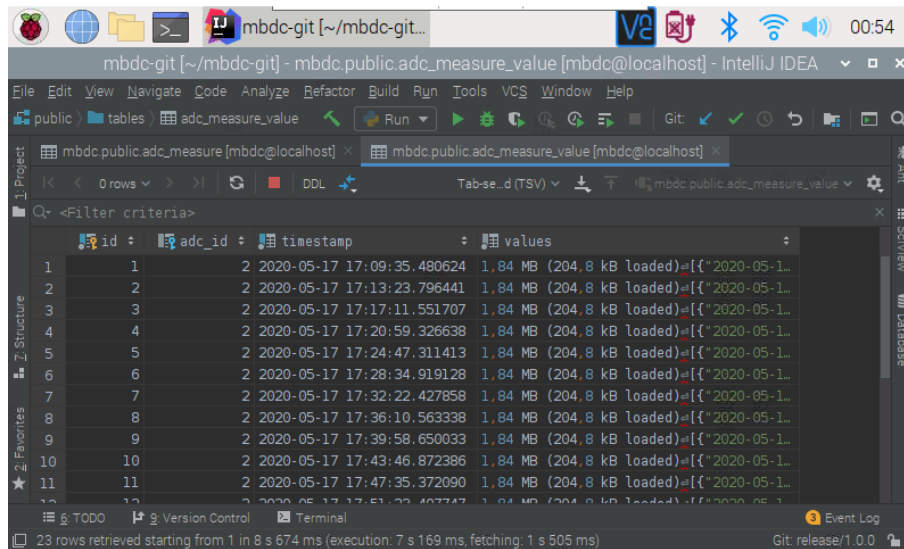
Další metody jako ukládání hodnot, vytvoření serveru atd. jsou sestrojeny na podobném principu jako metody uvedené v této kapitole. Celý zdrojový kód systému uveden v příloze na CD.

6.6 Procesy běžící na pozadí

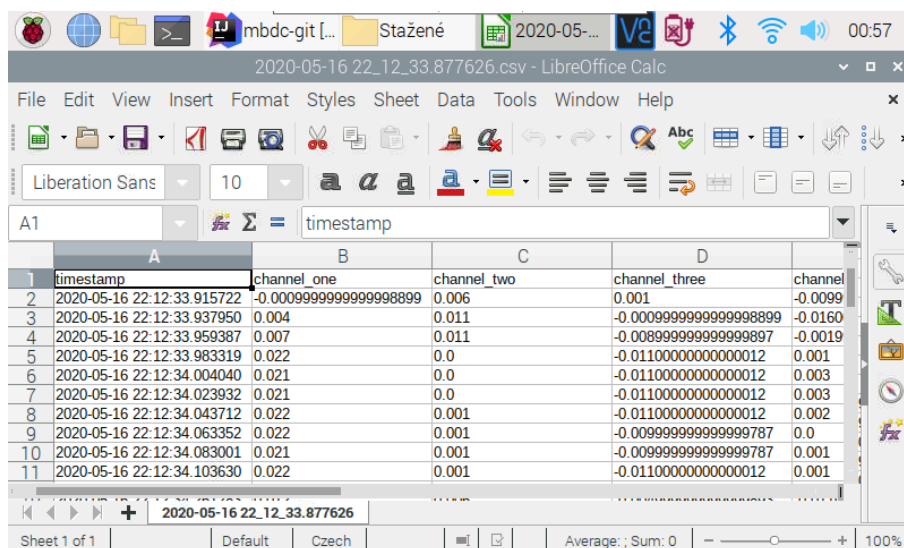
Na pozadí běžícího softwaru je vytvořena databáze, ze které jsou data zobrazována uživateli. Níže uvedené obrázky 6.10 – 6.12 obsahují zobrazení databází vytvořené v programovacím prostředí IntelliJ Idea, které obsahují sbíraná data a ukládají ID měření. Z těchto databází je poté vytvořen finální soubor naměřených dat, který má k dispozici uživatel.

id	status	start_date	end_date
1	FINISHED	2020-05-17 15:57:01.155976	2020-05-17 15:58:26.387863
2	FINISHED	2020-05-17 17:05:47.653345	2020-05-17 17:54:46.588432
3	FINISHED	2020-05-17 19:59:32.183370	2020-05-17 20:02:05.503462
4	FINISHED	2020-05-18 07:18:32.389729	2020-05-18 07:22:06.296762
5	FINISHED	2020-05-18 07:29:32.126251	2020-05-18 07:30:28.586560
6	FINISHED	2020-05-18 07:30:58.509420	2020-05-18 07:33:44.639979
7	FINISHED	2020-05-18 07:41:34.330744	2020-05-18 07:41:52.806746
8	FINISHED	2020-05-18 07:42:27.970923	2020-05-18 07:44:31.240445
9	FINISHED	2020-05-18 07:46:40.210679	2020-05-18 07:50:35.449276

Obrázek 6.10: Příklad databáze vytvořené v IntelliJ Idea, která obsahuje informace o jednotlivých měření. Obsahuje ID měření, informace o startu a ukončení měření a především zda je měření ukončeno, nebo stále probíhá (IN_Progress)



Obrázek 6.11: Informace o jednotlivých hodnotách konkrétního měření – ID konkrétní hodnoty, ID příslušného měření, časový údaj, velikost a konkrétní data zaslána AD převodníkem.



Obrázek 6.12: Příklad naměřených hodnot v souboru .csv vytvořeném mobilní jednotkou.

7 Návrh ověření systému

Ověření funkčnosti a porovnání mobilní jednotky s dílčími běžně používanými metodami záznamu bylo provedeno pomocí měření v laboratoři 1.LF UK. V první části proběhlo ověření funkčnosti a přesnosti pomocí generátoru signálu, kdy byla otestována funkčnost jednotky a přesnost měření. V druhé části měření byla sbírána data EKG a arteriální tlak z monitoru vitálních funkcí LifeScope a dále náklon lůžka pomocí inklinometru.

7.1 Metody ukládání dat

Pro ukládání měřených dat byl připojen přístroj, na kterém probíhalo měření, k systému sběru dat PowerLab a zároveň ke zkonstruované mobilní jednotce pomocí koaxiálních kabelů.

7.1.1 Použité přístroje

Tabulka uvedená níže 7.1 obsahuje využitě přístroje pro generaci a měření signálu, kdy na těchto signálech byla ověřována funkčnost mobilní jednotky a její přesnost naměřených hodnot. Jedná se o generátor signálu a další přístroje zaznamenávající fyziologická data, konkrétně EKG, arteriální tlak a náklon lůžka.

Tabulka 7.1: Použité přístroje pro otestování mobilní jednotky

Zařízení	Výrobce	Produkt	Parametry			
Generátor signálu	RIGOL	DG1022	-	-	-	-
Sběr dat	ADInstruments	PowerLab	<i>Počet kanálů</i>	<i>Vstup/Výstup</i>	<i>Vzorkovací frekvence</i>	<i>Počet bitů</i>
			16	Analog/Digital	400kS/s	16 bit
Měření EKG a tlaku	SpaceLabs Healthcare	91496	<i>Vstupy pro měření</i>	<i>Výstup</i>	<i>Zesílení</i>	
			EKG	Analog	1000	
			IBP		0	
Měření EKG a tlaku	Nihon Koden	LifeScope mu-631rk	<i>Vstupy pro měření</i>	<i>Výstup</i>	<i>Zesílení</i>	
			EKG	Analog	1000	
			IBP		0	

7.1.2 PowerLab ADInstruments

Jako referenční sběrač dat byl zvolen hardwarový systém PowerLab (ADInstruments). Systém PowerLab ukládá data do typu souboru *.adicht*, který je kompatibilní pouze se softwarem Labchart. Po otevření souboru v LabChart jej lze exportovat v podobě souboru *.mat* (soubor MATLAB).

7.1.3 Mobilní jednotka pro sběr dat

Mobilní jednotka sestavená v této práci disponuje vlastním softwarem pro ukládání dat do souboru *..csv*. Pro ukládání dat byla potřeba přístroj propojit koaxiálním kabelem s mobilní jednotkou. Vzorkovací frekvence AD převodníku byla nastavena na maximální hodnotu 240 Hz, lze však vzorkovací frekvenci upravit na nižší, ale to vzhledem k povaze signálu nebude využito. AD převodník mobilní jednotky snímá „surová“ data a poté jsou softwarově přepočítávána na hodnoty napětí (více v dokumentaci [19] a [35]). Mobilní jednotka byla ovládána pomocí displeje, kde má uživatel možnost zvolit z jakých kanálů budou data ukládána, spustit a zastavit měření a vybrat jaké měření si přeje stáhnout do svého zařízení (pokud jich je v historii více). Ovládání jednotky je popsáno v příloze A. Dále mobilní jednotka ukládá data do souboru *..csv*, kdy první sloupec značí časovou hodnotu, druhý sloupec obsahuje naměřené hodnoty napětí pro první kanál, třetí sloupec obsahuje hodnoty druhého kanálu, čtvrtý sloupec značí hodnoty třetího kanálu a pátý sloupec hodnoty čtvrtého kanálu. Tento soubor byl po ukončení nahrávání stažen z databáze do interní paměti Raspberry Pi do složky Stažené, ze které si uživatel může data stáhnout na vlastní externí medium.

7.2 Metody ověření funkčnosti mobilní jednotky

Pro ověření funkčnosti byla mobilní jednotka nejdříve připojena na generátor signálu, kde byla testována funkčnost jednotlivých kanálů a hodnocena přesnost měření. Byly zvoleny signály sinu, čtvercového signálu a trojúhelníkového signálu o frekvenci 5 Hz a 55 Hz s amplitudou 2 V, aby bylo zohledněno vstupních hodnot AD převodníku. Tato funkčnost byla ověřována bez implementace děliče napětí pro kanály 2, 3, a 4 (kanály pro arteriální tlak a dvouosý inklinometr). Přehled těchto signálů představuje tabulka 7.2.

Pro měření fyziologických dat bylo nejdříve nutno naměřená data synchronizovat v časové ose. Byl tedy pro aplikován takzvaný synchronizační pulz. V případě EKG byla odpojena a následně připojena elektroda a pro měření tlaku bylo prvních 10 sekund měření odpojen simulátor. Pro analýzu naměřeného signálu vybrán vždy vzorek dat (jedna vlna sinu, EKG apod.). Vybrané vzorky nebyly stejné délky z důvodu odlišné vzorkovací frekvence systému PowerLab a mobilní jednotky. Referenční signál měřen přístrojem PowerLab disponuje vzorkovací frekvencí 400 vzorků za sekundu a mobilní jednotka disponuje maximální vzorkovací frekvencí 240 vzorků za sekundu. Pro docílení

stejně délky hodnocených vektorů byl v programovacím prostředí MATLAB referenční signál „převzorkován“ pomocí metody *resample* na frekvenci mobilní jednotky.

7.2: Signály aplikované pro ověření funkčnosti mobilní jednotky generátorem signálu.

Signál	Frekvence [Hz]	
Sinus	5	55
Čtvercový signál	5	55
Trojúhelníkový signál	5	55

Srovnání naměřených fyziologických dat bylo provedeno s výstupem ze systému PowerLab. Výstupní soubory byly načteny do programovacího prostředí MATLAB, kde se provedlo statistické zhodnocení koeficientu determinace R^2 .

7.2.1 Metody měření EKG

Signál EKG byl měřen na probandovi pomocí monitoru vitálních funkcí Nihon Kodan LifeScope mu-631rk. Pro měření EKG lze využít 10svodový, 5svodový nebo 3svodový EKG kabel (pro každou elektrodu se u kabelů používají rezistory o odporu $1\text{ k}\Omega \pm 10\%$ zapojené v sérii). Maximální vstup tohoto modulu je $\pm 5\text{ mV}$ ($\pm 10\%$), kdy amplituda EKG je v jednotkách mV [8]. Při síťové frekvenci disponuje tento modul mírou potlačení součtového signálu (CMRR) do 110 dB. Zesílení signálu EKG monitorem vitálních funkcí je 1000. [36]

Sledování signálu EKG pomocí monitoru SpaceLabs je využito ve zdravotnickém zařízení VFN (Všeobecná fakultní nemocnice), kde měla být mobilní jednotka testována v řádu dnů.

Rozšiřující modul monitoru vitálních funkcí SpaceLabs snímá signál EKG ze 12 svodů. Výstup z monitoru vitálních funkcí je analogový (Obrázek 7.1), a proto je potřeba propojit monitor vitálních funkcí se systémem pro sběr dat pomocí koaxiálního kabelu. [36]



Obrázek 7.1: Zobrazení modulu SpaceLabs 91496. Analogový výstup je označen jako HLO – High Level Output. Převzato z [36].

7.2.2 Metody měření arteriálního tlaku

Arteriální tlak byl simulován pomocí připojené stříkačky. Data budou snímána jako v předchozí kapitole pomocí monitoru vitálních funkcí Nihon Koden LifeScope mu-631rk. Tento modul dokáže snímat rozsah hodnot -50 až 350 mmHg ± 2 mmHg. Pro přepočítání na napětí platí vztah 10 mV/mmHg [36].

7.2.3 Metody měření náklonu lůžka

Pro měření náklonu lůžka byl zvolen dvouosý inklinometr DOG2 MEMS SERIES CAN. Tento inklinometr měří náklon lůžka v rozsahu $\pm 90^\circ$, kdy -90° má hodnotu $4,5$ V, a $+90^\circ$ představuje hodnotu $0,5$ V, tedy hodnota 0° je $2,5$ V. Tento inklinometr disponuje vlastním AD převodníkem se vzorkovací frekvencí 100 Hz. [37]



Obrázek 7.2: Dvouosý inklinometr DOG2 MEMS SERIES CAN J1939. Převzato z [37].

7.2.4 Vyhodnocení podobnosti naměřených dat

Pro vyhodnocení spolehlivosti byla zvolena statistika R^2 , která patří mezi základní statistiky regrese a určuje takzvaný koeficient determinace Více o této statistické metodě v dostupné literatuře [38].

Výpočet této statistiky byl proveden pomocí metody *fitlm* obsažené v programovacím prostředí MATLAB, kdy výsledkem byla struktura obsahující všechny výpočty pro regresi včetně koeficientu determinace.

Jako reference pro ověření naměřených hodnot z generátoru signálu byl nasimulován v prostředí MATLAB referenční signál o stejných parametrech, které byly nastaveny generátorem signálu (Tabulka 7.5.1). Poté byl vybrán vzorek stejné délky jako referenční signál z naměřených dat jedním kanálem a byl porovnán s referencí. Z těchto hodnot byla vypočtena statistika R^2 .

Referenční signál pro měření fyziologických dat představují data naměřená přístrojem PowerLab, který disponuje vysokou vzorkovací frekvencí, a to 400 Hz. Pro srovnání s daty naměřenými mobilní jednotkou byly naměřené hodnoty systémem PowerLab exportovány pomocí příslušného softwaru LabChart do souboru *.mat* (soubor MATLAB), kdy jsou data a informace o měření uložena do struktury.

8 Výsledky

Před připojením mobilní jednotky ke generátoru signálu byla mobilní jednotka zapojena bez zatížení, aby byla ověřena správnost komunikace mezi softwarem a hardwarem. Fakta zjištěná při spuštění měření znázorňuje tabulka 8.1. Všechna měření byla provedena s nastavením vzorkovací frekvence 240 Hz. Data, která byla naměřena jsou uvedena v příloze na CD.

Tabulka 8.1.: Spuštění ukládání dat bez připojeného vstupního signálu

Počet měřených kanálů	Doba trvání měření [s]	Vzorkovací frekvence [Hz]	Počet hodnot za dobu trvání měření
1	76,60	188,00	14400
2	152,41	94,47	14400
3	227,86	63,20	14400
4	303,95	47,38	14400

Počet měřených kanálů představuje počet souběžně aktivních kanálů, kdy pro hodnotu 1 snímá a ukládá data pouze jeden z kanálů, pro hodnotu 2 jsou snímány hodnoty ze dvou kanálů atd. Doba trvání měření představuje časovou hodnotu, která reprezentuje čas, za který mobilní jednotka je schopna do databáze zapsat 14 400 hodnot.

8.1 Ověření pomocí generátoru signálu

Pro měření za účelem ověření přesnosti pomocí generátoru byly snímány hodnoty vždy pouze z jednoho kanálu pro dosažení maximální vzorkovací frekvence. Měření výstupního signálu z generátoru bylo provedeno bez implementace rozdílového zesilovače. Tabulky 8.2 – 8.7 uvedené v této kapitole reprezentují informace o konkrétním měření.

Tabulka 8.2: Informace o měření sinusového signálu pro jednotlivé kanály 5 Hz

Číslo kanálu	Doba trvání měření [s]	Vzorkovací frekvence [Hz]	Koeficient determinace
1	76,38	188,55	0,97
2	76,23	188,90	0,97
3	75,72	190,18	0,99
4	77,58	189,35	0,99

Tabulka 8.3: Informace o měření sinusového signálu pro jednotlivé kanály 55 Hz

Číslo kanálu	Doba trvání měření [s]	Vzorkovací frekvence [Hz]	Koeficient determinace
1	76,21	188,88	0,83
2	76,75	187,61	0,84
3	76,60	187,99	0,89
4	76,32	188,67	0,91

Tabulka 8.4: Informace o měření čtvercového signálu pro jednotlivé kanály 5 Hz

Číslo kanálu	Doba trvání měření [s]	Vzorkovací frekvence [Hz]	Koeficient determinace
1	76,56	188,08	0,96
2	76,63	188,69	0,97
3	76,53	188,15	0,86
4	76,41	188,46	0,95

Tabulka 8.5: Informace o měření čtvercového signálu pro jednotlivé kanály 55 Hz

Číslo kanálu	Doba trvání měření [s]	Vzorkovací frekvence [Hz]	Koeficient determinace
1	76,48	188,30	0,90
2	76,51	188,20	0,59
3	76,58	188,03	0,36
4	76,56	188,09	0,46

Tabulka 8.6: Informace o měření trojúhelníkového signálu pro jednotlivé kanály 5 Hz

Číslo kanálu	Doba trvání měření [s]	Vzorkovací frekvence [Hz]	Koeficient determinace
1	76,30	188,74	0,99
2	76,47	188,29	0,99
3	76,35	188,59	0,88
4	76,63	187,91	0,98

Tabulka 8.7 Informace o měření trojúhelníkového signálu pro jednotlivé kanály 55 Hz

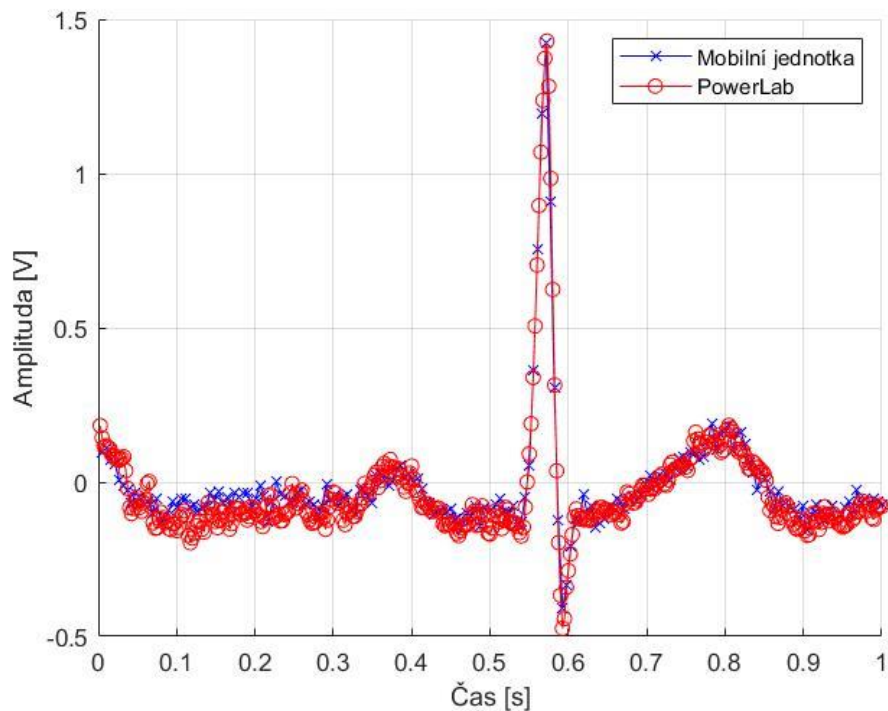
Číslo kanálu	Doba trvání měření [s]	Vzorkovací frekvence [Hz]	Koeficient determinace
1	76,12	189,10	0,98
2	76,33	188,65	0,97
3	76,18	189,02	0,98
4	76,33	188,66	0,99

8.2 Měření fyziologických dat

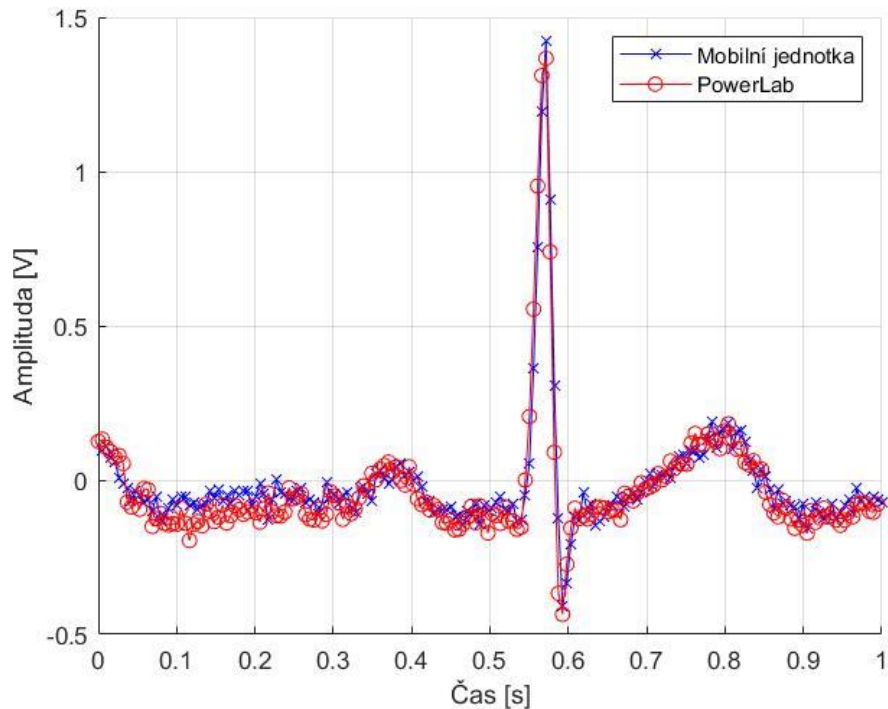
Pro měření fyziologických dat bylo zvoleno měření každého signálu samostatně na jednom kanálu, aby bylo dosaženo maximální vzorkovací frekvence mobilní jednotky. Na kanál číslo 2,3 a 4 byl implementován rozdílový zesilovač z důvodu korekce vstupních hodnot do rozsahu AD převodníku.

8.2.1 Měření EKG

Pro vyhodnocení naměřených dat EKG byla křivka časově synchronizována a poté vybrán jeden průběh EKG impulsu o délce trvání 1s. Pro přístroj PowerLab to značí 400 vzorků a pro mobilní jednotku 190 vzorků. Pro synchronizaci signálu byl proveden synchronizační puls, a to odpojení elektrody.



Graf 8.1: EKG vlna naměřena zařízením PowerLab (vzorky označené kroužkem) a sestavenou mobilní jednotkou pro sběr dat.



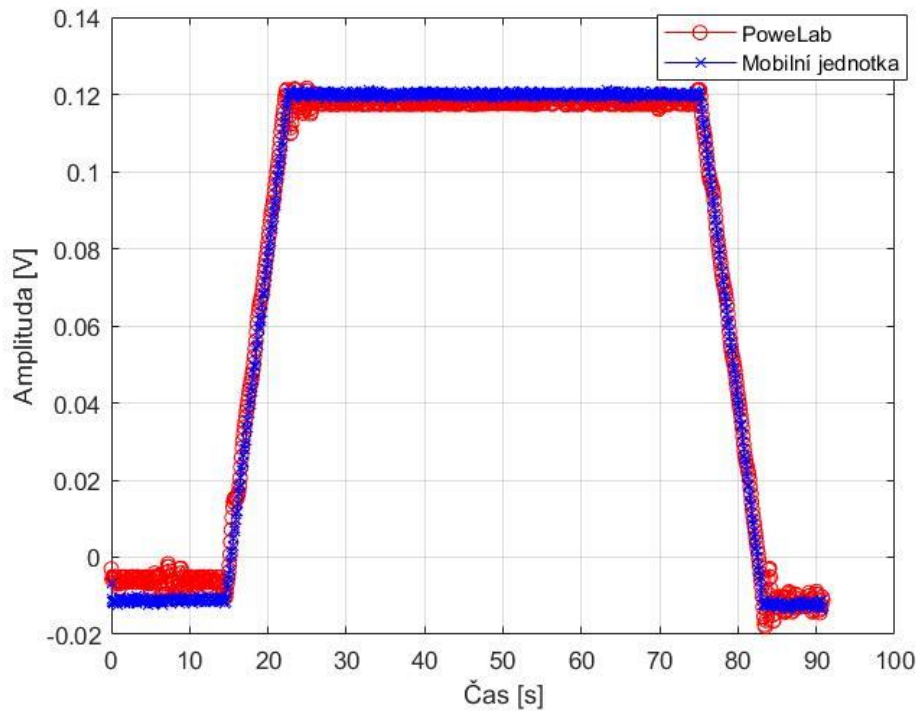
Graf 8. 2: EKG vlna naměřena zařízením PowerLab (vzorky označené kroužkem) a sestavenou mobilní jednotkou pro sběr dat po převedení vzorkovací frekvence PowerLab na hodnotu vzorkovací frekvence mobilní jednotky.

Tabulka 8.8: Minimální a maximální hodnota obsažena v signálu a koeficient determinace

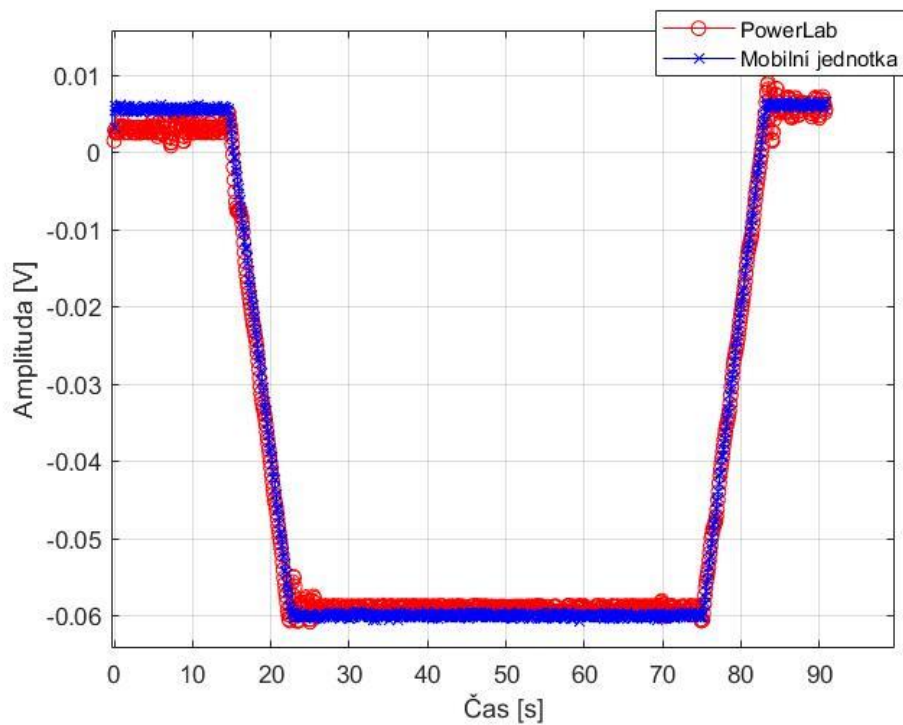
	PoweLab	Mobilní jednotka
Nejnižší hodnota naměřených dat	- 0,47 V	- 0,44 V
Nejvyšší hodnota naměřených dat	1,43 V	1,42V
Koeficient determinace	0,88	

8.2.2 Měření dvouosého inklinometru

Pro vyhodnocení naměřených dat arteriálního tlaku byla křivka časově synchronizována a poté vybrán jeden průběh změny polohy. Pro měření dvouosého inklinometru jsou vybrány kanály 3 a 4 z důvodu přítomnosti diferenciálního zesilovače. Tyto kanály zaznamenávají data souběžně pomocí stejné frekvence. Vzorkovací frekvence mobilní jednotky byla 66 Hz.



Graf 8.3: Průběh změny polohy na ose 1 ve vodorovné poloze. Z důvodu vybraného velkého časového úseku již zobrazen průběh s upravenou vzorkovací frekvencí PowerLab.



Graf 8.4: Průběh změny polohy na ose 2 ve svislé poloze. Z důvodu vybraného velkého časového úseku již zobrazen průběh s upravenou vzorkovací frekvencí PowerLab.

Tabulka 8.9: Minimální a maximální hodnota obsažena v signálu a koeficient determinace pro vodorovnou osu 1

Zařízení	PoweLab		Mobilní jednotka	
Nejnižší hodnota naměřených dat	-0,02V	11,40°	-0,01V	11,30°
Nejvyšší hodnota naměřených dat	0,12V	10,20°	0,12V	10,20°
Koeficient determinace	0,98			

Tabulka 8.10: Minimální a maximální hodnota obsažena v signálu a koeficient determinace pro svislou osu 2

Zařízení	PoweLab		Mobilní jednotka	
Nejnižší hodnota naměřených dat	-0,060 V	11,53°	-0,070 V	11,58°
Nejvyšší hodnota naměřených dat	0,009 V	11,22°	0,007 V	11,23°
Koeficient determinace	0,98			

8.2.3 Měření arteriálního tlaku

Pro vyhodnocení naměřených dat arteriálního tlaku byla křivka časově synchronizována pomocí synchronizačního pulzu a poté vybrán jeden průběh tlakové vlny. Byl vybrán vzorek o délce jedné sekundy. V této podkapitole jsou uvedeny informace o měření a výpočet koeficientu determinace.

Tabulka 8.11. Srovnání parametrů měření použitých zařízení

Zařízení	Vzorkovací frekvence	Počet vzorků
PoweLab	400 Hz	120
Mobilní Jednotka	99 Hz	99

Tabulka 8.12: Minimální a maximální hodnota obsažena v signálu a koeficient determinace

Zařízení	PoweLab	Mobilní jednotka
Nejnižší hodnota naměřených dat	0,86V	0,84V
Nejvyšší hodnota naměřených dat	1,48V	1,48
Koeficient determinace	0,74	

9 Diskuze

Cílem této práce bylo navrhnout a zkonstruovat mobilní systém pro měření analogových biosignálů z experimentálního a klinického prostředí. Požadavky na systém pro sběr dat zahrnují alespoň 4 analogové vstupu s dostatečnou vzorkovací frekvencí vzhledem k charakteristice sbíraných signálů. Primárně byly sbírány signály EKG, arteriální tlak a dva kanály byly obsazeny dvouosým inklinometrem. Z důvodu povahy signálu bylo nutno využít vhodnou vzorkovací frekvenci, aby nedošlo k podvzorkování a následnému zkreslení signálu. Frekvence 240 Hz byla zvolena jako dostačující na základě Nyquistova teorému. Mezi další požadavky patřilo sestavení robustního ovládání, zvolit dostatečnou velikost interní baterie a zvolení dostatečné velikosti paměťového média. V poslední řadě bylo nutno sestavený systém otestovat a následně porovnat s již s využívaným sběračem dat. Diskuze je rozčleněna do podkapitol kvůli přehlednosti. Každá podkapitola se věnuje specifické části práci.

9.1 Hardwarová část

Mobilní jednotka pro sběr dat aktuálně není ve vývoji příliš řešeným tématem, jelikož již trh nabízí několik možných řešení pro ukládání dat z měřicích přístrojů a aktuální studie se spíše zabývají bezdrátovým sběrem dat, což znamená zasílání dat pomocí mobilní brány, nebo jiné sítě, přímo do databáze pacienta. Požadavkům zadaným společnou experimentální laboratoří 1.LF UK a Nemocnice na Homolnce však žádný ze systému nevyhovoval z důvodu velkých rozměrů, absence zobrazovacího prvku a baterie, nebo potřeby dalšího příslušenství, jako je počítač, nebo software pro manipulaci s naměřenými daty.

Jako řídicí jednotka byl zvolen jednodeskový počítač Raspberry Pi 4B, jelikož disponuje dostatečnou RAM pamětí (4 GB). V případě, že by se jednalo o mobilní jednotku, která by nemusela obsahovat zobrazovací médium, tedy nemusela by disponovat uživatelským rozhraním, jako řídicí prvek by se dalo využít i Arduino. Arduino disponuje rozšiřovacími moduly pro sběr dat, ale pro zobrazení a převedení dat je potřeba právě další příslušenství, jako je počítač s programem LabView. Díky tomu, že Raspberry disponuje operačním systémem lze systém rozšířit o zobrazovací zařízení, na kterém uživatel bude moci obsluhovat aplikaci pro ukládání dat.

Z důvodu vstupních analogových signálů byl zařazen AD převodník. AD převodník ADC Differential Pi byl vybrán z důvodu kompatibility se systémem Raspberry Pi a přítomnosti dvou mikročipů o čtyřech vstupech, kdy je možnost rozšíření jednotky. AD převodník pro systém Raspberry Pi lze i sestavit napájením vybraného čipu na pájivé pole, ale složitost spočívá v nastavení komunikace se systémem Raspberry.

Pokud by se systém odpojil od zdroje bez záložního zdroje v podobě baterie, mohlo by dojít k poškození a ztrátě dat. Jako baterii lze použít i powerbanku o dostatečné kapacitě, ale uživatel by musel tuto baterii zvlášť nabíjet. Nevýhoda powerbanky, nebo jiné baterie spočívá v tom, že pokud se baterie vybije, systém bude odpojen od zdroje bez upozornění. Proto byla vybrána baterie s vlastním mikrokontrolerem, který zajišťuje bezpečné vypnutí systému v případě poklesu baterie na určitou hodnotu.

9.2 Softwarová část

Pro správné ukládání byla sestavena webová aplikace, která zajišťuje spuštění ukládání dat do databáze, výběr kanálů, ze kterých budou data sbírána. Webová aplikace byla využita z důvodu rozsáhlé dokumentace využitých frameworků a online zpracování dat (ukládání do databáze).

V kapitole 6. Realizace softwarové části jsou uvedeny frameworky a metody využitě při sestavování aplikace. Při sestavování aplikace bylo vyzkoušeno několik postupů a možností. Pro online sběr a zobrazení dat lze využít i hardwarové podpory pro Raspberry Pi programovacího jazyka MATLAB. Tato knihovna však využívá propojení počítače s plnohodnotným softwarem MATLAB se systémem Raspberry Pi skrz ssh (Secure shell) a vyžaduje připojení k síti, což je často v laboratořích nemožné. Ovládání a sběr dat probíhá vzdáleně na počítači a nelze ovládání označit za robustní. Proto nebyla tato alternativa dále rozvíjena. Je však ideální pro dlouhodobé sledování hodnot ze senzorů (teplota, vlhkost).

Pro dokonalejší aplikaci lze využít postupů jako je MVC (Model View Controller) nebo MVVM (Model View – View Model). Tyto postupy řeší komunikaci mezi zpracování dat a administrace (backend) a uživatelským rozhraním (frontend). V počátku tyto postupy byly vyzkoušeny při sestavování aplikace pomocí jazyku Python a využití knihovny GTK+ (knihovna pro tvoření GUI). Pro každou komponentu, kterou uživatel vidí (modální okna s chybou, domovská stránka, tlačítka atp.) bylo potřeba vytvořit tři skripty – model, handler a view. Model obsahoval veškeré metody a úkony, které by daná komponenta měla vykonávat. View obsahuje pouze vzhled dané komponenty a komunikuje skrz handler s modelem. Handler obsahuje metody pro volání metod z modelu, které jsou navěšené na View. Tato metoda vývoje aplikace je velmi složitá, ale spolehlivá. Vývoj aplikace mobilní jednotky pomocí MVC nebo MVVM nebyl zvolen z důvodu náročnosti a složitosti struktury

Pro sestavení uživatelského rozhraní byla sestaven návrh pomocí webové aplikace frameworku Vu.js, který obsahuje příjemné prostředí pro ovládání jednotky. Z důvodu složité konfigurace s REST API není komponenta GUI využita. V kapitole 6.5 je uveden tento návrh jako obrázek a popsány veškeré metody, které GUI uživateli nabízí. V příloze na CD je obsažen i kompletní kód tohoto návrhu ve složce GUI, ale bohužel není dokončena komunikace s REST API, která zajišťuje backendové procesy (start měření,

volání threadů atd.). REST API je plně funkční a je sestrojeno také jako webová aplikace (pomocí JSON) a pro uživatele splňuje robustní ovládání, které popsáno v příloze A.

Sestavený software byl v průběhu měření plně funkční a provedl všechny požadované akce, které vedly ke stažení výsledného souboru ve formátu .csv. V rámci vývoje lze software obohatit o navržené GUI a zdokonalit časovou náročnost, která se odvíjí od množství interně volaných metod, nebo importování potřebných knihoven. Dále by se funkčnost softwaru mohla zdokonalit na základě hardwarových komponent, kdy by se zaměnila SD karta za SSD disk. Tímto by se celý systém mohl zdokonalit a zrychlit. Dále by se práce mohla věnovat zabezpečení sítě a mobilní jednotka by tak mohla posílat naměřená data bezdrátově do databáze zdravotnického nebo experimentálně založeného zařízení a tím by ulehčila uživateli ukládání dat

9.3 Experimentální část

Experimentální měření bylo provedeno v laboratoři 1.LF UK, kdy fyziologická data byla snímána na dobrovolníkovi (arteriální tlak byl simulován pomocí injekční stříkačky, která vytvářela na senzoru tlak). Při experimentálním měření bylo zjištěno, že mobilní jednotka není schopna splnit požadovanou frekvenci 240 Hz, což může být způsobeno softwarem, který má vysokou časovou náročnost z důvodu velkého množství interních metod. Je také velmi pravděpodobné, že došlo při měření k přetížení systému, a proto vzorkovací frekvence není ideální.

Naměřené hodnoty EKG a tlaku byly v časové oblasti synchronizovány pomocí synchronizačního pulzu, který je popsán v kapitole 7.2. Po synchronizaci v časové oblasti byl vybrán jeden průběh signálu (jedna vlna EKG a jedna tlaková vlna), který byl dále statisticky hodnocen pomocí koeficientu determinace R^2 .

9.3.1 Ověření pomocí generátoru signálu

Nejdříve byla funkčnost jednotky ověřena pomocí generátoru signálu. Byl otestován každý kanál zvlášť, aby bylo docíleno nejvyšší možné vzorkovací frekvence. V kapitole 8.1. jsou uvedeny tabulky obsahující číslo měřeného kanálu, dobu měření, vzorkovací frekvenci a statistiku R^2 . Hodnoty jako je doba měření a vzorkovací frekvence jsou získány ze softwaru, kam byl přidán tzv. *log* vypisující tyto informace do konzole programovacího prostředí IntelliJIdea na základě zápisu informací do databáze. Pro měření hodnot tlaku a náklonu lůžka bylo potřeba vstupní signál přepočítat na hodnoty napětí, což bylo provedeno dle vztahů uvedených v dokumentaci příslušných přístrojů (dvousý inklinometr, monitor vitálních funkcí).

Tabulka 8.2 obsahuje informace o měření sinusového signálu o frekvenci 5 Hz. Výsledný koeficient determinace pro všechny kanály je od 0,97 do 0,99. Informace o měření sinusového signálu o frekvenci 55 Hz obsahuje tabulka 8.3. Na základě hodnoty

koeficientu determinace je patrné, že přesnost klesá. Hodnoty tohoto koeficientu se pohybují od 0,83 do 0,91.

Dalším testovaným signálem byl čtvercový signál. Informace o měřeném signálu o frekvenci 5 Hz obsahuje tabulka 8.4. Hodnoty koeficientu determinace nabývají hodnot od 0,87 do 0,97. Informace o měření čtvercového signálu 55 Hz obsahuje tabulka 8.5. Koeficient determinace nabývá hodnot 0,39 až 0,89. Tento rozsah může být takto velký z důvodu synchronizace dat a vybraného vzorku. Lze tedy tvrdit, že podobnost naměřených dat se odvíjí od vzorkovací frekvence a také synchronizace naměřených dat s referenčním signálem.

Tabulky 8.6 a 8.7 obsahují informace o měření trojúhelníkového signálu o frekvencích 5 Hz a 55 Hz. V tabulce 8.6 jsou uvedeny informace o měření signálu s frekvencí 5 Hz. Hodnoty koeficientu determinace nabývají hodnot od 0,97 do 0,99. Pro trojúhelníkový signál o frekvenci 55 Hz (tabulka 8.7) nabývá koeficient determinace hodnot od 0,97 do 0,99.

9.3.2 Měření fyziologických dat

Synchronizace signálů v časové ose byla provedena pomocí synchronizačního pulzu. Poté byl proveden výběr vzorků, který reprezentuje jeden průběh EKG nebo tlakové vlny a změny polohy ve vodorovné a svislé ose. Z důvodu odlišných vzorkovacích frekvencí jsou vektory vzorků jinak dlouhé a pro vyhodnocení koeficientu determinace bylo nutné referenční signál z přístroje PowerLab upravit pomocí změny vzorkovací frekvence. Nasnímaný signál z přístroje PowerLab byl převzorkován pomocí vzorkovací frekvence mobilní jednotky a poté byl vypočten koeficient determinace pomocí uvedených funkcí MATLAB.

Kapitola 8.2.1 se věnuje vyhodnocení EKG signálu. Vybraný průběh EKG vlny představuje pro systém PowerLab 400 vzorků a pro mobilní jednotku 190 vzorků. Graf 8.1 představuje vybranou konkrétní vlnu před provedením úpravy původní vzorkovací frekvence systému PowerLab a lze tak sledovat trend křivky naměřených hodnot mobilní jednotkou. Graf 8.2 znázorňuje srovnání hodnot po úpravě vzorkovací frekvence. Tabulka 8.8 obsahuje informace o naměřených hodnotách a koeficient determinace. Hodnota minima naměřeného mobilní jednotkou se liší o 0,03 V (relativní chyba je 0,03) a hodnota maxima o 0,01 V (relativní chyba je 0,01). Koeficient determinace je 0,88, což je pro potřeby vyhodnocení dostačující. Koeficient je nižší z důvodu odlišné vzorkovací frekvence.

Kapitola 8.2.2 se věnuje naměřeným hodnotám dvouosého inklinometru. V tomto případě bylo měření provedeno na dvou kanálech souběžně, kde byl implementován diferenční zesilovač, který korigoval vstupní hodnoty signálu pro inklinometr, tedy na kanálech 3 a 4. Pro hodnocení tohoto výstupu byl zvolen delší časový úsek, aby bylo možné sledovat trend křivky, a proto je na grafech 8.3 a 8.4 znázorněno porovnání s již

přepočítanou vzorkovací frekvencí referenčního signálu. Tabulka 8.9 obsahuje informace o naměřených hodnotách a koeficient determinace náklonu lůžka ve vodorovné ose. Hodnota minima se liší od referenčního signálu o $0,1^\circ$ (relativní chyba je 0,08) a hodnota maxima se liší o 0° (relativní chyba je tedy 0). Koeficient determinace pro toto měření dosahuje hodnoty 0,98. Tabulka 8.10 obsahuje informace o měření náklonu lůžka pro svislou osu. Hodnota minima se liší od referenčního signálu o $0,05^\circ$ (relativní chyba je 0,39) a hodnota maxima se liší o $0,01^\circ$ (relativní chyba je 0,19). Koeficient determinace má stejnou hodnotu jako v předchozím případě 0,98. Signál náklonu lůžka nedosahuje tak vysokých frekvencí jako arteriální tlak a EKG, proto má koeficient determinace takto vysokou hodnotu.

Kapitola 8.2.3 obsahuje naměřené hodnoty arteriálního tlaku, které byly měřeny na kanálu 2, kde je implementován diferenční zesilovač, který koriguje vstupní hodnoty signálu. Tabulka 8.11 představuje informace o vybraném vzorku signálu. V této kapitole není uvedeno grafické znázornění z důvodu posunu časové osy. K časovému posunu došlo pravděpodobně z důvodu zpomalení zapisování časových údajů do databáze, z důvodu přetížení systému. Vyhodnocení bylo provedeno vybráním stejného úseku signálu. Hodnota minima naměřené mobilní jednotkou se liší o 0,02V (relativní chyba je 0,03) a hodnota maxima se nijak neliší. Koeficient determinace je 0,74. Nižší hodnota koeficientu determinace je způsobena posunem časové osy hodnot naměřených zkonstruovaným systémem.

9.3.3 Limitace měření z důvodu pandemie Covid-19

Z důvodu vyhlášení nouzového stavu vládou ČR nebylo možné provést několikadenní záznam pro vyhodnocení kvality a spolehlivosti mobilní jednotky. Měření byla provedena těsně před odevzdáním diplomové práce v experimentální laboratoři 1.LF UK, která disponuje odlišnými měřicími přístroji, než je tomu v klinické praxi. Z důvodu časové tísně nebyla možnost přeměření fyziologických hodnot, což se projevilo na souboru měření arteriálního tlaku, kdy byl systém přetížen a časová osa byla značně posunuta. Z tohoto důvodu není v kapitole 8.2.3 uvedena grafická reprezentace naměřených hodnot. Pro ideální otestování spolehlivosti a kvality záznamu s běžně používanými metodami by bylo vhodné v další práci připojit zkonstruovanou mobilní jednotku pro sběr dat do klinického prostředí a snímat tak několikahodinové záznamy.

10 Závěr

V této diplomové práci byla navržen, zkonstruován a otestován mobilní systém pro sběr biosignálů v experimentálním prostředí. Zkonstruovaná mobilní jednotka disponuje kompaktností, dotykovým displejem pro robustní ovládání a interní baterií, která zajistí možnost manipulace během ukládání měřených dat. Mobilní jednotka disponuje čtyřmi aktivními analogovými vstupy, ze kterých lze snímat signál jednotlivě, ale i zároveň, dle nastavení uživatele. V této práci je zakomponován diferenční zesilovač, který koriguje vstupní signál do AD převodníku, aby nebyl překročen jeho vstupní rozsah. Kanál 1 je určen pro snímání EKG signálu, který je již zesílen modulem využitým při měření. Kanál 2 obsahuje zapojení diferenčního zesilovače pro korekci vstupních hodnot arteriálního tlaku. Kanály 3 a 4 také disponují zapojením diferenčního zesilovače pro korekci vstupních hodnot využitého inklinometru. Robustní ovládání je řešeno pomocí dotykového displeje s využitím webové aplikace REST API, která uživateli umožňuje nastavení aktivních kanálů, spustit a ukončit měření a stáhnout naměřené hodnoty uložené v souboru typu *.csv*. Zkonstruovaná mobilní jednotka byla otestována s využívaným systémem pro sběr dat v laboratoři 1. LF UK, kdy pro zhodnocení kvality byl využit výpočet koeficientu determinace. Koeficient determinace pro měření EKG, arteriálního tlaku a náklonu lůžka dosahoval hodnot 0,88, 0,98 a 0,74.

Přínos této práce je především v klinické praxi, kdy řešení sběru dat pomocí zkonstruované jednotky představuje pro zdravotnický personál pohodlnější obsluhu systému pro sběr dat. Výhodu představují hardwarové komponenty, které jsou na trhu dostupné. Další přínos této práce představuje zkonstruovaný software pro ukládání dat, který lze pomocí požadovaných úprav využít pro jakýkoliv hardwarový základ, jehož úkolem bude uložení dat.

11 Seznam použité literatury

- [1] MÜNZ, Jan, *Informační technologie ve zdravotnictví: informační systémy*. 3. Praha: České vysoké učení technické, 2011. ISBN 978-80-03-04720-0.
- [2] JOVANOV, Emil, Amanda O'DONNELL a Andy MORGAN. PROLONGED TELEMETRIC MONITORING OF HEART RATE VARIABILITY USING WIRELESS INTELLIGENT SENSORS AND A MOBILE GATEWAY. In: *Proceedings of the Second Joint EMBS/BMES Conference*. TX,USA: Houston, 2002, s. 1875-1876. ISBN 0-7803-7612-9. ISSN 0-7803-7612-9/02/\$17.00 © 2002 IEEE.
- [3] MUŽÍK, Jan a Karel HÁNA. Real-time BSPM processing system. *Springer*. Berlin, 2009, **2009**(22), 377-380. DOI: https://doi.org/10.1007/978-3-540-89208-3_89.
- [4] PENHAKER, M., M. STANKUS, M. PRAUZEK, O. ADAMEC, T. PETEREK, M. CERNY a V. KASIK. Advanced Experimental Medical Diagnostic System Design and Realization. *Elektronics and Eletrical Engineering*. 2012, **2012**(1117), 89-94. DOI: [dx.doi.org/10.5755/j01.eee.117.1.1060](https://doi.org/10.5755/j01.eee.117.1.1060).
- [5] MUZNY, Miroslav, Andre HENRIKSEN, Alain GIORDANENGO, Jan MUZIK, Astrid GRøTTLAND, Håvard BLIXGÅRD, Gunnar HARTVIGSEN a Eirik ÅRSAND. Wearable sensors with possibilities for data exchange: Analyzing status and needs of different actors in mobile health monitoring systems. *Elsevier*. 2020, **2020**(133). DOI: <https://doi.org/10.1016/j.ijmedinf.2019.104017>.
- [6] AL-ALI, A.R., M. AL-ROUSAN a T. OZKUL. Implementation of experimental communication protocol for health monitoring of patients. *Elsevier*. 2006, 28(5), 523-530. DOI: 10.1016/j.csi.2005.01.021. ISSN 09205489. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0920548905000152>

- [7] Real-time Biosignal Acquisition and Telemedicine Platform for AAL based on Android OS. *Proceedings of the 1st International Living Usability Lab Workshop on AAL Latest Solutions, Trends and Applications*. SciTePress - Science and and Technology Publications, 2011, , 111-121. DOI: 10.5220/0003303801110121. ISBN 978-989-8425-39-3. Dostupné z: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0003303801110121>
- [8] PENHAKER, Marek, Martin IMRAMOVSKÝ, Petr TIEFENBACH a František KOBZA. *Lékařské Diagnostické Přístroje*. Ostrava, 2004.. Učební Texty. VYSOKÁ ŠKOLA BÁŇSKÁ.
- [9] *PicoLog 1000 Series* [online]. [cit. 2020-01-22]. Dostupné z: <https://www.picotech.com/data-logger/picolog-1000-series/multi-channel-daq>
- [10] *PowerLab - Data with Integrity* [online]. [cit. 2020-01-22]. Dostupné z: <https://www.adinstruments.com/products/powerlab>
- [11] BHATTACHARJEE, Nirjhar. *A Real-Time Data Acquisition and Digital Signal Processing System for Neuromonitoring*. Silchar, 2015.. Master of Science. University of Cincinnati.
- [12] *USB-6002 Multifunction I/O Device* [online]. [cit. 2020-01-22]. Dostupné z: <https://www.ni.com/cs-cz/support/model.usb-6002.html>
- [13] REK, Tomáš. *Analyzátor AD převodníků*. Pardubice, 2009. Bakalářská práce. Univerzita Pardubice. Vedoucí práce Ing. Martin Hájek.
- [14] MLČEK, Mikuláš a João Batista BORGES. *Individualization of PEEP and Body Position in SARS-CoV-2 Associated ARDS*. 2020.
- [15] ROZMAN, Jiří. *Elektronické přístroje v lékařství*. 1. Praha: Academia, 2006. ISBN 80-200-1308-3.
- [16] HOTÁREK, Lukáš. *KOAXIÁLNÍ KABELY V KOMUNIKAČNÍCH SÍTÍCH*. Brno, 2013.. Bakalářská práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ.
- [17] DESMOND, Theventhira, Leow CHEE SENG a Vincent LEONG. Power Divider Rule: AC circuit analysis. *International Journal of Engineering Trends and Technolog*. 2019, **2016**(39-5), 274-283. DOI: 10.14445/22315381/IJETT-V39P246.

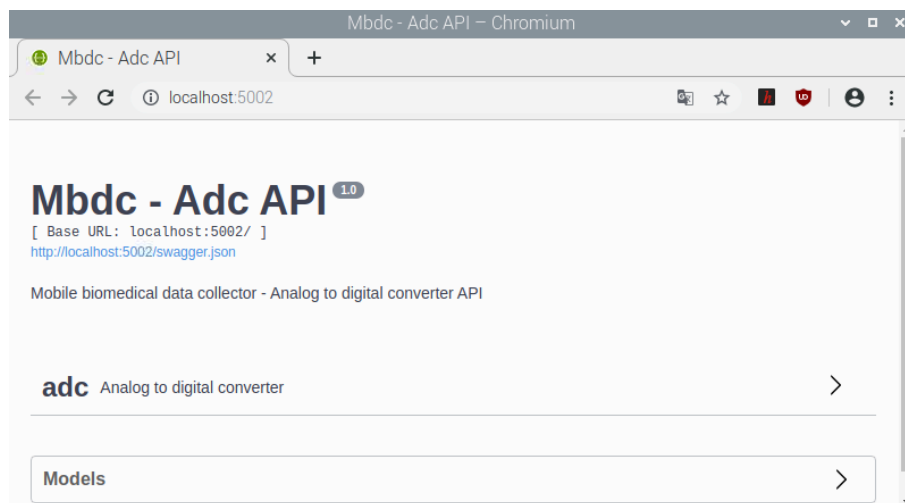
- [18] MICHAL, Cipryš. *SYSTÉM PRO SBĚR DAT S RASPBERRY PI*. Brno, 2019.. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce Ing. Tomáš Macho, Ph.D.
- [19] *ADC Differential Pi* [online]. In: . [cit. 2020-03-24]. Dostupné z: <https://www.abelectronics.co.uk/p/65/adc-differential-pi-raspberry-pi-analogue-to-digital-converter>
- [20] NEVORAL, Jan. *OVLÁDACÍ PANEL S DOTYKOVÝM DISPLEJEM*. Brno, 2015.. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ.
- [21] *PiJuice Quick Start Guide and FAQ* [online]. 2020 [cit. 2020-05-14]. Dostupné z: <https://learn.pi-supply.com/make/pijuice-quick-start-guide-faq/>
- [22] CHAUDHARI, Harshada. Raspberry Pi Technology: A Review. *International Journal of Innovative and Emerging Research in Engineering*. 2015, **3**(2), 83-87. ISSN 2394 - 3343.
- [23] BALL, Bill a Stephen SMOOGEN. *Sams' Teach Yourself LINUX in 24 Hours*. 1. Indianapolis, Indiana: Sams Publishing, 1998. ISBN 0-672-31162-3.
- [24] HALL, Tim a J-P STACEY. *Python 3 for Absolute Beginners*. 1. United States of America: Apress, 2009. ISBN 978-1-4302-1633-9.
- [25] *Vue: The Progressive JavaScript Framework: Documentation* [online]. Vue, 2020 [cit. 2020-04-20]. Dostupné z: <https://vuejs.org/>
- [26] FIELFING, Roy. *Architectural Styles and the Design of Network-based Software Architectures*. California, 2000.. Disertační práce. UNIVERSITY OF CALIFORNIA, IRVINE.
- [27] *Flask REST plus: Documentation* [online]. Python Software Foundation, 2019 [cit. 2020-04-19]. Dostupné z: pypi.org
- [28] TELNAROVÁ, Zdeňka. *ÚVOD DO DATABÁZÍ I*. Ostrava, 2020.. Distanční výuková opora. Ostravská univerzita.
- [29] *Flask SQLAlchemy: Documentation* [online]. palletsprojects, 2019 [cit. 2020-04-20]. Dostupné z: <https://flask-sqlalchemy.palletsprojects.com/en/2.x/>

- [30] RELAN, Kunal. *Building REST APIs with Flask: Create Python Web Services with MySQL* Kunal. 1. New Delhi, Delhi, India: Apress, 2019. ISBN 978-1-4842-5022-8.
- [31] MATTHEW, Neil a Richard STONES. *Beginning databases with PostgreSQL: from novice to professional*. 2nd ed. Berkeley: Apress, 2005. The expert's voice in open source. ISBN 15-905-9478-9.
- [32] *PostgreSQL: The World's Most Advanced Open Source Relational Database: Documentation* [online]. PostgreSQL Global Development Group, 2020 [cit. 2020-04-20]. Dostupné z: <https://www.postgresql.org/>
- [33] *MQTT: Documentation* [online]. Google groups, 2019 [cit. 2020-04-19]. Dostupné z: <http://mqtt.org/>
- [34] HYDE, Paul. *Java Thread Programming*. 1. Minnesota: Sams Publishing, 1999. ISBN 0-672-31585-8.
- [35] *ABElectronics_Python_Libraries/ADCDifferentialPi/ADCDifferentialPi.py: Alan Kinnaman* [online]. [cit. 2020-05-14]. Dostupné z: https://github.com/abelectronicsuk/ABElectronics_Python_Libraries/blob/master/ADCDifferentialPi/ADCDifferentialPi.py
- [36] *SpaceLabs Healthcare Ovládací modul (91496): Documentation*. 2014.
- [37] TE CONNECTIVITY. *DOG2 MEMS SERIES CAN J1939 INCLINOMETER: Documentation*. 2017.
- [38] MRKVIČKA, Tomáš a Vladimíra PETRÁŠKOVÁ. *Úvod do statistiky*. České Budějovice: Jihočeská univerzita, 2006. ISBN 80-704-0894-4.
- [39] *Raspberry Pi* [online]. [cit. 2020-03-06]. Dostupné z: <https://www.raspberrypi.org/>
- [40] SEMMLOW, John. *Biosignal and biomedical image processing: MATLAB-based applications*. New York: Marcel Dekker, 2004. Signal processing and communications series. ISBN 08-247-4803-4.
- [41] *HBMQTT: Documentation* [online]. Nicolas JOUANIN, 2015 [cit. 2020-04-19]. Dostupné z: <https://hbmqtt.readthedocs.io/en/latest/#>
- [42] AGGARWAL, Shalabh. *Flask Framework Cookbook*. 1. Birmingham: Packt Publishing, 2014. ISBN 978-1-78398-340-7.
- [43] *Flask: Documentation* [online]. Pallets, 2010 [cit. 2020-04-19]. Dostupné z: <https://flask.palletsprojects.com/en/1.1.x/#user-s-guide>

- [44] HORVÁTH, Jakub. *Implementace aplikací pro MQTT komunikaci*. Ostrava, 2017.. Bakalářská práce. VYSOKÁ ŠKOLA BÁŇSKÁ. Vedoucí práce Ing. Jakub Šafařík.
- [45] *SQLite: Documentation* [online]. SQLite, 2020 [cit. 2020-04-20]. Dostupné z: <https://www.sqlite.org/index.html>
- [46] *Signal Processing of Random Physiological Signals*. 1. Texas: Morgan & Claypool, 2006. ISBN 1598290398.
- [47] WEITKUNAT, Rolf, ed. *Digital biosignal processing*. Amsterdam: Elsevier Science Publishers, 1991. Techniques in the behavioral and neural sciences, vol. 5. ISBN 0-444-89144-7.
- [48] ŠILHÁNEK, Jan. *Zpracování signálů v elektroenergetice*. Praha, 2017.. Diplomová práce. České Vysoké Učení Technické. Vedoucí práce Jan Kyncl.
- [49] ČERNÍK, Martin. *Elektrické obvody: teorie a příklady*. Liberec, 2014. ISBN 978-80-7494-161-0.. Skripta. TUL.

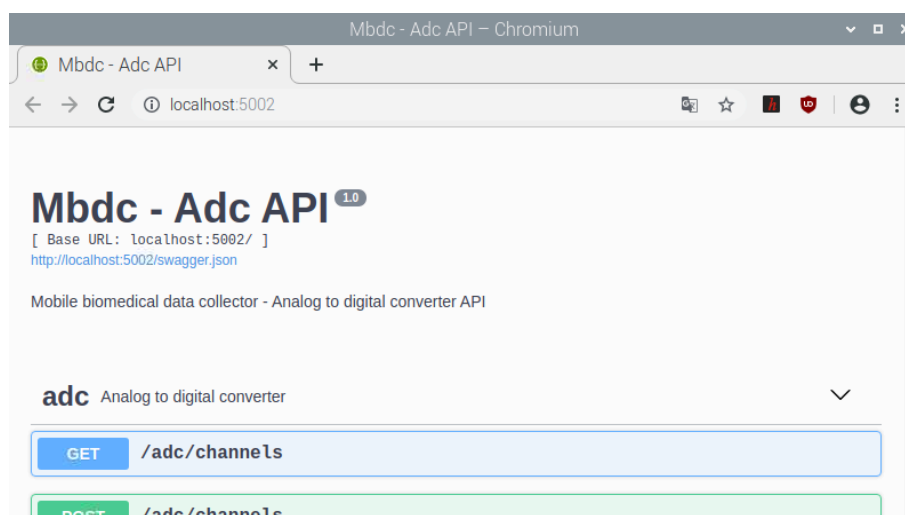
Příloha A: Ovládání jednotky

Před zapnutím mobilní jednotky je potřeba zkontrolovat zapojení zdrojových kabelů od baterie a napájení. Po připojení mobilní jednotky ke zdroji se Raspberry Pi automaticky zapne a spustí se webová aplikace s uživatelským rozhraním na obrázku 1.



Obrázek 1: Server pro obsluhu měření a ukládání dat

Po zvolení možnosti „adc“ se zobrazí dostupné metody pro ukládání dat a další možnosti, které mobilní jednotka nabízí, jako je například ověření stavu, zobrazení aktivních kanálů atd.



Obrázek 2: Zobrazení metod pro správu měření a ukládání dat. Pro zobrazení všech metod, které jsou k dispozici, uživatel může pomocí lišty vlevo pohodlně mezi metodami přecházet.

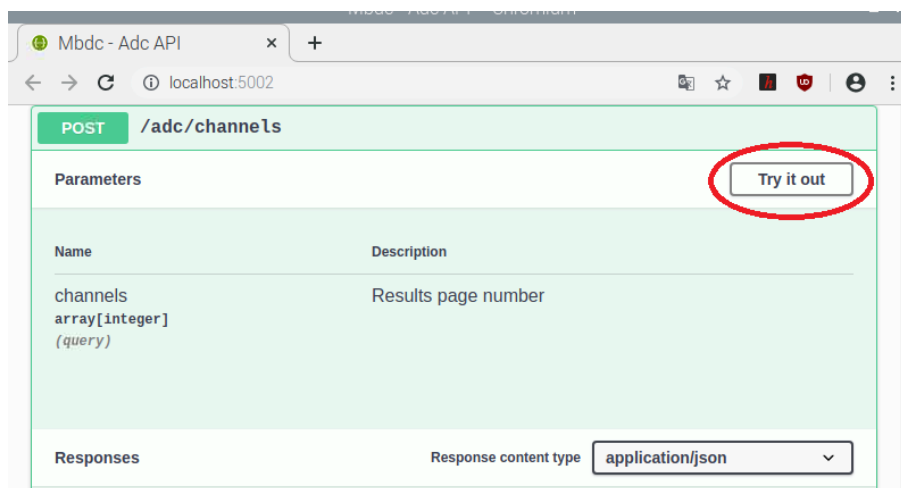
Spuštění ukládání dat

Na běžícím serveru zobrazeného na obrázku 5 je nutno rozkliknout možnost „adc“. Po zvolení této možnosti se rozbalí metody pro obsluhu ukládání dat. Jsou obsaženy metody: „Get ADC channels“, „Post ADC channels“, „Download“, „Measures“, „Start“, „Status“, „Stop“. Podkapitoly níže popisují využití každé metody. Pro spuštění měření stačí vybrat kanály pomocí metody „Post Channels“, zvolit metodu „Start“ a po skončení měření zvolit metodu „Stop“, zjistit ID měření pomocí metody „Measures“ a v metodě „Download“ pomocí ID stáhnout hodnoty do souboru *.csv*. Stažený soubor se automaticky uloží do složky stažené, odkud si jej uživatel může pohodlně přenést na vlastní externí úložné médium.

Po rozkliknutí požadované metody je nutno potvrdit výběr pomocí tlačítka „Try it out“ (vyznačeno na obrázku 7) a opět metodu potvrdit pomocí tlačítka „Execute“ (vyznačeno na obrázku 8). Tento postup potvrzení je u všech metod stejný.

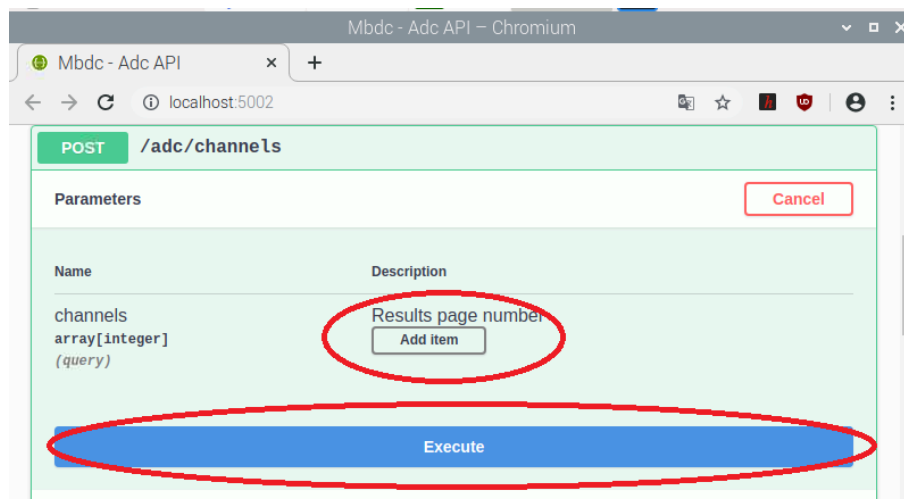
Výběr kanálů pro uložení

Pro výběr kanálů, ze kterých budou ukládána data je k dispozici metoda „Post channels“. Po rozkliknutí nabídky je potřeba pro načtení nastavení zvolit možnost „Try it out“.

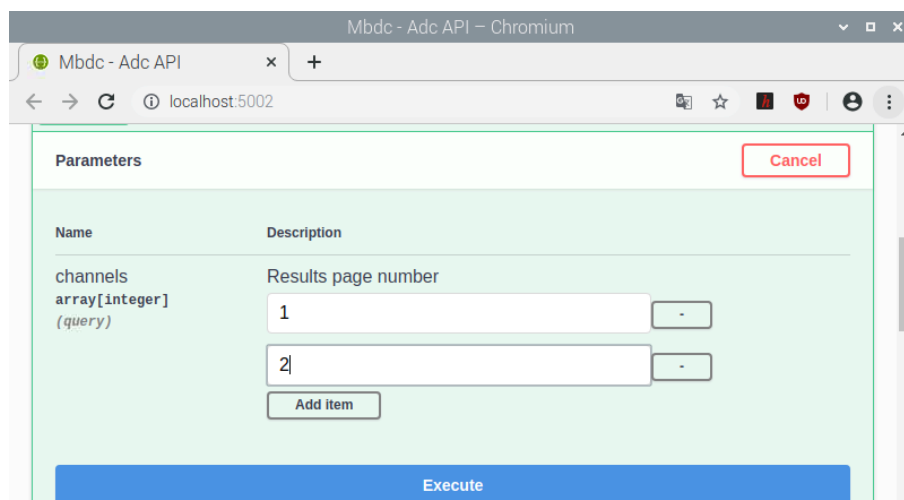


Obrázek 3: Načtení nastavení výběru kanálů (vyznačeno)

Po načtení nastavení je nutno zvolit možnost „Try it out“ a zvolit číslo kanálu. Pro přidání dalšího kanálu je nutno zvolit možnost „Add item“. Potvrzení výběru se provede pomocí tlačítka „Execute“.



Obrázek 4: Přidání kanálu (Add item) a potvrzení výběru pomocí tlačítka Execute (vyznačeno)

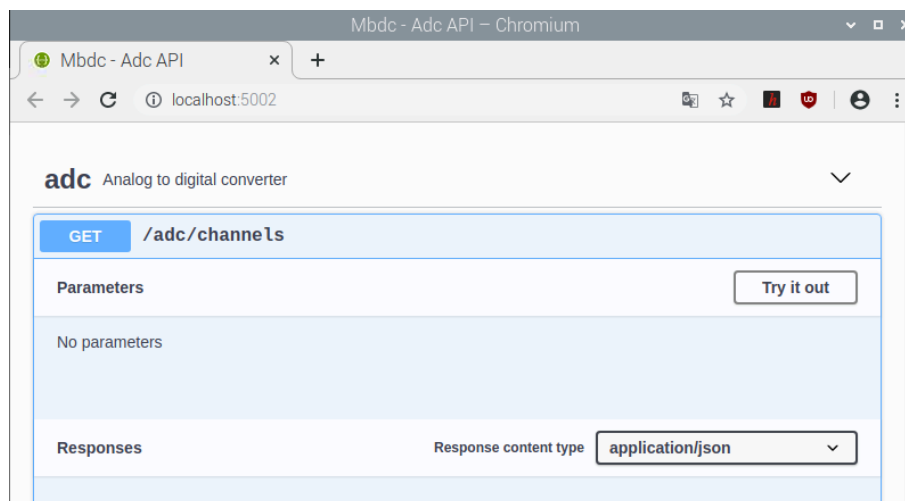


Obrázek 5: Potvrzení výběru tlačítkem Execute

Pro ověření aktivních kanálů je k dispozici metoda „Status“, která obsahuje informace o vybraných kanálech, které jsou aktivní.

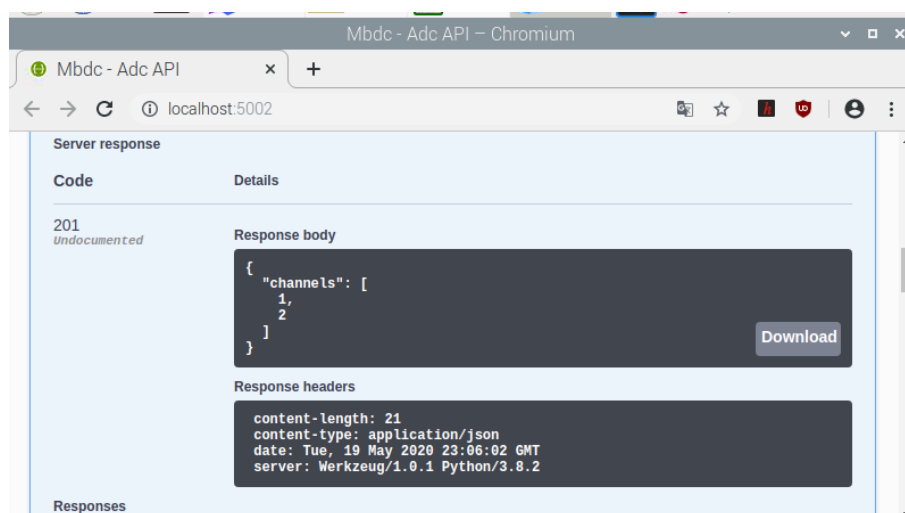
Přehled aktivních kanálů AD převodníku

Pro zjištění aktivních kanálů AD převodníku je k dispozici metoda „get ADC channels“. Po zvolení této metody je potřeba zvolit možnost „Try it out“ a potvrdit metodu tlačítkem „Execute“.



Obrázek 6: Zvolení metody Get channels. Je potřeba zvolit možnost Try it out a následně tlačítko Execute, které se objeví po zvolení možnosti Try it out.

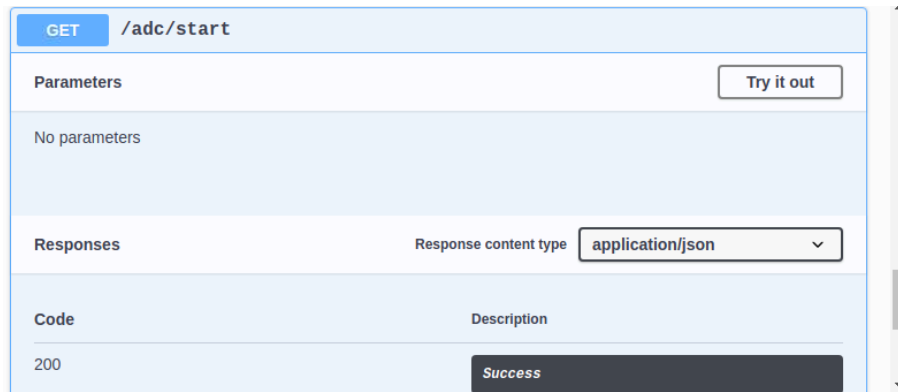
Pro potvrzení zvolení metody pomocí tlačítka „Execute“ se zobrazí aktivní kanály, ze kterých je prováděno uložení.



Obrázek 7: Zobrazení aktivních kanálů.

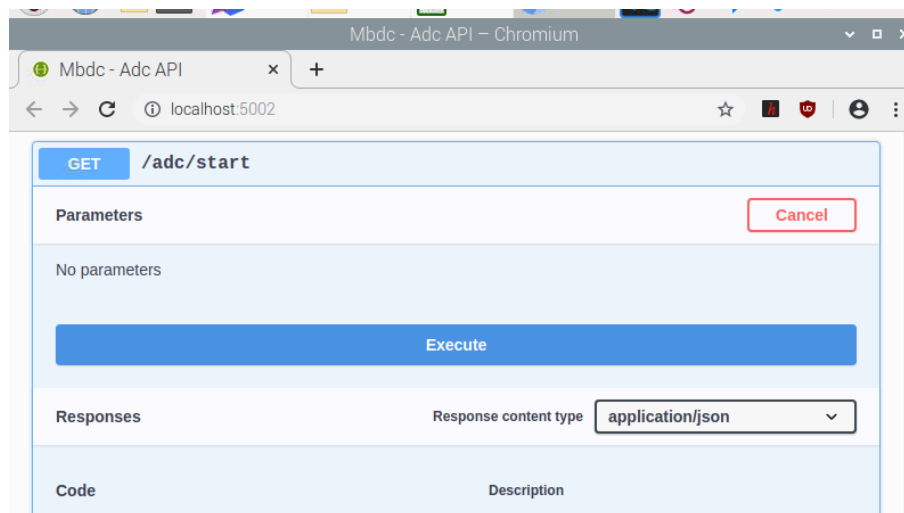
Start měření

Pro zvolení začátku měření je potřeba zvolit metodu Start. Potvrzení výběru se provede pomocí tlačítka „Try it out“.



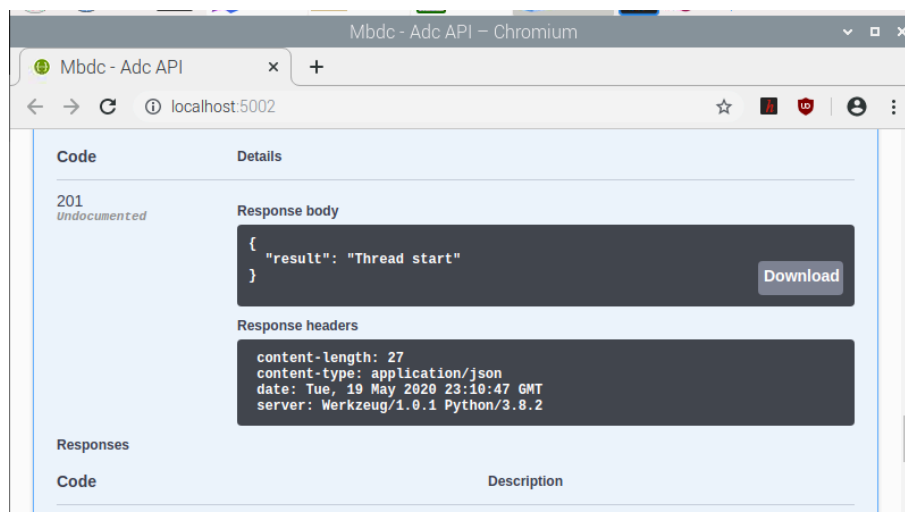
Obrázek 8: Pro potvrzení výběru metody je potřeba zvolit možnost *Try it out*, aby bylo možno potvrdit a načíst vybranou metodu.

Dále pro spuštění nahrávání je potřeba zvolit začátek měření pomocí tlačítka „Execute“.



Obrázek 9: Možnost *Execute*

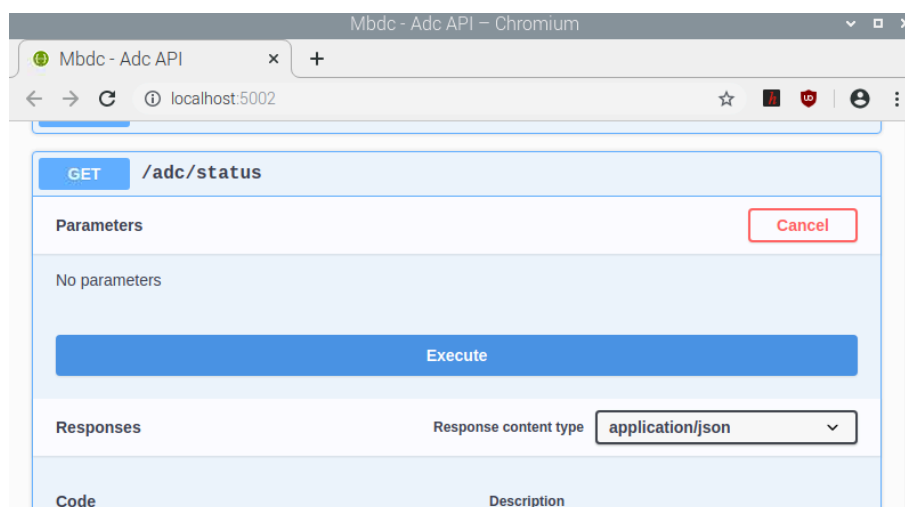
Po potvrzení začátku měření je vypsána informace o spuštění threadu, který snímá data z AD převodníku a ukládá data do databáze.



Obrázek 10: Informace o spuštění ukládání hodnot z AD převodníku.

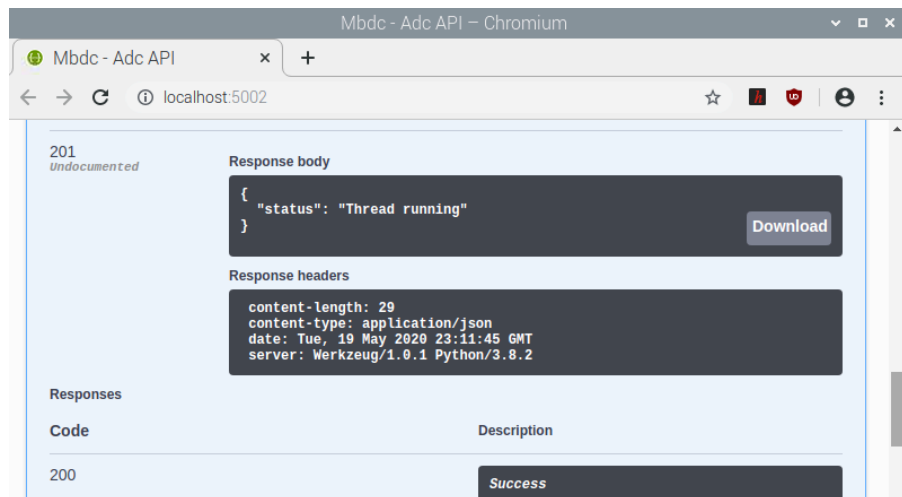
Informace o měření

Pro informace o spuštění, nebo zastavení měření je k dispozici metoda „Status“. Tato metoda zobrazí, zda na pozadí běží měření hodnot z AD převodníku. Po zvolení metody je potřeba provést opět potvrzení o výběru metody pomocí „Try it out“ a „Execute“.



Obrázek 11: Zvolení metody status o informaci o měření.

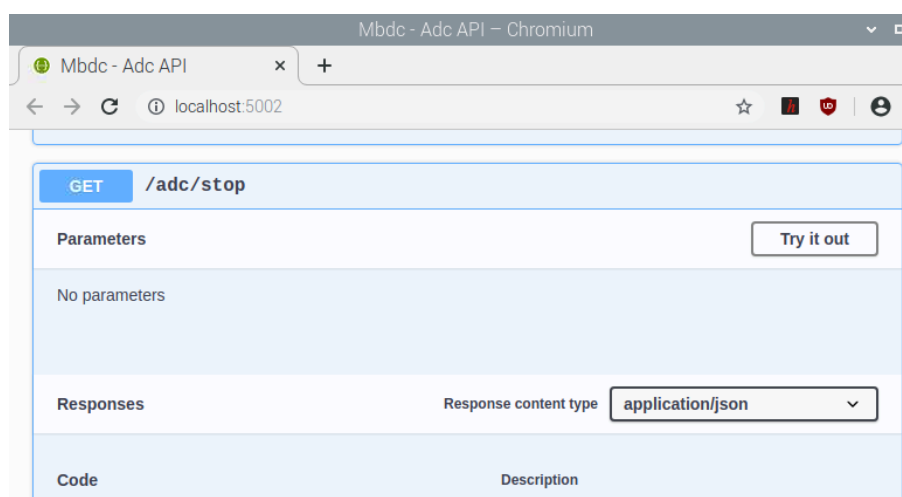
Po zvolení metody status a její načtení pomocí „Try it out“ je potřeba zvolit pro vypsaní výsledku tlačítko „Execute“.



Obrázek 12: Vypsání informací ohledně měření.

Zastavení měření

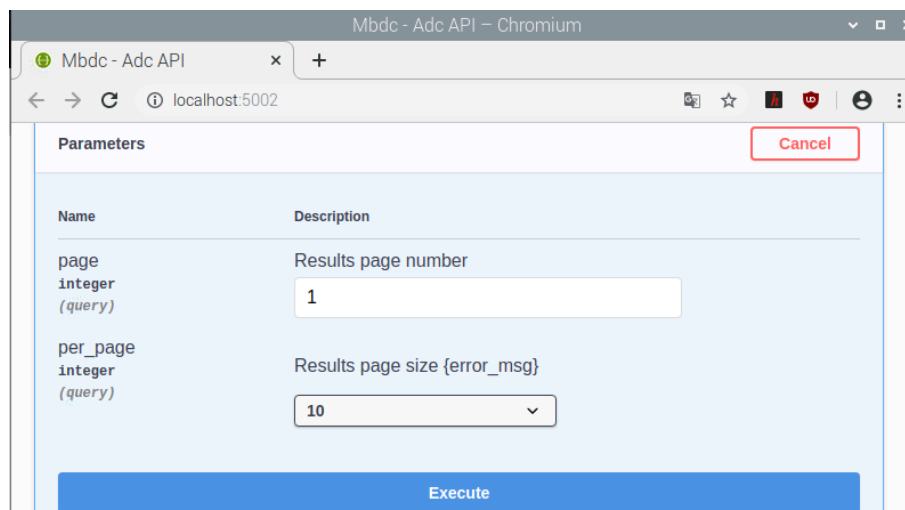
Pro zastavení měření je vytvořena metoda „Stop“. Postup pro zastavení měření je stejný jako v předchozí kapitole, pouze je zvolena metoda „Stop“.



Obrázek 13: Zvolení možnosti zastavení měření. Po zvolení možnosti Try it out a následně Execute se zobrazí informace o zastavení měření.

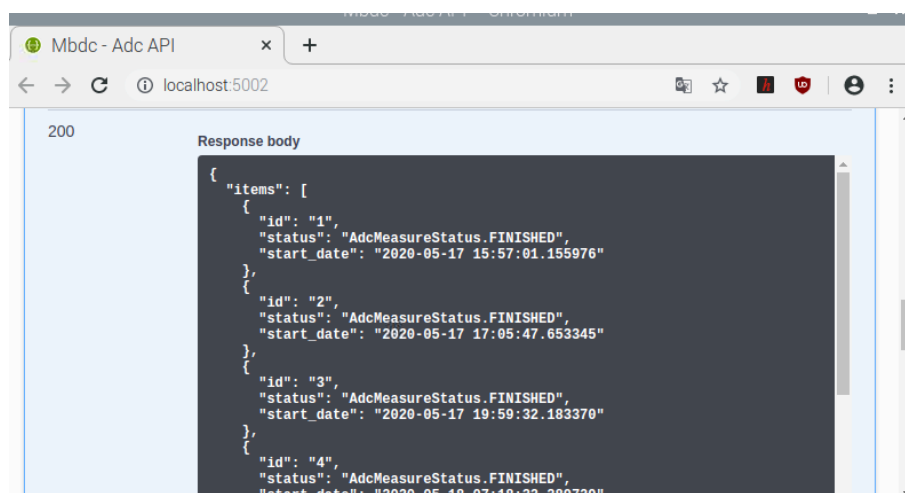
Získání ID měření

Aby bylo možno konkrétní měření stáhnout, je potřeba získat jeho ID. Toto ID je uloženo v metodě „Measures“, která obsahuje všechna měření uložena v historii.



Obrázek 14: Výběr metody measures po potvrzení pomocí tlačítka Try it out

Pro výběr měření je potřeba vybrat počet měření zobrazených na stránku a poté potvrdit výběr pomocí tlačítka „Execute“. Po potvrzení se zobrazí všechna měření uložená v databázi.

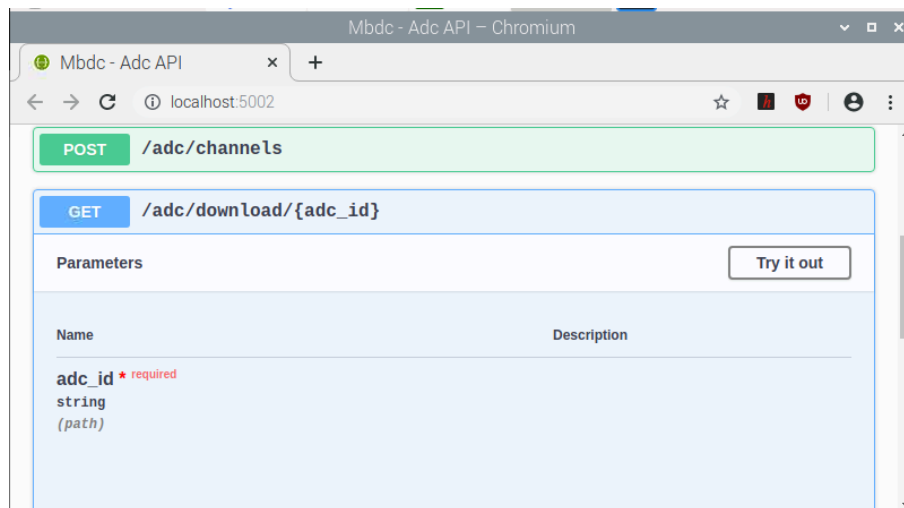


Obrázek 15: Zobrazení měření v databázi, ID, jeho informace zda je ukončen (Finished), případně v procesu (In Progress) a startu měření.

Pro stažení ve formátu .csv je nutno si zapamatovat ID požadovaného měření. Samostatné stažení provádí metoda „Download“.

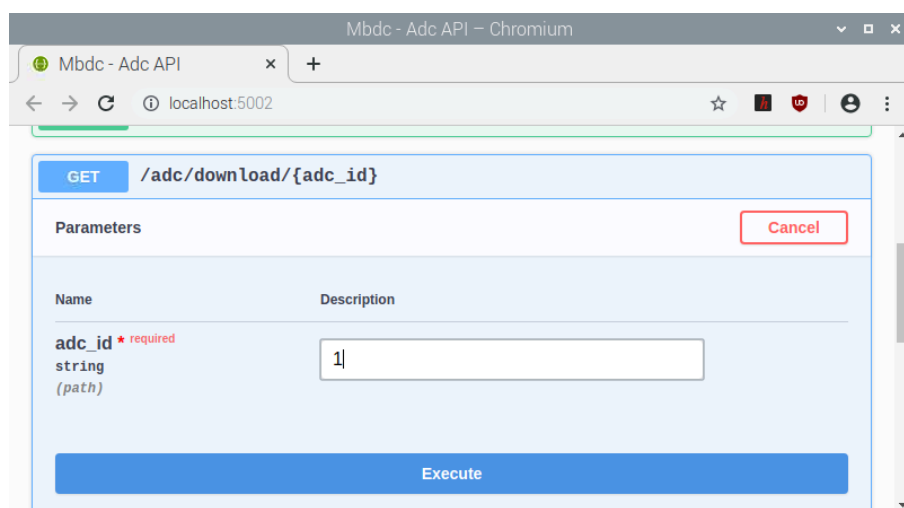
Stažení měření ve formátu .csv

Pro stažení konkrétního měření je potřeba znát jeho ID. Tento proces zajišťuje metoda Download.



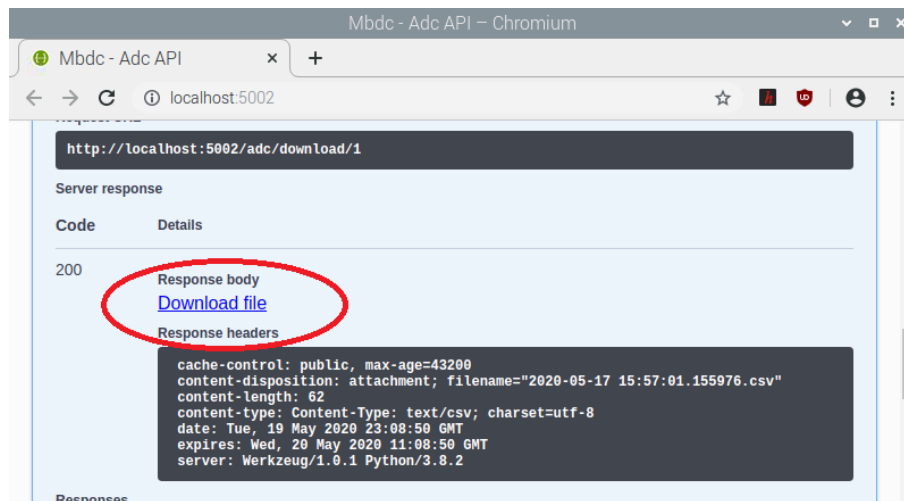
Obrázek 16: Metoda Download, kdy pro její načtení je potřeba opět zvolit Try it out.

Po načtení metody se zobrazí pole pro zadání hodnoty ID požadovaného měření a tlačítkem „Execute“ se tato volba potvrdí.



Obrázek 17: Zadání ID do vstupního okna a potvrzení výběru pomocí tlačítka Execute.

Po potvrzení výběru se následně zobrazí možnost „Download file“. Tato metoda zajistí stažení naměřených hodnot do souboru ve formátu .csv do složky Stažené na Raspberry Pi, odkud si jej uživatel může zkopírovat do svého přenosného úložiště.



Obrázek 18: Zvolení odkazu Download file

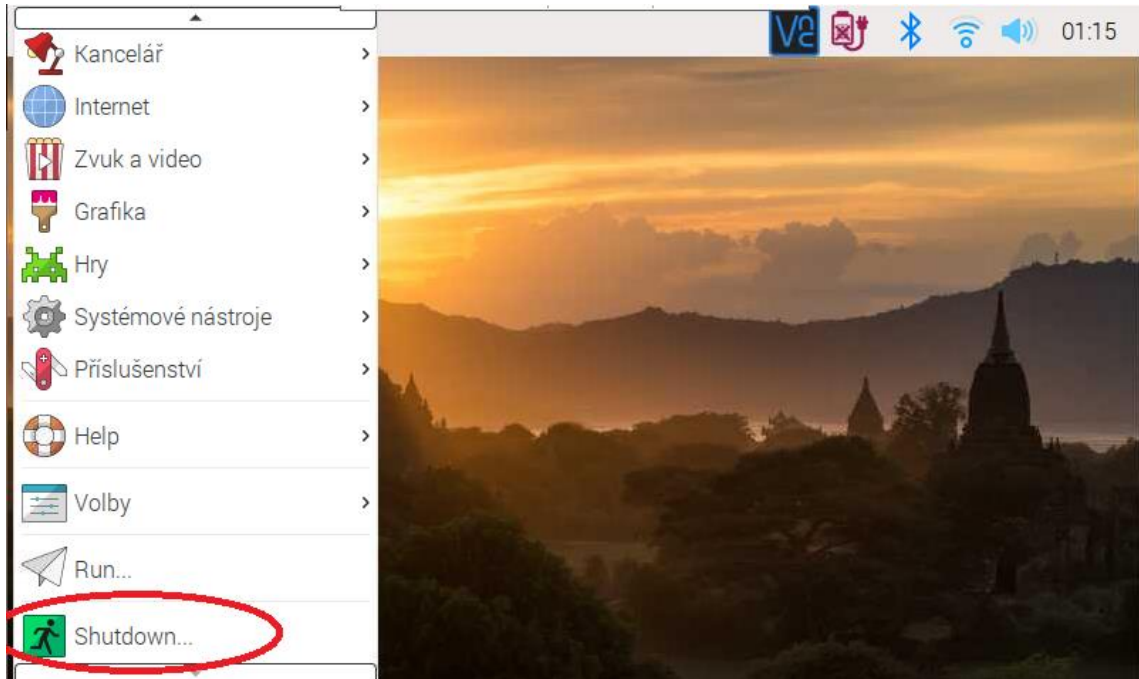
První sloupec souboru obsahuje časovou stopu, druhý sloupec obsahuje hodnoty naměřené prvním kanálem, třetí sloupec obsahuje hodnoty naměřené druhým kanálem, čtvrtý sloupec obsahuje hodnoty naměřené třetím kanálem a poslední sloupec obsahuje hodnoty naměřené čtvrtým kanálem. Příklad výstupu je na obrázku níže.

	A	B	C	D	
1	timestamp	channel_one	channel_two	channel_three	channel
2	2020-05-16 22:12:33.915722	-0.0009999999999998899	0.006	0.001	-0.0099
3	2020-05-16 22:12:33.937950	0.004	0.011	-0.0009999999999998899	-0.0160
4	2020-05-16 22:12:33.959387	0.007	0.011	-0.008999999999999897	-0.0019
5	2020-05-16 22:12:33.983319	0.022	0.0	-0.011000000000000012	0.001
6	2020-05-16 22:12:34.004040	0.021	0.0	-0.011000000000000012	0.003
7	2020-05-16 22:12:34.023932	0.021	0.0	-0.011000000000000012	0.003
8	2020-05-16 22:12:34.043712	0.022	0.001	-0.011000000000000012	0.002
9	2020-05-16 22:12:34.063352	0.022	0.001	-0.009999999999999787	0.0
10	2020-05-16 22:12:34.083001	0.021	0.001	-0.009999999999999787	0.001
11	2020-05-16 22:12:34.103630	0.022	0.001	-0.011000000000000012	0.001

Obrázek 19: Příklad naměřených hodnot v souboru vytvořeném mobilní jednotkou.

Ukončení provozu

Po skončení měření stačí zavřít okno prohlížeče a vypnout zařízení pomocí menu, kde se vybere položka „Shutdown“. Mobilní jednotka se vypne.



Obrázek 20: Zvolení příkazu Shutdown, který zajistí bezpečné vypnutí mobilní jednotky.

Příloha B: Realizace mobilní jednotky

Pro lepší představu vzhledu a rozměrů mobilní jednotky obrázky 1 – 3 znázorňují realizaci hardwaru.



Obrázek 1: Rozměry přední strany mobilní jednotky obsahující displej. Metr je přiložen pro reálnější představu rozměrů mobilní jednotky.



Obrázek 2: Rozměry šířky strany obsahující redukci pro koaxiální kabel. Metr je přiložen pro reálnější představu rozměrů mobilní jednotky.



Obrázek 3: Výška mobilní jednotky Metr je přiložen pro reálnější představu mobilní jednotky.



Obrázek 4: Vstupy pro externí úložné médium USB.

Příloha C: Obsah přiloženého CD

Název	Popis
Abstrakt_cz.pdf	Abstrakt v českém jazyce
Abstract_aj.pdf	Abstrakt v anglickém jazyce
Klicova_slova.pdf	Klíčová slova v českém jazyce
Key_words.pdf	Klíčová slova v anglickém jazyce
Zadani.pdf	Naskenované zadání
17PMBDP_456597_Daniela_Kolikova.pdf	Kompletní vypracování diplomové práce
Prilohy_DP	Složka obsahující přílohy
Zdrojove_kody	Složka obsahující software mobilní jednotky
Namerena_data	Složka obsahující soubory naměřené mobilní jednotkou