

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Student:** Alena Kufnerová  
**Studijní program:** Kybernetika a robotika (bakalářský)  
**Obor:** Robotika  
**Název tématu:** Rehabilitace jemné motoriky rukou

### Pokyny pro vypracování:

1. Seznamte se, zejména po technické stránce, s možnostmi rehabilitace jemné motoriky rukou (horních končetin). Vyberte některé rehabilitační postupy / metody, které lze vhodně převést na počítačové programy doplněné vhodně vytvořeným a jednoduchým hardwarem.
2. Na základě získaných informací navrhnete programovou a hardwarovou podporu pro tento typ rehabilitace. Zaměřte se nejen na možnost rehabilitace v ordinaci lékaře, ale hlavně na přístupnost této rehabilitace i v domácím prostředí.
3. Vytvořte prototyp navrženého hardwaru pro ověření jeho činnosti a rovněž vytvořte minimálně dvě ukázkové programové aplikace pro využití sestaveného hardwaru pro vlastní rehabilitaci.
4. Nedílnou součástí práce musí být vhodný manuál pro budoucí uživatele v podobě přehledné a hlavně obrázkové dokumentace pro použití.

### Seznam odborné literatury:

- [1] Trojan S., Druga R., Pfeiffer J., Votava J., Fyziologie a léčebná rehabilitace motoriky člověka, Grada, 2005.
- [2] WWW stránky rehabilitačních organizací (Cerebrum, JUS, Arpida,...).
- [3] WWW stránky výrobců (Sanomed, Setrans, NetMedik, Asker, ...).
- [4] Matthew MacDonald, Pro WPF in C# 2010, Apress, 2013.
- [5] Nagel Ch., Evjen B., Glynn J., Watson K., Skinner M.: Professional C# 4 and .NET 4, Wrox, 2010.

**Vedoucí bakalářské práce:** Ing. Petr Novák, Ph.D.

**Platnost zadání:** do konce letního semestru 2014/2015

L.S.

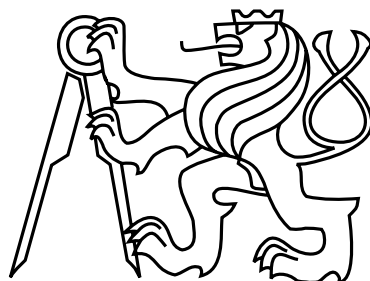
doc. Dr. Ing. Jan Kybic  
**vedoucí katedry**

prof. Ing. Pavel Ripka, CSc.  
**děkan**

V Praze dne 16. 12. 2013



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra kybernetiky



Bakalářská práce

## Rehabilitace jemné motoriky rukou

*Alena Kufnerová*

Vedoucí práce: Ing. Petr Novák Ph.D

Studijní program: Kybernetika a robotika

Obor: Robotika

2014



## Poděkování

Ráda bych poděkovala Ing. Petru Novákovi Ph.D za vedení mé bakalářské práce, cenné rady a velkou ochotu při konzultování práce. Mé poděkování patří též mé rodině a příteli za obrovskou podporu a trpělivost.



## Prohlášení

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

V Praze dne 23. 5. 2014

.....





# Abstract

The bachelor thesis deals with the design and production of a prototype device for fine motor skills rehabilitation, particularly those of hands, and several exercises related to it. The rehabilitation method proposal focuses on low cost, easy installation and controllability of the device. Although the device is primarily intended for home use, it may certainly be used also at a doctor's office. Motion is detected by a triaxial accelerometer. Rehabilitation applications are made in the form of fun games, but there are also some basic diagnostic exercises. The main aim of this thesis is to design an affordable prototype device for a rehabilitation method of fine hand motor skills, intended primarily for household environment.

**Keywords:** fine motor skills, stroke, complicated fractures, ergotherapy

**Title translation:** Rehabilitation of fine hand motor skills

# Abstrakt

Bakalářská práce se věnuje návrhu a tvorbě prototypu zařízení a několika úloh pro rehabilitaci jemné motoriky zejména rukou. Návrh celé metody rehabilitace klade důraz na nízkou cenu, snadnou instalaci a ovladatelnost zařízení. Přestože je zařízení určeno především pro použití v domácím prostředí, tak může být samozřejmě využito rovněž i v ordinaci lékaře. K detekci pohybu se využívá tříosý akcelerometr. Rehabilitační aplikace jsou tvořeny formou zábavných her, ale nechybí však ani základní diagnostická úloha. Výsledkem této práce je tedy prototyp cenově dostupného zařízení pro metodu rehabilitace jemné motoriky rukou určeného především do domácího prostředí.

**Klíčová slova:** jemná motorika, mozkové příhody, komplikované zlomeniny, ergoterapie



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Využívaná zařízení / nástroje pro rehabilitaci . . . . .	1
1.1.1	Základní výhody a nevýhody . . . . .	2
1.1.2	Dostupnost zařízení pro možnost rehabilitace v domácím prostředí . . . . .	3
1.2	Návrh přípravku pro rehabilitace i v domácím prostředí . . . . .	3
1.3	Možnost využití platformy Arduino . . . . .	4
1.3.1	Historie . . . . .	4
1.3.2	Hardware . . . . .	5
1.3.3	Software . . . . .	5
1.3.4	Arduino Uno . . . . .	6
1.4	Využití akcelerometru pro detekci pohybu . . . . .	7
1.4.1	Piezorezistorové akcelerometry . . . . .	8
1.4.2	Piezoelektrické akcelerometry . . . . .	8
1.4.3	Tepelné akcelerometry . . . . .	8
1.4.4	Kapacitní akcelerometry . . . . .	9
<b>2</b>	<b>Záměr této práce</b>	<b>11</b>
<b>3</b>	<b>Návrh hardwarové části zařízení</b>	<b>13</b>
3.1	Hlavní části prototypu zařízení . . . . .	13
3.1.1	Implementace funkce <code>setup()</code> a <code>loop()</code> . . . . .	15
3.2	Způsob signalizace stavu zařízení . . . . .	15
<b>4</b>	<b>Senzor použitý v prototypu zařízení</b>	<b>17</b>
4.1	Akcelerometr MMA7455 . . . . .	17
4.2	Měření statické převodní charakteristiky . . . . .	19
<b>5</b>	<b>Zprávy předávané mezi zařízením a aplikací (PC)</b>	<b>21</b>
5.1	Tvar zpráv . . . . .	21
5.2	Zpracování zpráv v zařízení . . . . .	22
5.3	Testování přenosové rychlosti dat . . . . .	22
<b>6</b>	<b>Zpracování naměřených hodnot v aplikaci</b>	<b>25</b>
6.1	Průměrování naměřených hodnot . . . . .	27
6.2	Možné orientace použití měřícího zařízení . . . . .	27
6.3	Stanovení skutečného úhlu náklonu a určení platnosti rozsahu . . . . .	28

6.4	Předávání hodnot rehabilitačním úlohám . . . . .	28
<b>7</b>	<b>Implementace programové aplikace</b>	<b>29</b>
7.1	Použité technologie . . . . .	29
7.2	Využití os rotací v aplikaci / úlohách . . . . .	29
7.3	Framework pro tvorbu úloh / aplikací . . . . .	30
7.4	Vytvářené rehabilitační úlohy . . . . .	31
7.4.1	Úloha Stabilita . . . . .	32
7.4.2	Úloha Chytání . . . . .	35
<b>8</b>	<b>Popis použití prototypu pro rehabilitaci</b>	<b>39</b>
8.1	Části zařízení . . . . .	39
8.2	Požadavky na operační systém osobního počítače . . . . .	39
8.3	Zapnutí/vypnutí zařízení . . . . .	40
8.4	Významy světelných signálů kontrolky na zařízení . . . . .	40
8.5	Základní konfigurace aplikace . . . . .	40
8.6	Orientace zařízení . . . . .	42
8.7	Typy průměrování . . . . .	42
8.8	Dostupné rehabilitační úlohy . . . . .	43
8.8.1	Základní diagnostické úlohy . . . . .	44
8.8.1.1	Úloha Test . . . . .	44
8.8.1.2	Úloha Stabilita . . . . .	45
8.8.2	Úlohy pro rehabilitaci pohybu / otáčení okolo jedné osy . . . . .	45
8.8.2.1	Úloha Tenis . . . . .	45
8.8.2.2	Úloha Chytání . . . . .	46
8.8.3	Úlohy pro rehabilitaci otáčení okolo dvou os . . . . .	47
8.8.3.1	Úloha Plocha s kuličkou . . . . .	47
8.8.3.2	Úloha Bludiště . . . . .	48
8.8.4	Speciální úlohy . . . . .	48
8.8.4.1	Úloha Hrneček . . . . .	48
<b>9</b>	<b>Poznatky z realizace a případná vylepšení</b>	<b>49</b>
<b>10</b>	<b>Závěr</b>	<b>51</b>
<b>A</b>	<b>Obsah přiloženého CD</b>	<b>55</b>

# Seznam obrázků

1.1	Rukavice Hand Tutor od firmy Medi Touch. . . . .	2
1.2	Zařízení Libra pro rehabilitaci stability. . . . .	2
1.3	Arduino Uno (přední strana, zadní strana). . . . .	5
1.4	Arduino IDE (s předpřipraveným zdrojovým kódem nástroje Arduino IDE - blikání LED diodou). . . . .	6
1.5	Struktura piezorezistorového akcelerometru. . . . .	8
1.6	Struktura piezoelektrického akcelerometru. . . . .	8
1.7	Řez ohřívací částí a struktura tepelného akcelerometru. . . . .	9
1.8	Struktura kapacitního akcelerometru. . . . .	9
3.1	Elektrické schéma prvního prototypu zařízení. . . . .	14
3.2	První prototyp zařízení pro rehabilitaci jemné motoriky zejména rukou sestavený na tzv. nepájivém poli. . . . .	15
4.1	Akcelerometr MMA7455 (modul s akcelerometrem, piny akcelerometru, I <sup>2</sup> C připojení k mikrokontroléru). . . . .	17
4.2	Hodnoty poskytující akcelerometr MMA7455 ve směru jednotlivých os pro daná otočení senzoru. . . . .	18
4.3	Konstrukce pro měření závislosti naměřené hodnoty akcelerometrem na úhlu otočení okolo dané osy akcelerometru (zepředu, rotace okolo osy X, rotace okolo osy Y). . . . .	19
4.4	Závislost náklonu akcelerometru MMA7455 na naměřené hodnotě tímto akcelerometrem. . . . .	20
5.1	Grafické znázornění principu zpracování dat z aplikace či akcelerometru. . . . .	23
6.1	Implementace třídy „PohyboMetrBase“. . . . .	26
7.1	Osy rotací v rámci rehabilitačních úloh aplikace. . . . .	30
7.2	Přehled tlačítek v hlavním menu aplikace. . . . .	31
7.3	Ukázka (pseudokód) z úlohy „Stabilita“ - určení rozměrů čárek na číselníku, jejich tvorba (tvorba objektu typu „Line“) a umístění na plátno. . . . .	33
7.4	Implementace metody pro příjem a zpracování hodnot ze zařízení a nastavení šipek na daný úhel. . . . .	34
7.5	Ukázka z úlohy „Stabilita“ (na obrázku vpravo je v rámci ukázky na číselníku modře zvýrazněn maximální měřitelný náklon zařízení). . . . .	34

7.6	Ukázka (pseudokód) z úlohy „Chytání“ - testování kolize padajících objektů s předmětem ovládaným pacientem. . . . .	36
7.7	Implementace metody pro příjem a zpracování hodnot ze zařízení a výpočtu x-ové souřadnice pozice ovládaného objektu. . . . .	36
7.8	Ukázka z úlohy „Chytání“. . . . .	37
8.1	Hlavní menu rehabilitační aplikace (zvýrazněné tlačítko pro nastavení). . . . .	41
8.2	Dialog pro základní konfiguraci aplikace. . . . .	42
8.3	Hlavní menu rehabilitační aplikace (zvýrazněná tlačítka různých kategorií úloh). . . . .	44
8.4	Úloha „Test“. . . . .	44
8.5	Úloha „Stabilita“ (na obrázku vpravo na číselném ciferníku modře zvýrazněn maximální měřitelný náklon zařízení). . . . .	45
8.6	Úloha „Tenis“ (vlevo nižší obtížnost, vpravo vyšší obtížnost hry). . . . .	46
8.7	Úloha „Chytání“. . . . .	46
8.8	Úloha „Plocha“ (vlevo režim „Pohyb postavou“ na nižší obtížnost, vpravo režim „Pohyb deskou“ na vyšší obtížnost hry). . . . .	47
8.9	Úloha „Bludiště“ (vlevo režim „Pohyb postavou“ na vyšší obtížnost, vpravo režim „Pohyb deskou“ na nižší obtížnost hry). . . . .	48
8.10	Úloha „Hrneček“. . . . .	48
A.1	Obsah příloženého CD. . . . .	55

# Seznam tabulek

3.1	Významy jednotlivých signálů světelných kontrol na zařízení. . . . .	16
5.1	Přehled příkazových zpráv posílaných z aplikace do zařízení. . . . .	22
5.2	Přehled datových zpráv posílaných ze zařízení do aplikace. . . . .	22
5.3	Výsledek měření maximálního počtu zpráv (obsahující vzorky naměřené akcelerometrem) přijatých aplikací za vteřinu. . . . .	24
8.1	Významy světelných signálů jednotlivých kontrol na zařízení. . . . .	40





# Kapitola 1

## Úvod

Rehabilitace jemné motoriky zejména rukou je cvičení pro dosažení co možná nejvyššího stupně tzv. funkční schopnosti rukou, neboli nejlepší dosažení požadovaných a plynulých pohybů horních končetin. Pacienti jsou nejčastěji děti s dětskou mozkovou obrnou, starší lidé po mozkových příhodách a v současné době stále častěji lidé po různých nehodách a úrazech (kolo, motocykl, automobil), kdy byla částečně narušena komunikace mezi mozkiem a nervovými zakončeními v horních končetinách.

### 1.1 Využívaná zařízení / nástroje pro rehabilitaci

Jemnou motoriku horních končetin lze procvičovat jak klasickými rehabilitačními metodami (opakované pohybové sekvence), tak i různými výtvarnými dovednostmi, jako je běžné kreslení, velmi často práce s keramickou hlinou nebo dokonce různé stolní hry jako například oblíbené pexeso. Dále lze použít canisterapie (léčebná metoda, využívající k terapii psa), kdy rehabilitovaný může psovi házet míček, krmit jej pamlsky atd. Rovněž je možné využít při rehabilitaci i různých elektronických zařízení, se kterými lze vykonávat dokonce i sofistikované rehabilitace a to zejména interaktivní metodou. Toto v podobě různých her ovládaných od jednoduché klávesnice přes dotykové obrazovky až po využití speciálních předmětů s již integrovanými senzory většinou pohybu. S předměty obsahujícími takovéto senzory můžeme procvičovat i některé aktivity pro soběstačnost jako například pití z hrnku, jení přiborem, osobní hygienu a další. A právě způsobu rehabilitace pomocí takovýchto senzorů se specifickými předměty se bude tato práce věnovat.

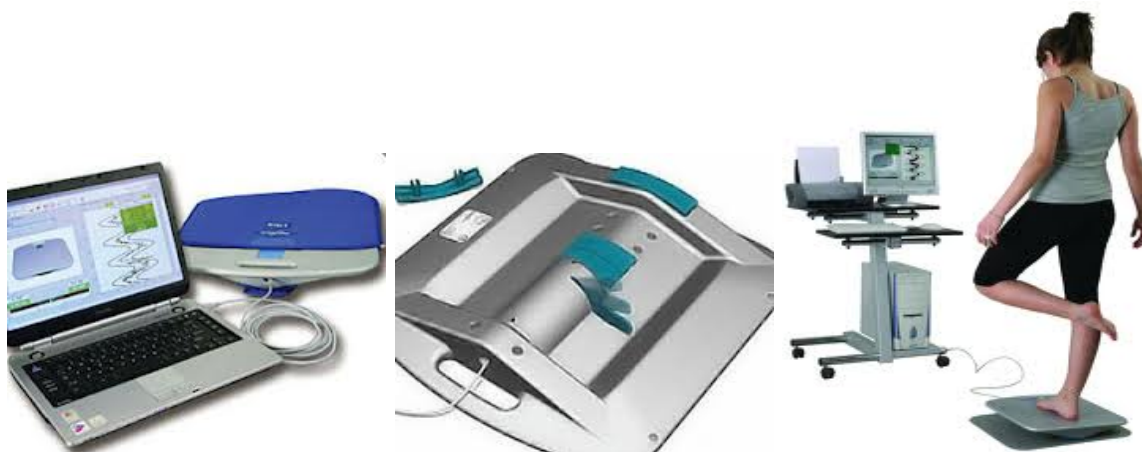
Mezi velmi sofistikované zařízení pro rehabilitaci jemné motoriky patří například rukavice Hand Tutor od firmy Medi Touch (viz. obrázek 1.1 na straně 2). Hand Tutor se skládá z ergonomické rukavice s několika senzory detekujícími pohyb ruky a speciálního rehabilitačního programu. Umožňuje tedy celou řadu biomechanických hodnocení, včetně rychlosti, pasivní a aktivní rozsah pohybu a analýzy pohybu prstů a zápěstí.[3] Výhodou tohoto zařízení je nejen sledování pohybu zápěstí, ale i jednotlivých prstů a samozřejmě celkově vysoká přesnost měření. Nevýhodou je však poněkud vysoká cena celého zařízení pohybující se v řádu desítek tisíc korun.

Pro rehabilitaci stability je používáno například zařízení Libra od firmy Easytech (viz. obrázek 1.2 na straně 2). Libra je deska o velikosti osobní váhy mající spodní stranu vy-



Obrázek 1.1: Rukavice Hand Tutor od firmy Medi Touch.

tvarovanou do desky tvoří jakousi houpačku. Připomíná tedy balanční hru, kdy člověk stojí na desce, která má na spodní straně uprostřed navíc polokouli. Na desce je nejčastěji ze shora vytvořeno trojrozměrné bludiště s kutálející se kuličkou a cílem této balanční hry je dopravit kuličku z jednoho konce bludiště do druhého. Libra pracuje na stejném principu jako balanční hra s rozdílem, že Libra se naklápí pouze okolo jedné osy. Zařízení se připojuje k běžnému počítači. Speciální aplikace pro zařízení Libra obsahuje jak diagnostické úlohy, tak i zábavné hry pro rehabilitaci.[4]



Obrázek 1.2: Zařízení Libra pro rehabilitaci stability.

### 1.1.1 Základní výhody a nevýhody

Základní výhody rehabilitace s využitím elektroniky spočívají v:

- zaznamenávání a uchovávání dat z průběhu rehabilitace například pro sledování pokroků v rehabilitačním procesu,
- částečné kontroly vykonávaných cvičení i bez přítomnosti lékaře,

- distribuce dat z průběhu rehabilitace lékařům,
- rehabilitace formou her (poutavé především pro mladší pacienty),
- trénování i aktivit pro dosažení soběstačnosti.

Nevýhodou této rehabilitace je:

- často nemalé prvotní náklady spojené s pořízením zařízení.

### 1.1.2 Dostupnost zařízení pro možnost rehabilitace v domácím prostředí

Možnost mít v domácím prostředí zařízení, která se využívají v rehabilitačních odděleních, je zatím v nedohlednu. Počty zařízení na osobu jsou příliš nízké, aby se tato zařízení půjčovala rehabilitovaným osobám i do domácího prostředí. Nyní rehabilitovaní doma nejčastěji používají pouze pasivní pomůcky (například speciálně vytvarované nádoby, hygienické potřeby atd.) nebo k nim do domácnosti dochází rehabilitační terapeut.

Jedním z řešení zvýšení dostupnosti zařízení pro domácí rehabilitaci jemné motoriky rukou je nalezení kompromisu mezi cenou tohoto zařízení na straně jedné a dostatečnou přesností na straně druhé pro tento typ rehabilitace. Výhody možnosti rehabilitace i v domácím prostředí jsou značné, například lze jmenovat:

- možnost velmi časté (i každodenní) rehabilitace,
- pohodlí pro rehabilitovaného (rehabilitovaný nemusí dojíždět na rehabilitační kliniku, je v pohodlí svého domova atd.),
- rehabilitovaný je v domácím prostředí méně stydlivý,
- znatelné zkrácení rehabilitačního procesu a tím i zrychlení návratu do pracovního procesu.

Za povšimnutí však stojí skutečnost, kdy při rehabilitaci v domácnosti vzniká jistá absence společnosti (kromě rodiny). Častý kontakt se společností je však velmi důležitý, obzvláště u dětí u kterých teprve probíhá sociální vývoj. Během tohoto období „dochází k postupné adaptaci dítěte na sociální prostředí“ [1], dítě se přizpůsobuje zásadám chování společnosti. Proto by určitý přesun rehabilitačního procesu do domácího prostředí, měl být současně kompenzován jinými společenskými interaktivitami.[2]

## 1.2 Návrh přípravku pro rehabilitace i v domácím prostředí

Základním požadavkem na přípravek umožňující rehabilitace jemné motoriky tedy zejména rukou i v domácím prostředí je samozřejmě jeho nízká cena, kvůli dostupnosti dostatečného počtu těchto přípravků pro rehabilitační zařízení, které tyto pomůcky právě pacientům mohou poskytovat / zapůjčovat. Dále samozřejmě jednoduchost ovládání zařízení, jelikož velkou část rehabilitovaných tvoří pacienti s mozkovou obrnou a starší lidé po mozkových příhodách. Rovněž je důležité, aby byla zaručena jednoduchost a rychlost jak používání výrobku, tak jeho zavedení do běžného provozu v rehabilitačním zařízení.

Hlavní hardwarová část zařízení, dále nazývaná měřicí modul, bude obsahovat vlastní senzory pro detekci / měření pohybu, bude umístěna přímo v předmětech sloužících jako rehabilitační pomůcky. Těmito předměty jsou myšleny speciální nadstavby určené pro konkrétní rehabilitační cvičení (základem bude krabička ve tvaru kvádra – ovladač her pohybem, dále pak například hrnek na pití, zvětšená hrací kostka, model autíčka atd.). Měřicí modul by měl být snímatelný, aby specifické předměty byly omyvatelné (důležité například pro hrnek na pití, který bude rehabilitovaný přikládat i k ústům).

Celé zařízení bude ve verzi prototypu s počítačem propojeno pomocí USB portu (napájení, komunikace s měřícím modulem, přenos dat). Měřicí modul bude obsahovat viditelné LED diody indikující stavy přístroje, jako jsou zapnutí přístroje, odesílání naměřených dat a některé další diagnostické funkce přístroje. Více v kapitole 3 na straně 13.

Mezi velmi často vykonávané rehabilitační metody patří cvičení soběstačnosti (například pití z hrnku, jedení přiborem, oblékání atd.) a úlohy tvořené formou zábavných her pro cvičení koordinace pohybů zejména rukou, přesnosti pohybů, udržení stability a samozřejmě zvětšování rozsahu a přesnosti těchto pohybů.

Programová aplikace vytvářená v rámci této bakalářské práce se bude zaměřovat na rehabilitaci procvičující koordinaci, přesnost, udržení stability a zvětšování rozsahu pohybů v podobě zábavných her. Dále se bude věnovat rehabilitaci procvičující pití z hrnku. Rehabilitace bude v podobě napodobování pohybu s hrnkem, který bude zobrazen na monitoru. Zařízení představující hrnek na pití bude určeno pro trénink manipulace s předmětem, stabilní držení těžšího předmětu a jemnou manipulaci s tímto předmětem. Cílem této úlohy bude tedy naučení rehabilitovaného samostatnému napití se z klasického či částečně upraveného hrnku. Programová aplikace bude samozřejmě obsahovat i úlohu pro základní diagnostiku rozsahu pohybu zejména rozsahu otáčení rukou (více v kapitole 8.8 na straně 43).

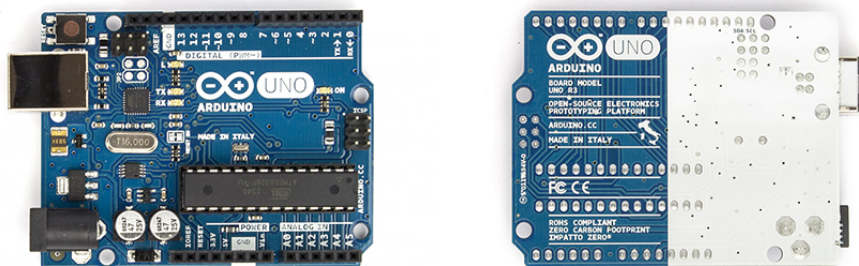
### 1.3 Možnost využití platformy Arduino

Při sestavování HW (hardware) části prototypu zařízení (obsahující senzor pohybu) byla využita vývojová platforma Arduino, která je vhodná především pro výukové účely a komunikaci s běžnými senzory.

Arduino představuje open-source platformu (v tomto případě volně dostupné zdrojové kódy i elektronická dokumentace používané a nabízené jako základ pro další vývoj) využívající nejčastěji mikrokontrolér typu ATmega od firmy Atmel. Platforma je založená na flexibilním a snadno použitelném hardware a software. Arduino je vhodné pro snadné získávání informací z různých senzorů či k řízení například světel, motorů atd. Vytvořené projekty mohou pracovat samostatně na Arduinu (například zobrazovat informace na připojeném displeji či řídit další zařízení) nebo mohou komunikovat se softwarem, který běží na počítači (například zpracování dat a jejich následné využití v aplikaci). Arduino si lze postavit tzv. „ručně“, nejčastěji tedy na tzv. nepájivém poli pro testování právě prototypů zařízení, nebo jej lze zakoupit již zkompletované na profesionálním plošném spoji (viz. obrázek 1.3 na straně 5).

#### 1.3.1 Historie

Projekt Arduino vznikl v roce 2005 v Itálii ve firmě Smart Projects. Tato firma rovněž vyvinula vývojové prostředí Arduino IDE. Cílem projektu bylo vytvořit jednoduchou pro-



Obrázek 1.3: Arduino Uno (přední strana, zadní strana).

totypovací platformu pro studenty umožňující rychlý vývoj a snadné používání. Projekt zaznamenal velký úspěch jak u začátečníků, tak i u pokročilých uživatelů. Postupem času začaly vznikat další verze a na začátku roku 2010 se prodalo více jak 120 tisíc kusů.

Arduino je na mikrokontrolérech ATmega od firmy Atmel založeno již od začátku. V oficiálních verzích se lze setkat s procesory ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560.[10, 11]

### 1.3.2 Hardware

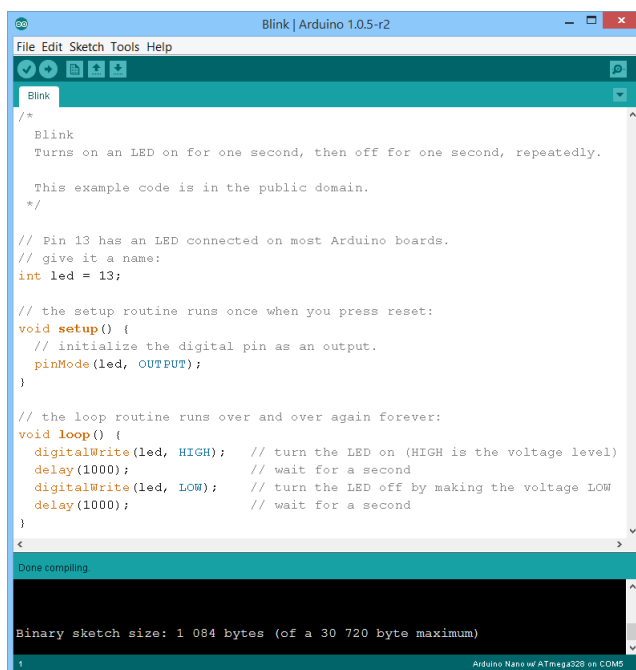
Základem Arduina, jak již bylo zmíněno, je mikrokontrolér z rodiny ATmega. Výrobce těchto čipů je norská firma Atmel. Mikrokontroléry ATmega patří do skupiny jednočipových mikropočítačů s architekturou AVR, což jsou osmibitové procesory typu RISC (Reduced Instruction Set Computing) s harvardskou architekturou (tj. má oddělený paměťový prostor pro program a pro data). Jednotlivé použité typy mikrokontrolérů (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) se liší především velikostí vnitřní paměti RAM (Random-Access Memory) a vnitřní paměti Flash (elektricky programovatelná paměť schopná uchovávat trvale data nezávisle na přítomnosti elektrické energie).

Každý mikrokontrolér obsahuje kromě CPU (Central Processing Unit) i některé periferie jako například časovače, sériové a paralelní porty, A/D převodníky pro měření analogových hodnot atd. Na desce Arduino je kromě mikrokontroléru i několik dalších podpůrných obvodů, to jsou především stabilizátory napájecího napětí a obvody zajišťující komunikaci s PC (u starších modelů sériový port RS-232, u novějších modelů USB).

V poslední době se objevují i klony, založené na jiných procesorech, například Microchip PIC, které nejsou kompatibilní softwarově, ale mívají stejné zapojení konektorů, lze k nim tedy připojit i originální nadstavby (shieldy) pro Arduino.[11]

### 1.3.3 Software

Programy pro Arduino lze psát, stejně jako pro jakoukoli jinou vývojovou desku s procesorem AVR, v nástrojích pro tento procesor určených (například Atmel Studio, AVR Studio atd.) nebo přímo v prostředí Arduino IDE (viz. obrázek 1.4 na straně 6). Nástroj Arduino IDE



```

Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

```

Done compiling

Binary sketch size: 1 084 bytes (of a 30 720 byte maximum)

Arduino Nano w/ ATmega328 on COM5

Obrázek 1.4: Arduino IDE (s předpřipraveným zdrojovým kódem nástroje Arduino IDE - blikání LED diodou).

je také otevřený, tedy volně ke stažení a používání. A co se týká operačního systému, Arduino IDE je multiplatformní (Win, Linux, Mac OS).

Autoři Arduina však nechtěli zatěžovat začínající zájemce o psaní software assemblerem nebo jazykem C, proto pro Arduino lze vyvíjet v jazyce zvaném Arduino Programmable Language. Tento jazyk je založen na jazyce Wiring, což je jazyk přímo vzniklý z C, odrážející specifické požadavky na vývoj software pro jednočipová zařízení. Dá se říci, že se jedná o „zjednodušené céčko“.[11]

### 1.3.4 Arduino Uno

Nejčastěji používanou verzí je Arduino Uno. Arduino Uno používá mikrokontrolér ATmega328. Deska obsahuje 14 digitálních vstupních / výstupních pinů (z toho 6 pinů může být použito jako výstupy PWM – Pulse Width Modulation), 6 analogových vstupů, 16 MHz krystal, připojení k PC pomocí USB typu B, napájecí konektor, ICSP (In-Circuit Serial Programming, možnost programování mikrokontroléru přímo v aplikaci sériovou linkou [12]), rozhraní a resetovací tlačítko.

Základní parametry:

- pracovní napětí 5 V (regulace napětí i na 3,3 V), vstupní napětí 7-12 V a maximální vstupní napětí 6-20 V,
- maximální proudový odběr na I/O pinu 40 mA DC (Direct Current) a na 3,3 V pinu 50 mA DC,
- paměť Flash 32 KiB<sup>1</sup>, ze které 0,5 KiB použito pro bootloader, paměť SRAM (Static Random Access Memory) 2 KiB a paměť EEPROM (Electrically Erasable Programmable Read-Only Memory) 1 KiB,
- největší rozměry desky Arduino Uno 75 mm na délku, 53 mm na šířku a 15 mm na výšku[14].

Arduino Uno se liší od všech předcházejících desek v tom, že nepoužívá čip FTDI pro USB-to-serial. Místo něj je použit běžný procesor Atmega16U2 naprogramovaný jako převodník USB-to-serial.[13]

## 1.4 Využití akcelerometru pro detekci pohybu

Akcelerometr je senzor pro současné měření:

- dynamického zrychlení – síla vzniklá změnou rychlosti pohybujícího se předmětu,
- statického zrychlení – síla vzniklá působením zemské gravitace.

Používá se nejen k měření akcelerace a náklonu, ale i k měření setrvačných sil (např. v automobilovém průmyslu – airbagy, systémy jízdní stability), měření odstředivé síly, měření a detekci vibrací (například při chodu stroje, může detekovat drhnoucí ložisko) nebo měření seismické aktivity a po zpracování získaných dat například využitím integrace i k měření rychlosti pohybu. S akcelerometry se velmi často setkáváme také v každodenním běžném životě. Například dnes již skoro každý Smartphone (chytrý telefon) využívá zabudované akcelerometry nejen ke hraní her ovládaných pohybem telefonu, ale především k otáčení obrazu na displeji dle aktuální orientace chytrého telefonu. V tomto případě není využívána hodnota dynamického zrychlení telefonu, ale informace o statickém gravitačním zrychlení. Hodnoty vypovídající o statickém zrychlení se pohybují v rozmezí hodnot odpovídajících  $[-1g, 1g]$ . Hodnoty odpovídající méně jak  $-1g$  a více jak  $1g$  naopak odráží velikost dynamického zrychlení. Prototypové zařízení konstruované v rámci této bakalářské práce bude stejně jako chytré telefony využívat hodnot statického gravitačního zrychlení pro určení náklonu zařízení dle zvolených os akcelerometru.

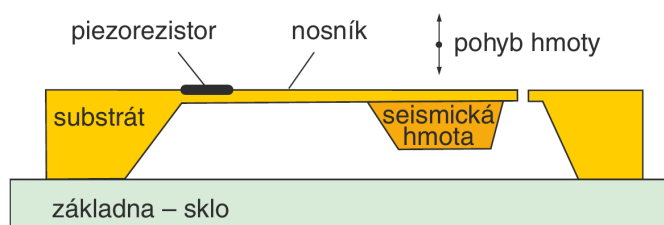
Akcelerometr se obecně skládá ze základny, která pro správné měření musí být pevně spojena s předmětem jehož pohyb měříme, z pružně uložené setrvačné hmoty, jejíž výchylka je vůči základně vyhodnocována a z viskózního tlumení. Akcelerometry podle vnitřní konstrukce, tedy podle principu detekce pohybu, dělíme do několika základních skupin.

---

<sup>1</sup>Kibibajt, 1 KiB =  $2^{10}$  B = 1 024 B

### 1.4.1 Piezorezistorové akcelerometry

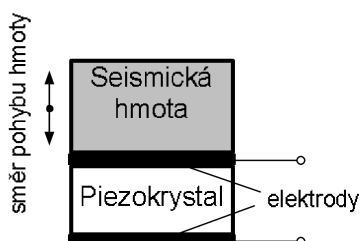
Piezorezistorové akcelerometry pracují na principu tenzometru, tedy změně ohmického odporu. Pohybem hmoty se zkracuje či prodlužuje piezoodporový element čímž dochází ke změně ohmického odporu (viz. obrázek 1.5 na straně 8). Tato změna je úměrná výchylce hmoty.



Obrázek 1.5: Struktura piezorezistorového akcelerometru.

### 1.4.2 Piezoelektrické akcelerometry

Piezoelektrické akcelerometry využívají piezoelektrický krystal generující náboj úměrný působící síle při zrychlení (viz. obrázek 1.6 na straně 8). Nevýhodou tohoto typu senzoru je nemožnost jeho použití pro měření statického gravitačního zrychlení.

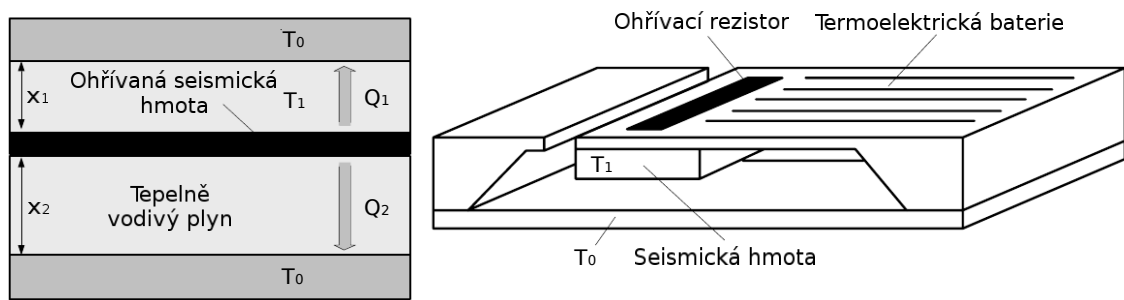


Obrázek 1.6: Struktura piezoelektrického akcelerometru.

### 1.4.3 Tepelné akcelerometry

V tepelných akcelerometrech je seismická hmota umístěna na tenkém nosníku a nachází se v blízkosti tepelné komory nebo mezi dvěma takovými komorami. Prostor mezi těmito komponenty je vyplněn teplotně vodivým plynem. Hmota je ohřívána na povrchu nebo zabudovaným ohříváčem na definovanou teplotu  $T_1$ . Pokud nepůsobí akcelerační zrychlení, potom je mezi hmotou a ohřívánými komorami teplotní rovnováha. Množství tepla  $Q_1$  a  $Q_2$  vedené od hmoty do komor přes plyn je funkcí vzdáleností  $x_1$  a  $x_2$  (viz. obrázek 1.7 na straně 9). Pro měření teploty je na nosníku umístěn teplotní senzor.

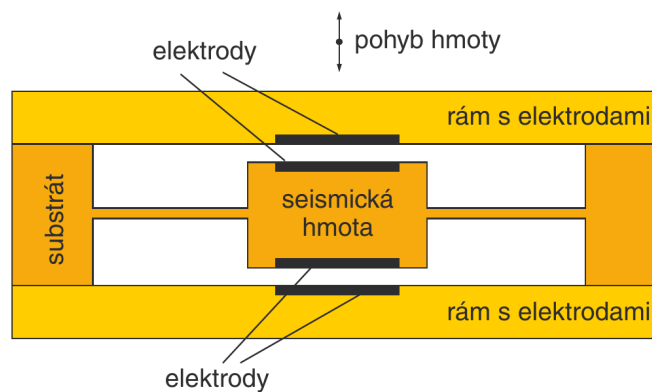




Obrázek 1.7: Řez ohřívací částí a struktura tepelného akcelerometru.

#### 1.4.4 Kapacitní akcelerometry

Kapacitní akcelerometry využívají změnu kapacity v důsledku výchylky seismické hmoty. Pevné elektrody tvoří vůči seismické hmotě dvě kapacity. Při pohybu dochází ke změně vzdáleností desek takto vytvořeného kondenzátoru, a tím tedy ke změně kapacit (jedna vzrůstá, druhá klesá; viz. obrázek 1.8 na straně 9). Kapacitní akcelerometry v porovnání s piezorezistorovými mají obecně vyšší stabilitu a citlivost.[5, 6]



Obrázek 1.8: Struktura kapacitního akcelerometru.



## Kapitola 2

# Záměr této práce

Jelikož rozsah práce může být vcelku obsáhlý, tak je nutno vymezit čemu se tato práce bude a čemu nebude věnovat. Bakalářská práce se tedy věnuje hlavně těmto tématům:

- návrhu prototypu zařízení pro rehabilitaci jemné motoriky zejména rukou po hardwarové stránce,
- prototypu zařízení levného a snadno použitelného,
- prototypu zařízení zaměřeného na rehabilitaci zejména v domácím prostředí,
- tvorbě ukázkových rehabilitačních úloh v programové aplikaci.

Nejedná se tedy nikterak o konečnou verzi zařízení určenou pro sériovou výrobu, ale o první verzi prototypu tohoto zařízení, které by se teprve zapůjčovalo pro několika měsíční testování na rehabilitačním pracovišti.

Práce se na druhou stranu nevěnuje těmto tématům:

- tvorbě zařízení pro komerční využití (tedy splňující veškeré podmínky takového zařízení),
- tvorbě zařízení pro přesnou diagnostiku a měřící účely,
- korekci měřených hodnot v závislosti na okolních podmínkách (například na teplotě),
- tvorbě databáze pacientů a poskytování statistik z měření na dálku,
- tvorbě plné dokumentace pro opakovanou výrobu zařízení.

Obsah úvodní kapitoly již seznamoval s možnostmi rehabilitace jemné motoriky rukou, s nyní používanými pomůckami / zařízeními, s jejich výhodami a nevýhodami, s jejich dostupností v domácím prostředí a s hlavními požadavky na takováto zařízení k domácímu používání. V následujících kapitolách bakalářské práce je popsán návrh hardwarové a programové podpory pro tento typ rehabilitace jak pro možné použití v ordinaci lékaře, tak ale

především pro využití v domácím prostředí. Dále jsou zařazeny i kapitoly s popisem sestavení prototypu zařízení pro ověření jeho činnosti a kapitoly s popisem implementace některých ukázkových rehabilitačních úloh pro skutečnou rehabilitaci s vytvořeným prototypem zařízení. Nechybí ani dokumentace pro použití pro budoucí uživatele.

## Kapitola 3

# Návrh hardwarové části zařízení

Prvotní prototyp zařízení pro rehabilitaci jemné motoriky zejména rukou je sestaven na tzv. nepájivém poli pro testování komunikace mikrokontroléru se senzorem a pro možnost snadné výměny senzoru za jiný v průběhu vývoje (viz. obrázek 3.2 na straně 15). Hardwarová část zařízení byla inspirována platformou Arduino Uno (elektrické schéma Arduino Uno naleznete na přiloženém CD nebo na stránkách [9]). Platforma Arduino Uno byla použita jako vzor (platforma Arduino je open-source), jelikož stejně jako všechny ostatní platformy od Arduino je vyvíjena především pro práci se senzory, tudíž umožňuje snadný a rychlý vývoj prototypů zařízení využívající běžné senzory. A v případě sestavování další verze prototypu zařízení pro testování přímo při rehabilitaci by bylo Arduino cenově přívětivou variantou.

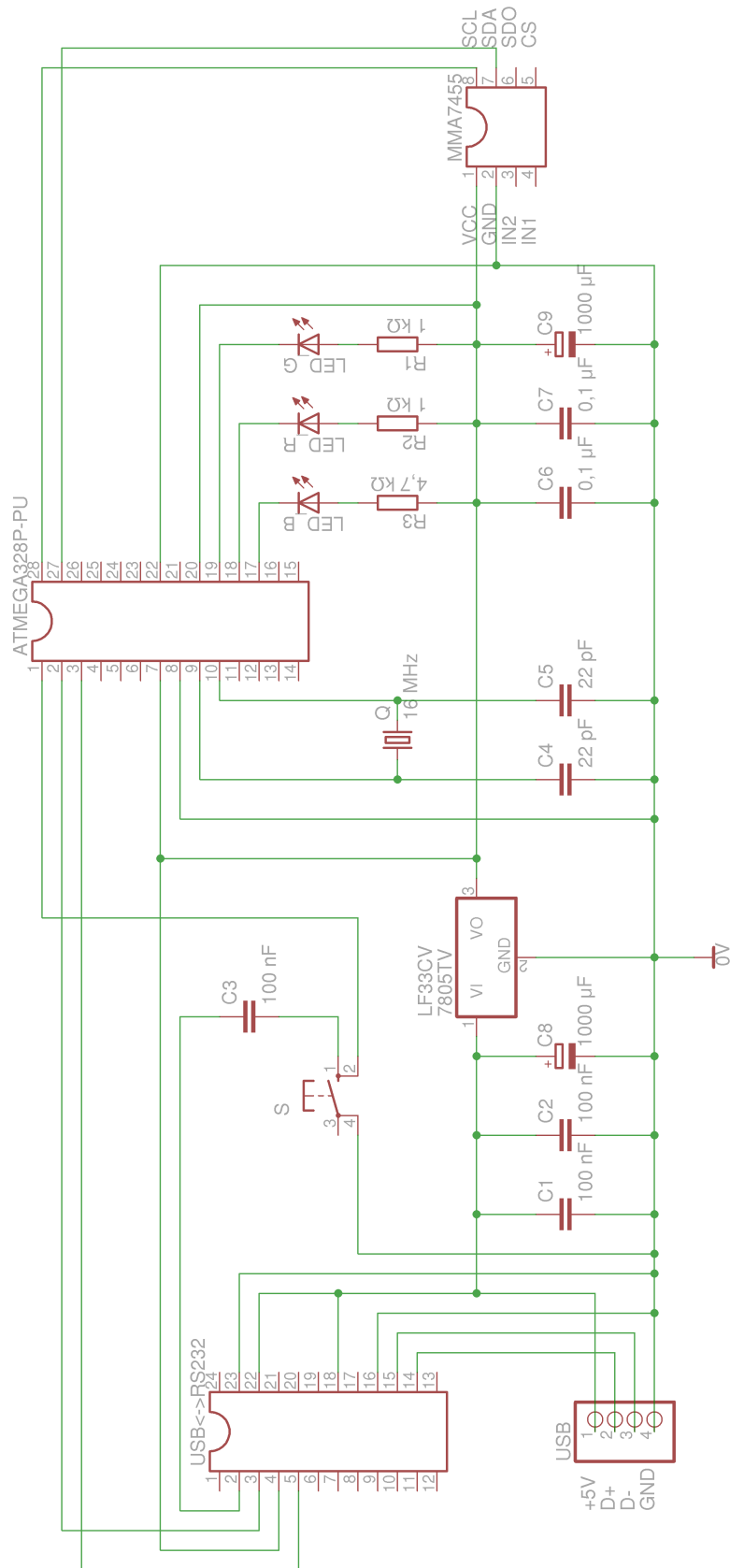
Jako výchozí senzor pro měření pohybu (přesněji řečeno náklonu) byl zvolen akcelerometr MMA7455. Tento akcelerometr byl vybrán pro velmi vhodný poměr ceny a přesnosti a pro jeho dobrou dostupnost. Modul se senzorem MMA7455 (viz. obrázek 4.1 na straně 17) je dostupný již za přibližně 50 Kč. A nejmenší senzorem rozlišitelná jednotka odpovídá otočení okolo dané osy akcelerometru o úhel přibližně 1,41 stupňů (podle úhlu náklonu), což je pro zamýšlený typ rehabilitace jemné motoriky rukou dostačující.

### 3.1 Hlavní části prototypu zařízení

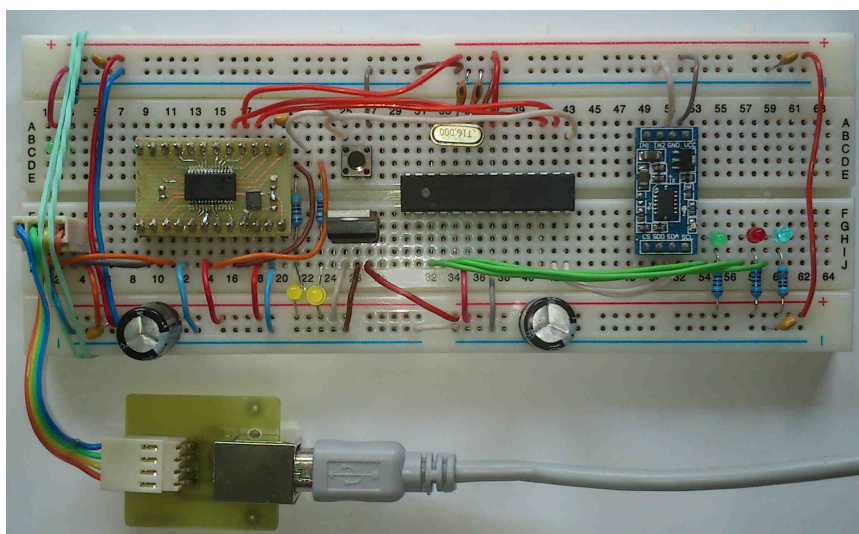
Tento první prototyp zařízení se skládá zejména z již zmiňovaného akcelerometru MMA7455, dále mikrokontroléru ATmega328 od firmy Atmel a na konec převodníku FT232RL (USB – RS232) od firmy FTDI. Konektor na prototypu zařízení pro propojení s počítačem je USB typu B. Detailnější elektrické schéma prototypu zařízení naleznete na obrázku 3.1 na straně 14).

Chování prototypu zařízení a zpracování dat je řízeno programem běžícím přímo v mikrokontroléru v zařízení. Pro psaní programu byl použit nástroj Arduino IDE.

Soubor se zdrojovým kódem (tzv. sketch) má příponu \*.ino a je zpravidla uložen v adresáři se stejným názvem jako je název souboru. Sketch je rozdělen do dvou povinných bloků,



Obrázek 3.1: Elektrické schéma prvního prototypu zařízení.



Obrázek 3.2: První prototyp zařízení pro rehabilitaci jemné motoriky zejména rukou sestavený na tzv. nepájivém poli.

přesněji řečeno funkcí. První je v podstatě inicializační funkce `setup()` a druhou je opakovaně vykonávaná funkce `loop()`. Vedle těchto povinných funkcí mohou být samozřejmě pro přehlednější zdrojový kód vytvořeny i další funkce využívané v ostatních funkcích sketchu.

### 3.1.1 Implementace funkce `setup()` a `loop()`

Ve funkci `setup()` se uskuteční veškerá potřebná nastavení jako je například nastavení konkrétních pinů Arduina na vstupní či výstupní, otevření sériového portu, nastavení přenosové rychlosti pro komunikaci s počítačem a další inicializační akce. Tato funkce je tedy vykonána pouze jednou při spuštění programu.

Pro komunikaci s rehabilitační aplikací (na PC) byla zvolena nejvyšší Arduinem doporučená přenosová rychlost, tedy 115 200 bitů za vteřinu (baudů).

Po dokončení funkce `setup()` se stále v nekonečné smyčce vykonává funkce s názvem `loop()`, jak už napovídá její název. V této části sketchu se uskutečňují čtení dat ze senzoru, základní potřebné výpočty, zpracovávání příkazů z řídicí aplikace (PC), ovládání světelných kontrol (LED) stavu zařízení a zaslání dat do PC.

## 3.2 Způsob signalizace stavu zařízení

Prototyp zařízení pro rehabilitaci je opatřen třemi světelnými LED diodami (zelená, modrá, červená) pro vizualizaci jeho aktuálního stavu. Význam jednotlivých signálů daných LED diod je podrobněji uveden v tabulce 3.1 na straně 16.

barva	význam	druhy signalizace	podrobnější popis
zelená	napájení	zapnuto/vypnuto	Svítili kontrolka, zařízení je zapnuté. Naopak je-li kontrolka zhasnutá, zařízení je bez napájení.
modrá	data	posílání dat spuštěno/zastaveno	Kontrolka se střídavě rozsvícuje a zhasíná se stejnou frekvencí jako je frekvence posílání dat.
červená	stav	potvrzení nastavení komunikace	Po dokončení nastavení komunikace kontrolka 5x za sebou krátce blikne.
		potvrzení přijetí příkazu	Při přijetí příkazu z aplikace kontrolka 2x za sebou krátce blikne.
		zanepřázdňný	Kontrolka svítí v průběhu vykonávání příkazu.

Tabulka 3.1: Významy jednotlivých signálů světelných kontrollek na zařízení.

V krabici dostatečně uzavřená verze prototypu zařízení by již mohla být v rámci experimentů zapůjčena lékařům například k několika měsíčnímu testování na některé rehabilitační pracoviště. Návrhy na změny, zlepšení zapůjčeného prototypu zařízení by pak byly využity pro tvorbu další verze prototypu zařízení pro rehabilitaci. Jedná se zde o oblast kdy pouze vzájemnou komunikací s lékaři a dlouhodobějším ověřováním navrženého zařízení lze dospět ke skutečně vyhovující a prospěšné pomůcce.



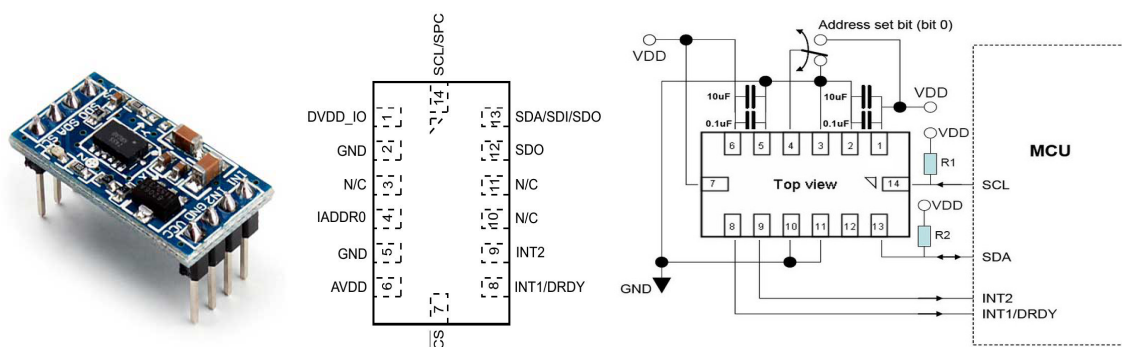
## Kapitola 4

# Senzor použitý v prototypu zařízení

V prototypu zařízení je použit tříosý akcelerometr MMA7455 firmy Freescale pro měření statického zrychlení, tedy náklonu vůči směru gravitačního zrychlení.

Pro další testování možnosti použití v prototypovém zařízení byl zvažován i tříosý akcelerometr ADXL345 firmy Analog Devices a devítiosý kombinovaný senzor MPU9150 firmy Inven Sense. Akcelerometr ADXL345 svými parametry spadá do stejné kategorie jako akcelerometr MMA7455 a nejnižší cena modulu s tímto senzorem se pohybuje již okolo 40 Kč. Senzor MPU9150 je kombinací tříosého akcelerometru, tříosého magnetometru a tříosého senzoru úhlové rychlosti. Patří však již mezi dražší senzory, modul s tímto senzorem lze pořídit za cenu od 275 Kč. Bohužel v dané době tento senzor nebyl dostupný.

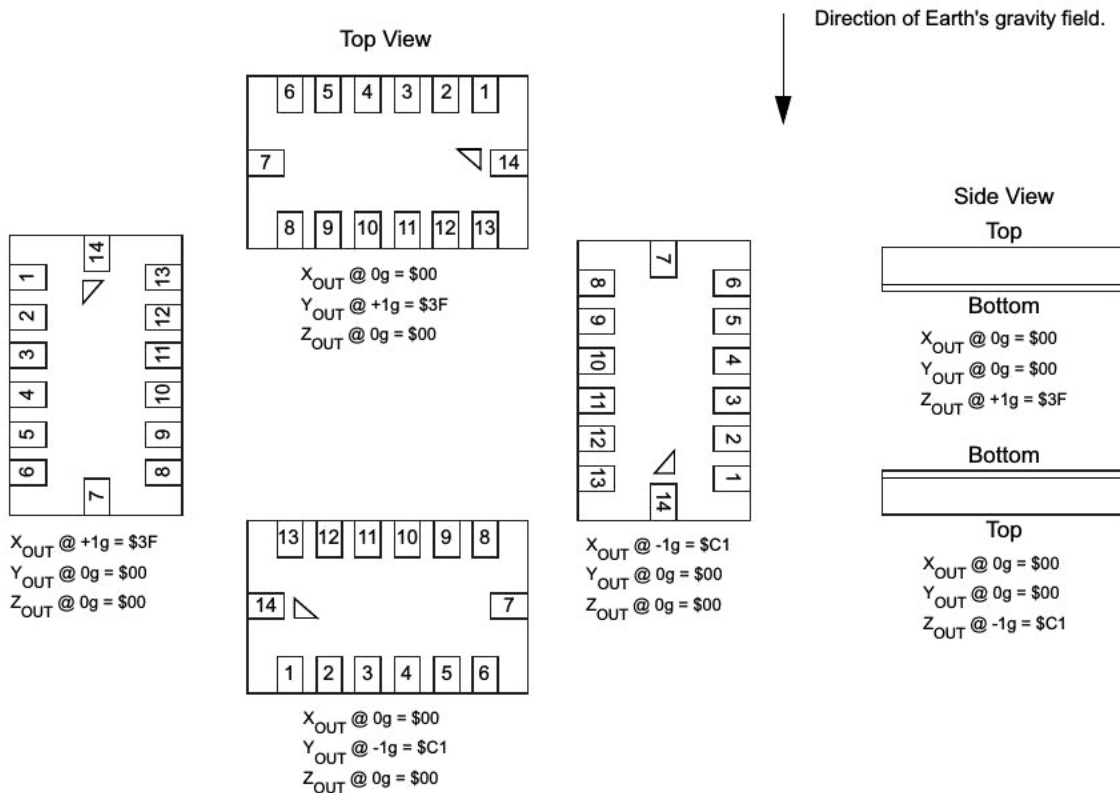
### 4.1 Akcelerometr MMA7455



Obrázek 4.1: Akcelerometr MMA7455 (modul s akcelerometrem, piny akcelerometru, I<sup>2</sup>C připojení k mikrokontroléru).

Akcelerometr MMA7455 poskytuje digitální výstup typu I<sup>2</sup>C/SPI (Inter-Integrated Circuit / Serial Peripheral Interface). Rozměry samotného integrovaného obvodu (senzoru) jsou 3 mm na šířku, 5 mm na délku a 1 mm na výšku, pouzdro typu LGA - 14. Napájecí napětí senzoru se může pohybovat v rozsahu 2,4 V až 3,6 V. Citlivost senzoru lze nastavit na  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ . Robustní provedení zaručuje životnost senzoru do zrychlení 5 000g.

Hodnoty poskytnuté akcelerometrem MMA7455 odpovídají míře působení gravitační síly proti směru dané osy akcelerometru. Čím nižší je poskytnutá hodnota z dané osy, tím více působí gravitační síla ve směru této osy. Pokud tedy některá osa akcelerometru směřuje přímo nahoru, senzor vrací hodnotu odpovídající zrychlení  $1g$ . Směřuje-li naopak tato osa přímo dolů, senzor vrací hodnotu pro zrychlení  $-1g$ . A tedy nulová hodnota bude vrácena z aktuálně vodorovně ležící osy akcelerometru (viz. obrázek 4.2 na straně 18). Více informací o akcelerometru MMA7455 naleznete v technické dokumentaci přiložené na CD nebo přímo na stránkách firmy Freescale [7].



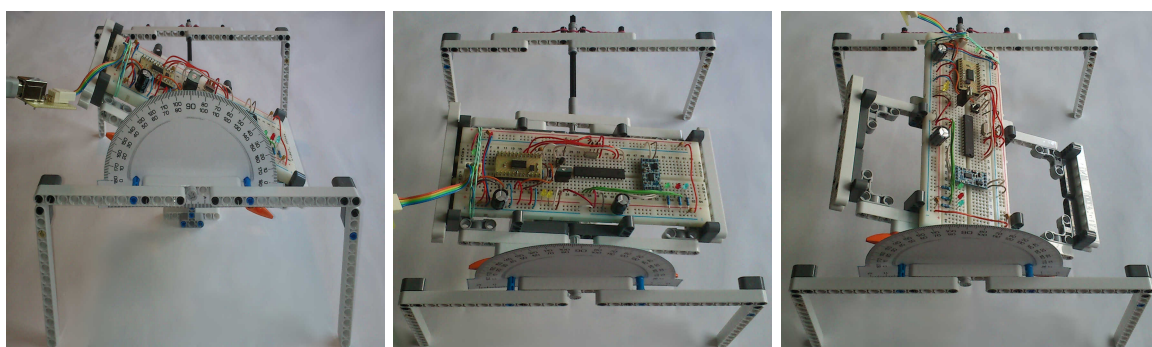
Obrázek 4.2: Hodnoty poskytující akcelerometr MMA7455 ve směru jednotlivých os pro daná otočení senzoru.

Akcelerometr MMA7455 byl vybrán pro jeho velmi nízkou a tedy přijatelnou cenu, dobrou dostupnost a snadné použití. Cena vybraného modulu s akcelerometrem MMA7455 (viz. obrázek 4.1 na straně 17) se pohybuje v rozmezí 50 Kč až 150 Kč. Snadné použití senzoru podporuje možnost využití přímo digitálního výstupu I<sup>2</sup>C (multi-masterová sériová sběrnice) pro čtení hodnot.

## 4.2 Měření statické převodní charakteristiky

Statická převodní charakteristika senzoru je dána funkční závislostí  $y = f(x)$  mezi vstupní veličinou  $x$  a výstupní veličinou  $y$  v časově ustáleném stavu. Tuto závislost lze, jak bylo zjištěno, popsat polynomem  $y = a_n x^n + \dots + a_2 x^2 + a_1 x + a_0$ . [8]

Měření statické převodní charakteristiky akcelerometru MMA7455, konkrétně závislosti naměřené hodnoty akcelerometrem na úhlu otočení okolo dané osy akcelerometru, bylo uskutečněno pomocí za tímto účelem experimentálně zhotovené konstrukce z Lega s úhloměrem (viz. obrázek 4.3 na straně 19). Měření rotace se uskutečnilo okolo dvou vodorovných os akcelerometru, tedy okolo osy X a osy Y.



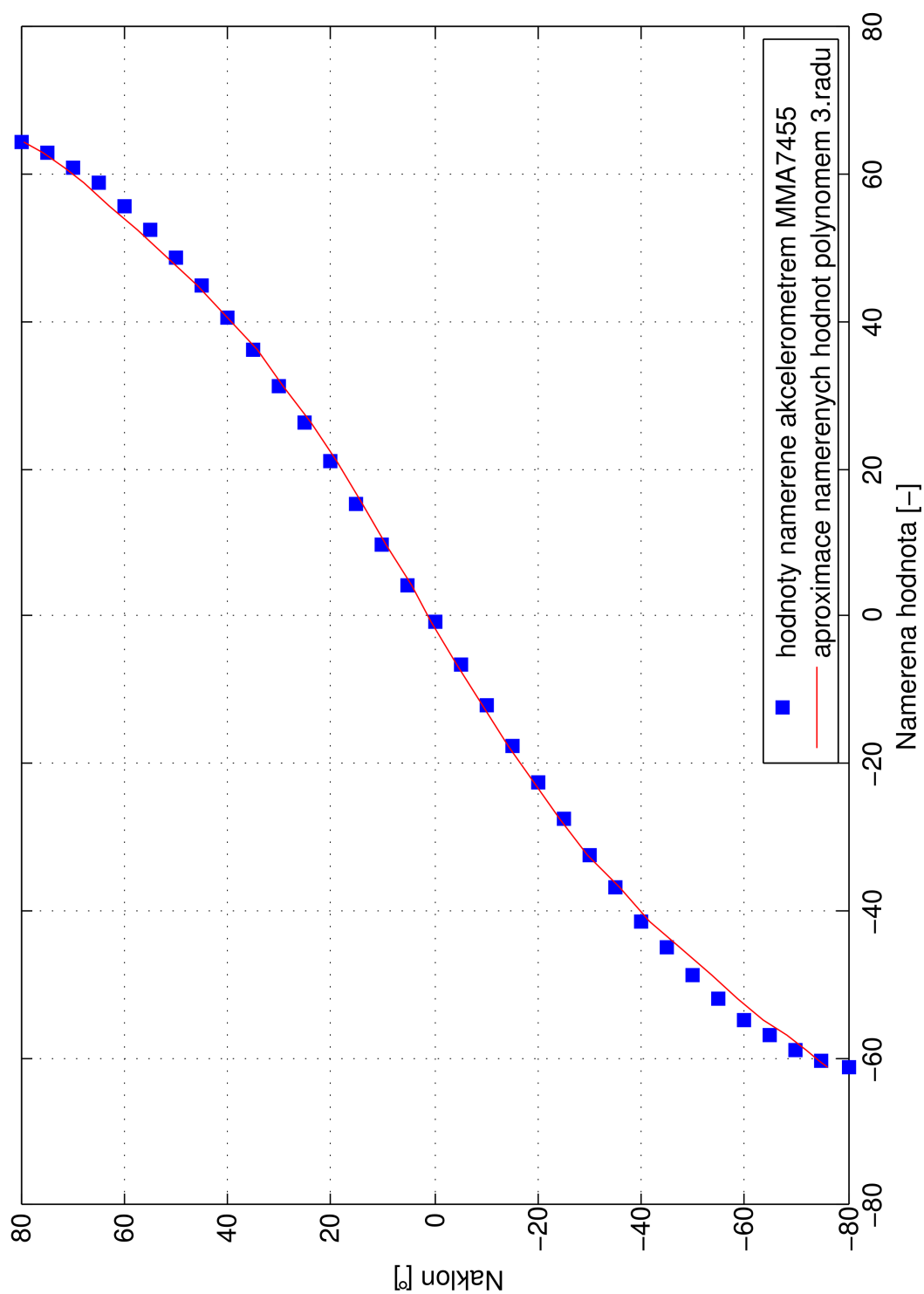
Obrázek 4.3: Konstrukce pro měření závislosti naměřené hodnoty akcelerometrem na úhlu otočení okolo dané osy akcelerometru (zepředu, rotace okolo osy X, rotace okolo osy Y).

Pro měření byl napsán program, který během jedné vteřiny vyčetl ze senzoru sadu sta hodnot a poskytl aritmetický průměr těchto hodnot. Toto vždy uskutečnil po desetivteřinových intervalech. V průběhu této desetivteřinové pauzy byl senzor pokaždé pootočen o konstantní úhel. Měření se uskutečnilo pro náklony od  $-80$  stupňů do  $+80$  stupňů s krokem  $5$  stupňů. Celé toto měření proběhlo vícekrát pro obě vodorovné osy akcelerometru a hodnoty pro dané náklony senzoru byly zprůměrovány, aby se minimalizovala chyba ručního nastavování úhlu náklonu senzoru.

V programu MATLAB byla k naměřené funkci nalezena inverzní funkce a ta aproximována polynomem 3. stupně (viz. obrázek 4.4 na straně 20). Do aplikace využívající akcelerometr pro rehabilitaci pak byla použita funkce (bez posunu pro správnou funkci kalibrace na nulovou hodnotu)

$$f : y(x) = 0,00010x^3 - 0,00092x^2 + 0,80920x + 1,40299, \quad x \in [-61, 65].$$

Funkce  $f$  tedy udává úhel otočení akcelerometru okolo dané vodorovné osy v závislosti na hodnotě poskytnuté akcelerometrem.



Obrázek 4.4: Závislost náklonu akcelerometru MMA7455 na naměřené hodnotě tímto akcelerometrem.

## Kapitola 5

# Zprávy předávané mezi zařízením a aplikací (PC)

Předávané zprávy mají buď příkazový charakter, nebo datový charakter. Mezi zprávy příkazového charakteru (přenos z PC do zařízení) patří například spuštění / zastavení měření hodnot akcelerometrem. Mezi zprávy datového charakteru (přenos ze zařízení do PC) patří například zprávy obsahující naměřené hodnoty ze všech tří os akcelerometru (i když jsou v současné verzi aplikace využity pouze dvě vodorovné).

### 5.1 Tvar zpráv

Zpráva předávaná mezi zařízením a aplikací (tvar zpráv je stejný pro oba směry komunikace) vždy začíná úvodním znakem „:“, následuje obsah zprávy a ukončení zprávy znakem „#“. Zvolení počátečního a koncového znaku je výhodné z hlediska nejen snadného zpracování dat, ale zejména k odlaďování samotné komunikace. Hlavní část zprávy je tvořena jedním znakem udávajícím konkrétní příkaz, následovaný případným parametrem obsaženým v následujících znacích až do koncového znaku zprávy. Implementované typy

- **příkazových zpráv** podrobněji popsány v tabulce 5.1 na straně 22,
- **datových zpráv** v tabulce 5.2 na straně 22.

Zprávy s příkazem „C“ a „Z“ aplikace pro rehabilitaci nevyužívá, byly využívány pouze při stavbě prototypu zařízení. Kalibrace byla přesunuta přímo do aplikace na PC a měření závislosti hodnoty naměřené akcelerometrem na náklonu akcelerometru slouží pouze pro zjištění statické převodní charakteristiky aktuálně použitého senzoru.

Zařízení při přijetí příkazu z aplikace (PC) pošle zpět tzv. potvrzovací zprávu obsahující pouze vrácený znak přijatého příkazu.

znak	význam zprávy	parametr	příklad zprávy
A	spustit posílání hodnot naměřených akcelerometrem	—	:A#
B	zastavit posílání hodnot naměřených akcelerometrem	—	:B#
C	zkalibrovat senzor na nulu a na hodnotu 1g pro svislou osu	svislá osa senzoru (1~X, 2~Y, 3~Z)	:C3#
S	nastavit rychlost měření	počet vzorků za vteřinu {1,..., 120}	:S100#
Z	spustit měření závislosti naměřené hodnoty na náklonu akcelerometru	krok měřených náklonů senzoru [°] {1, ... , 90}	:Z5#

Tabulka 5.1: Přehled příkazových zpráv posílaných z aplikace do zařízení.

znak	význam zprávy	parametr	příklad zprávy
D	hodnoty naměřené akcelerometrem	naměřené hodnoty ze všech os senzoru	:Dx=-1,y=4,z=51#
*	potvrzení přijetí příkazu	—	:*#

Tabulka 5.2: Přehled datových zpráv posílaných ze zařízení do aplikace.

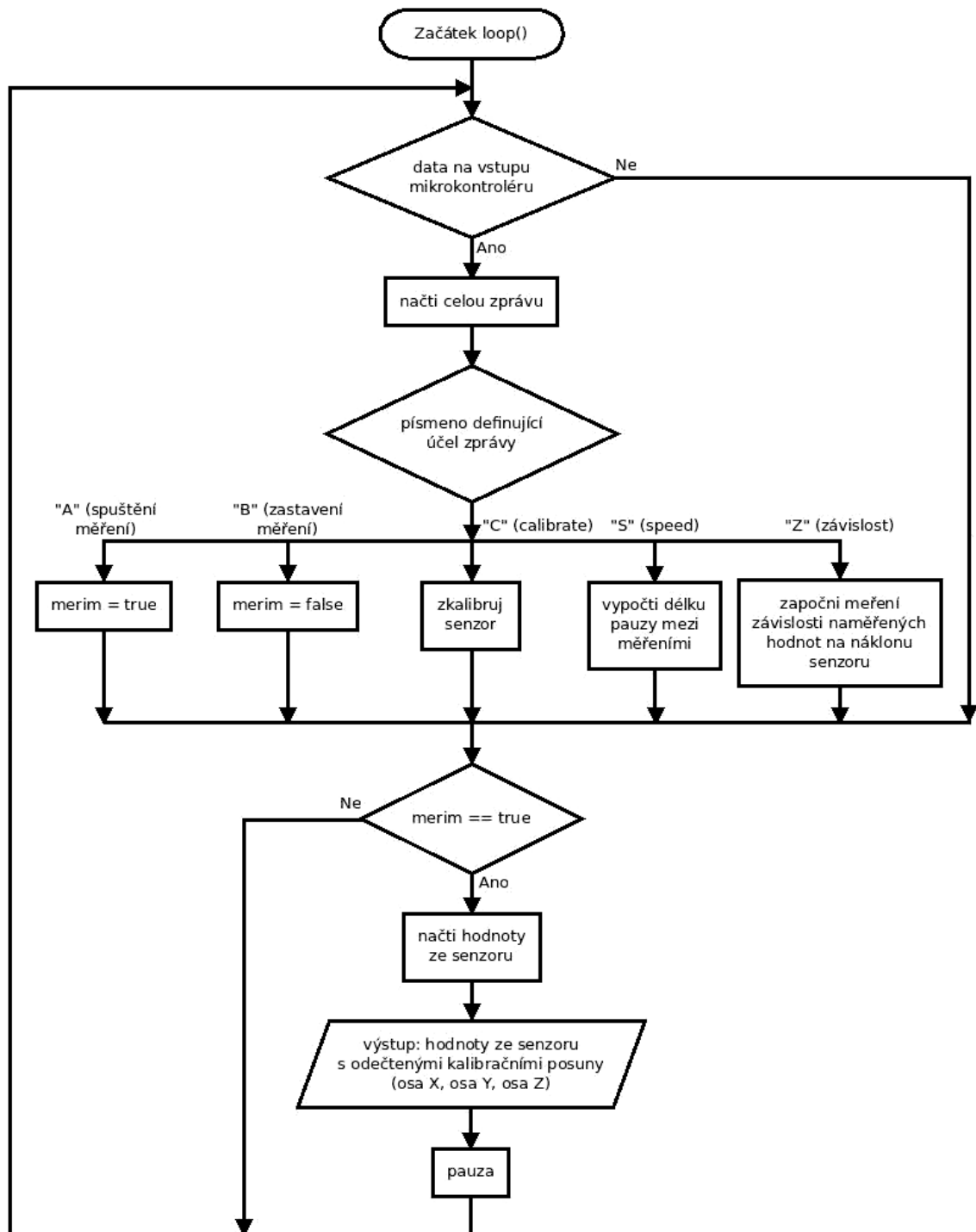
## 5.2 Zpracování zpráv v zařízení

Zpracování zpráv z aplikace (PC) ve sketchi (program v mikrokontroléru v zařízení) se zjednodušeně řečeno skládá z těchto částí (také viz. obrázek 5.1 na straně 23):

- **Načtení zprávy** - pokud je na vstupu mikrokontroléru nějaká zpráva z PC (aplikace) začne se postupně číst. Odpovídá-li první znak zprávy úvodnímu znaku „:“, je přečten další znak určující účel (typ / povel) zprávy a případná datová číselná hodnota až do načtení ukončovacího znaku „#“.
- **Zpracování dat / vykonání příkazu** - po přečtení celé (dříve zmíněné) zprávy začne její skutečné zpracování / vykonání dle znaku udávajícího účel zprávy.
- **Poslání měřených hodnot na výstup** - pokud není zpracovávána zpráva z PC, tak je přečtena naměřená hodnota z akcelerometru, převedena do vhodného tvaru a odeslána do PC.

## 5.3 Testování přenosové rychlosti dat

Požadovaný počet vzorků za vteřinu (rychlost měření) je nastavitelný příkazem „S“ z aplikace (PC). Existuje však maximální počet přenesených a tedy aplikací přijatých zpráv obsahujících vzorky naměřené akcelerometrem ze zařízení do aplikace na PC za vteřinu. Tento počet je ovlivněn především komunikační přenosovou (baudovou) rychlostí (počet bitů za



Obrázek 5.1: Grafické znázornění principu zpracování dat z aplikace či akcelerometru.

vteřinu), rychlostí čtení dat z pohybového senzoru a samozřejmě v poslední řadě efektivností zápisu dat v přenášené zprávě. Tedy významně záleží na tom, v jak stručném tvaru se hodnoty naměřené senzorem zasílají ze zařízení do aplikace na PC.

Maximální počet zpráv za vteřinu byl měřen v pomocném kódu umístěném přímo ve sketchi pro vyčítání dat z akcelerometru a současně v rehabilitační aplikaci. Rychlost měření, daná nastavitelnou pauzou po každém přečtení hodnot ze senzoru, byla nastavena na několiknásobně vyšší hodnotu, než je teoretická maximální dosažitelná rychlost měření uváděná výrobcem akcelerometru. Počet vzorků k naměření a odeslání ze zařízení byl nastaven na 10 000 vzorků. Rehabilitační aplikace při přijetí prvního a desetitisícátého vzorku vždy vypsalá aktuální absolutní čas. Maximální počet vzorků přijatých rehabilitační aplikací za vteřinu byl poté jednoduše spočítán jako podíl počtu přenesených vzorků a času (ve vteřinách) uplynulého mezi přijetím prvního a posledního vzorku. Toto měření proběhlo opakovaně a naměřené časy se aritmeticky zprůměrovaly. Výsledek měření maximálního počtu vzorků za vteřinu naleznete přehledně v tabulce 5.3 na straně 24.

<b>počet vzorků</b>	<b>průměrná uplynulá doba</b>	<b>výsledná průměrná rychlost</b>
10 000	83,23 s	<b>120 vzorků/s</b>

Tabulka 5.3: Výsledek měření maximálního počtu zpráv (obsahující vzorky naměřené akcelerometrem) přijatých aplikací za vteřinu.



## Kapitola 6

# Zpracování naměřených hodnot v aplikaci

Naměřené hodnoty (akcelerometrem) odeslané ze zařízení do aplikace v PC vstupují nejdříve do části programu starající se o jejich vyzvednutí ze zprávy, poté vykonání některých základních operací jako je průměrování a nakonec do části poskytující tyto hodnoty v požadované podobě jednotlivým rehabilitačním úlohám. Tato část programu je rozdělena na dvě třídy, základní třídu „PohyboMetrBase“ a třídu „PohybMetrArduino“ určenou konkrétně pro přístup k prototypu zařízení s akcelerometrem MMA7455. Rozdělení těchto tříd je zamýšleno tak, aby vedle třídy „PohybMetrArduino“ mohla být obdobná třída pro jiné zařízení (využívající jiný pohybový senzor).

Ve třídě „**PohyboMetrBase**“ je (konkrétněji na obrázku 6.1 na straně 26):

- konstruktor s přiřazením konkrétní třídy pro přístup k zařízení,
- definice obecných proměnných typu enum (výčet) používaných kterýmikoliv případnými třídami zařízení,
- rozhraní s metodami k implementaci ve třídě pro konkrétní zařízení se senzorem.

Třída „**PohybMetrArduino**“ obsahuje:

- nastavení komunikace mezi zařízením a aplikací, tentokrát na straně PC,
- přijímání zpráv obsahující naměřené hodnoty a jejich poskytnutí v požadované podobě jednotlivým rehabilitačním úlohám.

Nastavení komunikace na PC obsahuje samozřejmě stejné parametry jak komunikace na straně měřícího zařízení. Přijímané znaky jsou načteny až po ukončovací znak „#“, načte se tedy vždy celá zpráva najednou. Zpráva s naměřenými hodnotami obsahuje (nepočítáme zmiňované tzv. organizační znaky „:“, „D“, „#“) tři celočíselné hodnoty uvozené názvem jednotlivých os, které daná hodnota přísluší. Zpráva je v následujícím formátu:

: Dx = -1, y = 14, z = 51#

```

public class PohyboMetrBase
{
    // trida pro pristup k zarizeni
    public static IPohyboMetr iPohyboMetr = new AkcelerometrHW.PohybMetrArduino();

    static PohyboMetrBase() { iPohyboMetr.Open(); }
}

// enum orientaci zarizeni (dle umisteni zarizeni na tele)
public enum EnumOrientation { Hor, Ver } // horizontalni a vertikalni orientace

// vytvoreni objektu reprezentujiciho referenci na metodu pro predani namerenych hodnot
public delegate void DelegateOnDataFromDeviceNew(double osaX, double osaY, double osaZ,
    EnumPlatnostUhlu valid);

// enum (bitmask) pro platnosti hodnot uhlu v danych osach
// (prekracuje-li namerena hodnota mezni uhel naklonu, je prohlasena za neplatnou)
public enum EnumPlatnostUhlu
{
    PlatnyZadny = 0,    // zadna z hodnot pro danou osu není platna
    PlatnyX = 1,       // hodnota pro osu X je platna
    PlatnyY = 2,       // hodnota pro osu Y je platna
    PlatnyZ = 4,       // hodnota pro osu Z je platna
    Otoceno = 8,       // zarizeni je vzhuru nohama
    NoCalib = 16       // zarizeni není zkalibrovano
}

public interface IPohyboMetr
{
    bool Open();        // inicializace komunikace se zarizenim
    bool Start();       // spusteni komunikace se zarizenim
    void Stop();        // zastaveni komunikace se zarizenim
    void Close();       // ukončení komunikace se zarizenim

    void Configure();   // konfigurace zarizeni
    void Calibration(); // kalibrace zarizeni

    // metoda zpracovavajici namerenou hodnotu ve stupnich
    DelegateOnDataFromDeviceNew OnDataStupne { set; }
    // metoda zpracovavajici namerenou hodnotu v intervalu <-1, 1>,
    // dle nastaveni max. rozsahu pohybu pacienta
    DelegateOnDataFromDeviceNew OnData01 { set; }
}

```

Obrázek 6.1: Implementace třídy „PohyboMetrBase“.

Samotné naměřené hodnoty z tohoto znakového řetězce jsou vyzvednuty využitím regulárních výrazů a postupně uloženy do proměnných zastupujících jednotlivé osy. Například regulární výraz

$$(x=)[0-9\-\-]{1,4}(,)$$

z řetězce vyjme 1 až 4 číslice s případným znaménkem „-“ vyskytující se mezi „x=“ a „,“.

## 6.1 Průměrování naměřených hodnot

Naměřené hodnoty se po vyzvednutí z textového řetězce mohou průměrovat. Implementovány jsou čtyři typy průměrování sloužící pro různou míru odfiltrování drobných třesů těla pacienta. Typ průměrování si lze vybrat v základním nastavení aplikace. Implementované typy průměrování jsou:

- „None“ - bez průměrování,
- „Simple“ - prostý aritmetický průměr  $n$  posledních naměřených hodnot,
- „Weighted“ - vážený aritmetický průměr  $n$  posledních naměřených hodnot (novější hodnota má vždy větší váhu),
- „ZoneFilter“ - aritmetický průměr  $n$  posledních naměřených hodnot s tím, že poslední naměřená hodnota se do  $n$  průměrovaných hodnot zahrnuje pouze, pokud je její odchylka od aktuálního průměru větší než nastavená šířka filtrovaného pásma necitlivosti okolo aktuálního průměru.

Počet posledních  $n$  hodnot, které se budou průměrovat lze nastavit v konfiguraci aplikace. Čím je nastaven vyšší počet  $n$ , tím je na jednu stranu pohyb ovládaného předmětu v rehabilitační úloze plynulejší, však na druhou stranu s delší dobou reakce na změnu náklonu zařízení. Vážený aritmetický průměr tuto nežádoucí vlastnost mírně eliminuje. Typ průměrování „ZoneFilter“ je zaměřený na odstranění mírných třesů pacienta, čím je šířka pásma větší, tím větší třesy zařízení ignoruje.

## 6.2 Možné orientace použití měřícího zařízení

Dále se dle aktuálně použité orientace zařízení, vodorovně nebo svisle, prohodí hodnoty jednotlivých os za účelem prohození v osách jisté standardizace směru těchto os. Tato záměna slouží ke sjednocení směru os zařízení se směry os používaných v rehabilitačních úlohách, tedy k nezávislosti na orientaci (vodorovně nebo svisle) zařízení.

Některé rehabilitační úlohy jsou zaměřené na procvičování pohybu náklonu okolo pouze jedné vodorovné osy (dále jednoosé rehabilitační úlohy) nebo okolo dvou vodorovných os současně a to na sebe kolmých (dále dvouosé rehabilitační úlohy). Při procvičování pohybu náklonu okolo jedné osy, se v rehabilitačních úlohách jedná o osu X směřující od pacienta vpřed. Pro procvičování pohybu náklonu okolo dvou os, se jedná o osu X a osu Y, osa X je shodná s osou X u jednoosých rehabilitačních úloh a osa Y směřuje od pacienta vpravo (z pohledu pacienta). Zbývající osa Z v rehabilitačních úlohách směřuje nahoru a v současnosti není žádnou úlohou využita, neboť nelze snadno detekovat otáčení kolem této osy.

Spolu se zamýšlenými způsoby využití zařízení jsou definovány dvě základní orientace tohoto zařízení:

- **horizontální** - osy zařízení směřují shodně s osami v rehabilitačních úlohách,
- **vertikální** - osy zařízení směřují jinými směry, osa X zařízení směřuje vzhůru, osa Y vlevo od pacienta (z pohledu pacienta) a osa Z od pacienta vpřed.

Horizontální orientace prototypu zařízení je zamýšlena pro upevnění zařízení na zápěstí pomocí suchého zipu, nebo jeho držení v ruce před tělem pacienta (USB vývod směrem do rukávu). Vertikální orientace je uvažována pro upevnění zařízení například na hrudník osoby (USB vývod vedoucí dolů).

### 6.3 Stanovení skutečného úhlu náklonu a určení platnosti rozsahu

Následuje převod naměřených hodnot akcelerometrem z vodorovných os (os směřujících vodorovně při nastavení orientace zařízení) na skutečný úhel náklonu. Naměřené hodnoty jsou přečteny dle dříve změřené statické převodní charakteristiky konkrétního použitého akcelerometru. Pro použitý akcelerometr MMA7455 odpovídá statické převodní charakteristice polynom třetího řádu (viz. kapitola 4.2 na straně 19). Zároveň proběhne kontrola platnosti vypočtených úhlů, zkalibrovanost zařízení a zda-li není zařízení převráceno (tzv. vzhůru nohama) a výsledkem těchto kontrol jsou příslušné příznaky (v proměnné typu „EnumPlatnostUhlu“, definice typu „EnumPlatnostUhlu“ naleznete na obrázku 6.1 na straně 26). Úhly jsou platné, pokud jejich absolutní hodnota nepřesáhne povolený mezní úhel, kdy vypočtené úhly ještě dostatečně odpovídají reálnému náklonu zařízení. Zda-li není zařízení otočeno je detekováno z naměřené hodnoty svislé osy. Záporná hodnota odpovídá opačnému směru svislé osy ke směru gravitačního zrychlení, tedy zařízení je v tuto chvíli převráceno (tedy vzhůru nohama).

### 6.4 Předávání hodnot rehabilitačním úlohám

Na závěr se konkrétní rehabilitační úloze poskytnou hodnoty buď jako reálné stupně nebo jako interval  $\pm 1$ . Pro předání úhlu náklonu ve stupních se zkontroluje, není-li výsledný úhel větší než nastavený rozsah senzoru, tedy není-li větší než mezní využívaný úhel senzoru. Pokud vypočtený úhel v určité ose je větší, nastaví se její příznak na neplatný. U předání náklonu v druhé podobě, tedy v rozsahu  $\pm 1$  proběhne navíc převod reálného úhlu náklonu do zmiňovaného intervalu  $\pm 1$  v závislosti na nastaveném aktuálním rozsahu pro převod v konfiguraci aplikace. Pro náklon rovnající se konci povoleného rozsahu bude tedy přiřazena hodnota 1 (případně -1), pro náklon 0 stupňů hodnota 0.

Rehabilitačním úlohám jsou tedy předávány náklony v podobě buď jako hodnota reálných stupňů (základní diagnostické úlohy) nebo jsou převedeny na rozsah  $\pm 1$  (běžné rehabilitační úlohy). Rovněž jsou doplněny příznaky platnosti.

## Kapitola 7

# Implementace programové aplikace

Programová aplikace pro rehabilitaci je psaná v programovacím jazyce C#. Výběr programovacího jazyka pro psaní programů rehabilitačních úloh v rámci bakalářské práce byl dán jazykem, ve kterém byl vytvořen framework pro tvorbu různých typů aplikací poskytnutý vedoucím práce.

### 7.1 Použité technologie

Rehabilitační aplikace je vytvořena za použití technologie WPF v Microsoft .NET Framework 4 (Client Profile).

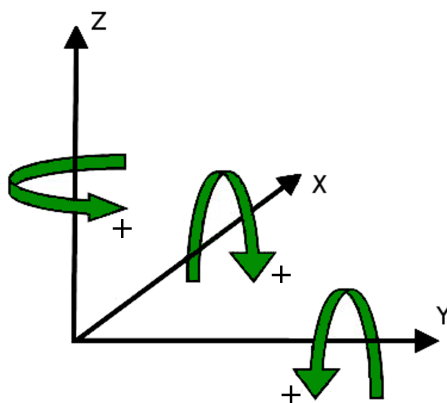
WPF (Windows Presentation Foundation) je podpora pro komplexní tvorbu bohatých formulářových aplikací, součástí .NET frameworku je od verze 3.0. Framework WPF používá značkovací jazyk XAML pro vytvoření uživatelského rozhraní. Nabízí spoustu hotových komponent jako například různá tlačítka, textová pole, výběrová pole, posuvníky, popisky a zároveň možnost tvorby vlastních komponent. Díky XAML jsou od sebe odděleny vzhled a funkčnost aplikace. Všechna grafika (včetně samotných WPF oken) pracuje s podporou Direct3D knihoven. To umožňuje hardwarovou akceleraci pomocí grafické karty a tudíž pokročilejší grafické schopnosti. Kromě WPF je v .NET frameworku stále přítomný starší formulářová podpora nazývaná Windows Forms. Ačkoli Microsoft Windows Forms ještě neoznačil jako zastaralý a v současné době se paralelně používají oba frameworky, WPF je technologicky mnohem dále.[15]

Aplikace by měla podporovat (podle dokumentace Microsoftu) mnoho verzí operačního systému Microsoft Windows (Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1). Aplikace byla vyvíjena a průběžně testována na Windows 7, Windows 8 a Windows 8.1.

### 7.2 Využití os rotací v aplikaci / úlohách

V rehabilitační aplikaci je logika uspořádání os poněkud odlišná od uspořádání os akcelerometru. Orientace os v aplikaci (tzv. virtuálních os) je neměnná (nezávislá na orientaci

měřícího zařízení) a byla zvolena s ohledem na lékaře a pacienty, aby byla intuitivní především pro ně jako uživatele. Osa X směřuje vždy z pohledu uživatele vpřed, osa Y vpravo a osa Z nahoru. Směr rotace okolo těchto os je zvolen tak, aby hodnota náklonu vzrůstala s rotací senzoru po směru hodinových ručiček (při pohledu ve směru osy), viz. obrázek 7.1 na straně 30.



Obrázek 7.1: Osy rotací v rámci rehabilitačních úloh aplikace.

Jelikož zařízení pro rehabilitaci využívá hodnoty z akcelerometru pouze pro určení náklonu, rotace okolo konkrétní osy senzoru (pouze vodorovných os). Z tohoto důvodu jsou hodnoty vystupující ze senzoru v zařízení přerovnány tak, aby byl vždy dodržen zmíněný směr rotací.

### 7.3 Framework pro tvorbu úloh / aplikací

Pro tvorbu rehabilitačních úloh byla vedoucím práce dodána tzv. základní aplikace a prázdná šablona pro tvorbu rehabilitačních úloh zajišťující sjednocený vzhled implementovaných úloh s celou rehabilitační aplikací.

Při spuštění celé rehabilitační aplikace se otevře okno v hlavním menu. Hlavní menu je rozděleno na tři základní části: tlačítka pro tzv. pohyb v rámci aplikace (přepínání mezi úlohami), tlačítka pro nastavení a hlavní zobrazovací část okna aplikace. Tlačítka pro pohyb v rámci aplikace se nachází v levé části hlavního menu aplikace. Patří mezi ně tlačítka pro návrat na tuto úvodní stránku, tlačítka pro informace o pacientu, několik tlačítek pro výběr kategorie rehabilitačních úloh a tlačítka pro zobrazení kontaktů uživatelů / poskytovatelů aplikace a tvůrce aplikace (viz. levá část obrázku 7.2 na straně 31). Tlačítka pro nastavení se nachází v horní části hlavního menu aplikace. Pomocí nich lze zadávat informace o pacientu, měnit nastavení pro rehabilitační úlohy (například aktuální orientace zařízení, rozsah pohybů pacienta, typ průměrování měřených hodnot), kontaktovat vývojáře aplikace či maximalizovat okno aplikace na celou obrazovku (viz. pravá horní část obrázku 7.2 na straně 31).



Obrázek 7.2: Přehled tlačítek v hlavním menu aplikace.

## 7.4 Vytvářené rehabilitační úlohy

Na ukázkou činnosti prototypu zařízení bylo vytvořeno několik základních rehabilitačních úloh. Obecně jsou tyto úlohy rozděleny do 4 kategorií dle zaměření:

- **„Diagnostické“** - Úlohy pro testování správné činnosti měřicího zařízení (senzoru) a základní diagnostiku pohybu rehabilitovaného. V aplikaci je náklon vizualizován rotací okolo virtuálních os X a Y.
  - „**Test**“ - pro kontrolu správné reakce předmětu v aplikaci na pohyby HW zařízení a pro kontrolu správného nastavení orientace zařízení,
  - „**Stabilita**“ - pro přesnější měření náklonu, rotace a sledování stability pacienta.
- **„Pohyb X“** - Úlohy zaměřené na rehabilitaci pohybu otáčením okolo jedné vodorovné osy zařízení. V aplikaci je náklon vizualizován rotací okolo virtuální osy X.
  - „**Chytání**“ - chytání padajících předmětů, zaměřené na dynamiku pohybu a postřeh.
  - „**Tenis**“ - odražení kuličky od ovládané posuvné desky (do barevných kostek, o zeď, se soupeřem), zaměřené na dynamiku pohybu a postřeh.
- **„Pohyb X a Y“** - Úlohy zaměřené na rehabilitaci pohybu otáčení okolo dvou vodorovných na sebe kolmých os. V aplikaci je náklon vizualizován rotací okolo virtuálních os X a Y.
  - „**Plocha**“ - sbírání předmětů na ploše, zaměřené i na rozvoj prostorového vnímání a odhadu důsledku svého pohybu.
  - „**Bludiště**“ - hledání a sbírání předmětů na konci cest v bludišti, zaměřené i na rozvoj prostorového vnímání, hlavolam.
- **„Speciální“** - Úlohy pro cvičení manipulace s předměty, v tomto okamžiku hrníčku na pití. V aplikaci je náklon vizualizován rotací okolo virtuálních os X a Y.
  - „**Hrneček**“ - pro cvičení opatrné manipulace s hrníčkem.

Framework pro tvorbu úloh byl vytvořen tak, aby umožňoval snadné přidávání rehabilitačních úloh, zdrojové kódy jednotlivých úloh jsou v samostatných souborech. Nově implementovaná rehabilitační úloha se do aplikace začlení v podstatě jen přidáním souboru se zdrojovým kódem této nové úlohy, samozřejmě přizpůsobeným tomuto frameworku a ve formátu „\*.cs“) mezi ostatní soubory s kódy jiných úloh.

Podrobnější popis a implementace dvou rehabilitačních úloh je popsána v následujících podkapitolách.

### 7.4.1 Úloha Stabilita

Úloha „Stabilita“ patří mezi základní diagnostické úlohy. Úloha slouží ke grafickému znázornění aktuálního náklonu, v podstatě rotace kolem konkrétní vodorovné osy a ke sledování stability tohoto náklonu pacienta. Měření rotace probíhá okolo dvou vodorovných na sebe kolmých os. Úhel náklonu zařízení je vypisován na obrazovce (ve stedu dolní části) a zároveň znázorňován natočením šipek na číselníku. Náklon je vizualizován pro každou osu zařízení zvlášť (osa X - zelená šipka, osa Y - modrá šipka). Viditelnost šipek lze vypínat a zapínat pomocí tlačítek „Osa X“ a „Osa Y“ v horním menu úlohy. Při překročení maximálního měřitelného náklonu používaného zařízení okolo dané osy příslušná šipka zesvětlá.

Zdrojový kód úlohy se skládá obecně ze tří částí. V první části kódu úlohy „Stabilita“ se připraví a nastaví prostředí potřebné pro úlohu. Do horního menu se vloží tlačítka pro spuštění / zastavení měření hodnot zařízení, zapínání / vypínání viditelnosti ukazatelů náklonu (šipek), kalibraci zařízení a návrat do hlavního menu rehabilitační aplikace. Dále se připraví plátno (canvas) pro vykreslování grafických objektů samotné úlohy. Velikost plátna je nastavena dle velikosti sloupce a řádku globální mřížky frameworku určeného pro úlohu (mřížka obsahuje jeden sloupec a jeden řádek).

V druhé části kódu se vytváří veškeré grafické objekty této úlohy a přidávají, umisťují se na plátno. Nejprve jsou vytvořeny šipky (ukazatele naměřeného úhlu náklonu zařízení). Každá šipka je nadefinována jako polygon charakterizovaný osmi body (poslední bod je na stejném místě jako bod první pro uzavření) s černým obrysem o dané tloušťce a barevnou výplní dle toho, jedná-li se o šipku pro náklon okolo osy X či okolo osy Y.

Následuje tvorba bodu (dále bod natáčení), podle kterého budou všechny objekty umisťovány a natáčeny na plátně. První souřadnici bodu je přiřazena hodnota poloviny velikosti sloupce zmiňované globální mřížky frameworku a druhé souřadnici hodnota tří čtvrtin velikosti řádku mřížky<sup>1</sup>. Bod natáčení je tedy uprostřed v dolní části plátna. V tuto chvíli se i uloží do pomocné proměnné menší rozměr plátna (dále jen velikost plátna), který bude dále používán jako výchozí velikost pro ostatní objekty na plátně. Ukládá se menší rozměr, aby se při jakýchkoliv rozměrech okna aplikace vešly všechny vykreslované objekty na plátno. Dále je vytvořen číselník pro odečet úhlu náklonu. Hodnoty na číselníku jsou od  $-90$  do  $90$  stupňů. Délky čárek na číselníku jsou závislé na úhlu, který představují. Názornou ukázkou určení rozměrů čárek na číselníku, jejich tvorby (tvorba objektu typu „Line“) a umístění na plátno naleznete na obrázku 7.3 na straně 33, určení rozměrů, tvorba a umístění na plátno je principiálně stejná pro všechny objekty této úlohy.

<sup>1</sup>Počáteční bod dvoudimenzionálního souřadného systému ve WPF je v levém horním rohu, osa x směřuje vpravo a osa y dolů.



```

int uhelOd = -90; int uhelDo = 90; // rozsah hodnot na číselníku
double kratka = 0.05; double stredni = 0.06; double dlouha = 0.08; // koeficient zmenšení pro délku čárky
double uzka = 0.003; double široka = 0.01; // koeficient zmenšení pro šířku čárky

// bod podle kterého se otáčí objekty na plátně
Point bodNatacení = new Point();
bodNatacení.X = GridTaskGlobal.ColumnDefinitions[0].ActualWidth / 2; // odpovídá polovině velikosti plátna
bodNatacení.Y = 0.75 * GridTaskGlobal.RowDefinitions[0].ActualHeight; // odpovídá 3/4 velikosti plátna
// velikost kreslicí plochy = menší rozměr okna aplikace
double velikostPlochy = Math.Min(taskGUIItems[EnumNumber.First].canvas.ActualWidth,
    taskGUIItems[EnumNumber.First].canvas.ActualHeight);

// vytvoření všech čárek na číselník
for (int i = uhelOd; i <= uhelDo; i++)
{
    // délka čárky představující úhel je závislá na úhlu, který představuje
    double delkaCarky = 0;
    if ((i % 10) == 0) { delkaCarky = dlouha; } // po 10 stupních - nejdelší čárka
    else if ((i % 5) == 0) { delkaCarky = stredni; } // po 5 stupních - střední čárka
    else { delkaCarky = kratka; } // po 1 stupni - nejkratší čárka

    // vytvoření jedné této čárky
    Line carka = new Line()
    {
        X1 = 0, Y1 = 0, // počáteční bod čárky vždy v [0, 0]
        X2 = 0, Y2 = -delkaCarky * velikostPlochy, // koncový bod čárky podle velikosti kreslicí plochy (plátna)
        Stroke = new SolidColorBrush(Colors.Red), // barva čárky červená
        StrokeThickness = (((i % 10) == 0) ? široka : uzka) * velikostPlochy // tloušťka čárky podle velikosti kreslicí plochy
    };

    // vložení čárky na plátno (canvas)
    taskGUIItems[EnumNumber.First].canvas.Children.Add(carka);

    // nastavení pozice a natočení čárky
    TransformGroup tgCarka = new TransformGroup();
    // posun čárky
    tgCarka.Children.Add(new TranslateTransform() { X = bodNatacení.X, Y = 0.15 * velikostPlochy });
    // natočení čárky okolo bodu natáčení
    tgCarka.Children.Add(new RotateTransform() { CenterX = bodNatacení.X, CenterY = bodNatacení.Y, Angle = i });
    // uložení transformace čárky
    carka.RenderTransform = tgCarka;
}

```

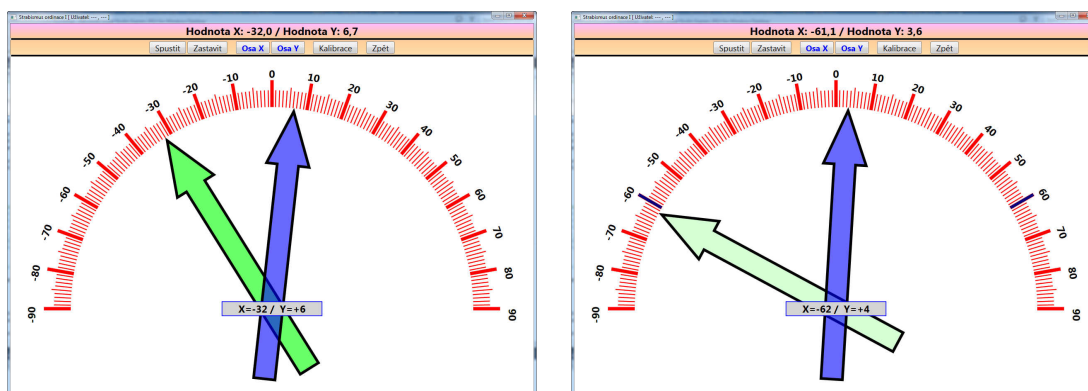
Obrázek 7.3: Ukázka (pseudokód) z úlohy „Stabilita“ - určení rozměrů čárek na číselníku, jejich tvorba (tvorba objektu typu „Line“) a umístění na plátno.

Jako poslední je vytvořeno orámované textové pole pro zobrazení hodnot aktuálních úhlů náklonu. Celý tento zobrazovač je sestaven jako objekt typu „TextBlock“ vložený do objektu typu „Border“ a ten pak ještě vložený do objektu typu „Viewbox“. Objekt „TextBlock“ umožní nastavit text, jeho velikost a zarovnání. Objekt „Border“ dovolí vytvořit okolo textu modrý rámeček o dané tloušťce a světle šedé pozadí textu. Následné vložení tohoto celého ještě do objektu „Viewbox“ slouží pro automatické zvětšení textu na maximální možnou velikost. Popisovaný zobrazovač je umístěn na bod natáčení.

V poslední části kódu je metoda pro příjem a zpracování hodnot ze zařízení dle potřeb konkrétní úlohy. Tato úloha přijímá naměřené hodnoty ve stupních a příznaky udávající aktuální platnosti naměřených úhlů pro jednotlivé osy a stav zařízení (definici výčtu „EnumPlatnostUhlu“ naleznete na obrázku 6.1 na straně 26). Hodnoty náklonů jsou poté jen přiřazeny do lokálních proměnných a následně je zavolána metoda pro nastavení aktuálního natočení šipek a zobrazení hodnoty úhlu náklonu v textovém poli ve středu otáčení šipek. Konkrétní implementaci metody pro příjem a zpracování hodnot ze zařízení naleznete na obrázku 7.4 na straně 34.

```
private void OnDataFromDeviceAngle(double inX, double inY, double inZ, EnumPlatnostUhlu valid)
{
    // vyvolání metody ve vlákně (této) aplikace (aby byl přístup k GUI prvkům)
    taskGUIItems[EnumNumber.First].canvas.Dispatcher.BeginInvoke(new Action(() =>
    {
        // nastavení úhlu natočení šipek
        rotateValueX.Angle = inX;
        rotateValueY.Angle = inY;
        // metoda pro natočení šipek na daný úhel a zobrazení aktuální hodnoty v 'TextBlock'
        ArrowSet();
    }));
}
```

Obrázek 7.4: Implementace metody pro příjem a zpracování hodnot ze zařízení a nastavení šipek na daný úhel.



Obrázek 7.5: Ukázka z úlohy „Stabilita“ (na obrázku vpravo je v rámci ukázkový na číselníku modře zvýrazněn maximální měřitelný náklon zařízení).

### 7.4.2 Úloha Chytání

Úloha „Chytání“ patří do kategorie „Pohyb X“, slouží pro rehabilitaci ve směru rotace, tedy náklonu okolo jedné vodorovné osy. Úloha je v podobě dvoudimenzionální zábavné hry zaměřené i na postřeh pacienta. Cílem hry „Chytání“ je pomocí zvířátka (například myš) či předmětu (například vozík, kyblík) ve spodní části obrazovky ovládaného nakláněním měřícího zařízení chytat padající věci (například sýr, jablko, banán, pytle s penězi). Ukázku z úlohy „Chytání“ naleznete na obrázku 7.8 na straně 37.

V první části zdrojového kódu úlohy „Chytání“ se opět připraví a nastaví prostředí pro úlohu. Nyní se do horního menu vloží tlačítka pro spuštění, zastavení příjmu hodnot ze zařízení, snížení zvýšení úrovně obtížnosti úlohy, kalibraci zařízení a navrácení se do hlavního menu rehabilitační aplikace. A opět se připraví plátno pro vykreslování objektů samotné úlohy.

V druhé části kódu se vytváří grafické objekty této úlohy, přidávají a umisťují se na plátno či se z něj odebírají. Jako první je implementována metoda pro generování padajících objektů. Ta se stará o tvorbu samotného objektu, nadefinování jak bude padat a vložení a odebrání objektu z plátna a seznamu padajících objektů. Padajícím objektům je nastavována náhodná velikost (omezená v poměru k šířce plátna a stupně obtížnosti hry). Souřadnice pozice jsou generovány tak, aby se padající objekt věšel celou svou šířkou na plátno a najel na plátno ze shora plynule (neobjevila se znenadání půlka padajícího objektu nahoře na plátně). Toho je docíleno nastavením x-ové souřadnice na hodnotu větší než je šířka padajícího objektu a zároveň menší než je rozdíl šířky plátna a šířky objektu a y-ové souřadnice na hodnotu výšky padajícího objektu se záporným znaménkem. Padání objektu tedy v podstatě začne z místa umístěného někde nad plátnem (v tuto chvíli objekt ještě není vidět). Animace padání objektu je definována počáteční a koncovou pozicí padajícího objektu a dobou animace. Nakonec je objekt přidán na plátno a do seznamu padajících objektů. V případě této úlohy je x-ová (vodorovná) souřadnice konkrétního padajícího objektu neměnná, všechny objekty padají pouze svisle dolů. V seznamu má objekt přiřazený identifikátor v podobě jména (řetězec znaků). Při zachycení i nezachycení padajícího objektu (jeho aktuální y-ová souřadnice pozice je větší než výška plátna) je objekt ze seznamu odebrán.

Následuje metoda k detekci chycení padajícího objektu. V té se prochází všechny položky ze seznamu padajících objektů a kontroluje se chycení objektu. Tato kontrola probíhá tak, že pokud je aktuální y-ová souřadnice dolní hranice padajícího objektu menší než y-ová souřadnice horní hranice předmětu ovládaného pacientem (padající objekt je výše než ovládaný předmět), pokračuje se v kontrole dalšího objektu. Pokud se ale padající objekt nachází byť i částí na stejné úrovni či níže než horní okraj ovládaného předmětu, proběhne tzv. testování kolize těchto dvou objektů. Kolize je testována na průnik intervalů objekty obsazených x-ových souřadnic (interval y-ové souřadnice od levého do pravého okraje padajícího objektu a ovládaného předmětu). Implementaci (pseudokód) testování kolize naleznete na obrázku 7.6 na straně 36. Při zachycení padajícího objektu je přehrán vhodný zvuk a objekt je okamžitě odebrán z plátna i ze seznamu padajících objektů.

A v poslední části je opět metoda pro příjem a zpracování hodnot ze zařízení dle potřeb této úlohy. Tato úloha oproti předešlé přijímá naměřené hodnoty v intervalu  $[-1, 1]$  a příznaky udávající aktuální platnosti naměřených úhlů a stav zařízení (definici výčtu

```

int i = pocetPadajicich;    // počet padajících objektů

// dokud je počet padajících objektů netestovaných na kolizi > 0
while (--i) >= 0
{
    // kontrola zda-li může dojít ke kolizi objektů (padajícího a ovládaného):
    // je-li dolní kraj padajícího objektu výše než horní kraj ovládaného předmětu
    // (hodnoty na ose y rostou směrem dolů)
    if (dolniPadajiciho[i] < horniOvladaneho)
    {
        continue; // ke kolizi nemůže dojít --> vykonávání těla smyčky pro tento objekt ukončeno
                  // pokračuje se testováním dalšího padajícího objektu
    }
    // Ke kolizi objektů může dojít --> kontrola zda-li došlo ke kolizi:
    // je-li levý nebo pravý kraj padajícího objektu mezi levým a pravým krajem ovládaného předmětu
    if (((levyPadajiciho[i] > levyOvladaneho) && (levyPadajiciho[i] < pravyOvladaneho))
        ||
        ((pravyPadajiciho[i] > levyOvladaneho) && (pravyPadajiciho[i] < pravyOvladaneho)))
    {
        padajiciChycen = true; // padající objekt chycen
    }
}

```

Obrázek 7.6: Ukázka (pseudokód) z úlohy „Chytání“ - testování kolize padajících objektů s předmětem ovládaným pacientem.

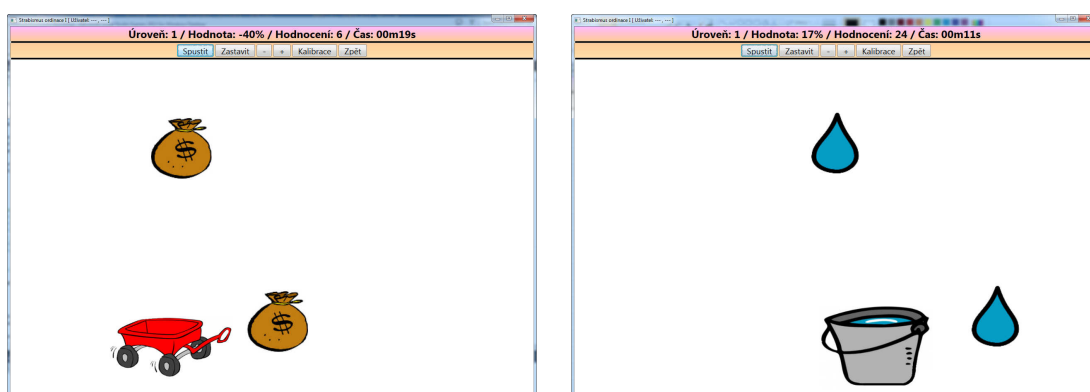
„EnumPlatnostUhlu“ naleznete na obrázku 6.1 na straně 26). Aktuální náklon zařízení je do tohoto intervalu přepočítán samozřejmě dle nastaveného maximálního úhlu náklonu pacienta podle konfigurace aplikace. Převod úhlu do intervalu  $[-1, 1]$  slouží pro zobecnění zpracování hodnot v rámci každé úlohy pro kterou nejsou vyhovující naměřené náklony ve stupních. V této metodě pro příjem a zpracování hodnot ze zařízení se vypočte pozice pacientem ovládaného předmětu na plátně. A to přičtením součinu poloviny šířky plátna a přijaté hodnoty v intervalu  $[-1, 1]$  k polovině šířky plátna. X-ová souřadnice polohy objektu je tedy posunuta od středu šířky plátna v závislosti na hodnotě (v tomto případě koeficientu) ze zmiňovaného intervalu. Pro hodnotu 0 je tedy objekt umístěn do středu spodní části plátna (vertikální umístění objektu je v této úloze neměnné, objekt se pohybuje pouze v ose x), pro hodnotu  $-1$  respektive 1 je objekt na levém respektive na pravém kraji plátna. Konkrétní implementaci metody pro příjem a zpracování hodnot ze zařízení a zmiňovaného výpočtu x-ové souřadnice pozice ovládaného objektu naleznete na obrázku 7.7 na straně 36.

```

private void OnDataFromDevice11(double inX, double inY, double inZ, EnumPlatnostUhlu valid)
{
    // vyvolání metody ve vlákne (této) aplikace (aby byl přístup k GUI prvkům)
    taskGUIItems[EnumNumber.First].canvas.Dispatcher.BeginInvoke(new Action(() =>
    {
        // výpočet pozice (x-ová souřadnice) ovládaného objektu na plátně
        poziceX = taskGUIItems[EnumNumber.First].canvas.ActualWidth / 2 // polovina šířky plátna
                + (taskGUIItems[EnumNumber.First].canvas.ActualWidth / 2) * inX;
        // posun z poloviny šířky plátna
        // dle přijaté hodnoty [-1, 1]
    }));
}

```

Obrázek 7.7: Implementace metody pro příjem a zpracování hodnot ze zařízení a výpočtu x-ové souřadnice pozice ovládaného objektu.



Obrázek 7.8: Ukázka z úlohy „Chytání“.



## Kapitola 8

# Popis použití prototypu pro rehabilitaci

Vytvořený prototyp zařízení je určen pro experimentální využití při rehabilitaci jemné motoriky zejména rukou. Cílem tohoto zařízení je využití především při rehabilitaci v domácím prostředí.

### 8.1 Části zařízení

Prototyp HW zařízení se skládá z:

- měřicího zařízení s páskem na připevnění k tělu pacienta,
- USB kabelu (typ A ↔ typ B) pro připojení k osobnímu počítači.

### 8.2 Požadavky na operační systém osobního počítače

Rehabilitační aplikace je vytvořena za použití Microsoft .NET Framework 4 (Client Profile) a měla by podporovat tyto verze operačního systému Microsoft Windows:

- Windows XP,
- Windows Vista,
- Windows 7,
- Windows 8, Windows 8.1.

Pro spuštění aplikace je tedy potřeba mít nainstalován Microsoft .NET Framework 4 (Client Profile) a pro komunikaci s externím měřicím zařízením USB ovladač od firmy FTDI.

### 8.3 Zapnutí/vypnutí zařízení

Zařízení se v podstatě zapíná pouhým připojením USB kabelu do osobního počítače. Zapnutí zařízení je indikováno rozsvícením zelené kontrolky (LED diody) na zařízení. Vypnutí zařízení opět pouze vytažením USB kabelu z počítače.

### 8.4 Významy světelných signálů kontrolky na zařízení

Na zařízení jsou umístěny tři kontrolky pro indikaci jeho stavu (zapnutí, inicializace z PC, přenos dat). Podrobný popis světelných signálů jednotlivých kontrolky naleznete níže v tabulce 8.1 na straně 40.

barva	význam	druhy signalizace	podrobnější popis
zelená	napájení	zapnuto/vypnuto	Svítili kontrolka, zařízení je zapnuté. Naopak je-li kontrolka zhasnutá, zařízení je bez napájení.
modrá	data	posílání naměřených hodnot ze zařízení spuštěno/zastaveno	Kontrolka se střídavě rozsvicuje a zhasíná se stejnou frekvencí jako je frekvence posílání dat ze zařízení.
červená	stav	potvrzení nastavení komunikace s PC	Po dokončení nastavení komunikace kontrolka 5x za sebou krátce blikne.
		potvrzení přijetí příkazu z aplikace	Při přijetí příkazu z aplikace kontrolka 2x za sebou krátce blikne.
		zařízení zaneprázdněné	Kontrolka svítí v průběhu, kdy zařízení vykonává příkaz z aplikace.

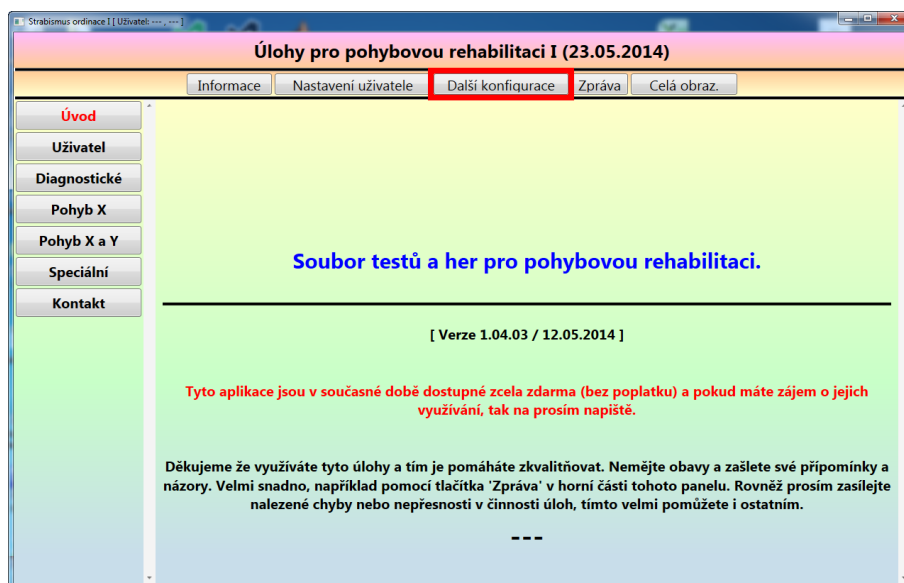
Tabulka 8.1: Významy světelných signálů jednotlivých kontrolky na zařízení.

### 8.5 Základní konfigurace aplikace

V nastavení rehabilitační aplikace jsou dostupné některé konfigurace společné pro všechny rehabilitační úlohy. Jedná se o nastavení základní orientace zařízení při používání (vodorovně nebo svisle), povolený úhlový rozsah zařízení, rozsah pohybu pacienta s převodem na  $\pm 1$ , typ průměrování a počet průměrovaných hodnot.

Dialog pro nastavení vyvoláte kliknutím na tlačítko „Další konfigurace“ na základním dialogu aplikace (viz. obrázek 8.1 na straně 41). Nacházíte-li se ve hře, je potřeba nejdříve hru opustit tlačítkem „Zpět“ vpravo nahoře.

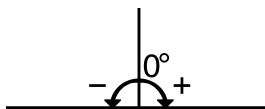




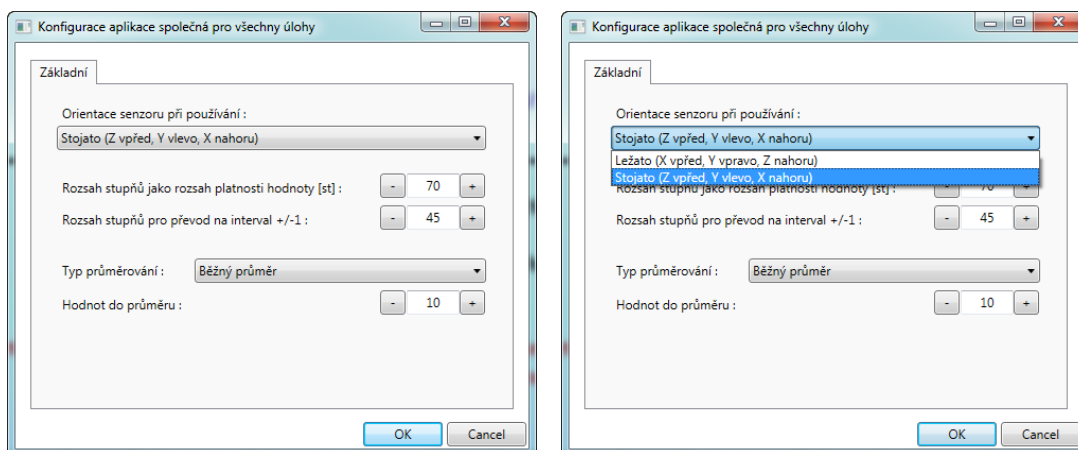
Obrázek 8.1: Hlavní menu rehabilitační aplikace (zvýrazněné tlačítko pro nastavení).

Na tomto dialogu lze nastavit (viz. obrázek 8.2 na straně 42):

- „**Orientace senzoru při používání**“ - základní poloha zařízení dle připevnění na tělo pacienta. Orientace zařízení a jak orientaci určit naleznete v bodu 8.6 na straně 42.
- „**Rozsah stupňů jako rozsah platnosti hodnoty [st]**“ - maximální skutečně měřitelný náklon používaného zařízení ve stupních (definován konkrétním zařízením, přesněji řečeno použitým senzorem).
- „**Rozsah stupňů pro převod na interval  $+/-1$** “ - maximální rozsah pohybu (rotace, náklon) pacienta ve stupních. Při pohledu po směru dané osy se úhel měří od svislice po/proti směru hodinových ručiček. Po směru hodinových ručiček je kladná hodnota úhlu, proti směru hodinových ručiček je záporná hodnota úhlu (viz. obrázek níže).



- „**Typ průměrování**“ - typ průměrování posledních  $n$  naměřených hodnot. Typy průměrování a jejich popis naleznete v bodu 8.7 na straně 42.
- „**Hodnot do průměru**“ - počet posledních  $n$  hodnot, které jsou zařazovány do průměrování. Doporučené počty hodnot do průměru naleznete opět v bodu 8.7 na straně 42.



Obrázek 8.2: Dialog pro základní konfiguraci aplikace.

## 8.6 Orientace zařízení

V současnosti jsou podporovány dvě základní orientace zařízení:

- **vodorovně** - určeno pro připevnění zařízení na zápěstí pacienta,
- **svisle** - určeno pro připevnění zařízení na hrudník pacienta.

Osy na zařízení směřují stejnými směry jako uvádějí popisky v nabídce nastavení orientace zařízení (viz. obrázek 8.2 vpravo na straně 42).

Po nastavení orientace zařízení či po každé změně umístění zařízení na těle pacienta je nutné zařízení v aktuální orientaci / poloze kalibrovat!

**Postup kalibrace zařízení:** Zařízení připevněte požadovaným způsobem na tělo pacienta, osa směřující dopředu a do strany by měla být co nejvíce vodorovně. Klikněte na tlačítko „Kalibrace“, které najdete v každé rehabilitační úloze v horním menu. Snažte se, aby se se zařízením po dobu kalibrace (zhruba po dobu 2 vteřin od stisku tlačítka „Kalibrace“) nehýbalo.

## 8.7 Typy průměrování

Pro různou míru odfiltrování drobných třesů ruky / těla pacienta jsou implementovány čtyři typy průměrování:

- „**Žádné průměrování**“ - bez průměrování. Velmi citlivé ovládání.
- „**Běžný průměr**“ – běžný aritmetický průměr  $n$  posledních naměřených hodnot. Citlivé ovládání, jemná korekce třesů těla. Doporučený počet hodnot do průměru: 10.

- **„Váhový průměr“** - vážený aritmetický průměr  $n$  posledních naměřených hodnot (poslední naměřená hodnota má největší váhu). Středně citlivé ovládání, střední korekce třesů těla.  
Doporučený počet hodnot do průměru: 10.
- **„Filtr třesů“** - aritmetický průměr  $n$  posledních naměřených hodnot s tím, že poslední naměřená hodnota se do  $n$  průměrovaných hodnot zahrnuje pouze tehdy, pokud je její odchylka od aktuálního průměru větší než nastavená šířka filtrovaného pásma okolo aktuálního průměru. Středně citlivé ovládání, vyšší korekce třesů těla.  
Doporučený počet hodnot do průměru: 10.

Čím méně hodnot do průměru, tím citlivější ovládání a kratší prodleva změny náklonu ovládaného předmětu v aplikaci.

## 8.8 Dostupné rehabilitační úlohy

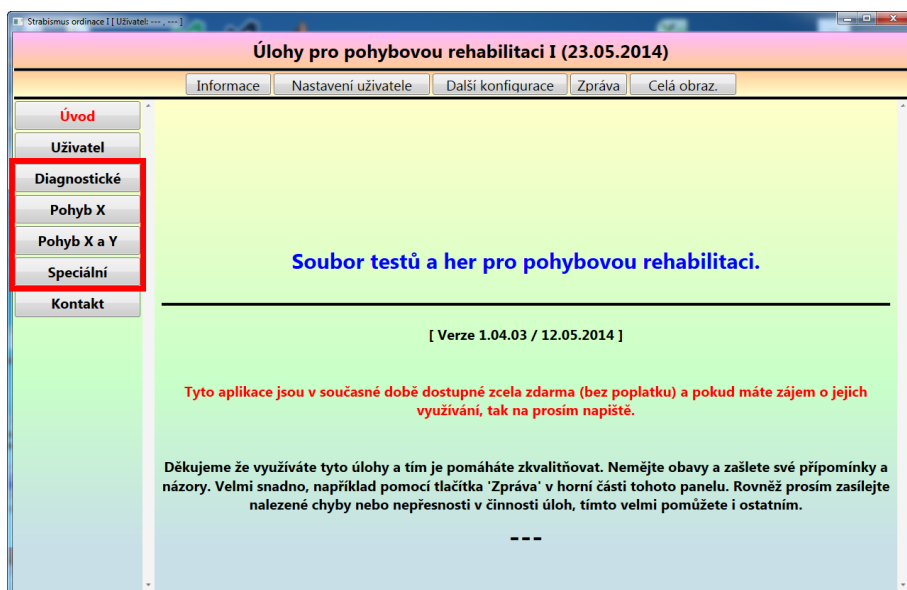
Jednotlivé rehabilitační úlohy jsou zaměřené na procvičování pohybu, přesněji řečeno náklonu okolo pouze jedné vodorovné osy nebo okolo dvou vodorovných na sebe kolmých os současně. Při procvičování pohybu náklonu okolo pouze jedné osy, se v rehabilitačních úlohách jedná o osu X směřující od pacienta vpřed. Pro procvičování pohybu náklonu okolo dvou os, se jedná o osu X a osu Y, osa X je shodná s osou X u jednoosých rehabilitačních úloh a osa Y směřuje od pacienta vpravo (z pohledu pacienta). Osa Z v rehabilitačních úlohách směřuje nahoru a zatím není využita. Tyto zmiňované osy nemusí být směřovány shodně jako osy na zařízení, jedná se o tzv. virtuální osy udávající orientaci v aplikaci.

Rahabilitační úlohy jsou rozděleny do 4 kategorií dle zaměření:

- **„Diagnostické“** - úlohy pro testování správné činnosti měřícího zařízení a základní diagnostiku pohybu rehabilitovaného.
- **„Pohyb X“** - úlohy zaměřené na rehabilitaci pohybu otáčením okolo jedné vodorovné osy zařízení.
- **„Pohyb X a Y“** - úlohy zaměřené na rehabilitaci pohybu otáčení okolo dvou vodorovných na sebe kolmých os zařízení.
- **„Speciální“** - úloha pro cvičení manipulace s předměty, v tomto okamžiku hrníčku na pití.

Jednotlivé úlohy jsou podrobněji popsány v následujících bodech dokumentace. Pro spuštění konkrétní úlohy klikněte v hlavním menu aplikace na příslušnou kategorii (viz. obrázek 8.3 na straně 44).

Všechny rehabilitační úlohy lze pozastavit a opět znovu spustit kliknutím na tlačítko „Zastavit“ a „Spustit“ v horním menu každé úlohy.



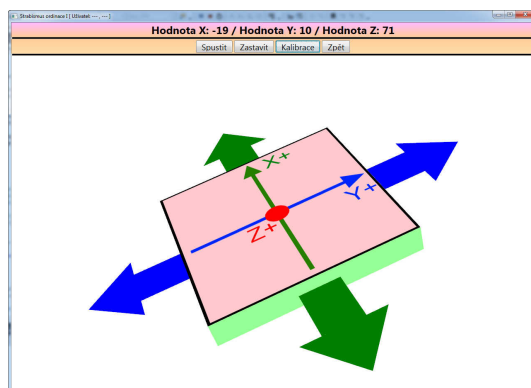
Obrázek 8.3: Hlavní menu rehabilitační aplikace (zvýrazněná tlačítka různých kategorií úloh).

### 8.8.1 Základní diagnostické úlohy

Obě testovací a diagnostické úlohy slouží pro grafické znázornění aktuálního náklonu, tedy rotace pacienta okolo dvou vodorovných na sebe kolmých os.

#### 8.8.1.1 Úloha Test

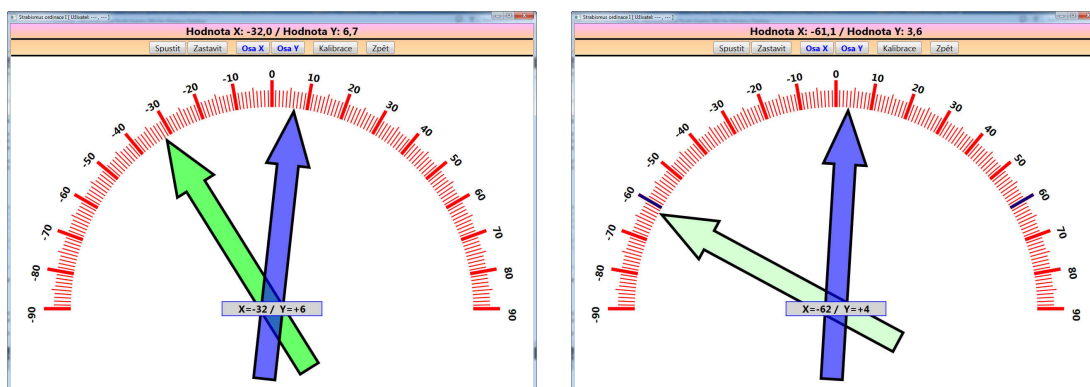
V úloze „Test“ lze pohyby pacienta pozorovat stejným způsobem jako v reálu. Úloha je spíše pro testování / kontrolu správné reakce předmětu v aplikaci na pohyby měřícího zařízení a pro kontrolu správného nastavení orientace zařízení.



Obrázek 8.4: Úloha „Test“.

### 8.8.1.2 Úloha Stabilita

Úloha „Stabilita“ je určena ke konkrétnějšímu měření náklonu, tedy rotace kolem jedné z vodorovných os a ke sledování stability tohoto náklonu pacienta. Úhel náklonu zařízení je vypisován na obrazovce a zároveň znázorňován šipkami na číselném ciferníku. Úhel náklonu je vizualizován pro každou osu zařízení zvlášť (osa X - zelená šipka, osa Y - modrá šipka). Viditelnost šipek lze vypínat a zapínat pomocí tlačítek „Osa X“ a „Osa Y“ v horním menu úlohy. Při překročení maximálního, dostatečně přesně, měřitelného náklonu používaného zařízení okolo dané osy příslušná šipka zesvětlá.



Obrázek 8.5: Úloha „Stabilita“ (na obrázku vpravo na číselném ciferníku modře zvýrazněn maximální měřitelný náklon zařízení).

## 8.8.2 Úlohy pro rehabilitaci pohybu / otáčení okolo jedné osy

Pro rehabilitaci otáčení okolo jedné osy jsou určeny 2D zábavné hry zaměřené i na postřeh pacienta.

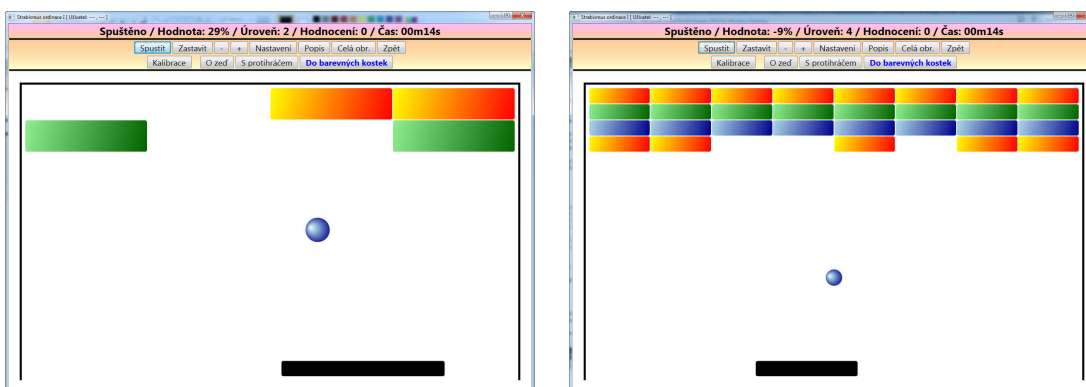
### 8.8.2.1 Úloha Tenis

V úloze „Tenis“ se opět v podstatě chytá padající předmět, nyní však předmět nepadá pouze svisle dolů, ale pohybuje se i šikmo. Padajícím předmětem je nyní malá kulička (míč), která je odražena deskou pohybující se opět po spodní straně obrazovky a ovládanou pohybem pacienta (pohyb ve smyslu rotace / otáčení zařízení, nikoli jeho posuvu). Dle zvolené varianty hry se kulička odráží zároveň i od stěn na bocích a předmětů v horní části hry. Varianty hry jsou:

- **„O zed“** - Kulička se odráží pouze od pacientem ovládané desky a nepohyblivých zdí na stranách a nahoře. Cílem této verze hry je neupustit kuličku (kuličku deskou vždy odrazit).
- **„S protihráčem“** - Kulička se odráží od pacientem ovládané desky, nepohyblivých zdí na stranách a od desky spoluhráče (počítače) pohybující se nahoře. Cílem této verze hry je opět neupustit kuličku.

- „Do barevných kostek“ - Kulička se odráží od pacientem ovládané desky, nepohyblivých zdí na stranách a od barevných kostek poskládaných u stropu. Nárazem kuličky do barevné kostky se kostka sebere (kostka zmizí). Cílem této verze hry je sesbírat všechny barevné kostky a udržet kuličku ve hře.

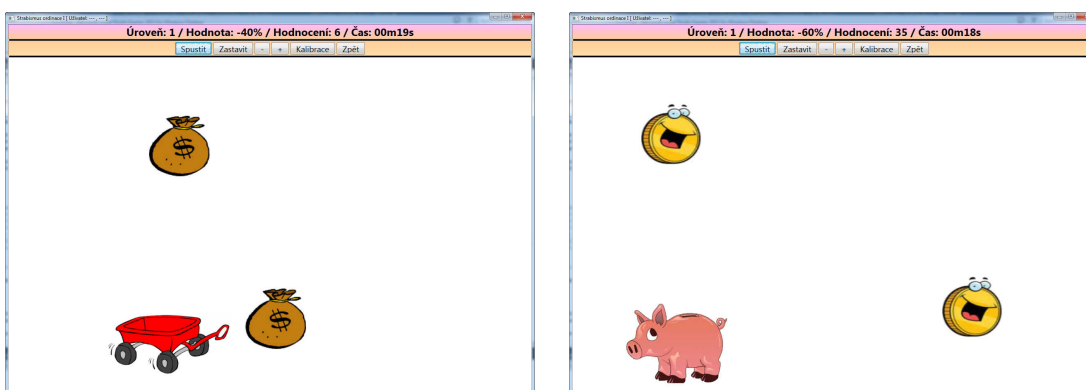
Obtížnost hry lze nastavit kliknutím na tlačítko „-“ či „+“ v horním menu hry. Tlačítkem „-“ obtížnost hry snižujeme, předměty ve hře se zvětšují. Naopak tlačítkem „+“ obtížnost hry zvyšujeme a předměty ve hře se zmenšují, hra je tedy náročnější.



Obrázek 8.6: Úloha „Tenis“ (vlevo nižší obtížnost, vpravo vyšší obtížnost hry).

### 8.8.2.2 Úloha Chytání

Cílem úlohy „Chytání“ je pomocí zvířátka (myš) či předmětu (vozik, kyblík) na spodu obrazovky ovládaného nákláněním zařízení, chytat padající věci například sýr, jablko, banán, pytle s penězi.



Obrázek 8.7: Úloha „Chytání“.

### 8.8.3 Úlohy pro rehabilitaci otáčení okolo dvou os

Pro rehabilitaci otáčení okolo dvou os současně jsou vhodné 3D hry zaměřené na prostorový pohyb pacienta a na rovnováhu.

#### 8.8.3.1 Úloha Plocha s kuličkou

V úloze „Plocha“ pacient nakláněním měřicího zařízení pohybuje buď přímo cílovým předmětem (kuličkou), nebo pouze naklání zobrazenou plochu na obrazovce a tím pohybuje kuličkou či zvířátkem po ploše, pomocí kluzného pohybu v podstatě z kopce a sbírá různé věci či se snaží dostat na cílový červený čtverec.



Obrázek 8.8: Úloha „Plocha“ (vlevo režim „Pohyb postavou“ na nižší obtížnost, vpravo režim „Pohyb deskou“ na vyšší obtížnost hry).

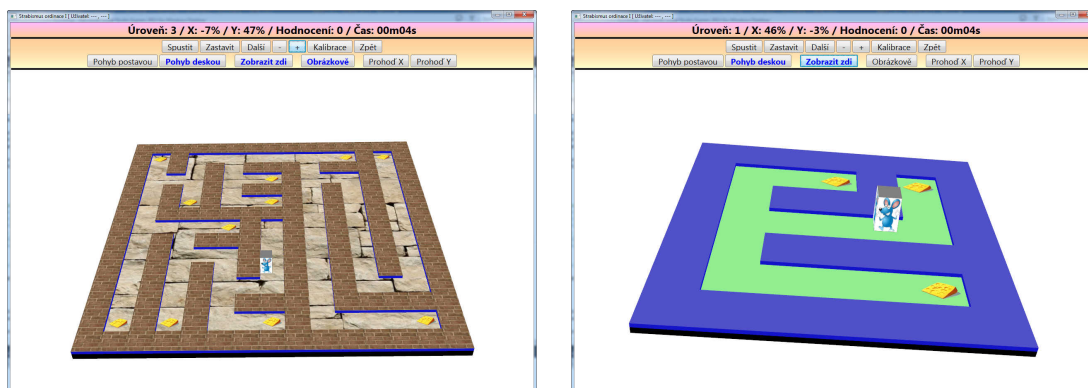
Hra „Plocha“ má dva režimy:

- **„Pohyb postavou“** - Pacient nakláněním měřicího zařízení pohybuje přímo kuličkou či zvířátkem po ploše, plocha na obrazovce se nenaklání. Pozice na ploše je úměrná náklonu zařízení v jednotlivých osách.
- **„Pohyb deskou“** - Pacient nakláněním zařízení naklání plochu, po které se posouvá kulička či sklouzává zvířátko (jako by z kopce).

Režim hry se nastaví kliknutím na tlačítko „Pohyb postavou“, „Pohyb deskou“ v horním menu hry. Opět lze nastavit i obtížnost hry kliknutím na tlačítka „-“, „+“. Čím vyšší obtížnost hry, tím větší plocha. Kliknutím na tlačítko „Obrázkově“ se přepne ovládaný předmět z kuličky na zvířátko.

### 8.8.3.2 Úloha Bludiště

Pohyb v úloze „Bludiště“ i její režimy a obtížnosti jsou principiálně shodné s úlohou „Plocha“. Jediným rozdílem je, že bludištěm prochází jen zvířátka a hledají (sbírají) jídlo.



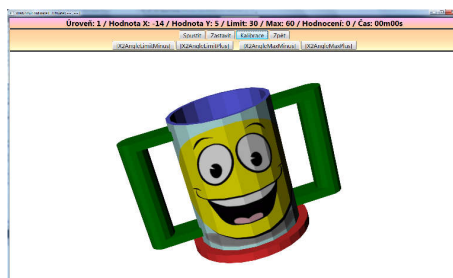
Obrázek 8.9: Úloha „Bludiště“ (vlevo režim „Pohyb postavou“ na vyšší obtížnost, vpravo režim „Pohyb deskou“ na nižší obtížnost hry).

### 8.8.4 Speciální úlohy

Speciální úlohy jsou zaměřené na cvičení činností z běžného života a cvičení soběstačnosti pacienta.

#### 8.8.4.1 Úloha Hrneček

Úloha „Pití z hrnku“ slouží ke cvičení manipulace s hrníčkem. Cíl této úlohy spočívá v naučení pacienta vzít hrníček do ruky, zvedat hrníček bez jeho velkého naklonění a u úst hrníček mírně naklonit. Při větším naklonění hrníčku se začne ozývat varovný zvukový signál a ten sílí tím více, čím více je hrníček nakloněn. Tato skutečnost má pacientu signalizovat překročení povoleného náklonu hrníčku (možnost vylití jeho obsahu) a tudíž nutnost jeho vrácení zpět do svislé polohy.



Obrázek 8.10: Úloha „Hrneček“.



## Kapitola 9

# Poznatky z realizace a případná vylepšení

Při realizaci prototypu popisovaného zařízení pro rehabilitaci jemné motoriky zejména rukou byly zaznamenány následující poznatky, které by v další verzi prototypu / zařízení bylo vhodné zohlednit a implementovat:

- Využít senzor poskytující větší přesnost ve větším rozsahu měření úhlu. U senzoru v tomto prvním prototypu zařízení bylo dosaženo maximálního měřitelného / rozeznatelného úhlu náklonu devadesát stupňů, ale pro skutečně použitelnou hodnotu úhlu náklonu odpovídající reálnému náklonu zařízení ve stupních bylo toto maximum dále omezeno na hodnotu pouze šedesáti stupňů kvůli malé strmosti statické převodní charakteristiky použitého akcelerometru pro náklony blížíící se pravému úhlu.
- Přenos naměřených hodnot z měřícího zařízení do aplikace vytvořit v binárním tvaru, místo současného v ASCII-HEX tvaru. ASCII-HEX tvar je na jednu stranu mnohem vhodnější na ladění komunikace a hledání případných chyb, ale na druhou stranu není optimální z hlediska využití komunikačního kanálu.

Vhodným vylepšením prototypu zařízení by bylo částečné zautomatizování některých nastavení, aby jej nemusel uživatel vykonávat ručně a tím rozšířit použitelnost zařízení i pro pacienty například se svalovými tiky. Jedná se například o tyto možnosti:

- Mezní úhel daného zařízení (podle parametrů použitého senzoru) vyčítat diagnostickou / rehabilitační aplikací přímo z měřícího zařízení, kde bude tento úhel uložen jako konstanta a tím odstranit nutnost jeho zadávání uživatelem ručně.
- Automaticky zvětšovat nastavený maximální rozsah pohybu pacienta při hraní hry podle jeho aktuálních schopností. Dostává-li se pacient opakovaně přes dříve nastavenou / limitní hranici úhlu, tak se tato mez automaticky zvýší, aby se přizpůsobila zvětšenému maximálnímu rozsahu jeho pohybu.
- Implementovat typ průměrování/filtr zaměřený na odstranění nežádoucích tiků pacienta, nechtěné záškuby končetin (v tomto případě nutnost detekce záškubu a jeho odfiltrování).



# Kapitola 10

## Závěr

Cílem bakalářské práce bylo zhodnotit současné možnosti domácí rehabilitace jemné motoriky, na tomto základě nejprve navrhnout a poté vytvořit prototyp především finančně nenáročného zařízení pro rehabilitaci jemné motoriky zejména rukou v tomto domácím prostředí. Současně vytvořit několik ukázkových rehabilitačních úloh využívajících tohoto vytvořeného prototypu zařízení.

Při sestavování prototypu zařízení byla využita vývojová open-source platforma Arduino, která je pro tento účel velmi vhodná. Jako senzor byl použit dobře dostupný tříosý kapacitní akcelerometr MMA7455 sloužící k měření statického gravitačního zrychlení, neboli síly vzniklé působením zemské gravitace, a to pro určení náklonu zařízení. Pro vlastní rehabilitaci byly v programovacím jazyce C# za použití technologie WPF v Microsoft .NET Framework 4 implementovány některé rehabilitační úlohy jak přímo pro skutečnou rehabilitaci, tak i pro základní diagnostiku pohybu rehabilitovaného. V průběhu řešení se na návrhy rehabilitačních terapeutů do práce zahrnula navíc i možnost částečné rehabilitace pohybu těla.

Prototyp zařízení je koncipován jako ověřený návrh zařízení pro vytvoření skutečného výrobku a pro zapůjčení lékařům k dlouhodobějšímu testování na některém rehabilitačním pracovišti. Na rehabilitačních odděleních jsou pacienti nejčastěji děti s dětskou mozkovou obrnou, starší lidé po mozkových příhodách a v současné době stále častěji lidé po různých nehodách a úrazech (kolo, motocykl, automobil), kdy byla částečně narušena komunikace mezi mozkiem a nervovými zakončeními v těle. Z tohoto účelu byly rehabilitační úlohy vytvořeny formou, pokud možno, zábavných her. Součástí práce je tedy i popis / manuál pro ovládání celého vytvořeného zařízení.

Výsledkem práce tedy není konečný výrobek, ale ověření navrženého a použitého HW a SW řešení pro tyto účely a zejména možnost vytvořit takovéto zařízení skutečně i pro domácí prostředí. Jedná se tedy pouze o prototyp, který by měl být po určité době experimentálního používání lékaři a pacienty doveden do fáze skutečného výrobku, což bylo hlavní náplní této práce.

Jelikož se jedná o velmi rychle se vyvíjející a měnící zařízení, tak přiložené CD obsahuje, mimo zdrojových kódů pro HW zařízení, pouze demonstrační aplikaci jako již spustitelné EXE a zdrojové kódy pouze vytvořených úloh. Aktuální verze HW a SW a další informace jsou na požádání dostupné u vedoucího této práce.



# Literatura

- [1] KOPŘIVOVÁ, Gabriela. *Léčebně – rehabilitační plán a postup u dětí s dětskou mozkovou obrnou* [online]. Brno, 2012 [cit. 2013-11-10]. Bakalářská práce. Masarykova univerzita, Lékařská fakulta. Vedoucí práce Mgr. Lucie Nowaková.  
Dostupné z: [http://is.muni.cz/th/326418/lf\\_b/Lecebne-rehabilitacni\\_plan\\_a\\_postup\\_u\\_deti\\_s\\_detskou\\_mozkovou\\_obrnou.pdf](http://is.muni.cz/th/326418/lf_b/Lecebne-rehabilitacni_plan_a_postup_u_deti_s_detskou_mozkovou_obrnou.pdf).
- [2] PATZELOVÁ, Martina. *Léčebně rehabilitační plán a postup u dětí s dětskou mozkovou obrnou* [online]. Brno, 2010 [cit. 2013-11-10]. Bakalářská práce. Masarykova univerzita, Lékařská fakulta. Vedoucí práce Mgr. Lenka Drlíková.  
Dostupné z: [http://is.muni.cz/th/258591/lf\\_b/DMO.dipareza.pdf](http://is.muni.cz/th/258591/lf_b/DMO.dipareza.pdf).
- [3] HandTutor Physical Therapy Product. *Physical Therapy Solutions* [online]. 2014 [cit. 2014-05-12].  
Dostupné z: <http://www.meditouch.co.il/en/HandTutor>
- [4] Proprioceptive Training Devices Libra, Proprioception Instruments by EASYTECH. *Strumenti per Fisioterapia. Apparecchi Riabilitazione, Crioterapia, Terapia Fisica e Medicina Sportiva EASYTECH* [online]. © 2008-2012 [cit. 2014-05-14].  
Dostupné z: <http://easytechitalia.com/eng/prodotti/libra.php>
- [5] HUSÁK, Miroslav. *Mikrosenzory a mikroaktuátory*. Vyd. 1. Praha: Academia, 2008, 540 s. ISBN 978-80-200-1478-8.
- [6] HUSÁK, Miroslav. *Užití MEMS v průmyslu* [online]. 2008, č. 12 [cit. 2014-02-12].  
Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/38277.pdf>.
- [7] *MMA7455L* [online]. 2009 [cit. 2014-05-09].  
Dostupné z: [http://cache.freescale.com/files/sensors/doc/data\\_sheet/MMA7455L.pdf?fpsp=1](http://cache.freescale.com/files/sensors/doc/data_sheet/MMA7455L.pdf?fpsp=1).
- [8] Senzory. *Masarykova univerzita* [online]. © 1996–2014 [cit. 2014-04-02].  
Dostupné z: [http://amper.ped.muni.cz/jenik/nejistoty/html\\_tree/node16.html#section00016210000000000000](http://amper.ped.muni.cz/jenik/nejistoty/html_tree/node16.html#section00016210000000000000)
- [9] Arduino - HomePage [online]. © 2014 [cit. 2014-03-10].  
Web: <http://arduino.cc>.
- [10] Arduino I. - Linux Software. *Linux Software* [online]. 2003-2014 [cit. 2014-03-21].  
Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=1881](http://www.linuxsoft.cz/article.php?id_article=1881)

- [11] Arduino: vývojový kit pro hrátky s hardware - Root.cz. *Root.cz - informace nejen ze světa Linuxu* [online]. © 1998 – 2014 [cit. 2014-03-21].  
Dostupné z: <http://www.root.cz/clanky/arduino-vyvojovy-kit-pro-hratky-s-hardware>
- [12] ICSP - význam zkratky, obor Technika | zkratky.cz. *Zkratky.cz - významy zkratek* [online]. 2006 [cit. 2014-03-22].  
Dostupné z: <http://www.zkratky.cz/icsp/14319>
- [13] Arduino - ArduinoBoardUno. *Arduino - HomePage* [online]. ©2014 [cit. 2014-03-25].  
Dostupné z: <http://arduino.cc/en/Main/ArduinoBoardUno>
- [14] Arduino Uno Rev3 :: HW Kitchen. *HW Kitchen - Open Source Electronics Cooking* [online]. [cit. 2014-03-25].  
Dostupné z: <http://www.hwkitchen.com/products/arduino-uno-rev3>
- [15] 1. díl - Úvod do WPF (Windows Presentation Foundation). *Devbook.cz - Programátorská sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další.* [online]. © 2014 [cit. 2014-05-12].  
Dostupné z: <http://www.devbook.cz/c-sharp-tutorial-wpf-uvod-a-prvni-formularova-aplikace>

## Příloha A

# Obsah přiloženého CD

	dokumentace.....	technické dokumentace
	src.....	zdrojové kódy:
		cteni_dat_akcelerometrMMA7455.....v HW prototypu zařízení
		RehabilitacePohybu.....rehabilitační aplikace (PC)
	text	
		Kufnerova_BP.pdf.....text bakalářské práce

Obrázek A.1: Obsah přiloženého CD.