

**Technická univerzita v Košiciach  
Fakulta elektrotechniky a informatiky**

**Analýza klávesových úderov pomocou  
špeciálnych vstupných zariadení**

**Diplomová práca**

**Technická univerzita v Košiciach  
Fakulta elektrotechniky a informatiky**

**Analýza klávesových úderov pomocou  
špeciálnych vstupných zariadení**

**Diplomová práca**

Študijný program: Informatika  
Študijný odbor: 9.2.1. Informatika  
Školiace pracovisko: Katedra počítačov a informatiky (KPI)  
Školiteľ: Ing. Štefan Korečko, PhD.  
Konzultant: Ing. Matúš Pleva, PhD.

**Košice 2021**

**Bc. Matúš Haluška**



## **Abstrakt v SJ**

Autentifikácia používateľa na základe tenzie svalstva pri zadávaní znakov reprezentujúcich heslo je zaujímavá prídavná vrstva bezpečnosti. Reprezentuje ďalší spôsob overenia identity osoby napríklad pri vstupe do systému. Pre vedenie výskumu v tejto oblasti je potrebné vytvoriť systém, ktorý zaznamená a uloží dáta, nad ktorými bude možné vykonať analýzu pre overenie relevantnosti takého systému. Práca sa venuje navrhnutiu a implementovaniu systému, ktorý prostredníctvom experimentov zaznamená dáta vhodné pre ich ďalšiu analýzu. Ich spracovanie a prípravenie za týmto účelom zabezpečuje pomocou post-processingu. Taktiež vyhodnocuje optimálnosť týchto nástrojov.

## **Kľúčové slová v SJ**

Myo, EMG, zber dát, post-processing dát

## **Abstrakt v AJ**

User authentication based on muscle tension during entering characters representing a password is an interesting additional layer of security. It represents another way of verifying a person's identity, for example when entering the system. To conduct research in this area, it is necessary to create a system that records and stores data on which it will be possible to perform an analysis to verify the relevance of such a system. The work is devoted to the design and implementation of a system that will record data through experiments, that are suitable for their further analysis. Their processing and preparation for this purpose is ensured by post-processing. It also evaluates the optimality of these tools.

## **Kľúčové slová v AJ**

Myo, EMG, data collection, data post-processing

## **Bibliografická citácia**

HALUŠKA, Matúš. *Analýza klávesových úderov pomocou špeciálnych vstupných zariadení*. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2021. 62s. Vedúci práce: Ing. Štefan Korečko, PhD.

**TECHNICKÁ UNIVERZITA V KOŠICIACH**  
**FAKULTA ELEKTROTECHNIKY A INFORMATIKY**  
Katedra počítačov a informatiky

**ZADANIE**  
**DIPLOMOVEJ PRÁCE**

Študijný odbor: **Informatika**

Študijný program: **Informatika**

Názov práce:

**Analýza klávesových úderov pomocou špeciálnych vstupných zariadení**

**Keystroke Analysis by Means of Special Input Devices**

Študent: **Bc. Matúš Haluška**

Školiteľ: **Ing. Štefan Korečko, PhD.**

Školiace pracovisko: **Katedra počítačov a informatiky**

Konzultant práce: **Ing. Matúš Pleva, PhD.**

Pracovisko konzultanta: **Katedra elektroniky a multimediálnych telekomunikácií**

Pokyny na vypracovanie diplomovej práce:

1. Analyzovať dostupné prostriedky a prístupy k získavaniu údajov zo senzorových náramkov MYO pre vedecké experimenty.
2. Na základe analýzy a požiadaviek so strany vedúceho a konzultanta práce navrhnúť a implementovať softvérové vybavenie pre zber a spracovanie údajov z náramkov MYO.
3. Softvérové vybavenie vyvinúť tak, aby v čo najväčšej miere zjednodušovalo realizáciu experimentov, kde sa pomocou MYO sleduje aktivita pri písaní na klávesnici.
4. Okrem údajov z MYO nech softvérové vybavenie ukladá aj údaje o stláčaní kláves a videozáznam.
5. Realizovať konkrétne experimenty s vyvinutým softvérovým vybavením.
6. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 23.04.2021

Dátum zadania diplomovej práce: 30.10.2020



prof. Ing. Liberios Vokorokos, PhD.  
dekan fakulty

## **Čestné vyhlásenie**

Vyhlasujem, že som záverečnú prácu vypracoval(a) samostatne s použitím uvedenej odbornej literatúry.

Košice, 23.4.2021

.....

*Vlastnoručný podpis*

## **Podakovanie**

Na tomto mieste chcem vyjadriť vďaku vedúcemu tejto diplomovej práce, *Ing. Štefan Korečko, PhD.*, za jeho čas, odborné vedenie a prínosné vecné pripomienky a návrhy, ktoré viedli ku úspešnému vypracovaniu tejto diplomovej práce.

Taktiež by som chcel poďakovať konzultantovi *Ing. Matúšovi Plevovi, PhD.*, ktorý taktiež prispel svojimi akademickými skúsenosťami.

Slová vďaky smerujem aj mojej rodine, priateľom a blízkym, ktorí ma podporovali počas celého môjho štúdia.

# Obsah

---

Úvod	1
<b>1 Analytická časť</b>	<b>3</b>
1.1 Autentifikačné technológie . . . . .	3
1.1.1 Autentifikácia založená na hesle . . . . .	3
1.1.2 Viacfaktorová autentifikácia . . . . .	4
1.1.3 Autentifikácia na základe certifikátu . . . . .	4
1.1.4 Autentifikácia pomocou tokenov . . . . .	4
1.1.5 Biometrická autentifikácia . . . . .	5
1.2 Myo náramok . . . . .	6
1.2.1 Hardvér a dizajn Myo náramku . . . . .	7
1.2.2 Základné typy dát poskytovaných Myo náramkom . . . . .	8
1.2.3 Príklady využitia Myo náramku . . . . .	8
1.3 Alternatívne EMG zariadenia . . . . .	12
1.3.1 Kai Gesture Controller . . . . .	12
1.3.2 Shimmer3 EMG . . . . .	12
1.4 Pôvodná aplikácia . . . . .	15
1.5 Softvérové technológie a programové vybavenie . . . . .	15
1.5.1 Vývojové prostredie Visual Studio 2019 . . . . .	15
1.5.2 Jazyk C# . . . . .	15
1.5.3 Myo SDK . . . . .	16
1.5.4 WindowsForms . . . . .	16
1.5.5 MyoSharp . . . . .	16
1.5.6 PyCharm . . . . .	17
1.5.7 Python . . . . .	17
1.5.8 Myo-Python . . . . .	17
1.5.9 Spracovanie videa . . . . .	17
1.6 Multi-threading a práca s dátami . . . . .	19
1.6.1 Vlákno . . . . .	19

---

1.6.2	ThreadPool	20
1.6.3	BackgroundWorker	21
1.6.4	Task Parallel Library (TPL)	21
1.6.5	Linq	21
1.7	Vyhodnotenie analýzy	22
<b>2</b>	<b>Experimentálne overenie knižníc pre Myo</b>	<b>24</b>
2.1	Test myosharp aplikácie	24
2.1.1	Výsledok testu	25
2.2	Test myo-python aplikácie	26
2.2.1	Výsledok testu	26
2.3	Vyhodnotenie	26
<b>3</b>	<b>Návrh systému pre realizovanie experimentov</b>	<b>28</b>
3.1	Zdroje dát	31
3.2	Návrh aplikácie	31
3.2.1	Ukladanie dát pomocou BackgroundWorkerov	31
3.2.2	Spracovanie dát	31
3.3	Návrh používateľského rozhrania	34
3.3.1	Hierarchická analýza úloh	34
3.3.2	Konceptuálny model	35
3.3.3	Model sekvencie obrazoviek	37
3.3.4	Prototyp používateľského rozhrania	38
<b>4</b>	<b>Implementácia systému pre realizovanie experimentov</b>	<b>42</b>
4.1	Inicializácia aplikácie	42
4.1.1	Inicializácia Myo Hub-u a vytvorenie komunikačných kanálov	42
4.1.2	Naplnenie zoznamu existujúcich subjektov	43
4.2	Pripojenie Myo náramkov	43
4.3	Vykonávanie kódu Myo BackgroundWorker-a	44
4.4	BackgroundWorker kamery a klávesnice	44
4.5	Zaznamenanie IMU dát	44
4.6	Zaznamenanie EMG dát	45
4.7	Uloženie dát na disk	45
4.8	Stojan kamery	45
<b>5</b>	<b>Návrh aplikácie pre dodatočné spracovanie dát</b>	<b>48</b>
5.1	Vystrihnutie dát	48

---

<b>6 Implementácia aplikácie pre dodatočné spracovanie dát</b>	<b>50</b>
6.1 Dodatočné spracovanie dát klávesnice . . . . .	51
6.2 Dodatočné spracovanie EMG a IMU dát . . . . .	52
<b>7 Vyhodnotenie</b>	<b>53</b>
7.1 Priebeh experimentov . . . . .	53
7.2 Výsledky experimentov . . . . .	54
<b>8 Záver</b>	<b>57</b>
<b>Literatúra</b>	<b>59</b>
<b>Zoznam príloh</b>	<b>63</b>
<b>A Používateľská príručka</b>	<b>64</b>
A.1 Realizácia experimentov . . . . .	64
A.1.1 Inštalácia Myo Connect . . . . .	64
A.2 Predpríprava experimentu . . . . .	65
A.2.1 Inicializácia experimentu . . . . .	65
A.2.2 Riadenie experimentu . . . . .	66
A.3 Dodatočné spracovanie dát . . . . .	67
<b>B Systémová príručka</b>	<b>68</b>
B.1 Aplikácia pre realizovanie experimentov . . . . .	68
B.1.1 Triedy . . . . .	68
B.1.2 Metódy . . . . .	69
B.2 Aplikácia pre dodatočné spracovanie dát . . . . .	71
B.2.1 Triedy . . . . .	71
B.2.2 Metódy . . . . .	71

# Zoznam obrázkov

---

1.1	Myo náramok. (Zdroj: [12]) . . . . .	7
1.2	Základná sada gest Myo náramku. (Zdroj: [16]) . . . . .	9
1.3	Návrh protézy spoločnosti BionIT Labs Company a Oddelenia inovatívneho inžinierstva University of Salento. (Zdroj: [10]) . . . . .	9
1.4	Napodobňovanie cvičení prezentovaných rozhraním na zber údajov pri správnom nasadení Myo náramku. (Zdroj: [17]) . . . . .	10
1.5	Kreslenie v aplikácii pomocou Myo náramku. (Zdroj: [10]) . . . . .	10
1.6	Používateľský test MYO-music prototypu. (Zdroj: [11]) . . . . .	11
1.7	Kai náramok nasadený na ruke. (Zdroj: [20]) . . . . .	13
1.8	Shimmer3 EMG nasadený na ruke. (Zdroj: [22]) . . . . .	14
1.9	Vykonanie operácie select nad kolekciami objektov rôznej veľkosti metódami for, foreach a LINQ. (Zdroj: [38]) . . . . .	22
2.1	Diagram zobrazujúci spracovanie dát aplikáciou. . . . .	25
3.1	Návrh systému. . . . .	30
3.2	Správne nasadenie Myo náramku. . . . .	30
3.3	Diagram hierarchickej analýzy úloh . . . . .	35
3.4	Konceptuálny model . . . . .	36
3.5	Sekvenčný diagram obrazoviek . . . . .	38
3.6	Prototyp hlavnej obrazovky . . . . .	40
3.7	Prototyp obrazovky výberu adresára . . . . .	40
3.8	Prototyp obrazovky výberu kamery . . . . .	41
4.1	Diagram tried implementovanej aplikácie. . . . .	43
4.2	Stojan pre kameru. Pohľad spredu. . . . .	46
4.3	Stojan pre kameru. Pohľad z perspektívy. . . . .	47
6.1	Procedúry vykonávané po spustení aplikácie. . . . .	51
7.1	Subjekt usadený v správnej polohe. . . . .	54



7.2	Vytvorené používateľské rozhranie. . . . .	56
A.1	Hlavná obrazovka. . . . .	66
A.2	Obrazovka výberu adresára. . . . .	66
A.3	Obrazovka výberu kamery. . . . .	67

# Zoznam tabuliek

---

1.1	Špecifikácia hardvéru Myo náramka. (Zdroj: [15]) . . . . .	8
1.2	Vzťah gest a akcií v projekte virtuálnej klávesnice. (Zdroj: [18]) . .	11
1.3	Technická špecifikácia Shimmer3 EMG Unit. (Zdroj: [21]) . . . . .	14

# Úvod

---

Autentifikačné technológie nás každodenne sprevádzajú pri vstupoch do rôznych systémov. Potreba overovania identity a následného umožnenia alebo zabránenia prístupu entity na základe rôznych politík, ku osobným dátam ale aj rôznym kardinálnym systémom ako riadenie letovej prevádzky alebo atómovej elektrárne rastie každým dňom. Neoprávnený prístup do systémov za účelom úmyselného poškodenia alebo aj neodbornej manipulácie s ním môže mať za následok stratu osobných dát alebo v horšom prípade stratu kontroly nad dôležitým kritickým systémom a spôsobenie katastrofy. Za účelom zabránenia vzniku podobných udalostí sú autentifikačné technológie vyvíjané a inovované a nasadzované všade tam, kde treba overiť oprávnenie vstupu externej entity do systému.

Zaujímavou metódou autentifikácie je použitie EMG dát pri procese identifikácie osoby, ktorá chce vstúpiť do systému. Ak by bolo možné pridať túto extra úroveň overenia ku klasickej metóde autentifikácie ako je použitie hesla, zaručilo by to, že osoba ktorá heslo zadala je skutočne osobou za ktorú sa vydáva. Tenzia svalstva ruky pri zadávaní hesla by mohla byť natoľko špecifická pre danú osobu, že by umožnila jej identifikovanie. Pre vedenie výskumu v tejto oblasti je ale potrebné vytvoriť nástroj, ktorý by umožnil zaznamenanie dostatočného množstva relevantných dát spojených s touto problematikou.

Myo náramok sa ukazuje ako zariadenie so širokým využitím v rôznych druhoch aplikácií a štúdií. Poskytuje EMG dáta z predlaktia ruky, čo sa javí ako vhodné pre použitie pri spomínanom spôsobe autentifikácie. Jeho jednoduché nasadenie z neho robí vhodného kandidáta na každodenné používanie.

## Formulácia úlohy

Táto diplomová práca má za úlohu analyzovať dostupné prostriedky a prístupy k získavaniu údajov z Myo náramkov, čomu sa venuje v analytickej časti. Použitím výsledkov analýzy je potrebné vytvoriť funkčnú aplikáciu, ktorá bude slúžiť na zaznamenávanie experimentov. Návrh tejto aplikácie je zahrnutý v časti návrh

systemy pre realizovanie experimentov. Počas vykonávania experimentov musí aplikácia zaznamenať a uložiť dostupné dáta z náramkov, klávesnice ako aj vytvoriť video záznam experimentu. Softvérové vybavenie má byť navrhnuté a vyvinuté tak, aby uľahčovalo realizáciu experimentov. Jeho vyhotovenie je popísané v časti implementácia systému pre vyhotovenie experimentov. V rámci práce je potrebné realizovať samotné experimenty. V kapitole vyhodnotenie sa nachádza výsledok realizovaných experimentov.

# 1 Analytická časť

---

Táto časť práce sa v úvode venuje popisu existujúcich spôsobov autentifikácie používateľa. Analyzované spôsoby autentifikácie sa nachádzajú v časti autentifikačné technológie. Ďalej sa venuje analýze Myo náramku. Uvádza jeho základné charakteristiky a parametre, ktoré je pre prácu s ním vhodné poznať. Tieto údaje sa nachádzajú v sekcii Myi náramok. Analýza sa pozerá aj na alternatívne technológie, ktoré by mohli Myo náramok v tejto práci nahradiť, čo opisuje v sekcii alternatívne EMG zariadenia. V časti pôvodná aplikácia sa analýza venuje popisu aplikácie, ktorá slúžila ako vhodný základ tejto práce. Dôležitou časťou analýzy je aj popis softvérových technológií, ktoré sú vhodné pre použitie v tejto práci. Vhodné použitie správnych nástrojov a programovacích techník zabezpečuje kvalitný a spoľahlivý finálny produkt. Uvádza a rozoberá sa tu aj popis pôvodnej aplikácie, ktorá slúžila ako základ pre aplikáciu navrhnutú a implementovanú v tejto práci. Experimentálna analýza týchto technológií to pomôže zabezpečiť. V svojom závere sa táto kapitola spätne pozerá na dostupné technológie a programovacie metódy a prístupy. Objasňuje, ktoré technológie a postupy sú zvolené a zároveň ich výber odôvodňuje.

## 1.1 Autentifikačné technológie

Dôležitou súčasťou systémov sú technológie, ktoré spravujú prístup k nim. Pri vstupe do systémov, ktoré majú dosah na kritickú infraštruktúru, ale aj osobné dáta, je na mieste overiť identitu používateľa, ktorý chce vstúpiť do systému. Za účelom kontrolovania prístupu a overovanie identity používateľa, boli vyvinuté rôzne technológie. Tieto často krát predstavujú vstupnú bránu do systémov.

### 1.1.1 Autentifikácia založená na hesle

Heslá sú najbežnejším spôsobom autentifikácie. Heslá môžu byť vo forme reťazca písmen, čísel alebo špeciálnych znakov. Pre potrebnú ochranu je nutné vytvorenie bezpečného hesla, ktoré obsahuje kombináciu rôznych znakov a je dostatočne

dlhé. Heslá sú však náchylné na phishingové útoky. Priemerný človek má asi 25 rôznych online účtov, ale iba 54% používateľov používa rôzne heslá vo svojich účtoch [1]. Skutočnosťou je, že veľké množstvo hesiel je ťažko zapamätateľné, čo je dôvod prečo ľudia opakovane používajú rovnaké heslá a často krát sú to jednoduché heslá. Heslá teda majú mnoho slabých stránok a nie sú dostatočné na ochranu. Jednoduché heslá sú náchylné na útoky hackerov, kedy hacker opakovane skúša rôzne kombinácie znakov, až kým neuhádne heslo.

### **1.1.2 Viacfaktorová autentifikácia**

Ďalšia bežná metóda autentifikácie je viacfaktorová autentifikácia, ktorá vyžaduje dva alebo viac nezávislých spôsobov identifikácie používateľa. Medzi príklady patria kódy vygenerované zo smartfónu používateľa, testy Captcha, odtlačky prstov alebo rozpoznávanie tváre.

Metódy a technológie viacfaktorovej autentifikácie zvyšujú dôveru používateľov pridaním viacerých vrstiev zabezpečenia [2]. Viacfaktorová autentifikácia môže byť dobrou obranou proti väčšine útokov na účet, ale má svoje vlastné úskalia. Ľudia môžu stratiť svoje telefóny alebo SIM karty a nebudú môcť vygenerovať autentifikačný kód.

### **1.1.3 Autentifikácia na základe certifikátu**

Technológie autentifikácie na základe certifikátu identifikujú používateľov, stroje alebo zariadenia pomocou digitálnych certifikátov. Digitálny certifikát je elektronický dokument založený na myšlienke vodičského preukazu alebo pasu.

Certifikát obsahuje digitálnu identitu používateľa vrátane verejného kľúča a digitálny podpis certifikačnej autority [3]. Digitálne certifikáty preukazujú vlastníctvo verejného kľúča a vydáva ich iba certifikačná autorita.

Používatelia poskytujú svoje digitálne certifikáty pri prihlasovaní na server. Server overuje dôveryhodnosť digitálneho podpisu a certifikačnej autority. Server potom pomocou kryptografie potvrdí, že používateľ má k certifikátu priradený správny súkromný kľúč.

### **1.1.4 Autentifikácia pomocou tokenov**

Technológie autentifikácie založené na tokenoch umožňujú používateľom raz zadať svoje poverenia za čo dostanú jedinečný šifrovaný reťazec náhodných znakov [4]. Token sa následne používa na prístup do chránených systémov namiesto opakovaného zadávania prihlasovacích údajov. Digitálny token dokazuje, že už máte

povolenie na prístup. Medzi prípady použitia autentifikácie na základe tokenov patria rozhrania RESTful API, ktoré používajú viaceré rámce a klienti.

### 1.1.5 Biometrická autentifikácia

Biometrická autentifikácia je bezpečnostný proces, ktorý sa spolieha na jedinečné biologické vlastnosti jednotlivca [5]. Kľúčové výhody použitia biometrických autentifikačných technológií sú nasledovné [6]:

- Biologické vlastnosti sa dajú ľahko porovnať s autorizovanými vlastnosťami uloženými v databáze.
- Biometrické overovanie môže riadiť fyzický prístup, ak je nainštalovaný na bránach a dverách.
- Do procesu viacfaktorovej autentifikácie je možné pridať biometriu.

Biometrické autentifikačné technológie používajú spotrebitelia, vlády a súkromné spoločnosti vrátane letísk, vojenských základní a štátnych hraníc. Medzi bežné metódy biometrickej autentifikácie patria:

#### Rozpoznávanie tváre

Táto metóda používa rôzne charakteristiky tváre jednotlivca, ktorý sa snaží získať prístup k schválenej tvári uloženej v databáze. Rozpoznávanie tváre môže byť nekonzistentné pri porovnávaní tvárí pod rôznymi uhlami alebo pri porovnávaní ľudí, ktorí vyzerajú podobne, napríklad blízki príbuzní.

#### Snímače odtlačkov prstov

Táto metóda používa jedinečný vzor na odtlačkoch prstov jednotlivca. Niektoré nové verzie skenerov odtlačkov prstov môžu dokonca posúdiť vaskulárne vzorce v prstoch ľudí [7]. Skenery odtlačkov prstov sú v súčasnosti najpopulárnejšou biometrickou technológiou pre každodenných spotrebiteľov aj napriek častým nepresnostiam.

#### Hlasová identifikácia

Táto metóda skúma rečové vzorce rečníka na formovanie špecifických tvarov a zvukových kvalít. Zariadenie chránené hlasom sa pri identifikácii používateľov zvyčajne spolieha na štandardizované slová, rovnako ako heslo.

## Očné skenery

Táto metóda zahŕňa technológie ako rozpoznávanie dúhovky a skenery sietnice. Skenery dúhovky premietajú jasné svetlo smerom k oku a vo farebnom prstenci okolo zrenice oka hľadajú jedinečné vzory. Vzory sa potom porovnávajú so schválenými informáciami uloženými v databáze. Ak osoba nosí okuliare alebo kontaktné šošovky, autentifikácia založená na očiach môže spôsobiť nepresnosti.

## EMG biometria

Medzi netradičné metódy biometrickej autentifikácie patrí rozpoznávanie a autentifikácia pomocou tenzie svalstva. Jej prevedenie spočíva v nasadení EMG senzorov na predlaktie ruky a zozbieraní EMG dát pri určitom geste. Následne je možné klasifikovať EMG dáta pri pokuse o autentifikáciu pomocou dát z kalibračných pokusov a identifikovať používateľa. Ako sa uvádza vo výsledku štúdie [8], realizovanej na University of Waterloo, táto metóda vykazuje vysokú presnosť autentifikácie. Bez postprocesingu dát sa presnosť identifikácie pohybovala v rozmedzí 83.8% až 92.2%. S použitím 13 vzoriek v postprocesingu došlo ku zvýšeniu tejto hodnoty takmer na presnosť 100%. Výnimkou boli len 2 gestá a síce zovretie päste, pri ktorom to bolo 93.4% a napnutie ukazováka kde sa pozorovala výsledná hodnota 96.2%.

## 1.2 Myo náramok

Toto zariadenie navrhnuté ako náramok na predlaktie používateľa Obr. 1.1 patrí medzi hlavné technológie použité v tejto práci. Je zdrojom EMG a IMU dát, ktoré sú bližšie popísané v sekcii základné typy dát poskytovaných Myo náramkom. Spoločnosť Thalmic Labs toto zariadenie vyvinula v júni 2013. Avšak predaj tohto zariadenia už skončil. Od 12.10.2018 je toto zariadenie dostupné už len na kúpu od tretej strany [9]. Podpora tohto zariadenia skončila s jeho predajom. Hlavné podporné webové stránky sú už nedostupné. Potrebné informácie sú ale stále jednoducho dopátrateľné hlavne vďaka množstvu nadšencov, ktorý toto zariadenie vlastní a zverejnili svoje projekty. Taktiež v rámci vedeckých článkov a štúdií je možné dopátrať sa potrebných informácií. Myo bol teda navrhnutý na rozpoznávanie gest ruky. To je uskutočnené pomocou ôsmich segmentov obsahujúcich EMG senzory, ktorými tento náramok obopína ruku v oblasti predlaktia [10]. Prenos EMG dát má frekvenciu 200 Hz [11]. Pri jeho nosení je dôležité nosiť ho na najširšej časti ruky, alebo použiť sťahovacie svorky dodávané s týmto náramkom.



Zvýši sa tak dotyková plocha senzorov s povrchom ruky. Predlaktím totiž vedú svaly ktorých tenzia spôsobuje ovládanie a pohyb ruky a jej prstov. Pri tenzii svalov na predlaktí náramok zaznamená zmenu povrchového napätia a podľa jeho intenzity klasifikuje gesto vykonané rukou [10]. Náramok taktiež sníma pohyb ruky pomocou trojosového akcelerometra, magnetometra a gyroskopu. Prenosová frekvencia orientačných dát alebo skrátene IMU dát je 50 Hz [11]. Spôsob vyhotovenia dizajnu tohto zariadenia umožňuje nosiť tento náramok pravidelne bez nutnosti znalosti odbornej manipulácie.



Obr. 1.1: Myo náramok. (Zdroj: [12])

### 1.2.1 Hardvér a dizajn Myo náramku

Ako bolo v predchádzajúcej kapitole spomenuté, Myo náramok obsahuje sadu ôsmich EMG senzorov vyrobených z ušľachtilej ocele. Tieto senzory spĺňajú medicínske štandardy [13]. Sú orientované smerom do vnútra tak, aby sa ich meracie plochy dotýkali ruky používateľa. Jednotlivé senzory sú umiestnené v obdĺžnikových dlaždiciach, ktoré sú navzájom spojené gumeným pásom. V tomto páse sú zároveň umiestnené dátové a napájacie káble pre jednotlivé senzory. Zariade-

nie disponuje trojosím akcelerometrom, magnetogramom a gyroskopom vďaka ktorým je možné určiť náklon alebo zmenu polohy zariadenia. Vďaka dizajnu zariadenia je možné prispôbiť veľkosť jeho rozťahnutia a teda padne na rôzne veľkosti ruky. Túto veľkosť je možné upraviť v rozmedzí 19-34 cm [14]. Toto je dôležité pre už spomínanú nutnosť zabezpečenia čo najväčšej dotykovej plochy s rukou. Detailne hardvérové parametre Myo náramku sú popísané v nasledujúcej tabuľke Tab. 1.1.

Tabuľka 1.1: Špecifikácia hardvéru Myo náramka. (Zdroj: [15])

Parameter	Hodnota
Váha	93 g
Hrúbka	1,14 cm
Rozťahnutelnosť	19 - 34 cm
Batéria: Typ	Li-Ion
Batéria: Výdrž	2x 260 mAh
Procesor	ARM Cortex M4
Konektivita	BLE
Spätná väzba	Vibrácie, LED diódy

## 1.2.2 Základné typy dát poskytovaných Myo náramkom

Myo náramok podporuje a poskytuje na spracovanie aplikácii dve základné typy dát, ktorými sú **EMG** a **IMU** dáta. EMG dáta predstavujú dáta z merania aktuálnej tenzie svalstva z každého senzora jednotlivo. Teda obsahujú set ôsmich hodnôt z meraní sensorov. IMU dáta poskytujú dáta z meraní akcelerometra, magnetogramu a gyroskopu. Na základe týchto dát je možné vypočítať rotáciu náramku a smer akým sa pohybuje. Oba sú odosielané do aplikácie prostredníctvom BLE (Bluetooth Low Energy). Ďalej je možné ich ľubovoľne spracovať. Myo podporuje rozpoznanie piatich základných gest [16]. Tieto gestá sú znázornené na obrázku Obr. 1.2.

## 1.2.3 Príklady využitia Myo náramku

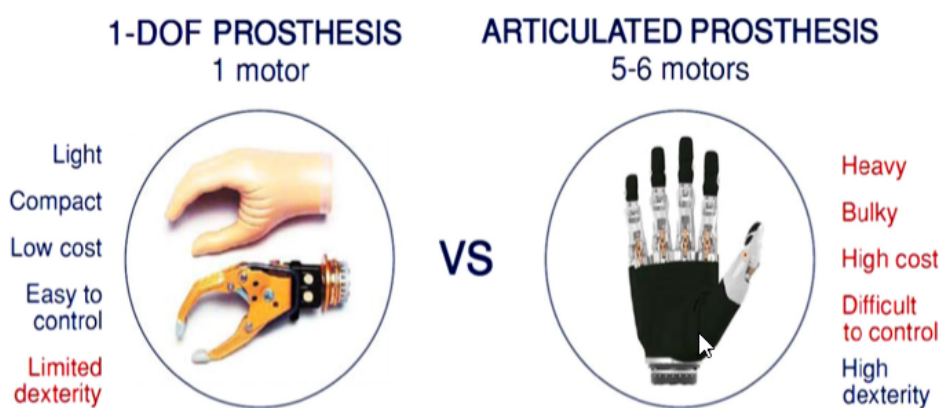
Ako bolo spomínané v predchádzajúcej kapitole, Myo náramok má široké uplatnenie v rôznych experimentálnych alebo praktických aplikáciách. Aj preto si množstvo vývojárov vybralo tento produkt ako zdroj EMG alebo IMU dát. Reálne príklady použitia tohto náramku sú po sumarizácii nasledovné.



Obr. 1.2: Základná sada gest Myo náramku. (Zdroj: [16])

### Využitie v biomedicíne

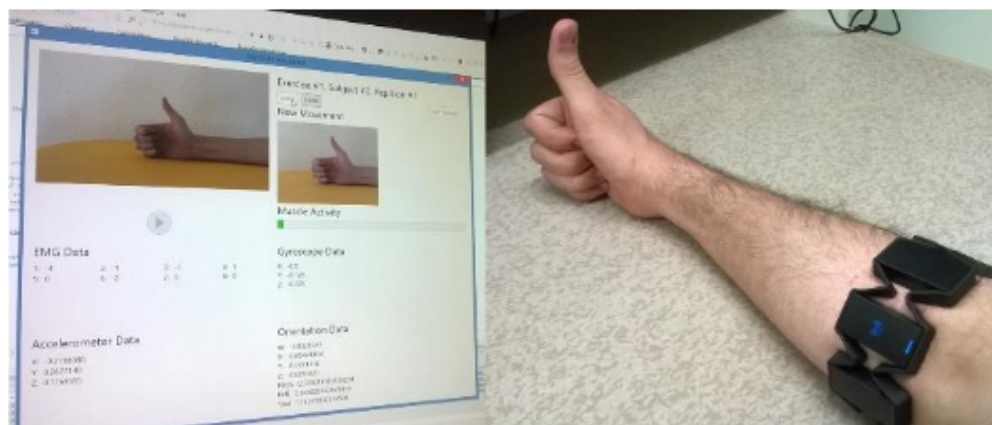
Veľmi zaujímavou aplikáciou tohto náramku je na poly biomedicíny. Mnohé práce sa venujú výskumu rehabilitácií pacientov po čiastočnej amputácii ruky. Podľa publikácie [10] oddelenia inovatívneho inžinierstva University of Salento v spolupráci s BionIT Labs Company, v ktorej zhrnuli existujúce publikácie týkajúce sa tejto témy je tu nádej pre mnohých pacientov s amputovanou spodnou časťou ruky. V práci predstavili aj vlastné riešenie, ktoré vyplynulo z tejto spolupráce. Je to protéza ruky s názvom Adam's Hand Obr. 1.3. EMG a IMU dáta z Myo náramku slúžia ako riadiace dáta pre DC servo motory. Tie hýbu z jednotlivými kĺbmi robotickej ruky a pacient tak môže nadobudnúť náhradu straty. Štúdia [17] z University of Canterbury upozorňuje, že pre dosiahnutie optimálnych výsledkov je potrebné správne umiestnenie EMG senzorov Obr. 1.4.



Obr. 1.3: Návrh protézy spoločnosti BionIT Labs Company a Oddelenia inovatívneho inžinierstva University of Salento. (Zdroj: [10])

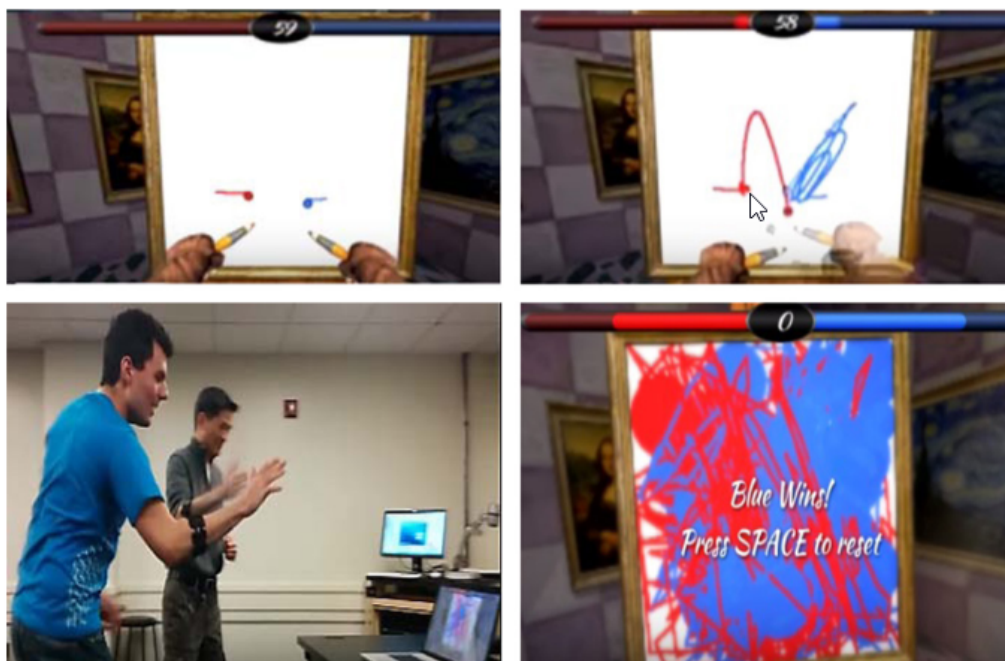
### Využitie pre zábavu

Okrem uplatnenia v biomedicínskej sa dá tento náramok použiť na ovládanie rôznych aplikácií slúžiacich na oddych a zábavu. To hlavne vďaka tomu, že je ovlá-



Obr. 1.4: Napodobňovanie cvičení prezentovaných rozhraním na zber údajov pri správnom nasadení Myo náramku. (Zdroj: [17])

danie pohybom ruky v mnohých prípadoch intuitívnejšie a prirodzenejšie ako ovládanie myši a klávesnice. Ako je popísané v článku [10], Myo náramok sa dá použiť na sekacie ovocia v hre Fruit Ninja alebo pri kreslení v aplikáciách na to určených ako je možné vidieť na obrázku Obr. 1.5. Článok [11] popisuje využitie Myo náramku ako rozhrania pre prácu s hudbou Obr. 1.6. Po experimente síce vyhodnocuje, že ich konkrétna implementácia má nedostatky ale zároveň predpokladá rozmach používania tohto náramku aj v tejto sfére.



Obr. 1.5: Kreslenie v aplikácii pomocou Myo náramku. (Zdroj: [10])



Obr. 1.6: Používateľský test MYO-music prototypu. (Zdroj: [11])

### Myo ako náhrada klasického používateľského rozhrania

Veľmi zaujímavým, aj z pohľadu tejto diplomovej práce, je využitie Myo náramku ako náhrady klasického používateľského rozhrania. Konkrétne klávesnice. O to sa postarala dvojica Md Abdur Rahim a Jungpil Shin zo školy počítačovej vedy a inžinierstva University of Aizu. V práci [18] sa im podarilo vytvoriť funkčnú virtuálnu klávesnicu. Klasifikovaním EMG a IMU dát dokázali predikovať rôzne gestá, ktoré boli priradené rôznym akciám klávesnice. Tento vzťah je popísaný v tabuľke Tab. 1.2. Pri priebehu experimentu mali zariadené experimentálne pracovisko pre zaistenie laboratórnych podmienok.

Tabuľka 1.2: Vzťah gest a akcií v projekte virtuálnej klávesnice. (Zdroj: [18])

Gesto	Popis	Funkcia
Dvojité klepnutie	Dvojité klepnutie ukazovák a palcom	Zadanie znaku
Držanie päste	Zovretie päste	Zmena znaku
Mávnutie vľavo	Mávnutie vľavo	Vymazanie znaku
Mávnutie vpravo	Mávnutie vpravo	Ukončenie riadka
Roztiahnutie prstov	Pohyb všetkých prstov ruky od seba	Znak medzery

## 1.3 Alternatívne EMG zariadenia

Okrem Myo náramku existujú ďalšie zariadenia, ktoré by bolo možné v tejto práci využiť. Od Myo náramku sa odlišujú hlavne spôsobom nasadenia a nosenia na požadovanom mieste.

### 1.3.1 Kai Gesture Controller

Spoločnosť Araciv Technologies Pvt. Ltd., ktorá vystupuje pod menom Vicara zaoberajúca sa nositeľnými zariadeniami vyvinula zariadenie s názvom Kai [19]. Toto zariadenie je navrhnuté pre umiestnenie na dlani Obr. 1.7. Primárne slúži na rozpoznávanie gest, ktoré používateľ vykonáva pomocou ruky a prstov. Vďaka jeho umiestneniu je klasifikácia gest vykonaných pohybom prstov presnejšia, ako klasifikácia pomocou Myo náramku. Vlastnosti Kai náramku:

- **Vysoká presnosť:** Kai je dostatočne citlivý na to, aby zachytil zmeny o rozsahu 1 mm. To umožňuje vykonávať mimoriadne presné akcie v rámci aplikácie.
- **Nízka latencia:** Tento náramok pracuje s latenciou nižšou ako 10ms.
- **Výdrž batérie:** Náramok sa dá používať po dobu 8 hodín bez dobíjania.
- **Haptická spätná väzba:** Spätná väzba vo forme vibrácii je programovateľná.
- **Bluetooth Low Energy:** Toto zariadenie je kompatibilné so štandardom Bluetooth 5.0
- **KaiSDK :** Poskytuje nástroje na vývoj pomocou KaiSDK a knižníc kai-dotnet a kai-python

Pre použitie v tejto práci by bolo potrebné zakúpenie Kai verzie pre developerov. Cena tejto verzie je 179\$. Nevýhodou tohto zariadenia je, že dokáže monitorovať len pohyb 4 prstov. Jeho umiestnenie, ktoré mu dáva výhodu v presnosti sledovania pohybu prstov mu zároveň znemožňuje sledovanie pohybu palca. Pri použití tohto zariadenia by priebeh experimentu musel obsahovať podmienku o nepoužívaní palca pri písaní.

### 1.3.2 Shimmer3 EMG

Za vývojom zariadenia Shimmer3 EMG je spoločnosť Shimmer [21]. Predstavila ho v roku 2013. Táto spoločnosť drží patent na platformu, ktorú používajú jej





Obr. 1.7: Kai náramok nasadený na ruke. (Zdroj: [20])

zariadenia. V spolupráci s mnohými inštitúciami sa osvedčila ako jedna z najinovatívnejších spoločností Írska v roku 2014. Jej platforma sa v súčasnosti používa aj pri výskume ochorenia COVID-19. Meracie senzory tohto zariadenia sa pripevnia na ľubovoľnú časť tela Obr. 1.8. Ich umiestnenie je teda variabilné podľa potreby experimentu. Technické vlastnosti Shimmer3 EMG sú popísané v tabuľke Tab. 1.3. Hlavné vlastnosti tohto zariadenia sú:

- Bezdrôtové a robustné snímače ktoré je možné pripevniť na telo.
- Surový a klinicky relevantný výstup inerciálnych a biofyzikálnych údajov súčasne.
- Presná synchronizácia medzi snímačmi.
- Synchronizácia s doplnkovými systémami prostredníctvom konfigurácie hodín v reálnom čase.
- Rýchle zacvakávacie pásky pre rýchle nastavenie.
- Navrhnuté pre ľahké použitie vo výskumných projektoch, opakovateľné klinické skúšky, systémovú integráciu a vývoj produktov.

Toto zariadenie je po technickej stránke ideálne na meranie EMG aktivity. Pre použitie na každodennej báze by ale nebolo príliš praktické. Výhoda tohto zaria-



Obr. 1.8: Shimmer3 EMG nasadený na ruke. (Zdroj: [22])

denia, spočívajúca v ľubovoľnom umiestnení senzorov, je zároveň aj nevýhoda. Nutnosť umiestňovania senzorov jednotlivo na telo je proces, ktorý sa pri Myo náramku nevyskytuje. Cena jedného Shimmer3 EMG je 554€.

Tabuľka 1.3: Technická špecifikácia Shimmer3 EMG Unit. (Zdroj: [21])

Špecifikácia zariadenia	
Spracovanie dát:	MSP 430 microcontroller(24MHz, 16Bit)
Komunikácia:	Bluetooth - RN42
Úložisko:	Integrovaná 8GB microSD karta Podpora až do 32GB
Batéria:	450mAh nabíjateľná Li-ion
Rozmery:	65mm x 32mm x 12mm. Váha: 31g
RAM:	16KB
Flash:	256KB
Frekvencia:	24MHZ
Počet kanálov	2-kanálové riešenie EMG
Vstupný rozdiel	Približne 800 mV (pre zisk = 6)
Šírka pásma	8.4 kHz
Ochrana vstupu	ESD a RF/EMI filtrovanie, obmedzenie prúdu,
Odporúčaný adaptér Bluetooth	Kensington K33956AM USB 2.0 Bluetooth 4.0
Pripojenie	Dotykové konektory IEC/EN 60601-1 DIN42-802, 1 mm



## 1.4 Pôvodná aplikácia

V rámci tejto práce je potrebné analyzovať aplikáciu, ktorá bola vytvorená Ing. Mariánom Hudákom. Je vhodným základom implementácie tejto diplomovej práce. Je to WindowsForms aplikácia postavená na .NET frameworku 4.7.2. Okrem iného, obsahovala funkčnú implementáciu komunikácie s Myo náramkom ako aj ukladanie dát do textových súborov. Ukladanie dát bolo realizované pri vyskytnutí sa udalosti, ktorá nastane pri získaní dát z náramka, čiže počas behu programu. Taktiež sú v nej vytvorené metódy, ktoré sa volajú pri stlačení klávesy a uvoľnení klávesy. Obsahuje počítadlo zadaných reťazcov do okna aplikácie, v ktoré slúži na zadávanie požadovaného reťazca v rámci experimentu. Bol implementovaný aj časovač merajúci čas od spustenia experimentu. Tento čas slúži na označenie dát. Pri zaznamenaní dát sa tak ku dátam priradí príslušný čas od spustenia aplikácie. Hlavná obrazovka obsahuje základné vstupné okná slúžiace na ovládanie a manipuláciu s aplikáciou. Projekt taktiež obsahuje ďalšiu obrazovku. Na nej sa zobrazujú grafy obsahujú vizualizáciu EMG dát generovaných Myo náramkom.

## 1.5 Softvérové technológie a programové vybavenie

Táto časť analyzuje dostupné softvérové technológie vhodné pre použitie v implementácii.

### 1.5.1 Vývojové prostredie Visual Studio 2019

Toto vývojové prostredie je plne vybavené pre vývoj rôznych druhov aplikácií. Ideálne však je pri vývoji aplikácií pre operačný systém Windows od spoločnosti Microsoft. Taktiež je vyvinuté spoločnosťou Microsoft [23]. Vďaka integrácii Windows Forms alebo Windows API je toto vývojové prostredie plne vybavené na rýchlu implementáciu oknových aplikácií. Okrem toho podporuje vývoj webových, mobilných alebo cross-platform aplikácií. V súčasnosti podporuje 36 rôznych programovacích jazykov. Medzi hlavné natívne podporované jazyky patria C#, F#, Visual Basic, JavaScript, C a v neposlednom rade C++.

### 1.5.2 Jazyk C#

Univerzálny programovací jazyk s viacerými paradigmami, ktorý obsahuje statické typovanie, silné typovanie, lexikálne škálovanie, imperatívne, deklaratívne,

funkčné, všeobecné, objektovo orientované a komponentovo orientované programovacie paradigmy.

C# bol vyvinutý okolo roku 2000 spoločnosťou Microsoft ako súčasť jeho iniciatívy .NET. Neskôr ho schválili ako medzinárodný štandard Ecma (ECMA-334) v roku 2002 a ISO (ISO / IEC 23270) v roku 2003. Bol navrhnutý Andersom Hejlsbergom a jeho vývojový tím v súčasnosti vedie Mads Torgersen. Je jedným z programovacích jazykov určených pre Common Language Infrastructure (CLI) [24]. Najnovšia verzia je 9.0, ktorá bola vydaná v roku 2020 v .NET 5.0 je zahrnutá vo Visual Studio 2019 verzii 16.8.

### 1.5.3 Myo SDK

Spoločnosť Thalmic Labs vyvinula tento nástroj, ktorý umožňuje prácu s Myo náramkom [25]. Cez toto SDK je riadené pripojenie náramkov aj zber dát. Samotné Thalmic Labs už na vývoji tohto SDK nepokračuje a vývoj sa zastavil. Všetky potrebné metódy však sú implementované. Existujú oficiálne verzie pre Windows, OSX, iOS a Android.

### 1.5.4 WindowsForms

Windows Forms (WinForms) je bezplatná a otvorená grafická knižnica tried (GUI) zahrnutá ako súčasť Microsoft .NET Framework, ktorá poskytuje platformu na písanie bohatých klientských aplikácií pre stolné počítače, notebooky a tablety [26]. Aj keď sa považuje za náhradu za staršiu a komplexnejšiu knižnicu Microsoft Foundation Class Library založenú na C ++, neponúka porovnateľnú paradigmu a slúži iba ako platforma pre vrstvu používateľského rozhrania vo viacúrovňovom riešení. Na podujatí Microsoft Connect 4. decembra 2018 spoločnosť Microsoft oznámila vydanie Windows Forms ako open source projektu na GitHubu. Je vydávaný pod licenciou MIT. S týmto vydaním je Windows Forms k dispozícii pre projekty zamerané na vývojový rámec .NET Core. Vývojový rámec je však stále k dispozícii iba na platforme Windows.

### 1.5.5 MyoSharp

Po publikovaní Myo SDK sa implementovali mnohé obalujúce knižnice pre zjednodušenie práce s ním. Dvaja prispievatelia, Nick Cosentino a Tayfun Unzun tak vytvorili knižnicu v jazyku C# [27]. Jednoducho sa vďaka nej dá pristupovať ku všetkým metódam, ktoré sú vo vnútri Myo SDK.

### 1.5.6 PyCharm

Integrované vývojové prostredie s názvom PyCharm je používané pri počítačom programovaní, špeciálne pre jazyk Python. Vyvíja ho česká spoločnosť JetBrains [28]. Poskytuje analýzu kódu, grafický debugger, integrované jednotkové testy, integráciu s riadiacimi systémami verzií a podporuje vývoj webových aplikácií v prostredí Django a dátovú vedu v prostredí Anaconda. PyCharm je multiplatformová aplikácia, ktorú je možné používať s verziami Windows, macOS a Linux.

### 1.5.7 Python

Tento jazyk patrí do skupiny interpretovaných, univerzálnych programovacích jazykov. Filozofia dizajnu Pythonu zdôrazňuje čitateľnosť kódu použitím výrazného odsadenia [29]. Jeho jazykové konštrukcie a objektovo orientovaný prístup majú za cieľ pomôcť programátorom napísať jasný a logický kód pre malé aj veľké projekty.

Python je dynamicky typovaný jazyk. Použitú pamäť počas behu aplikácie spravuje garbage collector. Podporuje niekoľko paradigiem programovania vrátane štruktúrovaného (najmä procedurálneho), objektovo orientovaného a funkcionálneho programovania.

### 1.5.8 Myo-Python

Knižnica ktorá obaluje Myo SDK a umožňuje prácu s týmto vývojovým nástrojom v jazyku python. Autorom tohto projektu je Niklas Rosenstein [30] a je publikovaný pod licenciou MIT.

### 1.5.9 Spracovanie videa

V rámci experimentu je realizované nahrávanie videa pomocou kamery, ktorá má za úlohu snímať klávesnicu. Preto sú v rámci analýzy skúmané dostupné knižnice pre prácu s videom.

#### AForge.NET

AForge.NET je open source C# framework určený pre vývojárov a výskumných pracovníkov v oblasti Computer Vision a umelej inteligencie. Tento framework umožňuje implementovať spracovanie obrazu, neurónové siete, genetické algo-

ritmy, fuzzy logiku, strojové učenie a v neposlednom rade aj robotiku [31]. Tento framework pozostáva z týchto hlavných častí:

- **AForge.Imaging** - knižnica s rutinami a filtrami spracovania obrazu.
- **AForge.Vision** - knižnica pre Computer Vision.
- **AForge.Video** - sada knižníc na spracovanie videa.
- **AForge.Neuro** - knižnica obsahujúca vybavenie pre implementáciu výpočtov podporujúcich neurónové siete.
- **AForge.Genetic** - knižnica obsahujúca programové techniky podporujúce implementáciu genetických algoritmov.
- **AForge.Fuzzy** - knižnica podporujúca fuzzy systémy.
- **AForge.Robotics** - podporná knižnica pre prácu s robotickými systémami
- **AForge.MachineLearning** - knižnica podporujúca implementáciu algoritmov strojového učenia

### **DirectX.Capture**

Táto knižnica je podobne ako predchádzajúca implementovaná v jazyku C# a .Net frameworku. Svoje využitie uplatňuje pri zázname video a audio záznamu, ktoré ukladá do AVI súborov [32]. Knižnica okrem iného obsahuje nasledujúce hlavné funkcie:

- List a výber hardvérových zariadení
- Prístup k bežným nastaveniam zvuku a videa (napr. Snímková frekvencia, veľkosť)
- Podpora kodekov na kompresiu zvuku a videa
- Podpora ukážky videa
- Podpora televíznych tunerov
- Podpora zvukových mixérov
- Načítanie možností zariadenia

## OpenCV

Najnovšia verzia knižnice OpenCV je licencovaná pod Apache 2 License. Knižnica OpenCV bola navrhnutá ako cross-platform knižnica. Knižnica bola teda napísaná v jazyku C a vďaka tomu je OpenCV prenosná na takmer akýkoľvek komerčný systém, od PowerPC Macs, až po robotické psy. Od verzie 2.0 obsahuje OpenCV svoje tradičné rozhranie v jazyku C aj nové C++ rozhranie. Nové algoritmy OpenCV sú väčšinou vyvinuté v C++. Boli vyvinuté aj wrappere pre jazyky ako Python, C# a Java, aby sa podporilo širšie nasadenie tejto knižnice. OpenCV funguje na stolných počítačoch (Windows, Linux, Android, MacOS, FreeBSD, OpenBSD) aj na mobilných zariadeniach (Android, iOS, Maemo) [33]. Medzi hlavné funkcie OpenCV patria:

- Čítanie a zápis obrázkov.
- Záznam a ukladanie videa.
- Spracovanie obrázkov (filtrácia, transformácia)
- Podpora rozpoznávania objektov na obrázkoch.
- Analýza videa, pohybujúcich sa objektov, odfiltrovanie pozadia.

## 1.6 Multi-threading a práca s dátami

Pre zabezpečenie plynulého toku dát zo všetkých zdrojov zároveň je vhodné spracovať tieto dáta nezávisle na sebe. Vďaka tomu nedôjde ku strate dát alebo ku možnému oneskorenému spracovaniu dát. To by mohlo spôsobiť skreslenie časových hodnôt získaných dát. Preto sa v tejto kapitole analyzujú dostupné riešenia vhodné pre implementáciu do tejto diplomovej práce. Multi-threading je koncept na súčasné spustenie viacerých operácií v rámci procesu, aby sa dosiahlo maximálne možné využitie výkonu procesora. Vlákno definuje cestu vykonania. Po spustení procesu sa spustí vlákno, ktoré sa nazýva hlavné vlákno. Ak má program veľa komplikovaných ciest, ktoré vykonávajú veľa časovo náročných a zložitých operácií, je možnosť každú z týchto ciest spustiť v samostatnom vlákne a dosiahnuť výsledok paralelne.

### 1.6.1 Vlákno

Cesta, na ktorej sa program spúšťa je definovaná ako vlákno a každé vlákno je definované jedinečným tokom kontroly. Pre viac vlákien sa musia nastaviť rôzne

cesty spúšťania. Každé vlákno je zodpovedné za konkrétnu úlohu. Táto metóda je efektívna keď sa aplikácia skladá z komplikovaných a časovo náročných operácií [34]. Samotné vlákna sú nenáročné a ich spustenie a beh nezaberá mnoho systémových zdrojov a času. Sú dobrým príkladom šetrenia plytvania cyklov CPU, pretože využívajú konkurenčné programovanie. Konštruktor tejto triedy tejto triedy akceptuje dva typy delegátov. Delegát je typ, ktorý predstavuje odkazy na metódy s konkrétnym zoznamom parametrov a návratovým typom.

Delegáti akceptované konštruktorom triedy Thread [34]:

- **ThreadStart** - Bez parametrická metóda.
- **ParameterizedThreadStart** - Akceptuje metódy s parameterom typu object.

Vlákna sa delia na dva základné typy:

- **Foreground Thread** - Pokiaľ je `IsBackground` vlastnosť triedy nastavená na `false`, `Foreground Thread` pokračuje v svojom vykonávaní aj pokiaľ sa ukončí rodičovský `Foreground Thread`.
- **Background Thread** - Pokiaľ je `IsBackground` vlastnosť triedy nastavená na `true`, `Background Thread` sa ukončí pokiaľ ukončí rodičovský `Foreground Thread`.

## 1.6.2 ThreadPool

Trieda `ThreadPool` je v .NET Frameworku k dispozícii od samého začiatku spolu s triedou `Thread` a spravuje ju CLR (Common Language Runtime). Trieda `ThreadPool` je wrapperom skupiny vlákien, ktoré sa dajú použiť na vykonávanie úloh, zverejňovanie pracovných úloh, spracovanie asynchrónnych vstupov a výstupov, čakanie na iné vlákna alebo procesné časovače. `ThreadPool` nie je dobrá voľba pokiaľ ide o dlhotrvajúce operácie [34]. Dôvodom je, že sa nedá zistiť či je práca, ktorú vykonáva ukončená a ani nevracia výsledok vykonávania.

`ThreadPool` poskytuje dva typy synchronizačných udalostí [34]:

- **ManualResetEvent** - Pomocou metódy `.Set()` je možné nastaviť stav ako otvorený. Metóda `.Reset()` nastavuje stav zatvorený. Metóda `.WaitOne()` blokuje thread, ktorý ju zavolať, pokiaľ sa nezavola metóda `.Set()`. Stav otvorený je nastavený pokiaľ sa nezavola metóda `.Reset()`.

- **AutoResetEvent** - Funguje podobne ako **ManualResetEvent**. Rozdiel je v tom, že sa automaticky na každom threade volá metóda `.WaitOne()` za ktorou sa vždy zavolá metóda `.Reset()`.

### 1.6.3 BackgroundWorker

Súčasťou .NET Frameworku je aj komponent **BackgroundWorker**. Umožňuje spúšťanie kódu v separátnych vláknach, vraciať stav kódu a jeho ukončenie [35]. Pri dostupnom viac jadrovom procesore je možné tieto thready spustiť na inom jadre ako na ktorom je spustená aplikácia, v ktorej sa **BackgroundWorker** vytvára.

Udalosti, ktoré sa vďaka implementácii tohto komponentu dajú využiť sú nasledovné [35]:

- **DoWork** - Pri zavolaní metódy `RunWorkerAsync()` sa vytvorí táto udalosť. V nej sa nachádza kód, ktorý sa má vykonať prostredníctvom tohto komponentu.
- **ProgressChanged** - Táto udalosť sa vyskytne pri zavolaní metódy `ReportProgress(Int32)`. Prostredníctvom tejto udalosti sa dá reagovať na zmenu progressu vykonaného kódu v **BackgroundWorker**.
- **RunWorkerCompleted** - Po ukončení vykonávania kódu **BackgroundWorkerom** sa vyskytne táto udalosť. Objaví sa po jeho úspešnom ukončení ale taktiež pokiaľ bolo vykonávanie zrušené alebo sa vyskytla výnimka počas behu programu.

### 1.6.4 Task Parallel Library (TPL)

Prvý krát sa táto trieda objavila v .NET 4.0 a jej úlohou je poskytnúť programátorom nástroje pre jednoduchú implementáciu paralelizmu a konkurencie v aplikáciách vyvíjaných prostredníctvom rámca .NET [34]. Obsahuje množinu verejných typov a API pre prácu v rámci namespace `System.Threading` a `System.Threading.Tasks`.

### 1.6.5 Linq

Táto trieda bola prvý krát predstavená v .NET Framework 3.5. Skratka **Linq** predstavuje **Language Integrated Query**. Integruje štandardné dopyty nad dátami priamo do jazyka **C#**. Dopyty na údaje sa tradične vyjadrujú ako jednoduché reťazce bez kontroly typu v čase kompilácie. Nie je potrebné učiť sa iný dopytovací jazyk pre

každý typ zdroja údajov. Napríklad databázy SQL, dokumenty XML a rôzne webové služby. Pri LINQ je dopyt prvotriednym jazykovým konštruktom, rovnako ako triedy, metódy, udalosti. Dopyty sa píšú na silne typované kolekcie objektov pomocou jazykových kľúčových slov a známych operátorov [36]. Rodina technológií LINQ poskytuje konzistentné prostredie dopytov na objekty, relačné databázy a XML.

Dôkazom o efektivite použitia knižnice LINQ oproti klasickým operáciám, ktoré sú bežne používané pri práci s kolekciami, ako slučky `for()` alebo `foreach()` je aj benchmark [37], zobrazený na obrázku Obr. 1.9, ktorý porovnáva tieto operácie s použitím LINQ. Rozdieli medzi jednotlivými operáciami sú síce malé a pri určitej veľkosti kolekcie, nad ktorou sú operácie vykonávané sa efektivita jednotlivých operácií strieda v prvenstve, ale LINQ má ďalšie výhody ako:

- **Stručnosť** - Použitím LINQ sa dá nahradiť rozsiahlejšia metóda jedným riadkom kódu.
- **Štýl toku dát** - Flexibilný a ľahko pochopiteľný.
- **Paralelizácia** - Stačí použiť PLINQ a zavolať metódu `AsParallel()`.



Obr. 1.9: Vykonanie operácie `select` nad kolekciami objektov rôznej veľkosti metódami `for`, `foreach` a LINQ. (Zdroj: [38])

## 1.7 Vyhodnotenie analýzy

Pre potreby tejto práce je dôležité použitie vhodných technológií a postupov, ktoré zaručia čo najvyššiu možnú konzistenciu dát na výstupe aplikácie. Výsledok ana-



lýzy dostupných knižníc pre spracovanie videa znie nasledovne. Analyzované boli tri knižnice, ktoré sú vhodné na použitie v implementácii aplikácie tejto diplomovej práce. Pre použitie s jazykom C# a .NET frameworkom sa ale ako najlepšia ukázala knižnica AForge.NET a to z niekoľkých dôvodov. Táto knižnica je open source a vývoj na nej stále pokračuje, narozdiel od knižnice DirectX.Capture kde sa vývoj zastavil v roku 2003. Časom sa tak môžu pridať ďalšie zaujímavé funkcie, ktoré budú rozšírením tejto knižnice. Taktiež bude táto knižnica držať krok s dobou a ostatnými technológiami. Široká škála funkcií, ktoré už obsahuje umožní ďalšie prípadné rozšírenia tejto aplikácie o analýzu obrazu alebo iné zaujímavé funkcie. V prípade použitia jazyka Python sa ako najlepšia alternatíva javí knižnica OpenCV. Táto knižnica taktiež obsahuje mnohé užitočné funkcie a je kompatibilná s jazykom Python.

Ako vhodná metóda pre rozdelenie operácií spracúvajúcich dáta sa javí BackgroundWorker. Pomocou operácií v jednotlivých vláknach sa zabezpečí, že sa v rámci jedného vlákna nebude blokovať zaznamenávanie dát z ostatných zariadení na úkor jedného zariadenia. BackgroundWorker obsahuje možnosti pre kontrolu vláken a v prípade WindowsForms aplikácií predstavuje prirodzený komponent na integráciu do aplikácie vďaka jeho podpore.

Zvolenie vhodnej knižnice a jazyka pre vyhotovenie hlavnej časti tejto diplomovej práce nie je možné len na základe analýzy dostupných informácií. Kombinácia MyoSharp a jazyk C# sa javí rovnako vhodná ako kombinácia Myo-Python a jazyk Python. Pre oba jazyky existujú vhodné efektívne riešenia implementácie tejto práce. Pred tým než sa teda začne návrh a implementácie tejto práce je nutné vykonať experimentálnu overenie, ktoré ukáže využitie týchto technológií v praxi a porovná výsledky jednoduchých testov týchto riešení.

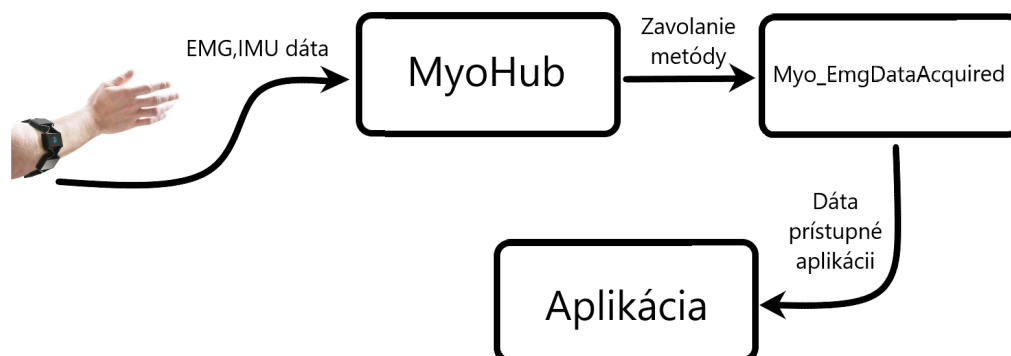
## 2 Experimentálne overenie knižníc pre Myo

---

Medzi hlavné ciele tejto práce patrí zabezpečenie zaznamenania a uloženia čo najväčšej množiny dát, ktoré generujú myo náramky. V rámci dosiahnutia tohto cieľa boli vytvorené aplikácie v jazykoch, v ktorých boli dostupné knižnice pre prácu s Myo náramkom. Prvou z nich je myosharp. Táto knižnica bola vytvorená v jazyku C# a funguje ako obalová trieda pre MyoSDK. Vývoj aplikácie teda prebieha pomocou jazyka C#. Druhou možnosťou, ktorá bola skúmaná bola knižnica myo-python. Táto knižnica je taktiež obalová trieda pre MyoSDK. Vývoj aplikácie pomocou tejto knižnice je realizovaný v jazyku Python v.0.2.x. V tejto kapitole sa nachádza popis jednotlivých implementácií ako aj vyhodnotenie, ktorá z knižníc je vhodná na použitie v rámci implementácie tejto diplomovej práce. Pri vyhodnotení sa hľadí najmä na množstvo dát za sekundu, ktoré sa podarilo zaznamenať z oboch náramkoch použitím jednotlivých knižníc ale taktiež podporu knižníc pre ďalšie potreby, ktoré finálna aplikácia na zaznamenávanie experimentu potrebuje.

### 2.1 Test myosharp aplikácie

Ako bolo uvedené v predchádzajúcej sekcii, prvou implementáciou, ktorá bola testovaná bola kombinácia jazyka C# a knižnice myosharp. Bol navrhnutý jednoduchý program, ktorý mal za úlohu uložiť EMG dáta vygenerované senzormi Myo náramku. Myo náramok generujúci dáta je pripojený na MyoHub ako je ilustrované na obrázku Obr. 2.1. Pri prijatí dát sa v aplikácii zavolá event *Myo\_EmgDataAcquired*, ktorá je implementovaná ako event listener na EventHandlerler *EmgDataAcquired*, ktorého argumentom sú dáta typu *EmgDataEventArgs*. Tento EventHandlerler je súčasťou rozhrania *IMyoEventGenerator* knižnice myosharp. *EmgDataEventArgs* nesú samotné dáta zosnímané senzormi Myo náramku a odoslané do aplikácie. Týmto krokom sa teda sprístupnia EMG dáta pre spracovanie aplikáciou.



Obr. 2.1: Diagram zobrazujúci spracovanie dát aplikáciou.

Pri testovaní efektivity získavania dát z Myo náramka touto aplikáciou bol ku dátam, ktoré sa spracovávali v metóde *Myo\_EmgDataAcquired* pridaný timestamp, ktorý označil čas, kedy boli dáta spracované od začiatku spustenia zberu dát. Teda čas 0 bol časom, kedy bol Myo náramok pripojený ku aplikácii, pripravený na prácu s nim a zároveň spustení záznam dát. Maximálna frekvencia odosielania dát Myo náramkom je podľa analyzovaných zdrojov 200Hz pre jeden komunikačný kanál pripojený na MyoHub. Výsledkom tohto experimentu teda je ako je v skutočnosti schopná kombinácia myosharp a C# zaobstaráť zber dát v porovnaní s referenčnou hodnotou 200Hz.

Testovanie pozostávalo z troch hlavných častí.

1. Použitie prvého Myo náramku označeného LIRKIS nálepkou.
2. Použitie druhého Myo náramku bez označenia.
3. Kombinácia oboch Myo náramkov.

### 2.1.1 Výsledok testu

Pri každom testovanom prípade bola realizovaná sada meraní. Oba náramky boli plne nabité ako aj notebook, ku ktorému boli pripojené bol napojený na zdroj energie. Výsledkom prvého prípadu, kde bolo použité iba Myo s LIRKIS nálepkou dopadlo s výsledkom v rozmedzí 190 až 200 dát za sekundu. Myo bez nálepky v rovnakom teste dopadlo s hodnotou pohybujúcou sa okolo 160 dát za sekundu. Pri kombinácii oboch náramkov bolo zistené, že sa počet dát rovná súčtu

dát získaných z oboch náramkov a teda sa na úrovni 350 až 360 dát za sekundu.

## 2.2 Test myo-python aplikácie

Druhou testovanou implementáciou bola aplikácia vyhotovená pomocou jazyka Python a knižnice myo-python. Podobne ako pri predchádzajúcej možnosti bol navrhnutý jednoduchý program na zaznamenávanie EMG dát. V rámci implementovanej triedy *Listener* je dôležitá metóda *on\_emg*. Táto metóda sa volá vždy pri prijatí dát aplikáciou. Na začiatku metódy je implementovaný časovač, ktorý sa spustí pri prvom zavolaní tejto metódy. Táto metóda ďalej sprístupňuje EMG dáta na manipuláciu pre aplikáciu. Spracovanie dát je teda realizované podobne ako pri myosharp aplikácii Obr. 2.1. Namiesto metódy *Myo\_EmgDataAcquired* je to ale uvedená metóda *on\_emg*. Ako aj v predošlej aplikácii, meria sa množstvo dát za sekundu od prijatia prvých dát. Testovanie pozostávalo z rovnakých troch častí ako testovanie myosharp aplikácie.

### 2.2.1 Výsledok testu

Podmienky testu boli zhodné s podmienkami myosharp aplikácie. Bola realizovaná sada meraní s plne nabitými náramkami a notebookom napojeným na zdroj energie. V prvom testovanom prípade, Myo s LIRKIS nálepkou, bol zaznamenaný výsledok blížiaci sa 200 dátam za sekundu. Výsledky Myo náramku bez nálepky sa nachádzali v rozmedzí 160 až 170 dát za sekundu. Pri kombinácii oboch náramkov sa výsledok rovnal súčtu dát samostatne a pohyboval sa na úrovni 360 dát za sekundu.

## 2.3 Vyhodnotenie

Experimentálna analýza mala za úlohu porovnať myosharp a myo-python aplikácie. Sústreďovala sa na efektivitu získania dát. Myo náramok bez LIRKIS nálepky v oboch prípadoch zaostával za náramkom s LIRKIS nálepkou. Predpokladaná príčina tohto zaostania je hardvérový nedostatok. Pri myo-python riešení bolo však v priemere zaznamenaných viac dát. Rozdiel je približne 10 záznamov za sekundu. Pri dátach z náramku s LIRKIS nálepkou sú výsledky zhodné. Vo výsledkoch jednotlivých riešení sa ukázalo, že myo-python aplikácia sa v celku pohybovala mierne nad hodnotami myosharp riešenia. Pre posúdenie vhodnosti riešenia je však nutné pozrieť sa aj na ďalšie dôležité vlastnosti knižníc. Knižnica

myo-python neobsahuje metódu pre identifikovanie náramku na základe unikátneho označenia. To je dôležité najmä pre zabezpečenie konzistencie dát počas priebehu experimentu. Je totiž možné, že Myo náramok sa na ruke počas experimentu chybné prekalibruje na opačnú ruku a teda pri priradení názvu ruky ku dátam by sa tieto znehodnotili. Preto je za potreby pri spustení experimentu zabezpečiť, že táto akcia nebude mať vplyv na dáta počas experimentu. Na druhej strane, knižnica myosharp poskytuje možnosť použiť identifikátor pripojenia náramku ako unikátneho označenia pre náramok. Toto označenie sa nemení počas celej doby pripojenia náramku. Rozdiel v získaných dátach je minimálne a skutočnosť potreby identifikovania Myo náramku jednoznačne ukazuje myosharp riešenie ako optimálne pre implementovanie v tejto diplomovej práci.

# 3 Návrh systému pre realizovanie experimentov

---

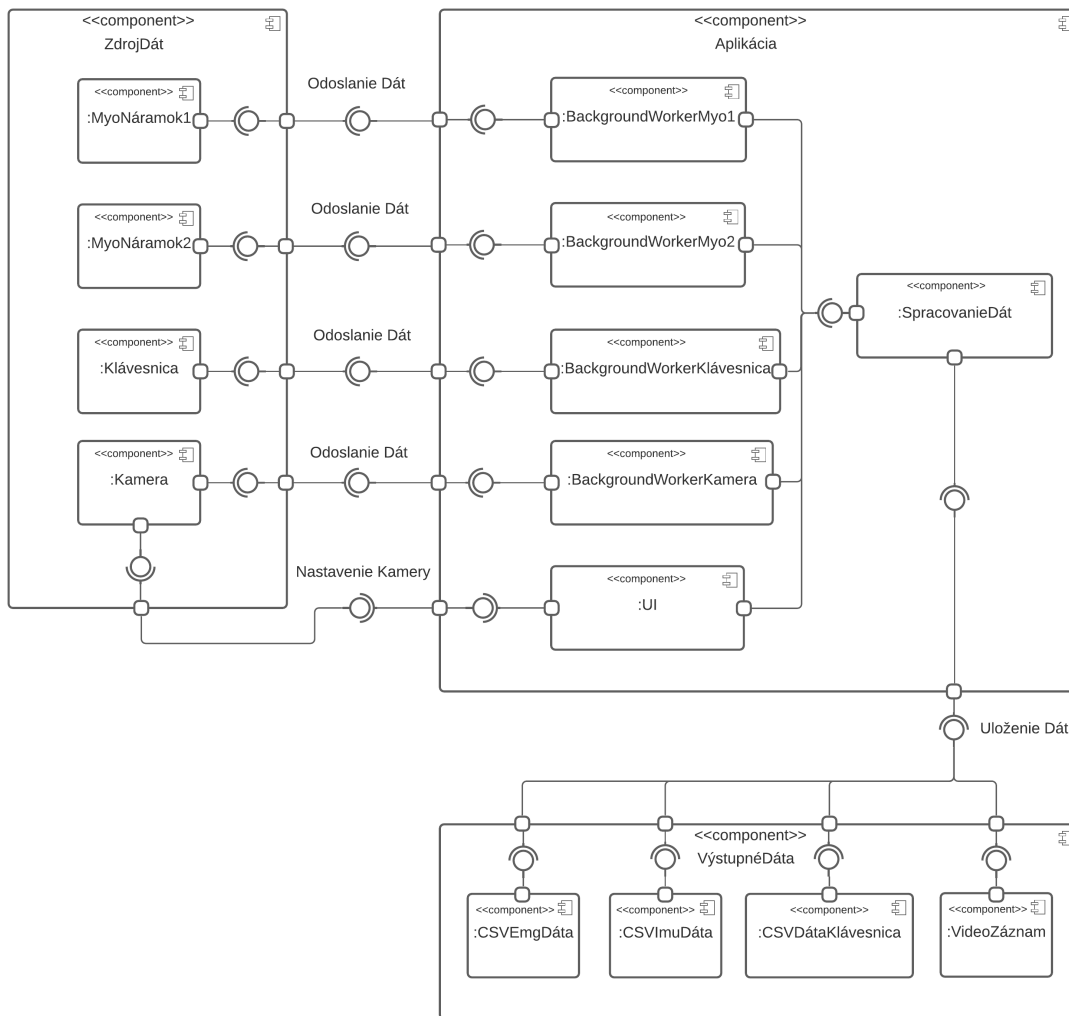
Po analyzovaní dostupných technológií a existujúcich riešení sa dá zostaviť návrh pre budúci funkčný systém. V tejto časti je teda opísaný návrh systému, ktorý slúži na riadenie a zaznamenávanie experimentov. Celý systém pozostáva z troch hlavných častí ako je možné vidieť na obrázku Obr. 3.1:

1. **Zdroj dát** - Dáta, ktoré sú predmetom experimentu, je potrebné získať z ich zdroja. V tejto časti sú definované zdroje dát.
2. **Aplikácia** - Zaznamenáva a spracováva dáta získané zo zdroja. Taktiež umožňuje riadiť experiment prostredníctvom používateľského rozhrania.
3. **Výstupné dáta** - Predstavujú uložené dáta ako výstup experimentu.

Pre realizovanie experimentu je taktiež vhodné navrhnuť postupnosť krokov, ktoré vedú k jeho úspešnému ukončeniu. Tieto kroky slúžia aj ako návod pre experimentátora na riadenie experimentu:

1. **Klávesnica** - Je vhodné použitie rovnakého modelu klávesnice pri všetkých experimentoch. Najvhodnejšie sú klávesnice s mechanickými spínačmi. To z dôvodu dlhšej dráhy stlačenia klávesy. Klávesnica sa zapojí do počítača a umiestni na okraj stola.
2. **Kamera** - Umiestni sa tak, aby snímala pohľad zhora na celú klávesnicu. Následne sa zapojí do počítača.
3. **Pripojenie Myo náramkov** - V ďalšom kroku sa pripoja Myo náramky ku počítaču. Skontroluje sa ich pripojenie v aplikácii Myo Connect. Pred pokračovaním na ďalší krok je potrebné skontrolovať stav batéria náramkov. Je vhodné aby bola batéria nabitá na svoju maximálnu kapacitu.

4. **Nasadenie Myo náramkov** - V tomto kroku sú Myo náramky nasadené na ruky subjektu vykonávajúceho experiment. Experimentátor sa uistí, že sú Myo náramky umiestnené správne, tak ako je znázornené na obrázku Obr. 3.2. Prvý náramok sa umiestni na ľavú ruku, druhý na pravú ruku. Je potrebné umiestniť náramok tak, aby držal na ruke čo najpevnejšie. V prípade, že má subjekt menší priemer predlaktí, je nutné použiť príslušenstvo určené na zmenšenie priemeru náramkov. Náramky je možné prehodiť v rámci rúk pred spustením experimentu. Počas vykonávania experimentu sa s nimi nesmie manipulovať.
5. **Spustenie aplikácie** - Po úspešnom zvládnutí predchádzajúcich krokov je potrebné spustiť aplikáciu pre zaznamenanie experimentov.
6. **Zadanie vstupných parametrov** - Experimentátor nastaví výstupný priečnik experimentu, meno subjektu, zvolí pripojenú kameru, ktorá slúži na nahranie videozáznamu experimentu.
7. **Usadenie subjektu** - Je potrebné, aby experimentátor dohliadol na usadenie subjektu za klávesnicu. Ak je to možné, treba nastaviť výšku posedu tak, aby ruky subjektu boli nad klávesnicou a neopierali sa o okraj stola. Subjekt by mal mať ruky v prirodzenej polohe, v oblasti lakťa ohnuté v uhle 90 stupňov. Lakte má subjekt zároveň držať pri tele.
8. **Spustenie záznamu experimentu** - Po vykonaní predchádzajúcich úkonov spustí experimentátor záznam experimentu a subjekt začne na klávesnici písať vopred určené slovo. Počas celého priebehu experimentu dohliada experimentátor na dodržanie na zachovanie polohy rúk podľa inštrukcií popísaných pri usádzaní subjektu. Explicitne má kontrolovať, aby sa subjekt neopieral rukami o okraj stola.
9. **Ukončenie záznamu experimentu** - Po dosiahnutí potrebného počtu správne zadaných textových reťazcov subjektom do aplikácie sa automaticky ukončí záznam experimentu. Záznam je možné ukončiť aj manuálne, pokiaľ je to potrebné.
10. **Pokračovanie v ďalších experimentoch** - Ak nasledujú ďalšie experimenty s rovnakým používateľom. Nie je potrebné vykonávať všetky kroky uvedené v tejto časti. Stačí pokiaľ sa znovu spustí záznam experimentu. Pred ďalším nahrávaním je vhodné aby si subjekt oddýchol a uvoľnil ruky. Ak je nasleduje iný subjekt, je potrebné začať na kroku 6.



Obr. 3.1: Návrh systému.



Obr. 3.2: Správne nasadenie Myo náramku.



Táto kapitola sa ďalej venuje podrobnejšiemu popisu návrhu jednotlivých komponentov systému.

### 3.1 Zdroje dát

Prvou súčasťou systému sú zdroje dát. Tie predstavujú Myo náramky, klávesnica a kamera. Každý z týchto zdrojov generuje rôzne dáta generované počas behu experimentu. Sú na sebe navzájom nezávislé.

Dáta generované zdrojmi sú nasledovné:

- **Myo náramky** - Tieto zdroje generujú EMG a IMU dáta.
- **Klávesnica** - V momente stlačenia klávesy generuje dáta o znaku klávesy ako aj či ide o fázu stláčania klávesy alebo uvoľňovania klávesy.
- **Kamera** - Generuje sekvenciu obrázkov v čase.

### 3.2 Návrh aplikácie

Výsledkom má byť aj aplikácia slúžiaca pre zaznamenanie experimentov. Je možné rozdeliť ju na tri hlavné časti:

1. **BackgroundWorker** - Táto trieda umožní zaznamenanie dát v separátnych vláknach a sprístupní dáta aplikácii.
2. **Spracovanie dát** - Uloženie dát do štruktúr a zapísanie do výsledných súborov experimentu.
3. **UI** - Používateľské rozhrnie slúžiace na obsluhu experimentu.

#### 3.2.1 Ukladanie dát pomocou BackgroundWorkerov

Pre každý zdroj dát je vhodné vytvoriť dedikovaný BackgroundWorker. Tie sa vytvoria pri spustení aplikácie a vykonávanie kódu, ktorý zabezpečí ukladanie dát do kolekcí sa začne v momente spustenia experimentu.

#### 3.2.2 Spracovanie dát

Po tom ako sú dáta uložené v kolekciách je potrebné vykonať nad nimi operácie ako zoradenie podľa času a v niektorých prípadoch priradenie označenia ruky

Myo náramku, ktorý ich vygeneroval. V prípade videa je potrebné len vytvoriť súbor do ktorého sa majú snímky v čase ukladať. Pre ostatné dáta je ideálnym riešením vytvorenie CSV súborov. Použitie CSV súborov ako aj ich štruktúru navrhol vedúci tejto diplomovej práce, ako ideálne riešenie ukladania dát obsiahnutých v kolekciami. Pri zadaní nového subjektu je vhodné vytvoriť adresárovú štruktúru v koreňovom adresári experimentu. Tá by mala obsahovať adresár s menom subjektu. V ňom by sa mal nachádzať priečinkom, ktorého názov bude tvoriť aktuálny dátum. Vo výsledku je teda na výstupe aplikácie 6 súborov.

### Video záznam

Obsahuje snímky obrazovky z priebehu experimentu. Formát súboru je WMV. Názov súboru tvorí [MenoSubjektu]-[DD-MM-YYYY]-[(HH-MM-SS)].

### Záznam klávesnice

Ďalším navrhovaným súborom na výstupe je súbor obsahujúci zaznamenané a spracované dáta z klávesnice. Jeho meno je vždy **keyboard.csv**. Štruktúra tohto súboru definuje názvy stĺpcov, ktoré sú oddelené čiarkou.

Stĺpce sú definované nasledovne:

1. **timestamp** - Stĺpec obsahuje čas, v ktorom boli dáta zaznamenané. Čas sa počíta od spustenia experimentu.
2. **event** - Dáta v tomto stĺpci môžu nadobudnúť hodnotu **keyPressed**, ktorá definuje dáta zaznamenané pri stlačení klávesy, alebo **keyReleased**, ktorá definuje dáta pri uvoľnení klávesy.
3. **key** - Obsahuje samotný znak stlačenej klávesy. Pokiaľ je hodnota **eventu** nastavená na **keyPressed**, znak je formátovaný ako malé písmeno, pokiaľ je to hodnota **keyReleased**, znak je formátovaný ako veľké písmeno.
4. **correct** - Označuje správnosť zadaného znaku v definovanej postupnosti, ktorá predstavuje slovo písané počas experimentu. Nadobúda hodnotu 1 pokiaľ je zadaný znak v postupnosti správny a hodnotu 0 pokiaľ je znak nesprávny.

### Záznam EMG dát náramku na ľavej ruke

Tento súbor obsahuje EMG dáta zaznamenané Myo náramkom na ľavej ruke. Názov toto súboru je **myoLeftEmg.csv**. Jeho konštrukcia vyzerá nasledovne:

1. **timestamp** - Stĺpec obsahuje čas, v ktorom boli dáta zaznamenané. Čas sa počíta od spustenia experimentu.
2. **event** - Obsahuje len textový reťazec **Left**. Slúži len na rýchlejšiu orientáciu, keďže určuje, že dáta sú z ľavej ruky.
3. **emg1,emg2,...,emg8** - Obsahuje samotný znak stlačenej klávesy. Pokiaľ je hodnota **eventu** nastavená na **keyPressed**, znak je formátovaný ako malé písmeno, pokiaľ je to hodnota **keyReleased**, znak je formátovaný ako veľké písmeno.
4. **correct** - Označuje správnosť zadaného znaku v definovanej postupnosti, ktorá predstavuje slovo písané počas experimentu. Nadobúda hodnotu 1 pokiaľ je zadaný znak v postupnosti správny a hodnotu 0 pokiaľ je znak nesprávny.

#### Záznam EMG dát náramku na pravej ruke

Ukladanie EMG dát z Myo náramku na pravej ruke je realizované podobne ako to bolo pri ľavej ruke. Rozdiel je v názve súboru a obsahu stĺpca **event**. Názov súboru teda je **myoRightEMG.csv** a stĺpec **event** obsahuje vždy textový reťazec **Right**.

#### Záznam IMU dát náramku na ľavej ruke

Pre ukladanie IMU dát z Myo náramka na ľavej ruke sa vytvorí súbor **myoLeftImu.csv**. Názvy stĺpcov a popis dát, ktoré obsahujú je nasledovný:

1. **timestamp** - Stĺpec obsahuje čas, v ktorom boli dáta zaznamenané. Čas sa počíta od spustenia experimentu.
2. **event** - Obsahuje len textový reťazec **Left**. Slúži len na rýchlejšiu orientáciu, keďže určuje, že dáta sú z ľavej ruky.
3. **roll** - Rotácia Myo náramka okolo osi x.
4. **pitch** - Rotácia Myo náramka okolo osi y.
5. **yaw** - Rotácia Myo náramka okolo osi z.

### Záznam IMU dát náramku na pravej ruke

Dáta z pravej ruky sú ukladané rovnako ako dáta z ľavej ruky. Názov súboru je však v tomto prípade **myoRightImu.csv** a stĺpec **event** obsahuje len hodnotu **Right**.

## 3.3 Návrh používateľského rozhrania

Vytváraná aplikácia je určená na vykonávanie experimentov. Pre zjednodušenie a zefektívnenie ich priebehu je vhodné navrhnúť grafické používateľské rozhranie, vďaka ktorému sa osoba vedúca experiment bude ľahko orientovať v organizácii experimentu. Taktiež subjekt(osoba), ktorý experiment vykonáva môže využiť výhodu jednoduchého a prehľadného používateľského rozhrania a to tak, že sa bude môcť orientovať či už v korektnosti napísaného slova, alebo v počte slov, ktoré ešte potrebuje napísať pre správne ukončenie experimentu. Toto grafické používateľské rozhranie je preto dôležitou súčasťou tejto práce.

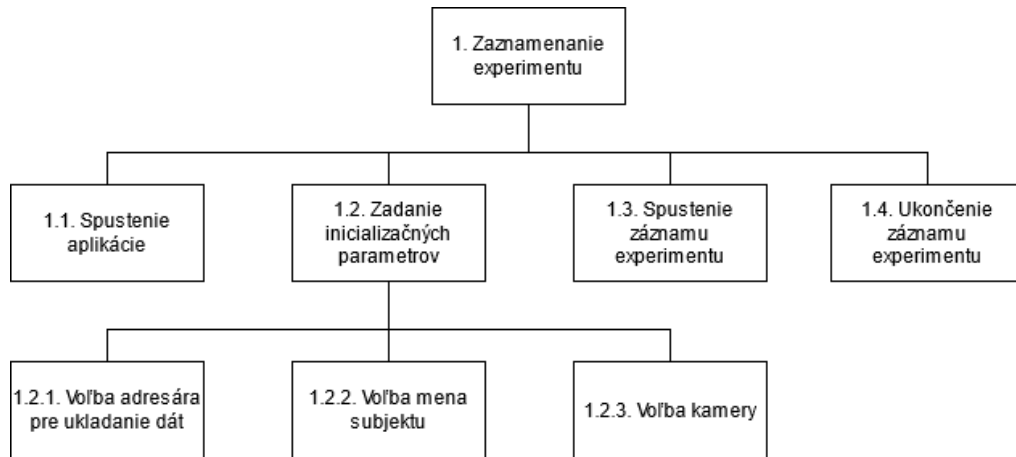
### 3.3.1 Hierarchická analýza úloh

V prvej časti návrhu používateľského rozhrania bol vytvorený diagram hierarchickej analýzy úloh. Tento diagram slúži na definovanie úloh, ktoré je potrebné vykonať v používateľskom rozhraní na dosiahnutie určitého cieľa, ktorý má aplikácia umožniť. Úlohy sú v ňom definované od generalizovaných úloh až po konkrétne kroky, ktoré je potrebné vykonať na dosiahnutie úloh.

Diagram Obr. 3.3 opisuje úlohy, ktoré musí používateľ vykonať pri obsluhu aplikácie. Hlavnou úlohou je zaznamenanie priebehu experimentu a dát, ktoré vstupné zariadenia použité v tomto experimente vygenerujú. Pre splnenie tejto úlohy je potrebné vykonať 4 úlohy, ktoré na seba navzájom nadväzujú.

Prvou z týchto úloh je spustenie aplikácie. Splnenie tejto úlohy umožní používateľovi prejsť na ďalšiu úlohu, ktorou je zadanie inicializačných parametrov aplikácie. Pre vykonanie tejto úlohy musí používateľ vykonať podsekvenciu 3 úloh. Voľba adresára pre ukládanie dát je prvou z nich. Po zvolení príslušného koreňového adresára experimentu nasleduje ďalšia podúloha, a síce voľba mena subjektu. Po zadaní alebo nevolení mena subjektu používateľ vykoná poslednú úlohu z podsekvencie úlohy pre zadanie inicializačných parametrov. Touto úlohou je voľba kamery. Po zvolení kamery, ktorú chce používateľ použiť na nahrávanie experimentu nasleduje úloha z hlavnej sekvencie ktorou je spustenie záznamu

experimentu. Poslednou úlohou ktorú musí používateľ vykonať pre zaznamenanie experimentu je ukončenie záznamu experimentu.



Obr. 3.3: Diagram hierarchickej analýzy úloh

### 3.3.2 Konceptuálny model

Ďalšou časťou návrhu používateľského rozhrania je zhotovený konceptuálny model, ktorý je znázornený na obrázku Obr. 3.4. Podľa jeho návrhu má používateľ kontrolu nad základnými interakciami s aplikáciou.

Prvou opísanou interakciou je voľba ľubovoľného adresára, do ktorého budú zaznamenané dáta z priebehu experimentu. V tomto kroku môže používateľ, podľa navrhnutého modelu, zvoliť z dvoch možností výberu adresára. Prvou možnosťou je manuálne zadanie celej cesty do textového poľa. Táto možnosť má výhodu v prípade, ak má používateľ presne určený adresár, kam sa majú dáta z experimentu ukladať. Takouto cestou môže byť napríklad aj zdieľaný sieťový disk. Druhou možnosťou je voľba adresára pomocou kontextového okna. Tu môže používateľ rýchlo zvoliť cestu v stromovej štruktúre adresárov. Táto možnosť pomôže používateľovi lepšie sa zorientovať v štruktúre adresárov pre jej grafické zobrazenie. V prehľadnom zobrazení si tak vyberie adresár, ktorý si zvolí ako koreňový adresár experimentu.

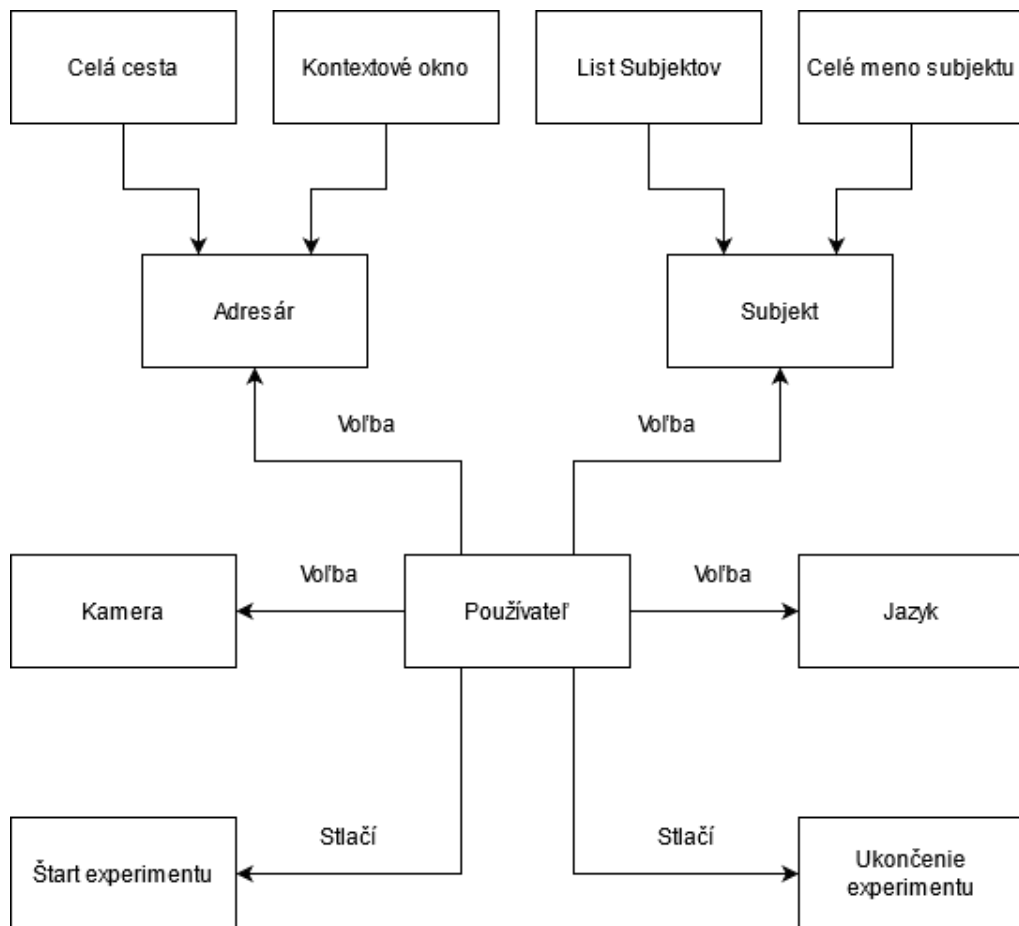
Druhou interakciou používateľa s grafickým rozhraním aplikácie popísanou v navrhnutom konceptuálnom modeli je výber subjektov. Výber subjektu predstavuje voľbu mena testovacieho subjektu (osoby). Tento výber je možné realizovať zvolením mena subjektu z zoznamu subjektov. Zoznam subjektov obsahuje mená subjektov, ktoré už experiment v minulosti absolvovali. Táto možnosť teda zrýchli fázu zadávania inicializačných parametrov experimentu. Výber subjektu je však možné realizovať aj napísaním mena experimentu. Zadanie celého mena

experimentu je vhodné v situácii prvej účasti konkrétneho subjektu na realizácii experimentu.

Ďalšou v poradí treťou opísanou interakciou používateľa s aplikáciou je v navrhnutom konceptuálnom modeli voľba kamery. Táto interakcia umožní používateľovi zvoliť prostredníctvom kontextového okna kameru, ktorej výstup vo forme video záznamu obsahuje priebeh experimentu. Používateľovi je taktiež umožnené nastaviť jazyk aplikácie.

Predposlednou interakciou opísanou v konceptuálnom modeli je interakcia používateľa a štartu experimentu. Po stlačení tlačidla štartu experimentu sa spustí záznam experimentu.

Poslednou interakciou, ktorá je obsiahnutá v konceptuálnom modeli je interakcia používateľa a ukončenia experimentu. Po stlačení tlačidla ukončenia experimentu sa ukončí záznam experimentu a uloží sa do zvoleného adresára.



Obr. 3.4: Konceptuálny model

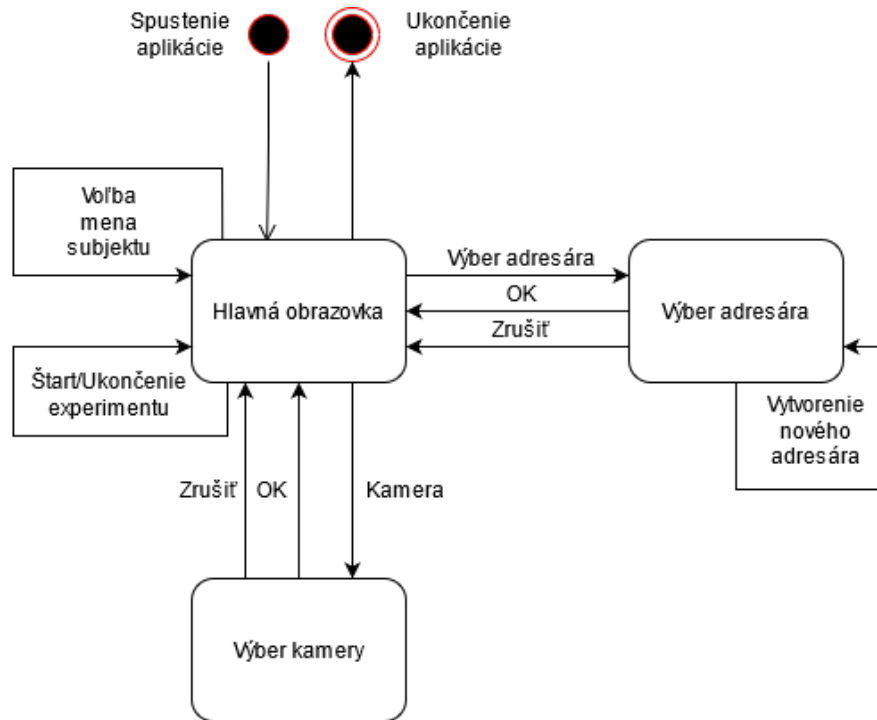
### 3.3.3 Model sekvencie obrazoviek

Na základe hierarchickej analýzy úloh a konceptuálneho modelu je navrhnutý stavový diagram sekvencie obrazoviek Obr. 3.5. Tento diagram slúži ako návrh akcií, ktoré je možné vykonať na jednotlivých obrazovkách. Zároveň opisuje ako sú realizované prechody medzi obrazovkami. Táto časť návrhu je dôležitá pre navrhnutie prototypu na základe ktorého je implementované používateľské rozhranie aplikácie.

Inicializačným stavom je spustenie aplikácie. Z tohto stavu je iniciovaná hlavná obrazovka. Hlavnú obrazovku je možné považovať za stav z ktorého je možné prejsť do ľubovoľného ďalšieho stavu, ktorý používateľské rozhranie môže nadobudnúť. V stave hlavnej obrazovky je možné vykonať dve akcie, ktoré nezmenia stav používateľského rozhrania. Prvou z týchto akcií je voľba mena subjektu. Druhou akciou je štart alebo ukončenie experimentu.

Po vykonaní akcie výber adresára používateľské rozhranie zmení svoj stav z hlavnej obrazovky na výber adresára. Zobrazí sa teda príslušná obrazovka. Na tejto obrazovke je možné vykonať akciu vytvorenie nového adresára, ktorá nezmení stav a teda ani obrazovku používateľského rozhrania. Okrem tejto akcie je možné vykonať dve ďalšie akcie, ktoré zmenia stav rozhrania. Prvou akciou je OK. Táto akcia potvrdí úspešné zvolenie adresára a zobrazí hlavnú obrazovku. Druhou akciou je zrušiť. Táto akcia zruší výber adresára a taktiež zobrazí hlavnú obrazovku.

Ďalšou akciou, ktorú môže používateľ vykonať na hlavnej obrazovke a zmení stav používateľského rozhrania je kamera. Po vykonaní tejto akcie nadobudne rozhranie stav výber kamery a zobrazí sa príslušná obrazovka. Na tejto obrazovke je možné vykonať akcie OK a zrušiť. Akcia OK potvrdí zvolenú kameru a zobrazí okno hlavnej obrazovky. Akcia zrušiť podobne zobrazí okno hlavnej obrazovky, avšak voľbu kamery nevykoná. Poslednou akciou, ktorú je možné vykonať na hlavnej obrazovke je ukončenie aplikácie.



Obr. 3.5: Sekvenčný diagram obrazoviek

### 3.3.4 Prototyp používateľského rozhrania

Po analyzovanej hierarchii úloh, vytvorení konceptuálneho modelu a definovaní vzťahov jednotlivých obrazoviek sú informácie získané v tomto procese využité pri návrhu prototypu používateľského rozhrania. Zostrojenie prototypu je vhodné najmä kvôli eliminácii možných nejasností budúceho implementovaného používateľského rozhrania, ktoré by mohli vyvstať na strane používateľa. Tieto nejasnosti sa môžu týkať rôznych aspektov používania aplikácie. Tým sa pozornosť používateľa zbytočne odtrhne od dôležitých vecí a upriami sa na ovládanie aplikácie, ktoré by malo byť čo najintuitívnejšie a najprirodzenejšie. Vytvorenie prototypu zároveň uľahčí implementáciu aplikácie, keďže jej vzhľad, rozloženie elementov ale taktiež funkcionality používateľského rozhrania je pevne definovaná. Eliminuje sa taktiež potreba častých úprav tohto rozhrania aplikácie, keďže je navrhnuté tak aby bolo čo najvhodnejšie.

#### Hlavná obrazovka

Návrh prototypu hlavnej obrazovky Obr. 3.6 je vytvorený s dôrazom na prehľadnosť a jednoduchosť, keďže na tejto obrazovke stráni používateľ najviac času.

Vertikálne je hlavná obrazovka farebne rozdelená na dve časti. Na ľavej časti obrazovky s nachádzajú všetky elementy pomocou ktorých používateľ zadáva

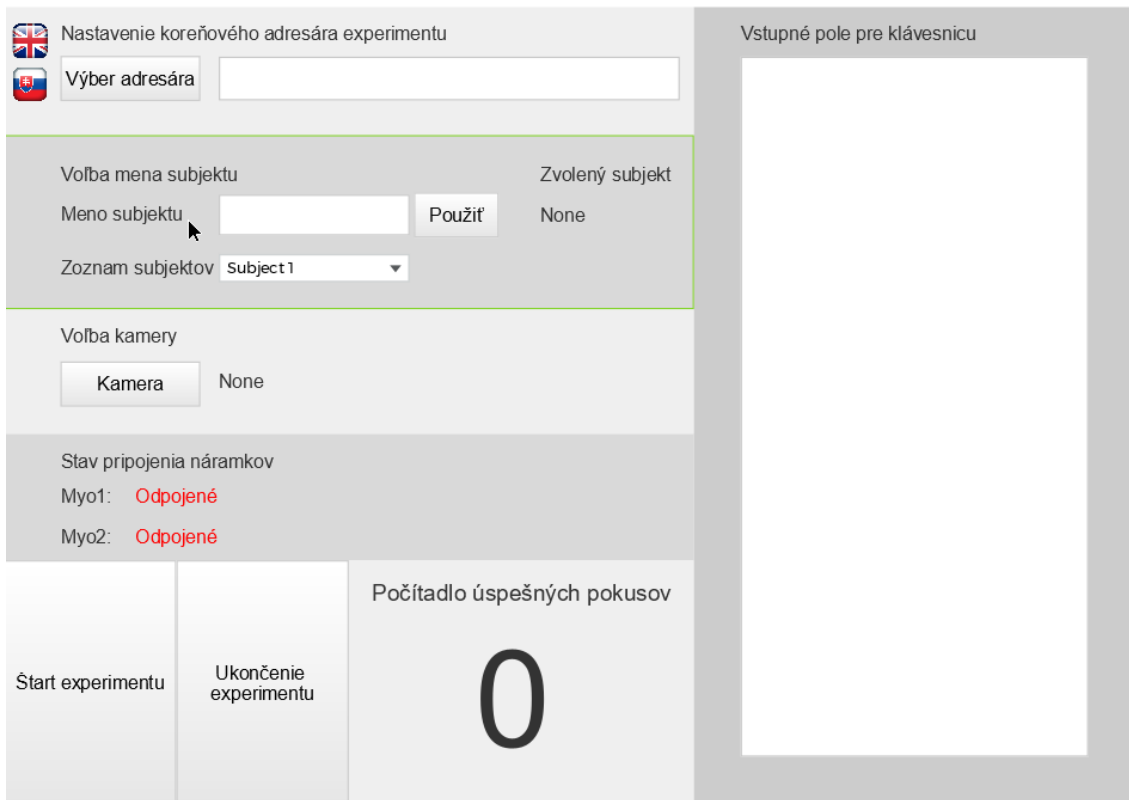


inicializačné parametre experimentu alebo samotný experiment riadi. Na pravej strane sa nachádza textové pole do ktorého vkladá znaky pomocou klávesnice, ktoré sú súčasťou spusteného experimentu.

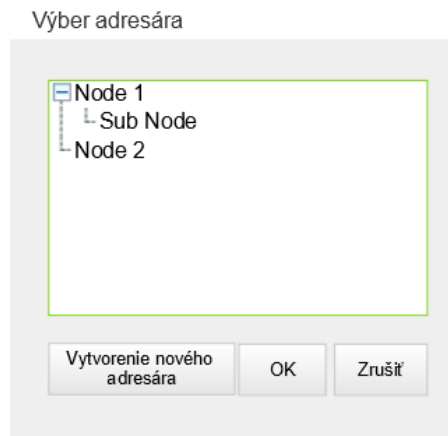
Ľavá strana hlavnej obrazovky je horizontálne rozdelená na päť sekcií. Tieto sekcie sú organizované vertikálne zhora nadol v poradí v akom má používateľ vykonávať akcie potrebné pre spustenie experimentu. V prvej sekcii sa nachádza nastavenie koreňového adresára experimentu na čo slúži tlačidlo pre výber adresára a textové pole, do ktorého je možné napísať celú cestu ku adresáru. Tak tiež sa tu nachádza možnosť voľby jazyka aplikácie. Druhá sekcia obsahuje voľbu mena subjektu. Tú je možné realizovať cez zadanie mena do textového poľa pre meno subjektu alebo pomocou výberu z listu subjektov, ktorý sa zobrazí v drop down menu. Treťou sekciou je voľba kamery. Tú vie používateľ realizovať pomocou tlačidla kamera. Po zvolení kamery je jej názov zobrazený pri tlačidle kamera. Štvrtou sekciou je Stav pripojenia náramkov. Má informatívny charakter. Používateľovi zobrazuje, či sú myo náramky pripravené na použitie v aplikácii. V piatej sekcii sa nachádzajú tlačidlá na riadenie experimentu. Sú to tlačidlá štart experimentu a ukončenie experimentu. Sú väčšie, aby ich používateľ vedel ľahko nájsť keďže riadia samotný experiment. Vedľa týchto tlačidiel sa nachádza počítadlo správne zapísaných slov požadovaných pre ukončenie experimentu. Toto počítadlo je podobne ako tlačidlo štart a ukončenie experimentu väčšie a to z dôvodu ľahšej orientácie používateľa v priebehu experimentu.

### **Obrazovka výberu adresára**

Táto obrazovka Obr. 3.7 sa zobrazí po stlačení tlačidla výber adresára na hlavnej obrazovke. Prototyp tejto obrazovky obsahuje zobrazenú stromovú štruktúru adresárov úložiska zariadenia. Pod touto stromovou štruktúrou sa nachádzajú tlačidlá. Prvé tlačidlo vytvorenie nového adresára vykoná rovnomennú akciu. Tlačidlo OK, ktoré sa nachádza na pravo od neho potvrdí zvolený adresár a zobrazí hlavnú obrazovku. Posledné tlačidlo zruší zruší výber adresára a zobrazí hlavnú obrazovku.



Obr. 3.6: Prototyp hlavnej obrazovky

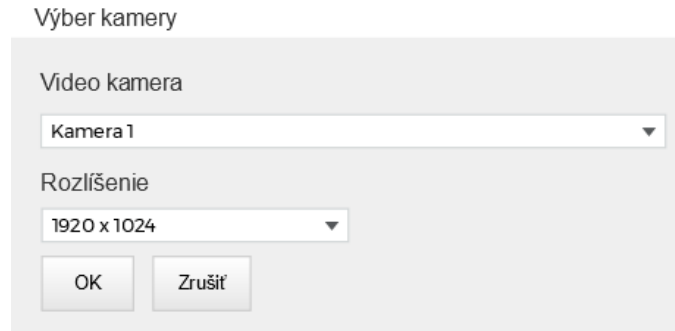


Obr. 3.7: Prototyp obrazovky výberu adresára

### Obrazovka výberu kamery

Posledným prototypom obrazovky aplikácie je obrazovka výberu kamery. Táto obrazovka sa zobrazí po stlačení tlačidla kamera na hlavnej obrazovke. Kamera sa zvolí pomocou drop down menu, ktoré obsahuje všetky kamery pripojené ku počítaču na ktorom je aplikácia spustená. Ďalšou možnosťou, ktorá je navrhnutá v prototypu tejto obrazovky je voľba rozlíšenia kamery. Táto voľba je taktiež re-

alizovaná pomocou drop down menu. V spodnej časti obrazovky sa nachádzajú tlačidlá OK a zrušiť. Oba tlačidlá zobrazia okno hlavnej obrazovky. Tlačidlo OK potvrdí voľbu kamery. Tlačidlo zrušiť zruší voľbu kamery.



The image shows a dialog box titled "Výber kamery" (Camera Selection). It contains two dropdown menus. The first is labeled "Video kamera" and has "Kamera 1" selected. The second is labeled "Rozlíšenie" (Resolution) and has "1920 x 1024" selected. At the bottom of the dialog are two buttons: "OK" and "Zrušiť" (Cancel).

Obr. 3.8: Prototyp obrazovky výberu kamery

## 4 Implementácia systému pre realizovanie experimentov

---

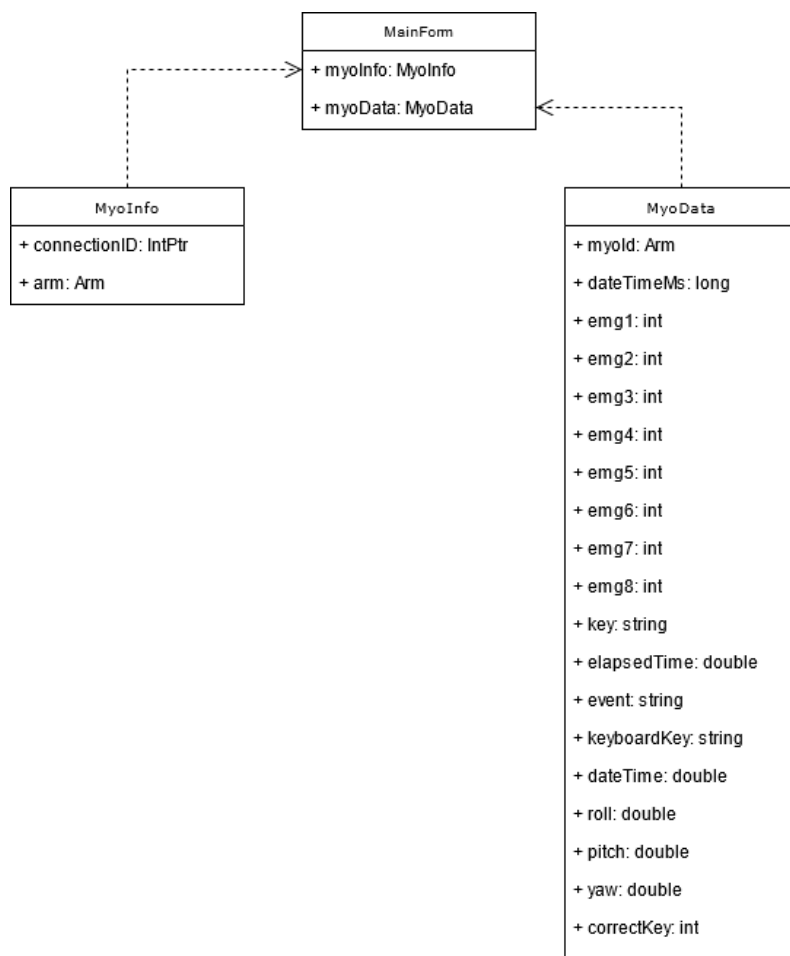
Na základe návrhu bola realizovaná implementácia aplikácie. Bol vytvorený WindowsForms projekt vo vývojovom prostredí VisualStudio 2019. V rámci tohto projektu bola vytvorená najdôležitejšia trieda, predstavujúca hlavnú obrazovku. Jej názov je **MainForm**. Ako je zobrazené na diagrame tried Obr. 4.1, aplikácia je okrem triedy **MainForm** tvorená triedou **MyoInfo** a **MainData**. Trieda **MainInfo** slúži na uchovanie informácií o pripojených náramkoch. Tieto sa menia vždy pri spustení nového experimentu a zostávajú konzistentné počas jeho vykonávania. Trieda **MyoData** slúži na spracovanie dát z Myo náramkov. Do tejto triedy sa ukladajú dáta z priebehu experimentu. Popis implementácie je rozdelený do viacerých častí, v ktorých sú opísané najdôležitejšie časti kódu.

### 4.1 Inicializácia aplikácie

Na začiatku, sú v triede **MainForm.cs** deklarované knižnice, ktoré boli analýzou vyhodnotené ako vhodné pre implementáciu tejto aplikácie. Inicializácia ďalej prebieha v konštruktore tejto triedy. Príznamky sú nastavené na východzie hodnoty a taktiež sú vytvorené objekty ako **Stopwatch**. Hneď na začiatku tohto konštruktora sa volá metóda **InitializeComponent()**, ktorá spustí udalosť **MainForm\_Load()**. Vďaka tomu sa zobrazí hlavné okno aplikácie. Počas inicializácie hlavného okna sa taktiež vytvoria objekty **BackgroundWorker**-ov pre komponenty systému tak ako je definované v návrhu tejto aplikácie.

#### 4.1.1 Inicializácia Myo Hub-u a vytvorenie komunikačných kanálov

Po úvodnej inicializácii aplikácie sa vytvorí objekt **MyoHub**. Do tohto hub-u sú priradené komunikačné kanály. Po ich vytvorení a priradení sa hubu priradia



Obr. 4.1: Diagram tried implementovanej aplikácie.

udalosti pre pripojenie a odpojenie Myo náramkov.

#### 4.1.2 Naplnenie zoznamu existujúcich subjektov

Metóda **MainForm\_Load()** obsahuje aj kód, ktorý naplní zoznam subjektov, ktoré už experiment absolvovali. Tie sa získajú zo zadaného koreňového adresára. Názvy podadresárov tohto adresára totiž obsahujú mená subjektov. Po ich uložení sa pridávajú objektu **comboBox1** ako jeho položky.

### 4.2 Pripojenie Myo náramkov

V momente pripojenia Myo náramku na **MyoHub** sa zavolá metóda s názvom **myoHub\_MyoConnected**. Jej vstupným parametrom je **MyoEventArgs** objekt. Celá táto metóda sa stará o rozpoznanie Myo náramkov a spustenie **BackgroundWorkerov**. Ak nie je spustený žiaden **BackgroundWorker**, tak sa vytvorí prvý objekt **MyoInfo**. Konštruktor tohto objektu čaká dva parametre. Prvým z para-

metrov je **e.Myo.Handle**, ktorý sa získa z objektu **MyoEventArgs** a reprezentuje identifikátor pripojeného Myo náramku. Druhým parametrom je **e.Myo.Arm**. Je získaný rovnako ako predchádzajúci parameter a predstavuje ruku na ktorej je náramok umiestnený. Vytvorí sa nové zoznamy pre uloženie dát z tohto Myo náramku. Tieto zoznamy sú typu **MyoData**. Pred spustením **BackgroundWorker** tohto náramku je vytvorený zoznam typu **object**. Do tohto zoznamu je pridaný objekt **MyoEventArgs** a taktiež zoznamy slúžiace na ukladanie dát Myo náramku. Následne je tento všeobecný zoznam použitý ako parameter pri spúšťaní **BackgroundWorker**.

Pokiaľ už existuje jeden **BackgroundWorker** pri zavolaní tejto metódy tak sa vykoná jej druhá časť, ktorá je obdobná ako prvá. Rozdiel je v tom, že sa všetky objekty vytvorí pre druhý náramok a existujú tak súčasne všetky naraz, spolu s objektami prvého náramku.

### 4.3 Vykonávanie kódu Myo BackgroundWorker-a

Vstupným parametrom tejto metódy sú objekty, ktoré boli vytvorené v predchádzajúcej časti. Na začiatku tejto metódy sú rozdelené do samostatných zoznamov a objektov. Dôvod celého tohto riešenia je, že metóda **myoBackgroundWorker\_DoWork** neprijme za parameter objekt iného typu ako **object**. Ak by ho prijala nedala by a použiť ako event na vykonávanie práce. Na konci tejto metódy sa vytvárajú eventy pre získanie dát, asíce **Myo\_EmgDataAcquired** a **Myo\_OrientationDataAcquired**, ktorých vstupné parametre tvoria zoznamy pre uloženie dát.

### 4.4 BackgroundWorker kamery a klávesnice

V oboch prípadoch sa v týchto metódach vytvorí eventy pre zaznamenávanie vstupu. V prípade klávesnice sú to eventy **textBox\_KeyPress** a **textBox\_KeyUp**. Pri kamere je to event **FinalVideo\_NewFrame**.

### 4.5 Zaznamenanie IMU dát

Dáta tohto eventu sú obsiahnuté vo vstupnom objekte **OrientationDataEventArgs** a sú spracované nasledovne:

- **roll** -  $(e.Roll + PI) / (PI * 2.0f) * 100;$

- **pitch** -  $(e.Pitch + PI) / (PI * 2.0f) * 100;$
- **yaw** -  $(e.Yaw + PI) / (PI * 2.0f) * 100;$

Tieto matematické operácie konvertujú dáta do eulerovských uhlov. PI predstavuje hodnotu konštanty pi. Objekt e, z ktorého sú získané dáta je spomínaným vstupným objektom metódy. V závere tejto metódy sú dáta uložené do zoznamu IMU dát pre príslušný Myo náramok.

## 4.6 Zaznamenanie EMG dát

Pre zaznamenanie EMG dát je použitá obdobná technika ako pri IMU dátach. Na vstupe má event **Myo\_EmgDataAcquired** objekt, ktorý obsahuje dáta. V tomto prípade je typu **EmgDataEventArgs**. Ďalším objektom na vstupe je zoznam do ktorého sa uložia dáta. Pred tým je však vytvorený objekt **MyoData**. Parametre konštruktora tohto objektu tvoria dáta z jednotlivých senzorov ako aj časová informácia o tom, kedy sa tento event spustil od začiatku nahrávania.

## 4.7 Uloženie dát na disk

Pri zadávaní vstupného reťazca sa sleduje, či je dosiahnutý potrebný počet správne zadaných textových reťazcov. Ak áno, ukončí sa získavanie a ukladanie dát do zoznamov a aplikácia prejde na uloženie týchto dát do súborov na pevný disk. Každý z typov dát má vlastnú metódu, ktorá sa stará o formátovanie súboru a o jeho uloženie na správnej ceste ako aj o jeho názov.

## 4.8 Stojan kamery

V rámci implementácie bol vytvorený aj stojan Obr. 4.2, na ktorom je pripevnená kamera. Dôvod vytvorenia tohto stojanu bola nedostupnosť profesionálneho stojanu na kameru. Kamera musí nahrávať klávesnicu pri pohľade z hora kolmo ku nej. Stojan je teda zostrojený tak aby bolo možné kameru umiestniť priamo nad klávesnicu. Je teda umiestnená na jeho vysunutú časť Obr. 4.3. Samotnú kameru je možné na stojan namontovať alebo ju odmontovať pretože sa skrutkuje do držiaka pripevneného ku stojanu. Dá sa tak odložiť napríklad pri prenose stojanu na miesto vykonávania experimentu.



Obr. 4.2: Stojan pre kameru. Pohľad spredu.





Obr. 4.3: Stojan pre kameru. Pohľad z perspektívy.

## 5 Návrh aplikácie pre dodatočné spracovanie dát

---

Po vykonaní úvodných testovacích experimentov bolo zistené, že je vhodné nad uloženými dátami vykonať dodatočné spracovanie dát. To hlavne z dôvodu, že nás zaujímajú len dáta, ktoré boli Myo náramkom zaznamenané v momente písania textového reťazca. Medzi dátami sa však nachádzali aj prestoje, kedy subjekt nezačínal reťazec, ale len presúval ruku nad klávesnicou alebo sa chystal na opakované zadávanie reťazca. Dáta taktiež obsahovali množstvo chybných pokusov, ktoré predstavujú vo výsledku falošné hodnoty pri ich analyzovaní. Bol teda vytvorený návrh riešenia dodatočného spracovania dát ako samostatnej aplikácie.

### 5.1 Vystrihnutie dát

Vhodnou metódou ako upraviť dáta do požadovaného formátu je vystrihnutie dát, ktoré boli vygenerované v momentoch, ktoré nie sú predmetom záujmu tejto práce. To ale nie je realizované na originálnych dátach. Tie je vhodné nechať v pôvodnom stave. Predstavujú unikátny záznam experimentu a je možné, že pre nich existuje využitie. Spôsob akým by bolo dáta vhodné upraviť vznikol po konzultácii s vedúcim práce a konzultantom.

Ak sa v postupnosti zadávaného požadovaného textového reťazca vyskytne čo i len jeden nesprávne zadaný znak, považuje sa tento reťazec za chybný. Každý chybný reťazec je potrebné vystrihnúť zo súboru keyboard.csv. Vystrihujú sa dáta od prvého zadaného znaku v rámci takéhoto reťazca, ktorý sa zaznamenal pri akcii "key pressed", až po jeho ukončenie po stlačení klávesy ENTER a jej uvoľnení, čo sa zaznamená v udalosti "key released".

Taktiež je potrebné na základe rezu v súbore keyboard.csv vystrihnúť dáta z EMG a IMU súborov, keďže dáta z času kedy bol zaznamenaný chybný reťazec nie sú potrebné. Vystrihnutie je v týchto súboroch vhodné realizovať tak, že za začiatok rezu sa zvolí najbližší menší záznam od pôvodnej udalosti "key pressed",

až po najbližší väčší záznam od pôvodnej udalosti "key released". To ale zabezpečí len vystrihnutie nesprávnych dát, ktoré boli zaznamenané presne počas zadávania zlého reťazca. V prípade, že bol hneď prvý zadaný reťazec v rámci pokusu nesprávny, je potrebné vystrihnúť dáta od samotného začiatku experimentu až po začiatok najbližšieho dobrého pokusu.

Ak je posledný zadaný reťazec v súbore nesprávny, vystrihnú sa všetky dáta nasledujúce za nim. Posledným tak v rámci súboru ostane najbližší správny pokus, ktorý bol vykonaný pred týmto nesprávnym pokusom.

Vhodné je taktiež orezať dáta z úvodu experimentu pokiaľ je prvý zadaný textový reťazec správny. To z dôvodu, že subjektu sa môže nástup do polohy písania zdržať a tak sa zbytočne zaznamenávajú dáta, ktoré predstavujú šum. Orezanie nastane na dátach menších ako 0.5s od od prvej správne stlačenej klávesy. Aj koniec aplikácie prejde orezaním pokiaľ je posledný pokus vykonaný správne. Taktiež sa tu realizuje rez 0.5s ale od udalosti "key released| klávesy ENTER.

V miestach rezu, kde sa dáta spájajú môže vznikáť skok medzi lokálnymi hodnotami. Preto bola navrhnutá aj metóda takzvaného "fade in/fade out". To je realizované pre násobením hodnôt v okolí rezu. Dáta priamo v bode rezu budú pre násobené hodnotu 0.25, dáta ku nim príľahlé budú pre násobené hodnotou 0.5 a dáta príľahlé ku týmto hodnotou 0.75.

## 6 Implementácia aplikácie pre dodatočné spracovanie dát

---

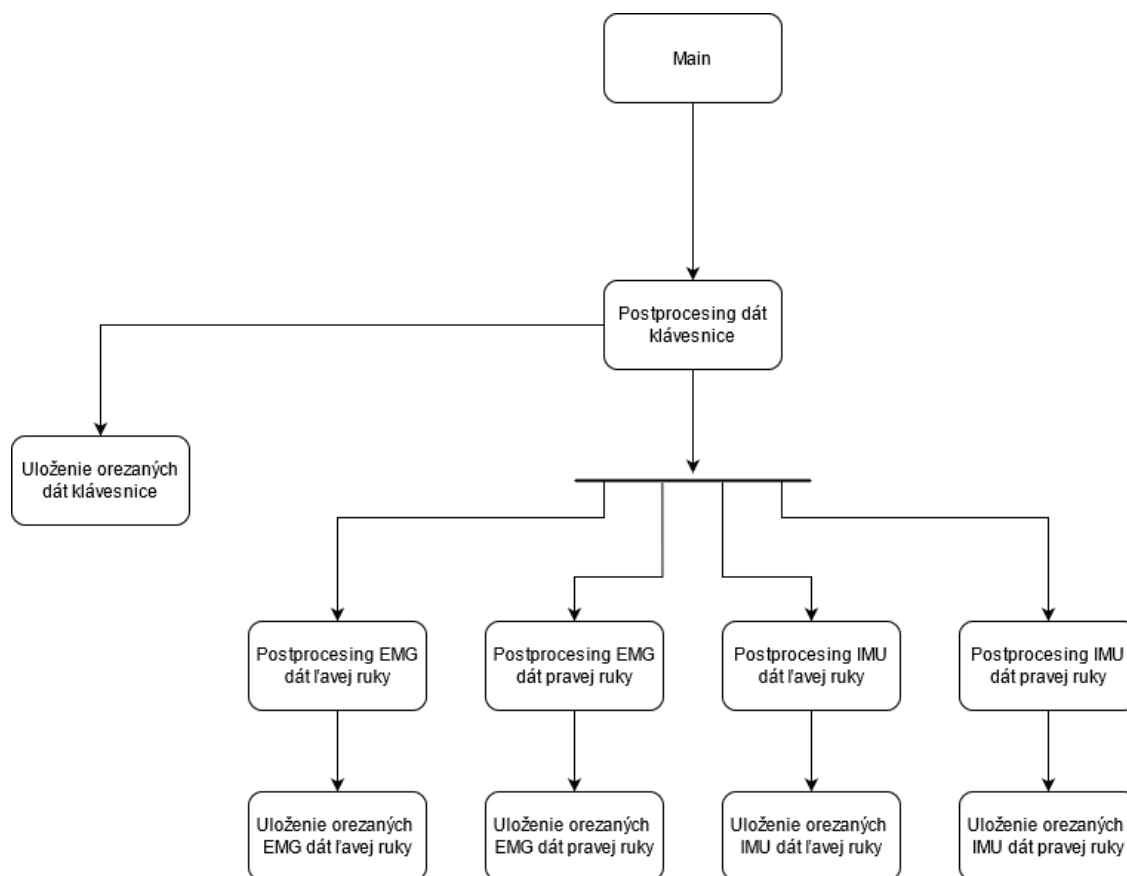
Vytvorená konzolová aplikácia, slúžiaca na dodatočné spracovanie dát, čaká vo vstupnom argumente cestu ku koreňovému adresáru súborov experimentov. Všetky .csv súbory v danom adresári spracuje a na výstupe vytvorí ich orezanú kópiu. Aplikácia nájde všetky .csv súbory, takže môžu byť uložené ľubovoľne hlboko v štruktúre koreňového adresára. Pri vytváraní upravených .csv súborov sa vytvorí aj kópia adresárovej štruktúry v ktorej je tento súbor umiestnený. Názvy nových adresárov ako aj nových súborov sú zmenené tak, že na ich konci je pridaný textový reťazec "PP", ktorý symbolizuje dodatočné spracovanie dát.

Celkovo je možné vytvorenú aplikáciu rozdeliť na 3 hlavné časti:

1. **Inicializácia** - V tejto časti sa spracuje vstupný parameter aplikácie, ktorý reprezentuje cestu ku zdrojovým súborom. Overí sa, či bol zadáný a ak áno, či existuje.
2. **Dodatočné spracovanie dát** - Skladá sa s podčastí dodatočného spracovania jednotlivých súborov. V nich sa upravujú dáta, ktoré sú neskôr na výstupe aplikácie
3. **Uloženie dát** - Táto časť sa stará o uloženie upravených dát. Vytvára štruktúry nových súborov a taktiež štruktúry adresárov kde ich uloží.

Podrobnejšie zobrazenie procedúr, ktoré aplikácia vykonáva je uvedené na obrázku Obr. 6.1. Vykonávanie sa začína procedúrou Main, ktorá predstavuje inicializáciu dát. Po jej vykonaní sa spustí procedúra dodatočného spracovania dát klávesnice. Táto procedúra sa volá ako prvá z množiny procedúr pre dodatočné spracovanie z dôvodu, že na úpravu EMG a IMU dát je potrebný výstup tejto procedúry. Po upravení dát klávesnice sa prejde na ich uloženie. Pri ich ukladaní sa vytvorí kópia pôvodnej adresárovej štruktúry a aj kópia samotného súboru **keyboard.csv**. Následne sa spustia procedúry pre dodatočné spracovanie EMG a

IMU dát získaných z oboch rúk. Tie upravujú pôvodné dáta na základe upravených dát klávesnice. Po každej vykonanej procedúre sa vykoná uloženie týchto dát pri čom sa vytvoria kópie pôvodných súborov do ktorých sa dáta uložia.



Obr. 6.1: Procedúry vykonávané po spustení aplikácie.

## 6.1 Dodatočné spracovanie dát klávesnice

Metóda `ProcessFilesInDirectory` zabezpečuje dodatočné spracovanie dát klávesnice. Na začiatku však vytvára kópiu pôvodnej adresárovej štruktúry, do ktorej sa neskôr uložia upravené dáta. Táto metóda zároveň spúšťa vykonávanie dodatočného spracovania dát nad všetkými súbormi, ktoré sú v pôvodnej adresárovej štruktúre. Dodatočné spracovanie dát klávesnice je vykonané vytvorením zoznamu nehodiacich sa dát na základe ktorého je skrútený zoznam s pôvodnými dátami. V tejto metóde sa pripravujú aj zoznamy pre dodatočné spracovanie EMG a IMU dát. V zásade sa na základe zoznamu nehodiacich sa dát vytvorí nový zoznam. V tomto zozname sa dáta upravujú tak, že sa zredukujú na počiatočné a koncové časové hodnoty nehodiacich sa dát. To predstavuje sériu dát, ktoré predstavovali chybu v zadávaní vstupného reťazca aplikácie na záznam experimentov.

Vytvára sa aj podobný zoznam pre úseky vhodných dát. Nakoniec sa zavolá metóda **PrintKeyData**, slúžiaca na uloženie dát klávesnice. Vytvára štruktúru súboru na výstupe aplikácie, do ktorej uloží orezané dáta. Za touto metódou sa zavolajú metódy pre dodatočné spracovanie EMG a IMU dát oboch rúk.

## 6.2 Dodatočné spracovanie EMG a IMU dát

Za účelom dodatočného spracovania EMG a IMU dát sú vytvorené metódy **ProcessMyoEmg** a **ProcessMyoImu**. Oba metódy na začiatku načítajú pôvodné dáta do zoznamu. Tento zoznam sa ďalej upravuje na požadovaný výsledný tvar. Na základe návrhu sa v týchto metódach kontroluje, či bol prvý zadaný znak súčasťou nesprávne napísanej sekvencie znakov. Ak je to pravda, tak sa zo zoznamu v ktorom sú uložené pôvodné dáta odstránia všetky dáta, ktorých časové označenie je menšie ako ako čas prvého znaku z dobrej sekvencie od ktorého sa ešte odčíta 0.5s. Následne sa vykoná aj podobné odstrihnutie dát z konca zoznamu. Pre lokalizovanie úsekov nepotrebných dát sa vyhladá v zozname tvorenom pôvodnými dátami časová hodnota najbližšia menšia ako časová hodnota zle zadaného znaku. Koniec tejto sekvencie dát je na časovej hodnote najbližšej väčšej hodnoty vzhľadom na časovú hodnotu posledného znaku v sekvencii označenej za sekvenciu chybne zadaných znakov. Po lokalizovaní všetkých takýchto úsekov v pôvodných dátach sa pristupuje na skrátenie listu na základe týchto dát. v miestach spojenia sa aplikuje metóda "fade in/fade out". Výsledkom sú upravené zoznamy dát, ktoré neobsahujú hodnoty počas chybne stlačených znakov. Tieto dáta sa následne ukladajú do súborov. Pri tomto procese sa vytvorí nový súbor, požadovaná štruktúra súborov a následne sa do nej dáta vložia.

## 7 Vyhodnotenie

---

Na základe implementovaných aplikácií zhotovených podľa ich návrhu je možné vykonať dôležitú časť tejto práce, ktorou je vyhodnotenie. Základ na ktorom táto časť stojí tvoria vykonané experimenty. Na týchto sa zúčastnilo 6 subjektov a boli vykonané v domácom prostredí. Ideálny počet účastníkov z pohľadu zabezpečenia čo najväčšieho počtu dát na ich ďalšiu analýzu je vyšší a ideálnym prostredím pre vykonanie experimentu je prostredie laboratória LIRKIS. Dôvodom týchto obmedzení bola nutná fyzická prítomnosť subjektov na vykonávaní experimentu. To nebolo možné kvôli štátnemu nariadeniu zákazu vychádzania a pretrvávajúcej pandémie koronavírusu. Pre vyhodnotenie úspešnosti implementácie tejto diplomovej práce je však počet účastníkov experimentu postačujúci.

### 7.1 Priebeh experimentov

Pred spustením experimentov boli každému subjektu nasadené Myo náramky tak, aby ich orientácia bola totožná s návrhom. Pokiaľ bolo potrebné boli aplikované sťahovacie svorky pre zabezpečenie čo najväčšej dotykovej plochy medzi Myo náramkom a rukou subjektu. Následne boli subjekty experimentátorom správne usadené. Experimentátor sledoval výšku usadenia. V prípade potreby stolička nastavená tak, aby ruky subjektu spĺňali predpísanú polohu vzhľadom ku klávesnici Obr. 7.1. Po správnom usadení bol subjekt inštruovaný o priebehu experimentu. V každom vykonanom experimente zadávali subjekty textový reťazec "password" pomocou klávesnice celkovo 25 krát pričom boli upozornené, že v prípade chyby sa nemá opravovať a má pokračovať ďalej až pokiaľ sa mu podarí tento reťazec zadať správne celkovo 25 krát. Zároveň bol každému subjektu vysvetlený zmysel experimentu a teda mu bolo zdôraznené aby pri zadávaní znakov používal prsty podľa prstokladu a aby sa rukami neopieral o kraj stola. Tieto podmienky kontroloval experimentátor počas vykonávania každého experimentu. V prípade, že niektorý subjekt nedodržel zadané podmienky bol experiment opakovaný. Každý subjekt zúčastnený na vykonávaní experimentu prešiel dokopy

ôsmimi iteráciami experimentov. Tie sa delili na 2 hlavné časti. Pri prvej časti bol subjekt inštruovaný experimentátorom, aby v čo najväčšej možnej miere zapájal svalstvo ruky pri stláčaní kláves. V druhej časti mu bolo vysvetlené aby používal naopak menšiu silu. Každá táto časť sa teda skladala zo 4 iterácií. Medzi jednotlivými iteráciami bol subjektom daný čas na oddych a uvoľnenie rúk.



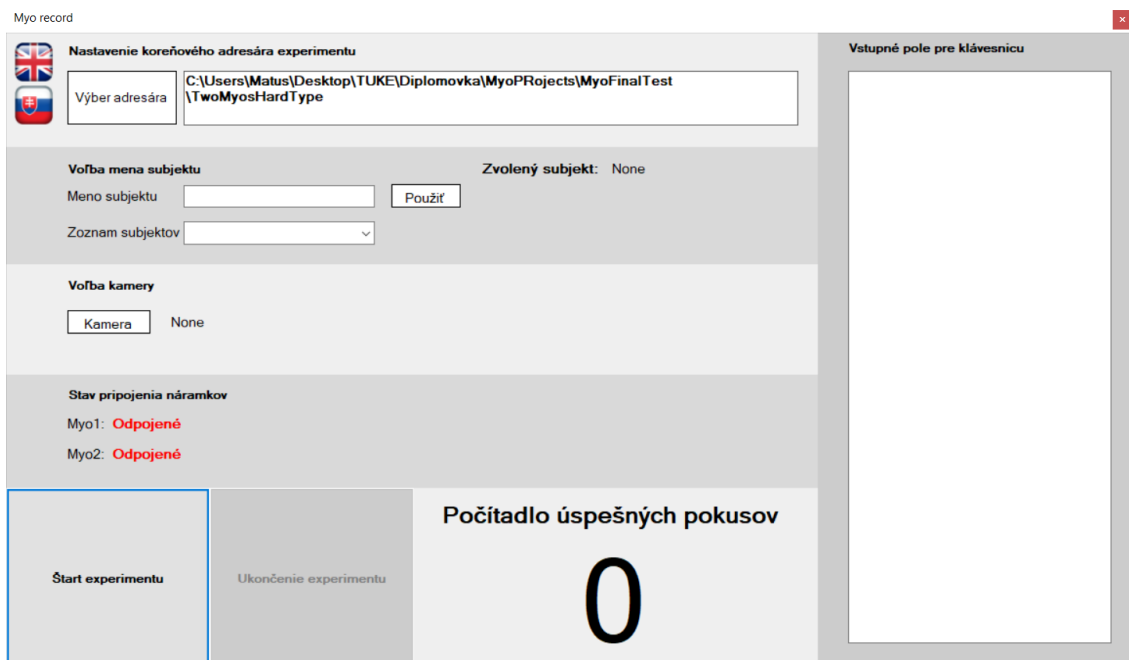
Obr. 7.1: Subjekt usadený v správnej polohe.

## 7.2 Výsledky experimentov

Počas vykonávania experimentov bolo dokopy vytvorených 288 súborov obsahujúcich pôvodné dáta experimentov. Po spustení dodatočného spracovania nad týmito dátami sú pripravené výsledne súbory vhodné na ich ďalšiu analýzu. Vý-



sledkom implementovaných návrhov sú 2 aplikácie. Prvá slúži na zaznamenanie experimentov, druhá na dodatočného spracovania dát vygenerovaných pri zaznamenaných experimentoch. Pri prvej aplikácii bolo úspešne vytvorené aj používateľské rozhranie Obr. 7.2. V rámci overenia používateľského rozhrania bol každému subjektu po ukončení experimentov položený SUS dotazník. Výsledkom tohto dotazníku je, že sa ho zúčastnili osoby vo veku od 17 do 49 rokov. Okrem jednej osoby boli všetci účastníci muži. Najlepšie hodnotenie dostala aplikácia od najmladšieho účastníka, kde dopadla so skóre 87.5. Naopak najhoršie dostala od najstaršieho účastníka. Tu aplikácia obdržala skóre 55. Celkové skóre aplikácie je 76.67, čo sa dá vyhodnotiť ako úspech. Aplikácia je dostatočne prehľadná a jednoduchá na používanie podľa väčšiny subjektov, čiže sa dá povedať, že implementované používateľské rozhranie bolo navrhnuté správne. Zaujímavým je aj výsledok NASA-TLX dotazníku. Ten bol zameraný na hodnotenie úlohy vykonania experimentu. Podľa výsledkov tohto dotazníka subjekty považovali vykonanie experimentu za fyzicky obtiažne. Na námahu, ktorú museli vynaložiť na vykonanie experimentu poukazyvali už počas jeho behu. Aj kvôli tomu bolo vhodné umožnenie oddychu subjektu medzi experimentmi. Pri masovejšom počte ľudí na experimente v jednom čase je vhodné rotovať subjekty tak, aby sa experiment neustále vykonával ale vždy s inou osobou. Vďaka tomu si bude môcť subjekt medzi experimentmi oddýchnuť a zároveň celkový čas vykonávania experimentov bude optimalizovaný tak, aby bolo možné rýchlo a efektívne zahrnúť do experimentu čo najviac ľudí. Celkové hodnotenie NASA-TLX dotazníku je 35.34, čo reprezentuje, že aj napriek väčšej fyzickej námahe je úloha vykonania experimentu ako celok jednoduchá úloha. Na základe uvedených informácií je možné konštatovať, že ciele práce boli naplnené úspešne.



Obr. 7.2: Vytvorené používateľské rozhranie.

## 8 Záver

---

Táto diplomová práca mala za cieľ navrhnúť a implementovať systém vhodný na zaznamenanie dát vygenerovaných počas zadávania znakov na klávesnici. Tieto dáta by malo byť neskôr možné v ďalších prácach analyzovať a prijať stanovisko ku tomu, aký majú EMG dáta generované EMG senzorom súvis so samotným pohybom ruky pri písaní.

V rámci analýzy je špeciálna pozornosť venovaná výberu vhodných technológií. Pre zabezpečenie efektívneho prenosu a ukladania dát boli skúmané dva programovacie jazyky, python a C#, spolu s knižnicami, ktoré sa dajú pri práci s Myo náramkom využiť. Z porovnania týchto dvoch vyšiel úspešnejšie jazyk C# a knižnica myosharp. Návrh aplikácie určenej na záznam experimentov myslí na potrebu vytvorenia prehľadného používateľského rozhrania. To je navrhnuté v niekoľkých krokoch definujúcich jeho vlastnosti. Taktiež sa rozoberá navrhnutie čo najefektívnejšieho spracovania dát. Implementácia aplikácie pre nahrávanie experimentov pojednáva o konkrétnom aplikovaní navrhovaných postupov. Na vznik potreby úpravy už nahraných dát reaguje návrh a implementácia konzolovej aplikácie určenej na post-processing dát. Návrh tejto aplikácie popisuje, akým spôsobom je vhodné spracovať dáta. Hlavným dôvodom potreby tejto podpornej aplikácie je odstránenie nerelevantných dát vygenerovaných počas vykonávania experimentu, čo zabezpečuje elimináciu vzniku možných šumov, ktoré by mohli budúce práce spôsobiť nerelevantné výsledky ich analýzy.

Vo výsledku je možné skonštatovať, že systém vytvorený za účelom efektívneho zberu dát generovaných zadávaním znakov kláves je úspešne implementovaný. Množstvo surových dát, z ktorými sa dá ďalej ľubovoľne pracovať alebo nad nimi vykonať post-processing je dostačujúce. Ich zdieľanie na analýzu je jednoduché vďaka ich uloženiu v .csv súboroch, ktoré sú zároveň prehľadnou kompaktnou štruktúrou pre ich uloženie. Výsledky z SUS dotazníkov dokazujú, že s aplikáciou sa dá jednoducho manipulovať a ovládať ju. Experimenty sú podľa NASA-TLX dotazníkov, vyplnených subjektami zúčastnenými na testovaní mierne fyzicky náročné, čo sa ale podarilo eliminovať. Zároveň však dokazujú, že úloha

vykonania experimentu nie je náročná a je dobre navrhnutá.

Tento systém je možné rozšíriť o ďalšie funkcie riadenia experimentov, ako voľba textového reťazca zadávaného v priebehu experimentu, alebo implementovania dotazníkov priamo do aplikácie. Následne by dáta z dotazníkov bolo vhodné ukladať v relačnej databáze. Taktiež samotné dáta generované počas experimentov by bolo vhodné ukladať v databáze. Ďalším rozšírením by mohlo byť aj pridanie analýzy videa na základe ktorej by bol vygenerovaný ďalší zdroj dát, ktorý by mohol byť relevantný pre celkovú analýzu dát.

# Literatúra

---

1. MAAYAN, Gilad David. *User Authentication Methods & Technologies to Prevent Breach*. 2021. Dostupné tiež z: <https://www.idrnd.ai/5-authentication-methods-that-can-prevent-the-next-breach/>.
2. CONTRIBUTOR, TechTarget. *What is multifactor authentication (MFA)? - Definition from WhatIs.com*. TechTarget, 2015. Dostupné tiež z: <https://searchsecurity.techtarget.com/definition/multifactor-authentication-MFA>.
3. OLENSKI, Julie. *Certificate-Based Authentication?* 2020. Dostupné tiež z: <https://www.globalsign.com/en/blog/what-is-certificate-based-authentication>.
4. N-ABLE. *How Does Token-Based Authentication Work?: N-able*. 2021. Dostupné tiež z: <https://www.n-able.com/blog/how-does-token-based-authentication-work>.
5. NICOLLS, Dean. *What is Biometric Authentication?* 2019. Dostupné tiež z: <https://www.jumio.com/what-is-biometric-authentication/>.
6. MITEK. *Advantages and disadvantages of biometrics*. 2021. Dostupné tiež z: <https://www.miteksystems.com/blog/advantages-and-disadvantages-of-biometrics>.
7. SHAHNEWAZ, Shaon. *Fingerprint vs vascular biometrics – What are the differences*. 2020. Dostupné tiež z: <https://www.m2sys.com/blog/important-biometric-terms-to-know/fingerprint-vs-vascular-biometrics-what-are-the-differences/>.
8. HE, Jiayuan; JIANG, Ning. *Biometric From Surface Electromyogram (sEMG): Feasibility of User Verification and Identification Based on Gesture Recognition*. *Frontiers in Bioengineering and Biotechnology*. 2020, roč. 8, s. 58. ISSN 2296-4185. Dostupné z DOI: 10.3389/fbioe.2020.00058.

9. LAKE, Stephen. *Ending Sales of Myo, Preparing for the Future*. Medium, 2018. Dostupné tiež z: <https://medium.com/@srlake/ending-sales-of-myo-preparing-for-the-future-281af9bbcac2>.
10. VISCONTI, Paolo; GAETANI, Federico; ZAPPATORE, GA; PRIMICERI, Patrizio et al. Technical features and functionalities of Myo armband: An overview on related literature and advanced applications of myoelectric armbands mainly focused on arm prostheses. *Int. J. Smart Sens. Intell. Syst.* 2018, roč. 11, č. 1, s. 1–25.
11. NYMOEN, Kristian; HAUGEN, Mari Romarheim; JENSENIUS, Alexander Refsum. Mumyo—evaluating and exploring the myo armband for musical interaction. 2015.
12. TEXAS AT AUSTIN, The University of. *Myo Gesture Control Armband*. 2021. Dostupné tiež z: <https://www.lib.utexas.edu/sites/default/files/styles/large/public/equipment/myo-armband.png?itok=d20Ex0KA>.
13. LABS, Thalmic. *TEDCAS uses the Myo armband to give surgeons gesture control*. thalmic, 2016. Dostupné tiež z: <https://medium.com/thalmic/tedcas-uses-the-myo-armband-to-give-surgeons-gesture-control-d24ce740e1ff>.
14. STERN, Becky; lady ada lady; NOSONOWITZ, Danny. 2016. Dostupné tiež z: <https://learn.adafruit.com/myo-armband-teardown>.
15. VOGELZANG, KH. *The tennis trainer: development of a Myo armband application*. 2017. B.S. thesis. University of Twente.
16. GUÉRIT, Robin; CIERRO, Alessandro; VANDERDONCKT, Jean; PEREZ MEDINA, Jorge Luis. Gesture Elicitation and Usability Testing for an Armband Interacting with Netflix and Spotify: Helping Teachers Develop Research Informed Practice. In: 2019, s. 625–637. ISBN 978-981-13-3009-4. Dostupné z DOI: 10.1007/978-3-030-11890-7\_60.
17. ABDUO, Mahmoud; GALSTER, Matthias. Myo gesture control armband for medical applications. 2015.
18. RAHIM, Md Abdur; SHIN, Jungpil. Hand movement activity-based character input system on a virtual keyboard. *Electronics*. 2020, roč. 9, č. 5, s. 774.
19. NAIR, Anupama. *Kai Gesture Controller: "The force" can now be with you!* Vicara, 2019. Dostupné tiež z: <https://medium.com/vicara/kai-gesture-controller-the-force-can-now-be-with-you-73a86f0301a4>.

20. NAIR, Anupama. *Kai Gesture Controller: "The force" can now be with you!* Vi-  
cara, 2019. Dostupné tiež z: [https://miro.medium.com/max/875/1\\*HQChvec-yU-I7Hp0xagzWA.jpeg](https://miro.medium.com/max/875/1*HQChvec-yU-I7Hp0xagzWA.jpeg).
21. SHIMMER. *Shimmer3 EMG Unit*. 2021. Dostupné tiež z: <https://www.shimmersensing.com/products/shimmer3-emg-sensor>.
22. MEDICALEXPO. 2021. Dostupné tiež z: [https://img.medicaexpo.com/images\\_me/photo-mg/107788-10137504.webp](https://img.medicaexpo.com/images_me/photo-mg/107788-10137504.webp).
23. MICROSOFT. *Visual Studio 2019*. 2020. Dostupné tiež z: <https://visualstudio.microsoft.com/cs/vs/>.
24. BILLWAGNER, Microsoft. *C# docs - get started, tutorials, reference*. 2021. Dostupné tiež z: <https://docs.microsoft.com/en-us/dotnet/csharp/>.
25. BERNHARDT, Paul. *#MyoCraft How To Get Started for #thalmichackd!* The Lab, 2015. Dostupné tiež z: <https://developerblog.myo.com/myocraft-hack-with-us-for-thalmichackd/>.
26. MICROSOFT. *Windows Forms for .NET 5 documentation*. 2021. Dostupné tiež z: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-5.0>.
27. COSENTINO, Nick; UZUN, Tayfun. *tayfuzun/MyoSharp*. 2017. Dostupné tiež z: <https://github.com/tayfuzun/MyoSharp>.
28. JETBRAINS. *PyCharm: the Python IDE for Professional Developers by JetBrains*. 2021. Dostupné tiež z: <https://www.jetbrains.com/pycharm/>.
29. FOUNDATION, Python Software. *Welcome to Python.org*. 2021. Dostupné tiež z: <https://www.python.org/>.
30. ROSENSTEIN, Niklas. *myo-python*. 2021. Dostupné tiež z: <https://pypi.org/project/myo-python/>.
31. AFORGE.NET. *Framework*. 2012. Dostupné tiež z: <http://www.aforgenet.com/framework/>.
32. LOW, Brian. *Capture Events*. 2002. Dostupné tiež z: <https://documentation.help/DirectX.Capture/DirectX.Capture.CaptureEvents.html>.
33. TEAM, OpenCV. *Platforms*. 2021. Dostupné tiež z: <https://opencv.org/platforms/>.
34. JOSHI, Muktak. 2019. Dostupné tiež z: <https://muktak.com/parallel-processing-in-c-using-threads-threadpool-and-tasks/>.

- 
35. ARTEMIOU, Artemakis. MSSQLTips, 2019. Dostupné tiež z: <https://www.mssqltips.com/sqlservertip/6024/compare-net-thread-class-threadpool-class-backgroundworker-class-and-task-parallel-library/>.
  36. BILLWAGNER. *Language-Integrated Query (LINQ) (C#)*. Microsoft, [b. r.]. Dostupné tiež z: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>.
  37. VICKERS, James. *Is Using LINQ in C# Bad for Performance?* The Startup, 2020. Dostupné tiež z: <https://medium.com/swlh/is-using-linq-in-c-bad-for-performance-318a1e71a732>.
  38. VICKERS, James. 2020. Dostupné tiež z: [https://miro.medium.com/max/1000/1\\*uSc7IAso299RGpfdlgRczA.png](https://miro.medium.com/max/1000/1*uSc7IAso299RGpfdlgRczA.png).
  39. ELBRUNO. *#MYO – Getting started, #Windows10 configuration*. 2016. Dostupné tiež z: <https://elbruno.com/2016/07/13/myo-getting-started-windows10-configuration/>.



# Zoznam príloh

---

**Príloha A** Používateľská príručka

**Príloha B** Systémová príručka

**Príloha C** CD médium – záverečná práca v elektronickej podobe

# A Používateľská príručka

---

Táto časť opisuje prácu s vytvorenými aplikáciami. Delí sa na 2 hlavné časti. V prvej sa nachádzajú inštrukcie pre vykonávanie experimentov a obsluhu používateľského rozhrania. Druhá časť obsahuje návod na používanie aplikácie pre dodatočné spracovanie dát.

## A.1 Realizácia experimentov

Pre realizáciu experimentov je v prvom rade potrebné zabezpečiť nasledujúce hardvérové vybavenie:

1. Myo náramky v počte 2 kusov.
2. Myo Bluetooth adaptér.
3. Klávesnica, najlepšie jedna pre všetky experimenty.
4. Kamera.
5. Stojan pre kameru.
6. Osobný počítač.

Softvérové vybavenie potrebné pre realizáciu experimentov:

1. Windows 10.
2. Myo Connect.
3. Aplikácia Myo record, vytvorená v tejto práci.

### A.1.1 Inštalácia Myo Connect

Aplikáciu Myo Connect je možné nainštalovať nasledovaním krokov v odkaze [39]. Odkaz popisuje aj inicializačnú kalibráciu Myo náramka.

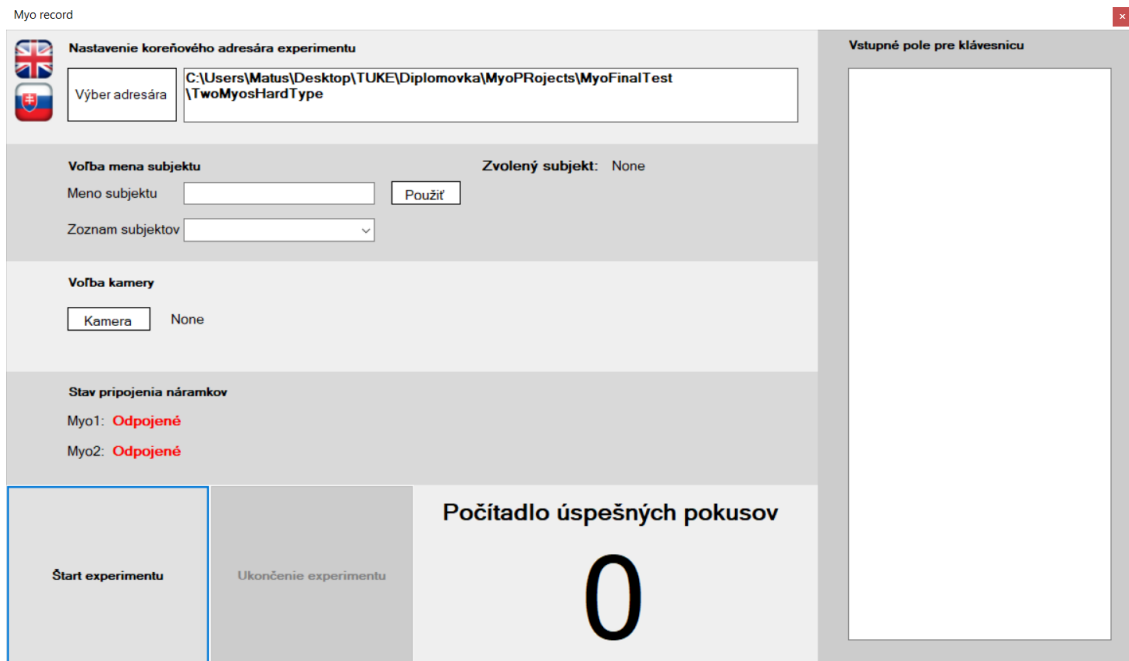
## A.2 Predpríprava experimentu

Pred začatím vykonávania každého experimentu je potrebné zabezpečiť rovnaké podmienky. Je nutné nabiť oba Myo náramky. Po ich nabití je potrebné ich pripojiť do aplikácie Myo Connect, ako je popísané v predchádzajúcej sekcii. Klávesnica sa po zapojení do počítača umiestni na okraj stola. Kamera sa namontuje na stojan, ktorý sa umiestni za klávesnicu tak, aby mala kamera dobrý záber na klávesnicu. Kamera sa zapojí do počítača. Pri každom subjekte je potrebné správne nasadenie Myo náramkov. V prípade potreby sa subjektu nastaví individuálna výška sedenia tak, aby nemal ruky opreté o okraj stola. Spustí sa aplikácia Myo record.

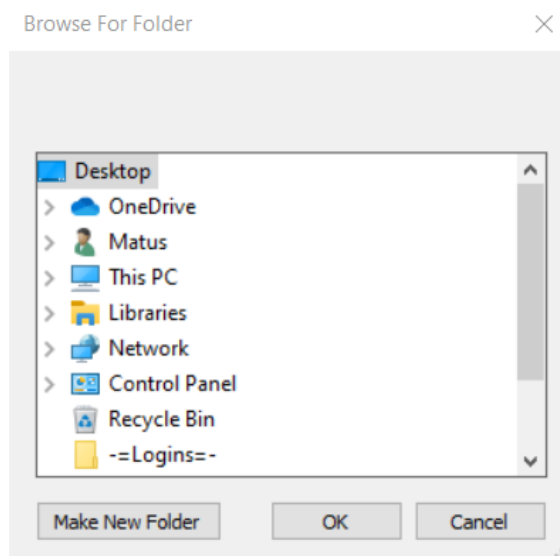
### A.2.1 Inicializácia experimentu

Po štarte aplikácie Myo record Obr. A.1 sa zobrazí hlavná obrazovka, na ktorej je potrebné vykonať nasledujúce kroky:

1. V prvom kroku sa v prípade potreby nastaví jazyk. Po stlačení tlačidla britskej vlajky sa nastaví angličtina. Po stlačení tlačidla slovenskej vlajky sa nastaví slovenčina.
2. Stlačenie tlačidla **Výber adresára** zobrazí kontextové okno Obr. A.2.
3. V kontextovom okne je možné vybrať alebo vytvoriť adresár, do ktorého sa experiment zaznamená.
4. Po stlačení tlačidla **OK** sa akcia potvrdí a okno sa kontextové zavrie.
5. Do textového okna **Meno subjektu** sa zadá meno subjektu. Pokiaľ sa subjekt zúčastnil experimentu v minulosti, je možné vybrať jeho meno zo zoznamu. Potvrdenie tohto mena sa vykoná stlačením tlačidla **Použiť**. Meno sa zobrazí pri sekcii **Zvolený subjekt**.
6. Výber kamery sa realizuje po stlačení tlačidla **Kamera** v kontextovom okne výberu kamery Obr. A.3.
7. Je potrebné vybrať kameru podľa názvu a nastaviť jej rozlíšenie. Potvrdením výberu sa kontextové okno zavrie.



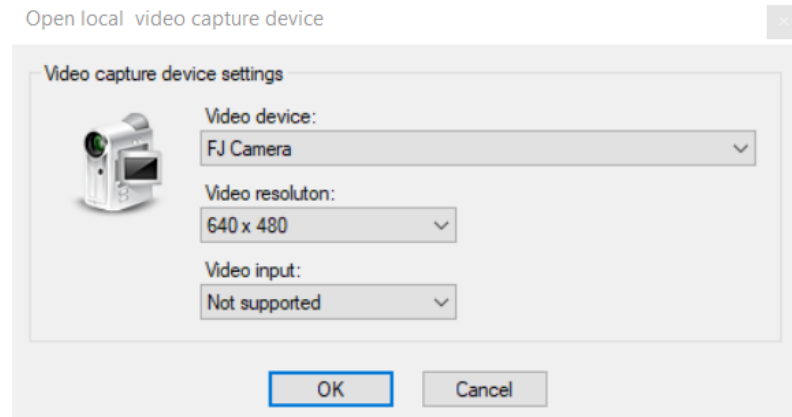
Obr. A.1: Hlavná obrazovka.



Obr. A.2: Obrazovka výberu adresára.

## A.2.2 Riadenie experimentu

Po zadaní inicializačných parametrov je potrebné kalibrovať Myo náramky na rukách subjektu vykonaním kalibračného gesta. Po skontrolovaní stavu pripojenia náramkov je možné spustiť experiment. Subjekt v ňom musí 25 krát napísať slovo **password** do textového okna **Vstupné pole pre klávesnicu**. Informácia o počte slov, ktoré zadal správne je v sekcii **Počítadlo úspešných pokusov**. Po zadaní posledného správneho slova sa experiment ukončí.



Obr. A.3: Obrazovka výberu kamery.

### A.3 Dodatočné spracovanie dát

Táto aplikácia nevyžaduje zložitú obsluhu. Pre vykonanie dodatočného spracovania dát je potrebné spustiť **CsvPostProcessing.exe** a ako argument použiť celú cestu ku dátam. Na úrovni zadaného adresára vytvorí nový adresár s rovnakým názvom a pridá mu koncovku **PP**. V tomto adresári sa nachádzajú upravené dáta.

## B Systémová príručka

---

Táto časť sa venuje popisu vytvorených aplikácií po systémovej stránke. Aplikáciu pre vykonanie experimentov a aplikáciu pre dodatočné spracovanie dát popisuje v separátnych sekciách.

### B.1 Aplikácia pre realizovanie experimentov

Na vytvorenie projektu bolo použité vývojové prostredie Visual Studio 2019. Pri implementovaní kódu sa využíva knižnica MyoSharp pre prácu s Myo náramkami a knižnica AForge pre prácu s videom.

#### B.1.1 Triedy

V rámci tejto aplikácie bolo vytvorených viacero tried, ktoré sú popísané nižšie.

##### MainForm

Táto trieda je hlavnou triedou. Vykonáva sa v nej inicializovanie parametrov. Vytvárajú sa objekty použité v aplikácii. Má bezparametrický konštruktor.

##### MyoInfo

Táto trieda sa používa na vytvorenie štruktúry pre informácie o Myo náramkoch. Jej konštruktor má 2 parametre. Prvým je **connectionID** typu **IntPtr**. Druhý je **arm** typu **Arm**

##### MyoData

Úlohou tejto triedy je vytvorenie štruktúry pre dáta z Myo náramkov a klávesnice. Jej konštruktor tvorí množina objektov z Myo náramkov, znaky zadané klávesnicou, čas zaznamenania dát a označenie správnosti dát.

## B.1.2 Metódy

Popis metód vytvorených v rámci aplikácie.

### **Form1\_Load**

Zavolá sa pri inicializovaní aplikácie. Vytvára niektoré základné objekty na otvorenie kontextových okien a prehľadávania adresárov.

### **Form1\_FormClosing**

Stará sa o zastavenie komunikácie s Myo náramkom. Zároveň ukončuje aplikáciu.

### **InitializeMyo**

V tejto metóde sa inicializuje Myo Hub a komunikácia s náramkami.

### **StopMyo**

Zastavenie komunikácie s Myo náramkami.

### **myoHub\_MyoConnected**

Metóda vytvára BackgroundWorker pre Myo náramki. Pripravuje dáta na posunutie do BackgroundWorker-ov.

### **myoBackgroundWorker1\_DoWork**

Vytvára udalosti pre prvý Myo náramok.

### **myoBackgroundWorker2\_DoWork**

Vytvára udalosti pre druhý Myo náramok.

### **keyboardBackgroundWorker\_DoWork**

Vytvára udalosti pre klávesnicu.

### **CameraBackgroundWorker\_DoWork**

Vytvára udalosti pre zaznamenávanie videa.

### **myoHub\_MyoDisconnected**

Metóda rieši odpojenie náramkovz Myo Hub-u.

### **Myo\_OrientationDataAcquired**

Spracovanie dát z Myo náramkov.

### **SortByTime**

Zoradenie zoznamu, ktorý je vo vstupnom parametri a jeho vrátenie.

### **SetText**

Nastavenie textov pripojenia.

### **EndExperiment**

Metóda ukončuje experiment. Dáta zo zoznamov zapisuje do dokumentov.

### **StartExperiment**

Spustenie získavania dát a zapisovania údajov do zoznamov.

### **WriteToFileCsv**

Volá metódy pre zápis do .csv súborov.

### **PrintEmgData**

Zapisuje EMG dáta do súborov. Vytvára štruktúru pre EMG dáta.

### **PrintImuData**

Zapisuje IMU dáta do súborov. Vytvára štruktúru pre IMU dáta.

### **PrintKeyData**

Zapisuje dáta klávesnice do súborov. Vytvára štruktúru pre tieto dáta.

### **textBox4\_KeyUp**

Zaznamenanie uvoľnenia klávesy a vytvorenie objektu v zozname s dátami pre klávesnicu.

### **textBox4\_KeyPress**

Zaznamenanie stlačenia klávesy a vytvorenie objektu v zozname s dátami pre klávesnicu.



### **cameraButton\_Click**

Vytvorenie BackgroundWorker-a pre kameru.

### **SvkButton\_Click**

Nastavenie textu rozhrania na slovenčinu.

### **EngButton\_Click**

Nastavenie textu rozhrania na angličtinu.

## **B.2 Aplikácia pre dodatočné spracovanie dát**

Na vytvorenie projektu bolo použité vývojové prostredie Visual Studio 2019. Projekt je konzolová aplikácia.

### **B.2.1 Triedy**

V rámci tejto aplikácie bolo vytvorených viacero tried, ktoré sú popísané nižšie.

#### **Program**

Hlavná trieda aplikácie. Obsahuje všetky ostatné triedy a metódy.

#### **MyoEmgData**

Definuje štruktúru EMG dát v zoznamoch.

#### **MyoImuData**

Definuje štruktúru IMU dát v zoznamoch.

#### **MyoKeyData**

Definuje štruktúru dát klávesnice v zoznamoch.

#### **BadData**

Predstavuje štruktúru pre zoznamy, v ktorých sú umiestnené dobré dáta.

### **B.2.2 Metódy**

Popis metód vytvorených v rámci aplikácie.

### **Main**

Hlavná metóda programu. Spúšťa jeho vykonávanie.

### **ProcessFilesInDirectory**

Spúšťa spracovanie dát. Stará sa o vytvorenie zoznamu zlých dát.

### **ProcessMyoImu**

Spracuje IMU dáta na výslednú podobu.

### **ProcessMyoEmg**

Spracuje EMG dáta na výslednú podobu.

### **PrintKeyData**

Metóda vytvára súbor pre dáta klávesnice a zapisuje do tohto súboru dáta zo zoznamu pre klávesnicu.

### **PrintEmgData**

Metóda vytvára súbor pre EMG dáta a zapisuje do tohto súboru dáta zo zoznamu pre EMG dáta.

### **PrintMyoImu**

Metóda vytvára súbor pre IMU dáta a zapisuje do tohto súboru dáta zo zoznamu pre IMU dáta.