

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra kybernetiky



Diplomová práce

Zařízení pro základní diagnostiku a rehabilitaci pohybového aparátu člověka

Jiří Vildman

Vedoucí práce: Ing. Petr Novák, Ph.D.

8. května 2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Jiří V i l d m a n
Studijní program: Biomedicínské inženýrství a informatika (magisterský)
Obor: Biomedicínské inženýrství
Název tématu: Zařízení pro základní diagnostiku a rehabilitaci pohybového aparátu

Pokyny pro vypracování:

1. Prostudujte (na WWW) dostupné projekty zaměřené na využití pohybových senzorů pro základní diagnostiku a rehabilitaci pohybového aparátu člověka.
2. Vytvořte potřebné HW komponenty využívající zmíněné pohybové senzory, navrhnete jejich vhodné rozmístění (na testované osobě) a propojení. Dále možnosti přenosu snímaných dat do PC a rovněž základní ukládání, zpracování a vizualizaci takto získaných dat.
3. Podrobněji popište/zhodnoťte schopnosti takto navrženého a vytvořeného zařízení ve smyslu: celková cena materiálu, přesnost získaných dat, rychlost měření, stabilita/spolehlivost výstupu, možnosti využití v ordinaci / domácím prostředí.

Seznam odborné literatury:

- [1] Webové stránky projektů obdobných zařízení.
- [2] Richard Barnett, Larry O’Cull and Sarah Cox: Embedded C Programming and the Atmel AVR. Delmar, Cengage Learning, 2007, ISBN 1-4180-3959-4.
- [3] Matthew MacDonald: Pro WPF in C# 2010, Apress, 2013, ISBN 978-1-4302-7205-2.
- [4] Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner: Professional C# 4 .NET 4, Wrox, 2010, ISBN 978-0-470-50225-9.

Vedoucí diplomové práce: Ing. Petr Novák, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 12. 12. 2014

Poděkování

Děkuji vedoucímu práce za obětavé vedení mé diplomové práce, cenné rady jak v oblasti programování, tak v oblasti tvorby HW a v neposlední řadě za zapůjčení mnohdy soukromých nástrojů a přístrojů pro zdárnou realizaci zmíněného HW.

Prohlášení autora práce

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

V Praze dne

.....

Jiří Vildman

Abstrakt

Tato práce navazuje na návrh zařízení z bakalářské práce, určeného k dohledu nad pohybovou rehabilitací základního pohybového aparátu člověka. Původní návrh však dále nejen rozvíjí, ale do jisté míry i realizuje. Komplexně řeší mnoho otázek a problémů při realizaci hardwaru a rozšiřuje software pro použití s nově vytvořeným hardwarem. Rovněž popisuje další možnosti budoucího rozšíření a optimalizace softwaru, které jsou však nad rámec této práce, ale vycházejí zcela z jejího řešení. V závěru se práce zabývá zhodnocením možností a vhodností tohoto systému.

Klíčová slova:

člověk, pohyb, snímání, přístroj, C#, Atmel.

Abstract

This master's thesis builds on the design of the device dealt with in my bachelor's thesis. The device is intended for diagnostics of basic human locomotion system. However, the design is further developed and partially implemented in the thesis. The thesis comprehensively deal with remaining issues and problems which may arise during implementation of HW and extends SW for use with the newly created HW. It also describes the additional options of expansion and optimization of the SW, which are beyond the scope of this work. At the end, the work evaluates the suitability of this solution.

Key words:

human, motion, capture, device, C#, Atmel.

1 Obsah

2	Seznam obrázků	xiii
3	Úvod	1
3.1	IGS-190-M.....	1
3.2	Osobní pohybový senzor od firmy Princip.....	2
3.3	Xsens MVN.....	2
3.4	IGS-180	3
3.5	Další projekty.....	3
3.6	Shrnutí	3
4	Cíle práce	5
5	Stav po bakalářské práci	7
6	Hardware	9
6.1	HW jako celek.....	9
6.2	Pohybové senzory	12
6.2.1	MPU 6050	12
6.2.2	MPU 9150.....	12
6.2.3	MPU 9250.....	12
6.3	Měřicí modul	13
6.4	SW v měřicím modulu	16
6.5	Centrální modul.....	18
6.5.1	První generace centrálního modulu	18
6.5.2	Druhá generace centrálního modulu	20
6.6	Komunikace s měřicími moduly	21
7	Obslužná úloha na PC.....	25
7.1	Implementace softwaru na PC	26
7.1.1	Hlavní třídy úlohy	26
7.1.2	Základní zobrazení 3D postavy	31
7.1.3	Transformace částí těla	34
7.2	Přepočty naměřených hodnot.....	36
7.3	Druhá postava a předcvičování	39
7.4	Detekce připojených senzorů	39

7.5	Přiřazení měřících modulů k částem těla	40
7.6	Výběr aktivních částí.....	41
7.7	Kalibrace	42
7.8	Nahrávání, ukládání a přehrávání pohybů	42
8	Hodnocení	45
8.1	Rychlost měření.....	45
8.2	Stabilita výstupu	45
8.3	Přesnost získaných dat	46
8.4	Cena.....	47
8.5	Možnosti rehabilitace.....	47
8.5.1	Předcvičování.....	48
8.5.2	Orientační kontrola	48
8.6	Cviky, pro které lze v současné době zařízení využít.....	48
8.7	Možnosti využití zařízení	50
9	Návrhy na další vývoj.....	53
10	Závěr	55
11	Zdroje.....	57
12	Obsah příloženého CD	59

2 Seznam obrázků

Obrázek 1 – Zařízení pro snímání pohybu člověka od firmy MetaMotion, GS-190-M [1].	1
Obrázek 2 – Moduly pro měření pohybu od firmy Princip [2].	2
Obrázek 3 – Obě varianty zařízení Xsens MVN spolu s obslužným SW [3].	3
Obrázek 4 – Vzhled aplikace pro vizualizaci dat po bakalářské práci.	7
Obrázek 5 – Topologie zapojení zařízení s připojenými třemi větvemi měřících modulů.	10
Obrázek 6 – Předpokládané umístění modulů na postavě uživatele.	11
Obrázek 7 – Pohybové senzory MPU 6050 [7] (vlevo), MPU 9150 [10] (uprostřed) a MPU 9250 (vpravo) [8].	12
Obrázek 8 – Princip propojení vodičů mezi konektory na měřícím modulu (* pouze při osazené propojce P5, podle obrázku 9).	13
Obrázek 9 – Vzhled měřícího modulu s adresou 12 s popisem důležitých částí (*3,3V pouze při propojce na P5).	15
Obrázek 10 – Schéma měřícího modulu, vytvořené v programu Eagle.	15
Obrázek 11 – Návrh plošného spoje měřícího modulu.	16
Obrázek 12 – Zobrazení komunikace v rámci celého zařízení při využití centrálního modulu 1. generace.	17
Obrázek 13 - Vzhled centrálního modulu 1. generace s připojenými 2 větvemi měřících modulů.	18
Obrázek 14 – Schéma centrálního modulu 1. generace a jeho základní části.	19
Obrázek 15 – Zobrazení propojení hradla AND v obvodu centrálního modulu.	19
Obrázek 16 – Zobrazení propojení vstupů budiče sběrnice v obvodu centrálního modulu.	20
Obrázek 17 – Vzhled úlohy po spuštění a její základní části.	25
Obrázek 18 – Princip rozlišení směru náklonu akcelerometru.	38
Obrázek 19 – Přepočtení hodnot akcelerometru pro měřící modul umístěný na těle.	38
Obrázek 20 – Zobrazení postupu přiřazování měřících modulů k částem postavy.	41
Obrázek 21 – Zobrazení postupu pro nahrávání pohybu.	44
Obrázek 22 – Zobrazení postupu pro přehrání pohybu, respektive pro spuštění předcvičování.	44
Obrázek 23 – Zobrazení průběhu filtrace dat.	46
Obrázek 24 – Postava se senzory na stehnech a lýtkách, zobrazená při vykonávání cviků: dřep, roznožení a přednožení.	48
Obrázek 25 – Postava se senzory na pažích a předloktích, zobrazená při vykonávání cviků vzpažení, předpažení a upažení.	49
Obrázek 26 – Postava se senzorem na těle, zobrazená při vykonávání cviků: záklon, úklon a předklon.	49
Obrázek 27 – Postava se senzorem na hlavě, zobrazená při vykonávání cviků: předklon, úklon a záklon.	50

3 Úvod

Ke snímání pohybu člověka, respektive pohybu jeho jednotlivých částí, se v současné době využívají různé techniky a způsoby jako například: optické snímání pomocí kamer (buďto s použitím tzv. „markerů“ neboli značek na člověku, případně i bez nich), využití dat z magnetometrů po umělém vytvoření vhodného magnetického pole nebo využití senzorů snímajících zrychlení, jako jsou akcelerometry a gyroskopy (inerciální senzory) atd.

S přihlédnutím k návaznosti na bakalářskou práci bych zde rád uvedl některé systémy, které se stejně jako tato práce věnují využití akcelerometrů, popřípadě gyroskopů, nebo dokonce magnetometrů. To však pouze v přirozeném gravitačním a magnetickém poli země, tedy bez dodatečně uměle vytvořeného například magnetického pole.

3.1 IGS-190-M

Zařízení (viz Obrázek 1) od firmy METAMotion využívá 19 gyroskopických senzorů. 18 z nich je součástí speciálního obleku a jeden je určen pro různé pomůcky (neboli předměty) dle přání uživatele (např. tenisová raketa nebo boxerská rukavice při hraní her). Přenos dat je řešen bezdrátově. Výdrž baterií udávaná výrobcem je 3 hodiny. Dle videí, která lze nalézt na oficiálních webových stránkách produktu, poskytuje zařízení velmi dobré rozlišení. Výrobce uvádí $0,1^\circ$. Cena zmíněného zařízení začíná na zhruba 42000 dolarech, což je v současné době (4/2015) cena lehce přesahující milion korun. [1]



Obrázek 1 – Zařízení pro snímání pohybu člověka od firmy MetaMotion, GS-190-M [1].

3.2 Osobní pohybový senzor od firmy Princip

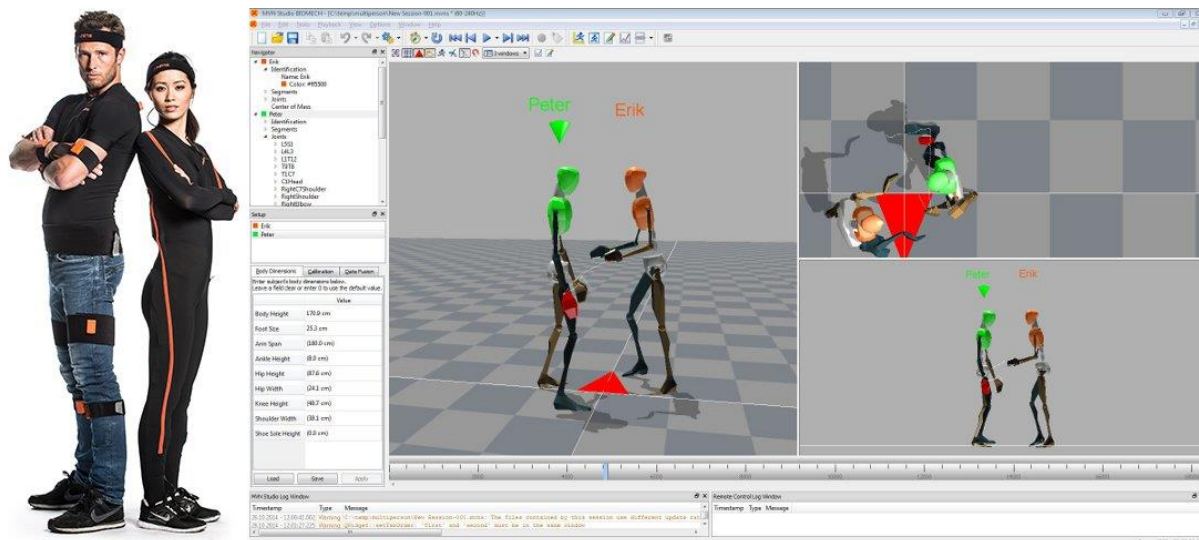
V tomto případě jde stále více pouze o projekt než o výsledné komerční řešení. Zakoupit jej tedy zatím nelze. Představuje však kompaktní zařízení obsahující tříosý akcelerometr, tříosý gyroskop a nová generace, podle dostupných informací, by měla obsahovat i tříosý magnetometr. Vše je v kompaktním pouzdře spolu s bateriemi, pamětí a rovněž doplněno o bezdrátové „pojítka“, jak lze vyčíst na oficiálním webu tohoto projektu. Zda jde o Wi-Fi nebo jinou bezdrátovou technologii, zde není uvedeno. Sensory by měly být schopny komunikovat jak vzájemně, tak i se vzdáleným počítačem. S tímto zařízením byly doposud zrealizovány dva projekty. První se zabýval porovnáním aktivity zdravé horní končetiny a horní končetiny s poškozenou pohyblivostí. K tomuto byly využity 3 senzory a speciálně pro tento popisovaný případ vyvinutý software. Druhý projekt se zabýval diagnostikou závratí, respektive kontrolou cvičení tzv. Sémontova a Epleyho manévru. Díky těmto manévřům se dá diagnostikovat a následně léčit benigní paroxysmální polohové vertigo, což je nejčastější příčina závratí. Pro tento projekt byl opět firmou vyvinut speciální software. Toto zařízení ukazuje Obrázek 2. [2]



Obrázek 2 – Moduly pro měření pohybu od firmy Princip [2].

3.3 Xsens MVN

Další zařízení (viz Obrázek 3) se prodává ve dvou variantách. První varianta obsahuje 17 bezdrátových senzorů, které se v podstatě přichycují každý zvlášť. Druhou možností je oblek, ve kterém jsou senzory (též 17) přímo umístěny a drátově propojeny. Každá varianta má své výhody i nevýhody. Bezdrátové řešení má samozřejmě vyšší spotřebu (potřeba 17 baterií, přesto ale kratší výdrž - šest hodin) a nižší výstupní vzorkovací frekvenci (60 Hz). Paměť této varianty stačí pouze na přibližně 10 sekund záznamu pohybu (paměť slouží pouze jako „buffer“). Tato varianta je však pro uživatele mnohem pohodlnější. Naopak druhá varianta ve formě celého obleku poskytuje vzorkovací frekvenci až 240 Hz a o více jak polovinu delší výdrž na baterie (9,5 hodiny), a to při použití pouze jedné baterie. Paměť („buffer“) je zde až pro 2 minuty záznamu pohybu. Data z obou variant jsou odesílána bezdrátově na vzdálený počítač. Cena těchto zařízení začíná na 12000 dolarech (v 4/2015 cca 300 tis. Kč). Firma k tomuto zařízení také vyvinula zobrazovací software MVN Studio. Ten slouží pro ukládání, vizualizaci a export dat pro další 3D animační programy. [3]



Obrázek 3 – Obě varianty zařízení Xsens MVN spolu s obslužným SW [3].

3.4 IGS-180

Toto zařízení vyvinula firma Animazoo. Jde o 17 gyroskopických senzorů implantovaných do speciálního obleku. Sensory jsou drátově propojeny a data jsou do PC přenášena přes Wi-Fi. K tomuto projektu nelze dohledat bližší informace a již vůbec ne k použitému hardwaru. Softwary jsou společně s hardwarem dodávány ve formě tři typů. AnimaView Foot step editor, Autocal custom skeleton generator a AnimaDemo. Bohužel výrobce neuvádí, k čemu jednotlivé typy softwaru slouží. Pro zajímavost: toto zařízení využívají firmy jako Ubisoft, Hyundai nebo Škoda auto. Zakoupit ho lze v přepočtu přibližně za 1,2 miliónu Kč (4/2015). [4]

3.5 Další projekty

Existuje rovněž několik dalších projektů, které začaly ve většině případů jako školní práce. Obvykle však nedosáhly použitelných výsledků a tudíž na ně bohužel zřejmě nikdo nenavázal. Dnes je tedy téměř nemožné k nim dohledat jinou dokumentaci než například text diplomové, či disertační práce. Zdrojové kódy programů a schémata obsahují tyto práce jen výjimečně.

3.6 Shrnutí

Na trhu tedy existuje několik projektů nebo dokonce již výrobků splňujících parametry, jaké byly stanoveny pro systém vytvářený v této práci. Některé parametry, těchto systémů, jsou samozřejmě mnohem lepší a tedy za hranicí možností této práce. Nevyhovující je však, a to bohužel ve všech případech, velká finanční náročnost a tím tedy nedostupnost těchto zařízení. Nejvýhodněji se z hlediska ceny jeví systém Xsens MVN. I jeho cena však přesahuje 300 tis. Kč. U tohoto přístroje nutno dále upozornit na vysoké HW požadavky pro cílový počítač. Výrobce hovoří o čtyř-jádrovém procesoru o frekvenci 2.7 GHz nebo vyšší, o SSD disku a o 8 GB operační paměti, což může být další problém. Pokud by se tedy pacientovi zapůjčil takový přístroj domů, muselo by se k němu nejspíše zapůjčit i vhodné PC. To by sice nemělo velký vliv na výslednou cenu zařízení (PC jsou řádově

levnější), ale nebylo by již snadno přenosné a bylo by potřeba celý systém u uživatele odborně instalovat. Další dvě uvedená řešení, IGS-180 a IGS-190-M, jsou pro běžnou ordinaci, natož domácí použití, cenově zcela nedostupná. Poslední z možností, zařízení od firmy Princip, by bylo možno využít, ale záleželo by na situaci, zda by se zapůjčení nebo prodej podařilo vyjednat, neboť toto zařízení není běžně k prodeji. Nicméně k tomuto zařízení není žádný vhodný software.

4 Cíle práce

Cíle této diplomové práce lze rozdělit do čtyř hlavních částí. Prvním cílem je prostudování obdobných komplexnějších projektů, zhodnocení vhodnosti jejich využití při diagnostice a zejména domácí rehabilitaci pohybového aparátu člověka. Druhým cílem jsou všechny úkony potřebné k upřesnění návrhu, vytvořeného v rámci bakalářské práce, a realizace hardwaru, tedy i jeho skutečné sestavení. Další částí práce je upravit stávající software (vytvořený v rámci bakalářské práce) pro použití s nově realizovaným hardwarem (novými měřicími moduly) a rozšířit jej o další potřebné části (například možnost více senzorů). Poslední cíl spočívá ve zhodnocení vlastností a schopností celého systému, včetně jeho finanční náročnosti. Konkrétně lze tedy cíle této práce popsat následovně:

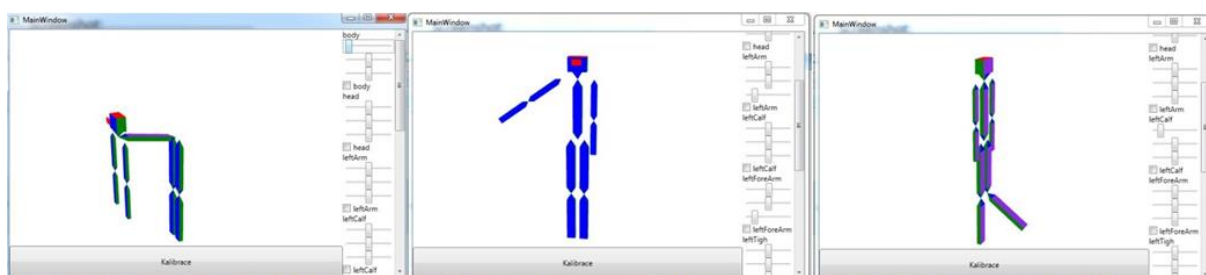
- **Projekty:** Jedná se o dohledání a zhodnocení některých existujících řešení pro snímání pohybu člověka. Jde však již pouze o systémy s inerciálními senzory, kterými jsou akcelerometr a gyroskop a to z důvodu, že jde o diplomovou práci, navazující na předešlou bakalářskou práci, ve které byl již základní princip měření zvolen.
- **Hardware (HW):** Vzhledem k tomu, že původní návrh (bakalářská práce) je již dva roky starý, je nutné znovu zmapovat a zhodnotit dostupnost použitelných senzorů, mikroprocesorů a dalších hardwarových prvků. S přihlédnutím k novým skutečnostem je potřeba upravit stávající návrh a tento nový návrh realizovat a otestovat.
- **Software (SW):** Jak již bylo uvedeno, bude nutná úprava SW pro „spolupráci“ s novým hardwarem, převážně pak při použití více pohybových senzorů současně. Software by měl být také doplněn o možnost předcvičování, kdy budou zobrazeny dvě postavy, z nichž jedna bude předcvičovat, respektive opakovat předem nahrané / uložené pohyby a druhá bude kopírovat skutečné pohyby uživatele (tzv. vizualizovat data ze senzorů). S tím také souvisí další rozšíření o možnost nahrávání, ukládání a přehrávání pohybů. Jelikož se zde bude využívat více měřících modulů (senzorů), ty však nemusí být vždy využity všechny současně, měla by aplikace umožňovat propojení libovolného měřícího modulu (senzoru) s libovolnou částí těla. Aplikace by též měla být schopna sama detekovat aktuálně připojené měřící moduly.
- **Zhodnocení:** V závěrečné fázi proběhne zhodnocení dosažených výsledků. Hodnotí se především, zda zařízení (návrh a prototyp) vyhovuje účelu, pro jaký bylo vytvořeno, tedy zda s ním lze zejména dohlížet na rehabilitaci pohybového aparátu člověka (převážně v domácím prostředí). Dále se také stanoví, na jaké konkrétní cviky lze zařízení vhodně využít. Neméně důležitá je současně finanční náročnost vyvinutého zařízení. Měl by se tedy zde také odrazit odhadovaný rozpočet celého řešení. V této části by rovněž měla být nastíněna možná rozšíření do budoucna takto navrženého systému.

Další důležitou věcí je vymezit, čemu se tato práce naopak nevěnuje:

- zabezpečení komunikace – zatím není důvod data šifrovat nebo jinak zabezpečovat
- minimalizaci – pro ladění a ověření činnosti systému může být i nežádoucí
- přesnosti – pro orientační kontrolu rehabilitace není přesnost kritická
- zdravotnické certifikaci – zařízení není v takovéto vývojové fázi

5 Stav po bakalářské práci

V rámci mé (předešlé) bakalářské práce byl před dvěma lety vytvořen návrh zařízení pro snímání pohybu člověka, určeného zejména pro domácí rehabilitaci, a byla rovněž vytvořena demonstrační aplikace pro vhodnou vizualizaci takto snímaného pohybu. Jelikož hardware byl v podstatě pouze navržen, nikoli zkonstruován, nebylo možno aplikaci na PC dostatečně využít. Z toho důvodu byla aplikace vytvořena, a tedy i odzkoušena, pouze s jedním senzorem, který byl původně určen pro zcela jiné účely, a který bylo možné si snadno zapůjčit. Vzhled původní aplikace pro vizualizaci pohybu člověka, snímaného pouze jedním senzorem znázorňuje Obrázek 4.



Obrázek 4 – Vzhled aplikace pro vizualizaci dat po bakalářské práci.

6 Hardware

Další část textu se věnuje popisu HW oblasti této práce a pro přehlednost bude využíváno tohoto názvosloví:

- „senzor“ – měřící součástka, v tomto případě kombinace akcelerometru, gyroskopu, popřípadě i magnetometru
- „měřící modul“ – část zařízení, na které je umístěn senzor a mikroprocesor poskytující data z detekce / měření pohybu
- „centrální modul“ – část zařízení určená pro sběr dat z jednotlivých měřících modulů a zajišťující komunikaci například se vzdáleným PC (přenos snímaných dat)
- „větev“ – jeden až tři měřící moduly zapojené v sérii

HW byl navrhován / realizován s těmito základními požadavky:

- pro použití s jedním až maximálně s patnácti měřícími moduly (zpočátku pouze se šesti)
- s možností paralelního čtení dat ze všech měřících modulů současně (do centrálního modulu)
- možnost bateriového napájení a bezdrátového přenosu dat
- jednoduchost a účelnost s postačující přesností
- minimální cena celého zařízení

Na činnosti celkového zařízení se podílejí dva základní typy hardwaru. Několik měřících modulů (s pohybovými senzory) a jeden centrální modul (pro sběr a přenos dat).

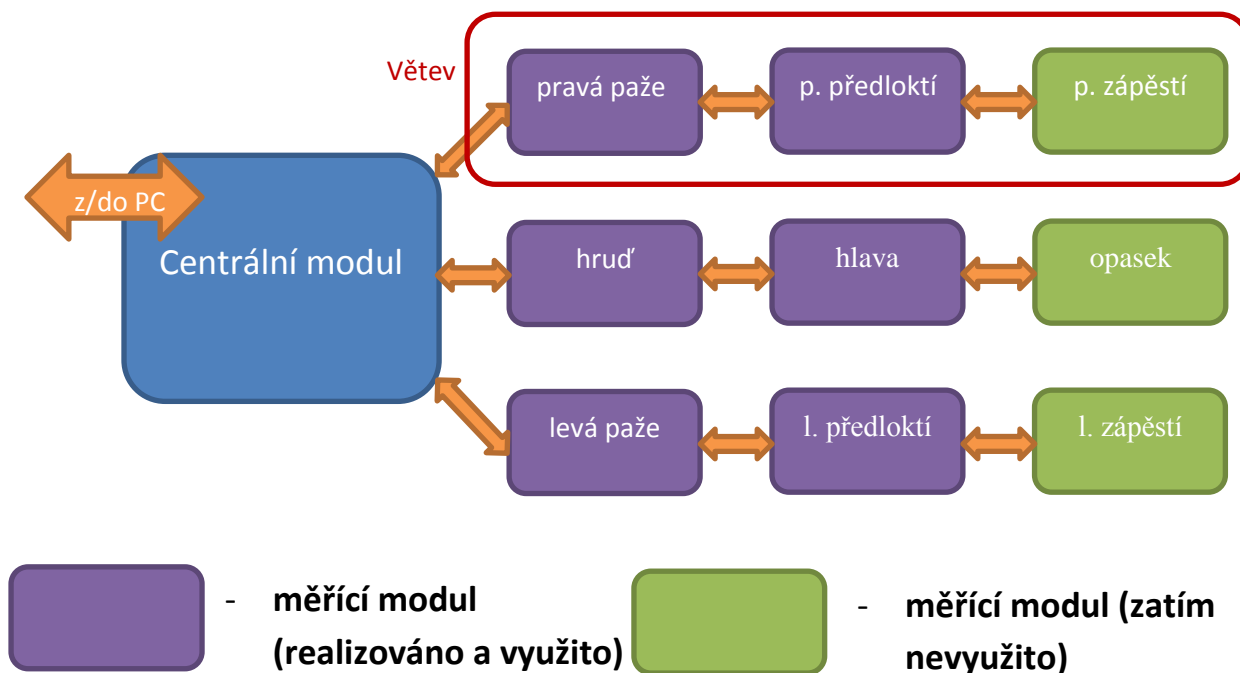
Měřících modulů (shodných) je navrženo celkem 10, což je postačující počet pro získání informací o poloze všech hlavních částí těla, respektive o jejich úhlech natočení vůči zemi. V základním nastavení projektu se počítá s jejich umístěním na hlavě, hrudi a párově na pažích, předloktích, stehnech a lýtkách. Blíže se tomuto rozmístění věnuje kapitola 6.1. Zařízení je možné i přes jeho jednoduchou konstrukci rozšířit dále ještě o dalších pět měřících modulů. Tedy na celkový počet 15 měřících modulů (dáno použitým rozsahem adresace a konstrukcí řetězení). Tyto dodatečné senzory jsou zamýšleny na chodidla (2x), zápěstí (2x) a opasek. Ty však nejsou v tomto popisu nikterak podrobněji zahrnuty.

Centrální modul slouží k propojení mezi měřícími moduly a vzdáleným PC. Nejprve byl realizován centrální modul první generace, aby mohlo dojít ke zkušebnímu provozu navržených měřících modulů (rychlost vzorkování, objem dat, spolehlivost, ...). V návrhu centrálního modulu druhé generace bylo zahrnuto bateriové napájení, případně paměťová karta a samozřejmě prvek pro bezdrátový přenos.

6.1 HW jako celek

Při spojení modulů zmíněných v kapitolách 6.3 a 6.5.1 vznikne zařízení schopné měřit polohu vůči zemi na šesti různých částech těla (počet stanovený pro rozsah této práce) a záznam o této poloze přenášet do PC. Šest částí těla je užíváno pro možnost sledování buďto kompletní horní části těla, tedy obě paže a předloktí, hrud' a hlavu, nebo spodní část těla, tedy obě stehna a lýtka, přičemž

dva moduly v tomto případě zůstanou nevyužity. Zařízení mezi sebou komunikují pomocí sériového rozhraní USART, rychlostí 230,4 kBaudů (první verze) nebo modifikovaným SPI rozhraním (druhá verze). Topologii celkového zapojení znázorňuje Obrázek 5.

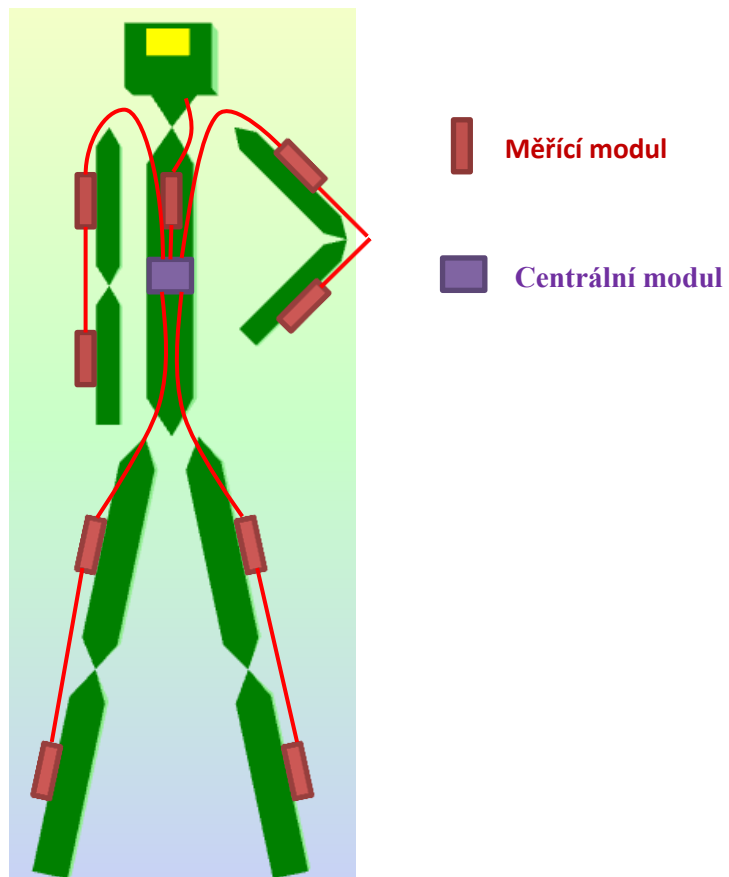


Obrázek 5 – Topologie zapojení zařízení s připojenými třemi větvemi měřících modulů (horní část těla).

Na centrální modul lze, mimo dvou základních měřících modulů, připojit do každé existující větve ještě třetí měřící modul a tím zvýšit počet senzorů až na již zmíněných 15. Naměřená data se budou přenášet do vzdáleného PC nebo jiného vhodného zařízení (tablet, „smart“ TV, „chytrý“ telefon, ...) pomocí USB nebo bluetooth. Případně se mohou ukládat na SD kartu (druhá generace centrálního modulu je navržena pro všechny tyto varianty). V rámci této práce však není centrální modul druhé generace zcela realizován (není dokončena a odladěna podpora ukládání dat na paměťovou kartu).

Celé zařízení je pak na člověka uchyceno pomocí pásků se suchými zipy a to následujícím způsobem (viz Obrázek 6):

- na paže, předloktí, stehna a lýtka jsou měřící moduly umístěny z vnější strany tak, aby nedocházelo ke křížení kabelů a měřící moduly nevadily v pohybu těchto částí těla
- měřící modul na tělo je umístěn v horní části hrudníku (popruh jde nad prsy)
- umístění modulu na hlavě připomíná „čelenku“, kdy je senzor v oblasti týlu
- centrální modul je umístěn ve spodní části hrudníku (popruh jde pod prsy), případně na pase / opasku



Obrázek 6 – Předpokládané umístění modulů na postavě uživatele.

6.2 Pohybové senzory

Pohybové senzory byly vybrány podle několika kritérií:

- digitální výstup – jednoduchá komunikace bez potřeby přesného A/D převodníku na pomocném procesoru
- dostupné jako minimální vývojová deska – některé senzory nelze v „amatérských“ podmínkách zapájet / osadit na plošný spoj
- kombinace alespoň akcelerometru a gyroskopu (magnetometr bude výhodou)
- dostupná dokumentace
- přijatelná cena a dostupnost

Dále jsou uvedeny pohybové senzory využitě v této práci, pro které je celý projekt (v současné verzi) navržen. Vzhled těchto senzorů ukazuje Obrázek 7.

6.2.1 MPU 6050

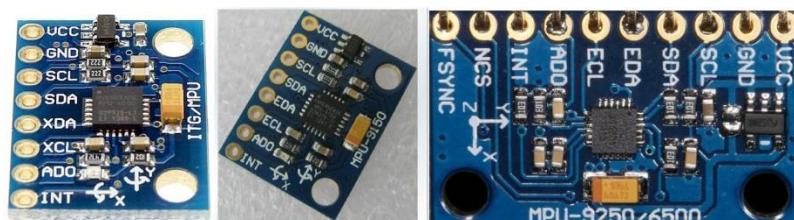
Jedná se o senzor [4] kombinující tříosý akcelerometr, tříosý gyroskop a signálový procesor (podle dostupných údajů výrobce). Data z os na každém senzoru jsou kvantována 16 bitovým AD převodníkem. Rozsah měření je nastavitelný. U akcelerometru na $\pm 2g$, $\pm 4g$, $\pm 8g$ nebo $\pm 16g$ a u gyroskopu na ± 250 , ± 500 , ± 1000 nebo ± 2000 $^{\circ}/s$. Vše je obsaženo v jednom čipu o velikosti $4 \times 4 \times 0.9$ mm. Pro komunikaci se využívá I²C sběrnice o rychlosti 400 kHz (model MPU 6000 poskytuje i sběrnici SPI). Rychlost měření je jak u akcelerometru, tak u gyroskopu rovněž nastavitelná. Maximální hodnota rychlosti měření u gyroskopu je 8 kHz a u akcelerometru 1 kHz. V našem případě je použita minimalistická vývojová verze s názvem „breakthrott board“, kdy je od výrobce měřící součástka umístěna na vhodném minimálním plošném spoji s vyvedenými potřebnými piny / signály. To vše se pohybuje v ceně od 50 do 150 Kč (na eBay.com).

6.2.2 MPU 9150

Tento senzor [5] je obdobný jako MPU 6050. Obsahuje však navíc i magnetometr. Parametry akcelerometru a gyroskopu jsou zcela shodné s verzí MPU 6050. Hodnoty z magnetometru jsou digitalizovány pomocí 13 bitového AD převodníku. Rozsah je zde $\pm 1200 \mu T$. Vizuálně jsou MPU 9150 a MPU 6050 také téměř totožné. Cena se zde pohybuje od 150 do 200 Kč (na eBay.com).

6.2.3 MPU 9250

Zde je vše realizováno v ještě menším provedení a to $3 \times 3 \times 1$ mm. Magnetometr má navíc již 16 bitový AD převodník a rozsah $\pm 4800 \mu T$. MPU 9250 je schopno komunikovat jak pomocí I²C rozhraní, tak i pomocí SPI. Akcelerometr dokáže měřit rychlostí až 4 kHz [6]. Cena je zde mezi 150 a 300 Kč (na eBay.com).



Obrázek 7 – Pohybové senzory MPU 6050 [7] (vlevo), MPU 9150 [10] (uprostřed) a MPU 9250 (vpravo) [8].

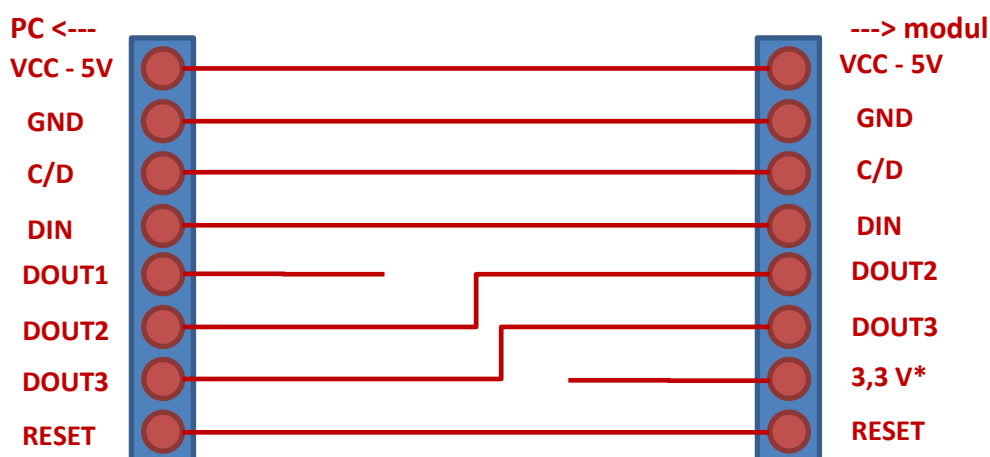
Pro zde popisované použití za účelem základní diagnostiky a domácí rehabilitace jsou uvedené senzory v podstatě rovnocenné, i když některé jejich parametry se nepatrně odlišují.

6.3 Měřící modul

V této části zařízení dochází k detekci, neboli měření polohy senzoru vůči zemi (tedy vůči směru gravitačního zrychlení). Pro tento účel je využito již zmíněných senzorů od firmy InvenSense. Ve vytvořené verzi lze použít tedy všechny senzory s označením MPU 6050, MPU 9150 nebo MPU 9250. Tohoto je dosaženo pouhým nastavením (podmíněné kompilace v programu) při překladu firmware do měřícího modulu. O rozdílech využitých senzorů vypovídají kapitoly 6.2.1 – 6.2.3. Měřící modul dále obsahuje jednočipový procesor od firmy Atmel, typu AVR (v současné verzi ATmega8), tři stavové LED diody, jedno tlačítko pro libovolné použití, dva konektory pro vzájemné propojení (řetězení) měřících modulů a možnost manuálního nastavení adresy konkrétního modulu pomocí 4 propojek (současně využito rovněž k programování osazeného procesoru).

Tlačítko je zde použité zejména z důvodu interakce s uživatelem při nastavování celého systému. Využito je tedy především pro přiřazování měřícího modulu ke konkrétní části těla. Toto je důležité v předpokladu, že každý měřící prvek lze umístit na libovolnou část těla. Je však možné, že své využití později najde i pro jiné účely. Například pokud by byl tento hardware použit v jiném projektu.

Hlavní konektory na měřícím modulu jsou dva a jsou téměř totožné. Jeden je určen pro připojení „směrem“ k centrálnímu modulu, druhý „směrem“ k dalšímu měřícímu modulu ve stejné větvi. Takto lze snadno zapojit / řetězit až tři měřící moduly za sebe. Tedy tzv. do jedné větve. Zatím se však počítá v každé větvi pouze se dvěma měřícími moduly. Obrázek 8 poskytuje přehled o signálech na konektorech měřícího modulu pro jejich propojení do větve.



Obrázek 8 – Princip propojení vodičů mezi konektory na měřícím modulu (* pouze při osazené propojce P5, viz Obrázek 9).

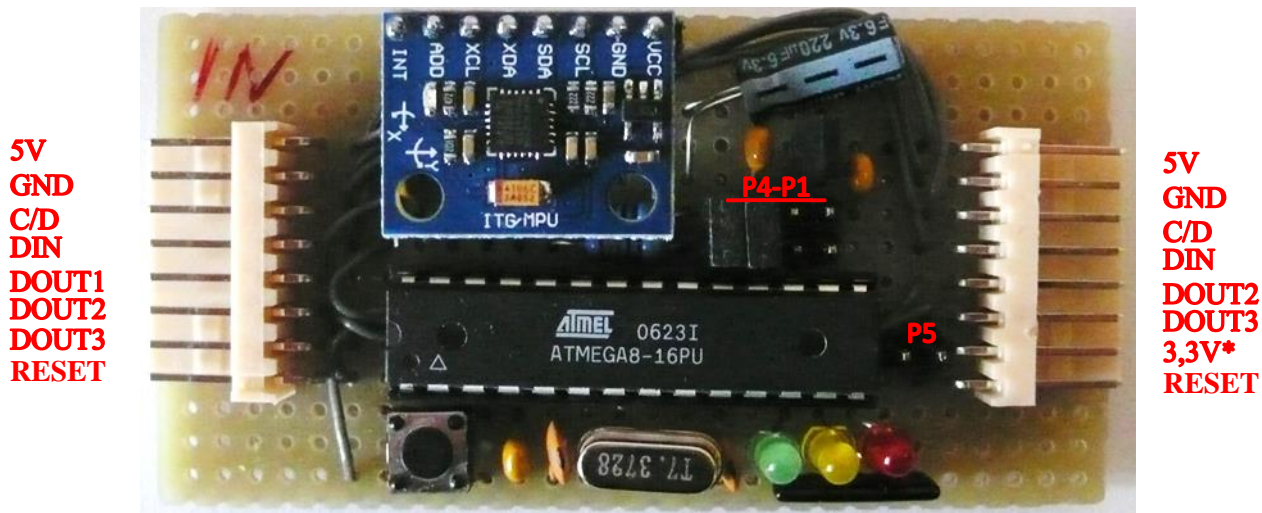
Konektor „směrem“ k centrálnímu modulu obsahuje tyto signály:

- VCC: napájení 5V
- GND: zem 0V
- C/D: (Command/Data) přepínání mezi příkazovou a datovou komunikací
- DIN: datový vodič z centrálního modulu do senzorů
- DOUT1 – DOUT3: tři vodiče s daty od až tří senzorů v jedné větvi směrem do centrálního modulu
- RESET: nulování konfigurace senzoru

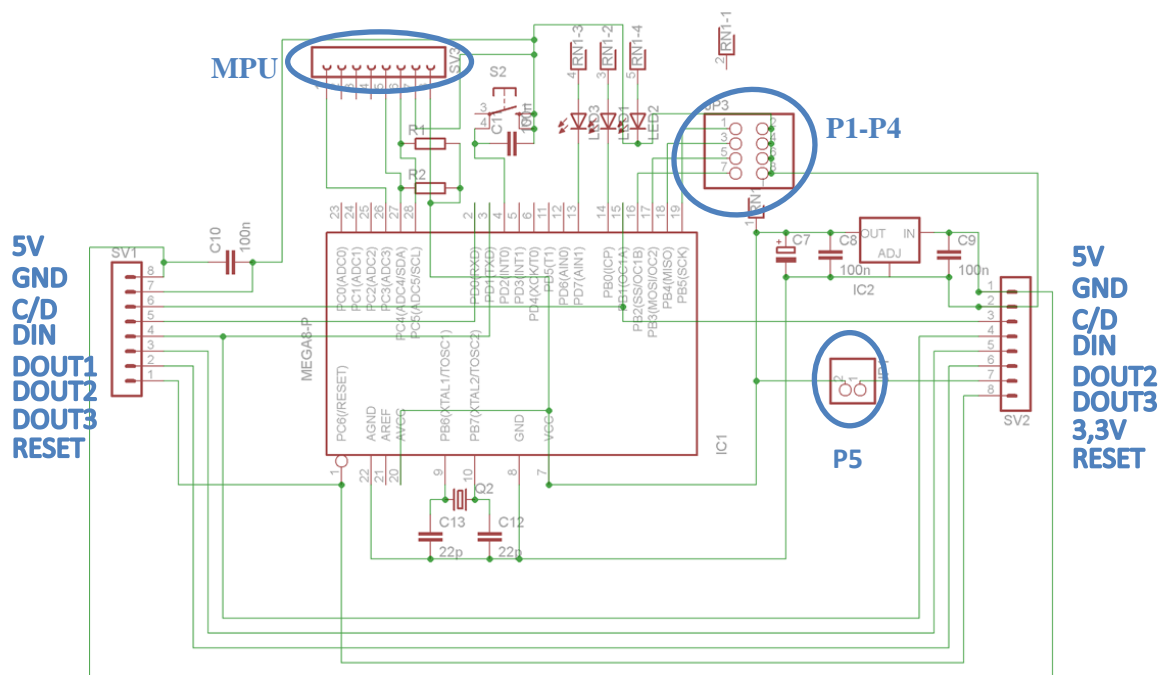
U konektoru určeného k propojení s dalším měřícím modulem zůstal nevyužit jeden z pinů pro data (DOUT1), který je ukončen vždy na měřícím modulu (na pinu procesoru). Je tam pouze z důvodu, sjednocení konektorů na obou stranách modulu a propojovací kabely tudíž byly volně zaměnitelné. Tento nevyužitý výstup je však propojitelný pomocí propojky P5 (viz Obrázek 9) na interní stabilizované napájení 3,3 V vytvořené v měřícím modulu, protože je tento konektor rovněž využíván při programování osazeného procesoru.

Adresní piny P1 až P4 (viz Obrázek 9) slouží k manuálnímu nastavení binární adresy (bity A0 až A3) každého měřícího modulu při komunikaci. Osazená propojka značí logickou jedničku, ačkoli je připojena na zem (na nízké a tedy častěji používané adresy tak stačí méně propojek, rovněž je toto řešení srozumitelnější). Při pohledu na modul tak, jak ukazuje Obrázek 9, je nejnižší bit adresy vpravo. Takovéto řešení adresování je využito z důvodu jednoduchosti využití a variability měřících modulů. Všechny moduly takto obsahují stejný hardware i software. Bylo by možné též nastavit adresu senzoru přímo v paměti mikroprocesoru, například v jeho paměti typu EEPROM, ale od této možnosti bylo v této prototypové fázi návrhu ustoupeno.

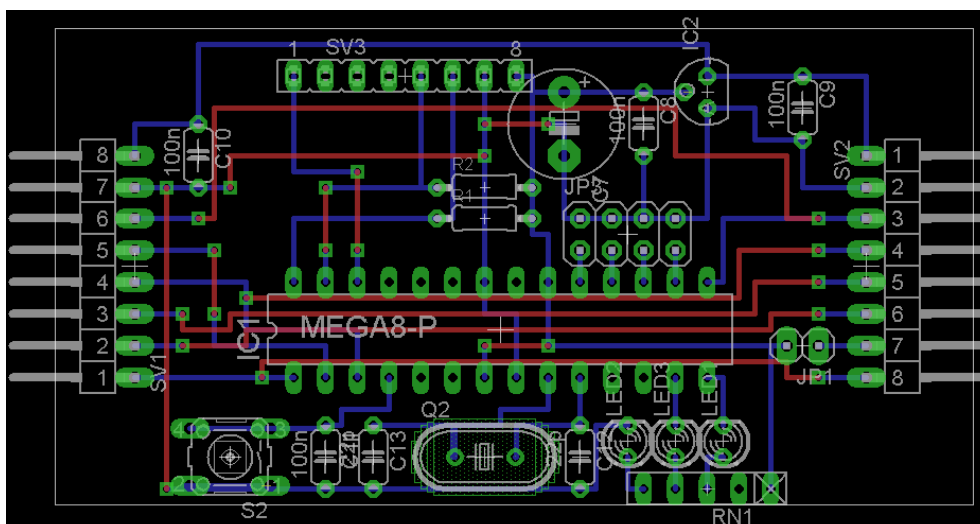
Při programování procesoru je potřeba všechny propojky vyjmout a na místo P1 – P3 v řadě blíže procesoru připojit programovací konektor. Konkrétně P1 – SCK, P2 – MISO, P3 – MOSI (použito při programování metodou ISP – in system programming). Zbylé vývody, potřebné pro programování osazeného procesoru, jsou na konektoru, který znázorňuje Obrázek 9 vpravo (propojení do dalšího senzoru ve větvi). Při programování je samozřejmě nutné osadit propojku na pozici P5 zajišťující výstup napětí 3,3 V na příslušném vývodu konektoru pro jeho detekci připojeným programátorem. Schéma měřícího modulu znázorňuje Obrázek 10. Návrh plošného spoje potom Obrázek 11.



Obrázek 9 – Vzhled měřícího modulu s adresou 12 s popisem důležitých částí (*3,3V pouze při propojce na P5).



Obrázek 10 – Schéma měřícího modulu, vytvořené v programu Eagle.



Obrázek 11 – Návrh plošného spoje měřícího modulu.

Pro měření polohy může být využito jednoho ze senzorů MPU 6050, MPU 9150 nebo MPU 9250. S těmito senzory se komunikuje pomocí sběrnice I²C. S celým měřícím modulem se komunikuje pomocí rozhraní USART při použití rychlosti 230,4 kBaudů. Vodič RxD, respektive DIN, přichází data do senzoru, je pro všechny měřící moduly společný a tedy jen jeden. TxD, odchází data z modulu, má pak každý měřící modul samostatná. Jde o vodiče DOUT1 – DOUT3. Takto je systém připraven vysílat data paralelně (i když ve skutečnosti je přenos sériový) ze všech měřících modulů do centrálního modulu současně. Tohoto není využito v centrálním modulu první generace, kde se komunikuje vždy pouze s jedním měřícím modulem (pouze adresovaný dotaz na data a příjem odpovědi). Paralelní přenos je využit až v centrálním modulu druhé generace.

Měřící modul také obsahuje 3 LED diody s tímto významem:

- zelená – indikace přítomnosti napájení
- oranžová – přenos dat / aktivita modulu
- červená – vykonávání povelu (bliká), chyba (svítí)

6.4 SW v měřícím modulu

SW do mikroprocesoru obsaženého v měřícím modulu byl nejprve vytvářen pomocí vývojové a výukové platformy zvané „Arduino“. Tato platforma byla zvolena pro její rozšířenost, jednoduchost a rychlost vývoje SW i pro osoby s malými znalostmi v tomto oboru. Postupem času se však narazilo na několik problémů, pro které byla platforma „Arduino“ zcela opuštěna:

- nutnost použití (pouze) doporučeného krystalového oscilátoru pro procesor
- nemožnost nastavit libovolnou přenosovou rychlost sériového kanálu (z předchozího důvodu)
- použití doporučeného vývojového prostředí s názvem „Arduino IDE“, které je velmi primitivní a pro větší projekty nevhodné
- nemožnost ladění (debugování) vytvářeného programu
- možnost použití pouze doporučených / podporovaných procesorů
- některé další nevýhody

Z těchto uvedených nevýhod bylo od dříve zvoleného řešení s využitím platformy Arduino tedy zcela odstoupeno. Software v mikroprocesorech je nyní psán v prostředí „Atmel Studio“ za použití jazyka C/C++. V programu je vytvořena určitá „abstrakce“, ve které je deklarována základní komunikace s pohybovým senzorem. Konkrétní implementace jsou pak vytvořeny pro jednotlivé typy senzorů. Zatím jde o již zmíněné pohybové senzory MPU6050, MPU9150 a MPU9250. Po menší úpravě lze tedy využít v podstatě téměř jakékoli senzory. Pokud se nekomunikuje se senzorem, běží na mikroprocesorech programová smyčka čekající na pakety s příkazy z PC, respektive z centrálního modulu. Program na zaslané příkazy reaguje. Každý měřicí modul obsahuje unikátní adresu. Nejdůležitější přenášené příkazy do měřicího modulu jsou následující:

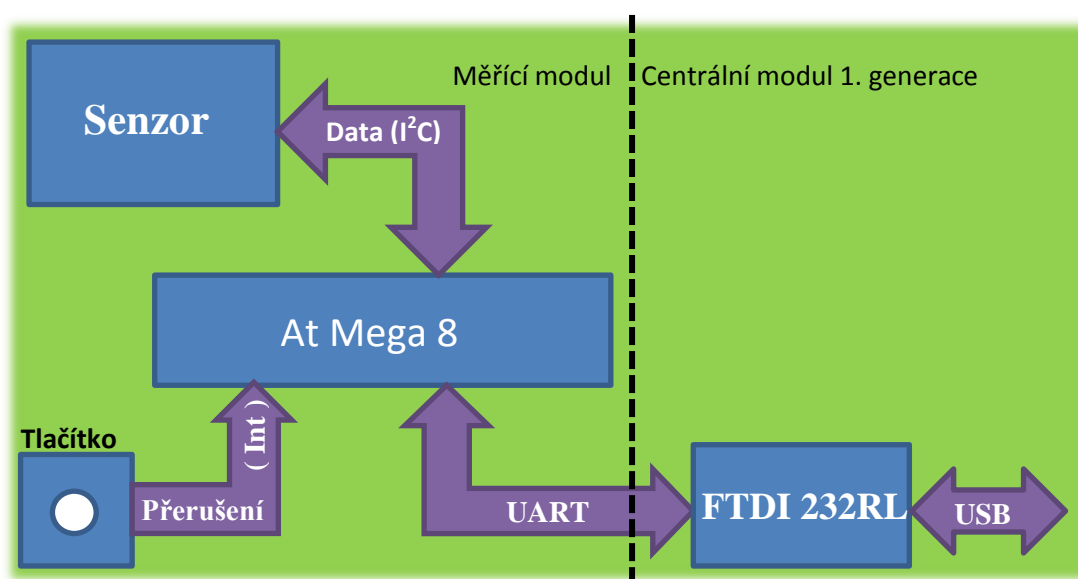
- start kontinuálního měření, kdy má procesor kontinuálně číst data ze senzoru a společně s jeho adresou je vysílat zpět do PC
- ukončení kontinuálního měření
- nastavení rychlosti kontinuálního měření
- dotaz na jednu aktuální hodnotu ze senzoru – využíváno pouze při adresní komunikaci s měřicím modulem
- žádost o informační paket, tedy dotaz na typ připojeného měřicího modulu nebo senzoru

Program v závislosti na obdrženém povelu vysílá:

- data, kdy v závislosti na připojeném senzoru vysílá hodnoty ze tří os akcelerometru a tří os gyroskopu nebo i ze tří os magnetometru
- informace o modulu, které obsahují adresu a typ modulu, typ senzoru a informace o poskytovaných datech

Poslední možnost, kdy měřicí modul vysílá nějaké informace, nastane při stisku na něm umístěného tlačítka. V tomto případě vyšle paket s adresou, číslem tlačítka a délkou stisku tlačítka v milisekundách. Zde již zcela závisí na obslužném programu, jak tuto informaci zpracuje a tedy využije.

Pro lepší přehled funkci mikroprocesoru a jeho komunikaci s PC znázorňuje Obrázek 12.



Obrázek 12 – Zobrazení komunikace v rámci celého zařízení při využití centrálního modulu 1. generace.

6.5 Centrální modul

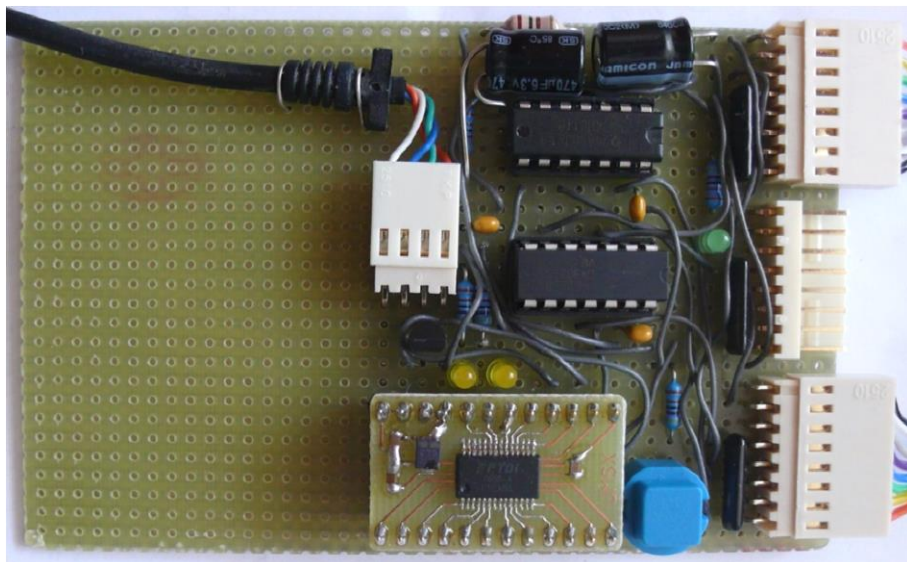
Centrální modul slouží především ke zprostředkování komunikace mezi měřicími moduly a vzdáleným PC (popřípadě tabletem, telefonem,...). Tento modul má navržené zatím dvě verze (generace). První generace zprostředkovává pouze drátovou komunikaci mezi PC (přes USB) a ostatními moduly (interní sériová komunikace), jinak neobsahuje další funkce. Druhá generace je již sofistikovanější a poskytuje možnost bezdrátové komunikace a bateriového napájení. Navíc je připravena pro záznam dat na SD kartu.

6.5.1 První generace centrálního modulu

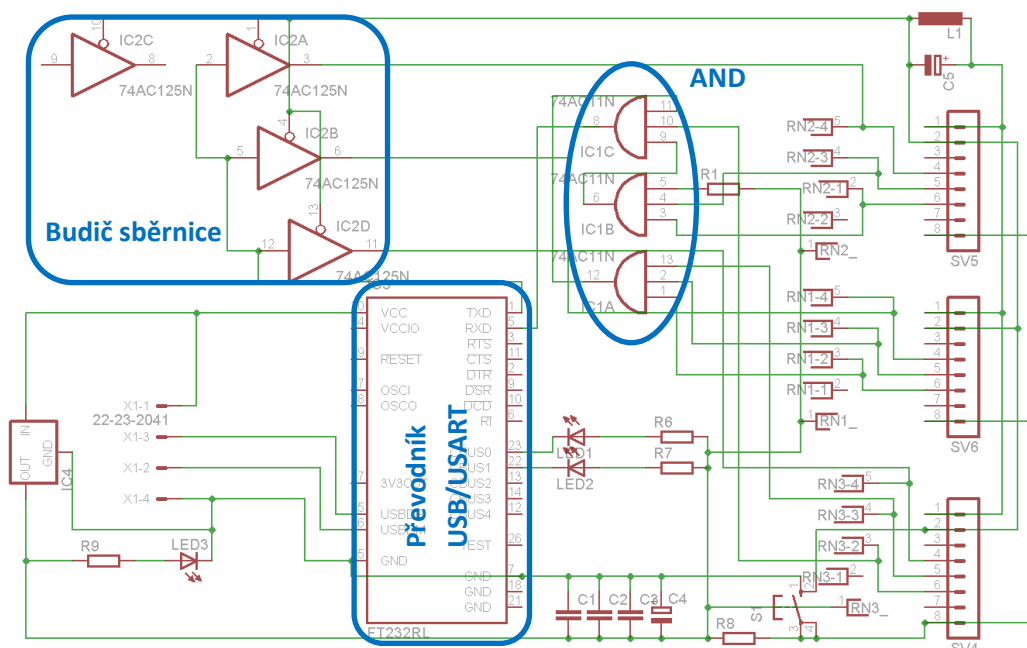
Tato verze byla zrealizována za účelem ověření činnosti měřících modulů a jejich součinnosti se softwarem v PC. Obsahuje pouze tři konektory, každý určený pro připojení jedné větve obsahující dva měřící moduly. Lze jich tedy připojit až šest. Dále centrální modul obsahuje:

- budič sběrnice 74HC125 pro data jdoucí do měřících modulů (jednotlivých větví)
- 3x3 vstupový „AND“, obvod 74HC11, který zajišťuje součin výstupních datových (sériových) komunikačních linek ze všech připojených měřících modulů. Vždy však musí komunikovat pouze jeden měřící modul, jinak dojde při běžné sériové komunikaci k nečitelnosti dat
- tlačítko pro reset všech měřících modulů
- USB kabel s převodníkem UART – USB (FTDI 232RL)
- stavové diody indikující připojené napájení a aktivitu komunikace přes USB

Tento modul ukazuje Obrázek 13. Volné místo na desce je ponecháno k dalšímu pozdějšímu rozšíření o bezdrátový modul (bluetooth) a baterii. Jeho schéma zapojení pak ukazuje Obrázek 14.

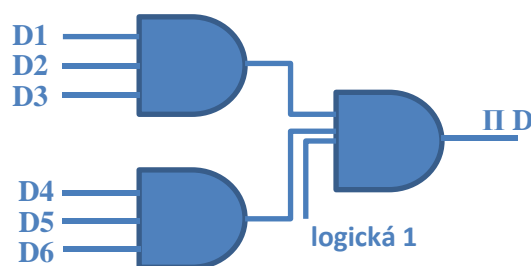


Obrázek 13 - Vzhled centrálního modulu 1. generace s připojenými 2 větvemi měřících modulů.



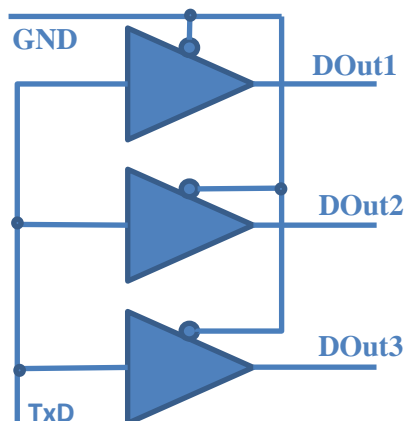
Obrázek 14 – Schéma centrálního modulu 1. generace a jeho základní části.

Data jdoucí ze sensorů do centrálního modulu jsou tedy přivedena do hradla AND, které uskuteční jejich logický součin. To znamená, že na výstupu je logická 1 jen v případě, že jsou všechny vstupy též v logické 1 (stav, kdy se nekomunikuje). Pokud kterýkoli ze vstupů změní svůj stav na logickou 0, výstup se změní taktéž na logickou 0. Signály na sériové lince jsou aktivní v logické 0. Lze tedy takto komunikovat s libovolným senzorem pomocí běžné sériové linky. Současně však tedy samozřejmě pouze s jedním měřicím modulem. Schéma zapojení tohoto hradla naznačuje Obrázek 15.



Obrázek 15 – Zobrazení propojení hradla AND v obvodu centrálního modulu.

Obvod 74HC125 je využit pro posílení sériové datové linky vedoucí do všech připojených měřicích modulů. Výstupy z tohoto obvodu tedy vedou data pro jednotlivé větve s měřicími moduly. Budič sběrnice je zde užít z toho důvodu, že každý signál je určen pro dva až tři měřicí moduly, kterým vedou delší vodiče (například z hrudi přes rameno až k předloktí nebo například z hrudi až na lýtko). V tomto případě jsou všechny datové vstupy 74HC125 připojeny na výstup (TxD) převodníku FTDI 232RL, jak naznačuje Obrázek 16.



Obrázek 16 – Zobrazení propojení vstupů budiče sběrnice v obvodu centrálního modulu.

Z činnosti první generace centrálního modulu bylo zjištěno, že pomocí dotazů po sériové lince při komunikační rychlosti 230,4 kBaudů, bylo dosaženo rychlosti měření zhruba 62 vzorků za vteřinu. A to však pouze s jedním měřícím modulem. Při použití více modulů se rychlost v podstatě přímo úměrou snižuje. Proto je nutné využít u druhé generace centrálního modulu paralelního přenosu naměřených dat pomocí modifikovaného SPI rozhraní (viz kapitola 6.6).

6.5.2 Druhá generace centrálního modulu

Pro plné využití měřících modulů je navržen centrální modul poněkud modifikovaný od předešlého, popsaného dříve. Tento další modul je navržen tak, aby byl schopen číst data z více senzorů současně, pracovat bezdrátově a měl případně možnost uložení záznamu cvičení přímo na sobě. Měřící moduly k němu jsou stále připojeny pomocí vodičů. Oproti první generaci tento modul obsahuje navíc vlastní mikroprocesor (Atmega644) a tedy případně i záznamové médium, SD kartu. Samozřejmostí je možnost bateriového napájení. Na rozdíl od předešlé verze, zde nejsou data z měřících modulů přijímána pouze na dotaz (tedy vždy pouze z 1 senzoru), ale každý měřící modul je připojen pomocí svého datového vodiče přímo na jeden pin / vstup procesoru v centrálním modulu. Tím lze v podstatě dosáhnout sběru dat ze všech měřících modulů současně, tedy s mnohem vyšší frekvencí (rychlostí). Komunikace tedy bude probíhat tak, že z procesoru na tomto centrálním modulu bude vyslán příkaz (impulz na vodiči C/D) pro odměr do všech připojených měřících modulů. Ty poté všechny poskytnou současně svá naměřená data zpět (každý však po své sériové lince). Zde již nejde o běžnou sériovou linku, ale v podstatě o vlastní implementaci SPI komunikace (hodinový a datový vodič).

6.6 Komunikace s měřicími moduly

Princip komunikace centrálního modulu první a druhé generace s měřicími moduly lze stručně shrnout následovně:

1. generace centrálního modulu:

- komunikace s měřicími moduly pomocí klasického sériového rozhraní USART
- vysílají se dotazy na jednotlivé měřicí moduly a přijímají se data vždy pouze z jednoho měřicího modulu
- napájení i přenos dat realizován pomocí USB z PC
- pro komunikaci s měřicími moduly se využívá těchto vodičů / signálů:
 - Reset
 - TxD – vysílání dat do všech senzorů
 - RxD – přijímání dat ze všech senzorů
 - Command/Data – stále na log. 0
- komunikace probíhá následovně:
 - A) Reset měřících modulů
 - B) vyslání žádosti (z PC) o data s konkrétní adresou (určené tedy pouze jednomu měřicímu modulu) po vodiči TxD,
 - C) čekání (v PC) na data přichozí po vodiči RxD
 - D) po příjmu dat opakování od bodu B s adresou dalšího měřicího modulu

Tato první generace byla vytvořena pro ověření činnosti měřících modulů, vytvořené aplikace v PC a rovněž pro ověření komunikace mezi PC a měřicími moduly

2. Generace centrálního modulu:

- autonomní zařízení
- bateriové napájení
- paměťová karta (podpora)
- bezdrátový přenos dat do vzdáleného PC
- komunikace s měřicími moduly buďto pomocí USART (zejména jejich konfigurace) nebo pomocí modifikovaného SPI (zejména přenos dat), umožněny však obě varianty
- komunikace pomocí USART – stejná jako u centrálního modulu první generace, ale s měřicími moduly nekomunikuje přímo vzdálené PC, ale právě tento centrální modul (tedy v něm obsažený procesor)
- komunikace pomocí modifikovaného SPI:
- využívají se vodiče / signály:
 - Reset – reset všech měřících modulů
 - Command/Data (C/D) – log. 0 pro dotazy / příkazy, log. 1 pro data
 - TxD – při C/D = 0: data pro všechny měřicí moduly, při C/D = 1: hodiny (CLK) pro všechny měřicí moduly při přenosu dat (paralelně po SPI ze všech měřících modulů)
 - RxD (počet dán počtem připojených měřících modulů) – paralelně SPI data ze všech měřících modulů
- komunikace probíhá následovně:
 - A) Reset – reset všech měřících modulů

- B) při Command/Data = 0 se používá stejná komunikace jako u první generace – slouží hlavně pro konfiguraci měřících modulů
- C) s každou změnou C/D z 0 na 1 měřící modul vykoná odměr na pohybovém senzoru a připraví data pro vyslání do centrálního modulu
- D) data se vyšlou z měřícího modulu do centrálního modulu pomocí modifikovaného SPI, kdy:
 - TxD – představuje hodiny (CLK) pro přenos dat ze senzorů (jeden puls hodin přenáší vždy jeden bit dat)
 - RxD – data z každého měřícího modulu mají svůj vlastní vstup do procesoru v centrálním modulu (data z měřícího modulu se nepřenášejí sériově ale paralelně ze všech měřících modulů současně)
- po vyčtení dat se proces opakuje od bodu C

Tento centrální modul druhé generace umožňuje rychlejší měření, jednodušší komunikaci s připojeným zařízením (PC, tablet, telefon, televize,...) a poskytuje již bezdrátové připojení. Navíc bude umožněno ukládání dat na SD kartu.

Jak už bylo uvedeno, komunikace mezi PC (první generace) nebo centrálním modulem (druhé generace) a měřícími moduly je realizována prostřednictvím sériového rozhraní USART případně modifikovaného SPI. Přes tato rozhraní jsou zasílány předem definované pakety. Jedná se o pakety buďto s daty, nebo příkazy. Protože s těmito daty, respektive příkazy, musí komunikovat i software v mikroprocesorech, umístěných na měřících modulech, byla jejich struktura již naznačena v kapitole 6.4. Protokol (struktura paketů) je tedy následující:

- paket začíná znakem dvojtečka „:“
- poté následuje adresa modulu, buďto pro který je paket určen, nebo ze kterého byl paket odeslán
- další znak určuje typ paketu:
 - C – příkazy (Command), do měřícího modulu
 - I – informace (Info), z měřícího modulu
 - D – data (Data), z měřícího modulu
 - B – stisk tlačítka (Button), z měřícího modulu
- poté následuje mezera a další část paketu se liší podle jeho typu
- každý paket končí znakem „#“
- konkrétně tento proces vypadá následovně:
 - Příkazy z PC (centrálního modulu):
 - „:AC V=v D=d#:“
 - : – začátek paketu
 - A – adresa
 - C – příkaz
 - V – číslo příkazu, v - hodnota
 - D – případná data, d - hodnota
 - # – konec paketu
 - základní příkazy jsou:
 - V=4 – žádost o informační paket
 - V=5 D=1 – spuštění kontinuálního měření

- V=5 D=0 – zastavení kontinuálního měření
- V=7 – nastavení rychlosti měření (D = vzorků za vteřinu)
- V=9 – žádost o aktuální data ze senzoru (1 hodnota)
- Odpovědi z měřícího modulu:
 - informační paket – „:AI T=t S=s D=d#“
 - : – začátek paketu
 - A – adresa
 - I – informační paket
 - T – typ modulu
 - S – typ senzoru
 - D – použitelná data (například AGM znamená, že lze posílat data z akcelerometru, gyroskopu a magnetometru)
 - # – konec paketu
 - datový paket – „:AD A=a_x, a_y, a_z G= g_x, g_y, g_z M= m_x, m_y, m_z #“
 - :A – stejné jako u informačního paketu
 - a_x, a_y, a_z – data z akcelerometru
 - g_x, g_y, g_z – data z gyroskopu
 - m_x, m_y, m_z – data z magnetometru
 - „Button“ paket – „:AB I=i T=t“
 - : A – stejné jako u předchozích paketů
 - B – reakce na stisk tlačítka
 - i – číslo tlačítka (pokud by časem bylo na modulu více než jedno tlačítko)
 - t – délka stisku tlačítka v milisekundách

Příklad základní komunikace se senzorem s adresou „1“ lze popsat takto:

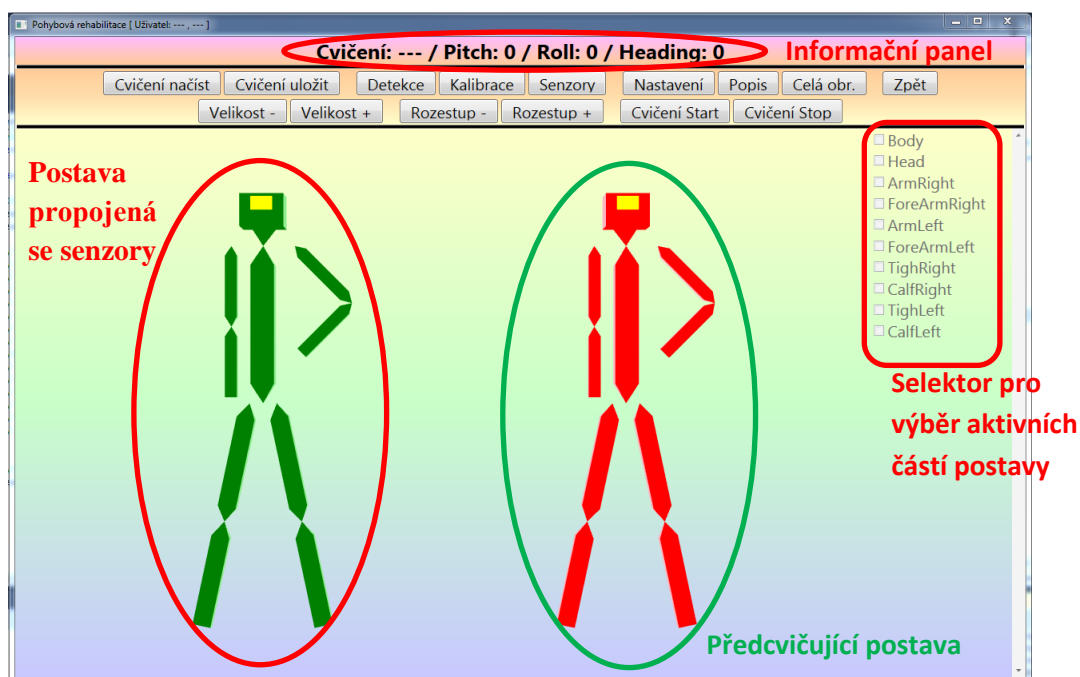
- Žádost o informační paket: „:1C V=4#:“
- Příjem informačního paketu: „:1I T=AccelGyroMag S=MPU6050 D=AGM#“
- Žádost o data (jeden odměr): „:1C V=9#:“
- Příjem dat (jednoho odměru): „:1D A=90, 60, 90 G= 0, 5, 10 M= 33, 20, 0 #“

Pakety / povely s adresou „0“ jsou určeny všem měřícím modulům současně (společná konfigurace) a na tyto pakety / povely měřící modul (v současné verzi) neodpovídá.

7 Obslužná úloha na PC

Jak již bylo uvedeno, práce na softwaru (obslužné úloze na PC) započala již v rámci bakalářské práce. Původní software byl ovšem určený pouze pro vytvoření 3D vizualizace naměřených dat, ve formě postavy člověka, a to pouze z jednoho připojeného měřicího modulu. Z výsledků mělo být patrné, zda je vhodné se dále vývojem takového zařízení zabývat. Právě proto byl software jednoduchý a neměl žádné skutečné spojení s reálnými senzory. Vzhledem k celkem experimentální povaze práce nebyl software také zcela implementačně přehledný. Nastala tedy potřeba software nejdříve výrazně přepracovat (zjednodušit a zpřehlednit), aby bylo možno na původní navázat a dále jej rozvíjet. Po této nutné úpravě byla část software pro zobrazení postavy člověka přidána do rozsáhlejší aplikace, jejímž tvůrcem je vedoucí této práce Petr Novák, a která obsahuje více úloh pro další různé pohybové senzory. Poté byly vytyčeny cíle, které by měla tato část aplikace splňovat (oproti verzi z bakalářské práce):

- zobrazit druhou postavu, která by sloužila k předcvičování
- umožnit nahrávání, ukládání a přehrávání pohybů pro snadnou tvorbu cviků
- umožnit logické svázání libovolného senzoru s libovolnou částí těla (vzhledem k rozmanitosti možných cviků, kdy na každý typ cviku může být potřeba odlišný počet senzorů)
- kalibrace používaných hodnot – každý člověk má jinak vytvarovanou postavu, navíc připevnění modulů obsahuje jistou vůli a moduly jsou tedy po každém nasazení samozřejmě v poněkud odlišné výchozí poloze



Obrázek 17 – Vzhled úlohy po spuštění a její základní části.

Činnost úlohy lze stručně popsat v těchto krocích:

- spuštění úlohy – nejsou zjištěny žádné připojené měřící moduly a tudíž v selektoru v pravé části obrazovky (viz Obrázek 17) nejsou ani zatrženy žádné části těla
- detekce připojených měřících modulů – po stisku tlačítka „Detekce“, aplikace detekuje aktuálně připojené senzory, a pokud tyto senzory byly při posledním ukončení aplikace přiřazeny k nějaké části těla, tak se tyto části těla v selektoru zatrhnou / povolí a lze je aktivovat
- přiřazení – pokud je potřeba změnit přiřazení mezi měřícími moduly a částmi těla, následuje tento postup:
 - A) stisk tlačítka „Senzory“ (v aplikaci)
 - B) kliknutí na část těla určenou k přiřazení (na postavě člověka)
 - C) stisk tlačítka na měřícím modulu určeném k tomuto přiřazení
 - D) pro další část těla opakovat od bodu B
 - E) nebo pro ukončení znovu stisknout tlačítko „Senzory“
 - F) přijmout nové nastavení
- kalibrace – nutno v podstatě vykonat po každém zapnutí aplikace:
 - A) stisk tlačítka „Kalibrace“
 - B) 5 vteřin stát v pozoru (všechny senzory ve výchozí poloze)
- výběr cvičení – nutno vykonat vždy na začátku nebo při změně cviku:
 - A) stisk tlačítka „Cvičení načíst“
 - B) v dialogu dostupných cvičení dvojklikem označit vybrané cvičení
- zahájení předcvičování – stisk tlačítka „Cvičení Start“
- ukončení cvičení – stisk tlačítka „Cvičení Stop“
- záznam cvičení – pro uložení nového cviku nebo pro vytvoření záznamu cvičení:
 - A) stisk tlačítka „Cvičení uložit“
 - B) 5 vteřin příprava do začátku nahrávání
 - C) nahrávání běží do opětovného stisku tlačítka „Cvičení uložit“
 - D) v zobrazeném dialogu vyplnit název cvičení a potvrdit jeho uložení

7.1 Implementace softwaru na PC

Celý software je psán v Microsoft Visual Studiu 2013 Express jazykem C#. Využívá se zejména část .NET Framework nazývaná WPF. Ta obsahuje spoustu užitečných knihoven pro práci s formuláři a grafikou (i 3D).

Jelikož se jedná o celkem složitou část technologie WPF pro zobrazení 3D objektů, tak zde jde pouze o popis struktury celé implementace, nikoli tedy o komplexní a podrobný popis vytvořeného zdrojového kódu. Tato kapitola je tedy určena zejména těm, kteří budou na tuto práci v budoucnu navazovat a tudíž ji dále rozvíjet. Podrobný popis implementace 3D postavy a vytváření jejího pohybu by mohl značně a nevhodně zvětšit obsah tohoto textu.

7.1.1 Hlavní třídy úlohy

Pro tvorbu této pohybové úlohy byl využit tzv. framework pro pohybové úlohy dodaný vedoucím této práce. Ten však zajišťuje pouze přenos dat ze senzorů na nejnižší úrovni a základní podporu pro

tvorbu vlastní úlohy. Každá vlastní tzv. mini-aplikace je v tomto frameworku vytvořena jako tzv. „úloha“. Základní činnost vytvořené úlohy obstarávají tyto třídy:

Třída „TaskCvičeníMain“: Zajišťuje chod celé této úlohy. Obsahuje metody, které úlohu spouštějí, inicializují a ukončují. Jsou v ní také definována tlačítka, včetně jejich obsluhy. Další důležitou činností je zpracování hodnot přicházejících z měřících modulů.

Třída „DataMovement“: Pracuje se surovými daty. Jsou v ní obsaženy metody pro výpočet úhlů z naměřených hodnot a rovněž pro průměrování nastavené v rámci celé aplikace.

Třída „Predcvicovani“: Sdružuje a definuje základní atributy a metody potřebné k předcvičování.

Třída „PredcvicWork“: Určena pro výpočet a vykonání pohybu jednou částí těla.

Zde jsou tyto třídy jednotlivě podrobně popsány. Především tedy základní metody, vlastnosti a jejich funkce:

třída „TaskCvičeníMain“:

TaskOnCreate() – Při prvním spuštění této úlohy v rámci celé aplikace inicializuje grafické rozložení úlohy a definuje:

- umístění ovládacích tlačítek
- inicializaci informačního panelu úlohy
- nastavení rozložení plochy úlohy
- spuštění komunikace se senzory

TaskOnInit() – Pokaždé před zobrazením úlohy, tedy při každém spuštění této úlohy ze základní aplikace se vykoná:

- nastavení komunikace s moduly
- načtení informací o propojení měřících modulů s částmi těla z uloženého XML souboru "SensorsToAddress.xml"
- inicializace pole instancí „DataMovement“ moveDataMy (vysvětleno dále) a nastavení průměrování na datech v něm uložených

TaskOnNew() – Přidá do „gridu“ celé úlohy:

- 3D scénu (ViewPort3D)
- panel pro tlačítka
- panel pro „selektor“
- volá funkci CreatePostavy() (viz dále), která vytvoří postavy a přidá je do 3D scény

OnChange() – Při příjmu dat z měřících modulů, respektive po jejich umístění do struktury moveDataMy (viz dále), u každého měřícího modulu spárovaného s některou částí postavy:

- zkontroluje, zda je příslušná část „selektoru“ (viz Obrázek 17) zatržena

- podle části postavy, ke které patří právě zpracovávaná data, přepočte úhly, jak ukazuje Obrázek 19
- tyto data vloží do slovníku s daty postavy (postava.Data – vysvětleno dále)
- po přepočtu všech částí provede obnovení grafiky (postava.Update() – viz dále), čímž se nastaví pozice / poloha částí postavy dle aktuálních hodnot

OnInfoToTask (...) – Při příjmu informačního paketu, což je užíváno při automatické detekci připojených modulů, uloží do proměnné addressReceived hodnotu přijaté / existující adresy – ta v tomto případě není dále využita, ale modul, jemuž adresa náleží, je označen za aktivní.

OnDataToTaskCoplex (...) – Při příjmu nových dat, před zavoláním funkce onDataChange() (viz dříve), předá zpracování dat do GUI vlákna tím, že:

- kopíruje přijatá data do již zmíněné struktury moveDataMy
- volá funkci onDataChange() (viz dříve)

OnCmdToTask (...) – Při stisku tlačítka na měřícím modulu, čehož je využíváno při párování částí postavy s adresami měřících modulů, volá funkci setSensorAndPosition (viz dále) s adresou modulu, na kterém bylo tlačítko stisknuto.

SensorsCallib() – Zajišťuje kalibraci po stisku tlačítka „Kalibrace“ s následující posloupností:

- spustí časovač, který „tiká“ s frekvencí 1 Hz
- zobrazí v informačním panelu odpočet s dobou 5s
- po 4 sekundách zavolá funkci Callib() (viz dále)
- po 5 sekundách vrátí informační panel do původního stavu, aktualizuje grafiku postavy a zastaví čítač

Callib() – Kopíruje hodnoty ze slovníku postava.Data do postava.dataCallib (hloubková kopie) – obě tyto struktury jsou vysvětleny dále.

SetSensorPositionAndAddress(...) – Při stisku tlačítka na měřícím modulu je tato funkce volána s adresou daného modulu a při kliknutí na část postavy je volána s informací o dané části postavy. Výsledkem je přiřazení části postavy k adrese měřícího modulu.

SensorsPosition() – Zajišťuje přiřazování měřících modulů k částem postavy.

Při prvním stisku tlačítka „Senzory“:

- označí tlačítko za stisknuté / aktivní
- v informačním panelu nastaví informaci o přiřazování senzorů
- v komunikaci se senzory nastaví, aby se nedotazovalo na data, ale aby se pouze sledovaly příchozí pakety
- nyní je přiřazování v režii již popsané SetSensorPositionAndAddress (...), která je střídavě volána z kliknutí na postavu a z příchozího paketu o stisku tlačítka na měřícím modulu

Při druhém stisku téhož tlačítka:

- zobrazení dotazu, zda chce uživatel přijmout nové nastavení
 - při NE – navráčeno vše do původního stavu a odznačení tlačítka „Senzory“

- při ANO – smazání původního přiřazení nahrazení novým
 - dále uložení nového nastavení do příslušného XML souboru
 - volání funkce updateSelect() (viz dále)

updateSelect() – Povolí a zaškrtně v „selektoru“ ty části těla, ke kterým je přiřazen měřící modul.

CviceniLoad() – Zajišťuje načtení cvičení po stisku tlačítka „Cvičení načíst“:

- vymaže vzorPohybList (viz dále)
- zavolá funkci GetCvicNames(), která zjistí všechny XML soubory s cviky uložené v adresáři a jejich názvy uloží do pole
- zobrazí dialog obsahující dostupné názvy cvičení
- při dvojkliku na nějaký název zavře dialog a název vybraného cvičení uloží do infoCvitName.Value, což je hodnota názvu aktuálního cvičení v informačním panelu
- data z vybraného souboru uloží do PredcvicListu (viz dále)

CviceniSave() - Řídí proces nahrávání pohybu po stisku tlačítka „Cvičení uložit“

- při prvním stisku:
 - tlačítko zobrazeno stisknuté / aktivní
 - v proměnné „int“ i je ukládán počet vteřin, po které nahrávání běží
 - i začíná na hodnotě -5, čímž je dána přípravná doba do začátku nahrávání
 - podle částí těla zaškrtnutých v „selektoru“ je vytvořen slovník, kde „selektor“ (checkbox), je klíčem a struktura „DataValueItemsTimeSpanPoint3DInt32“, do které lze ukládat hodnoty natočení (int32) společně s časem (TimeSpan), je hodnotou
 - spuštění časovače (zatím s periodou 0,2s)
 - při každém „Ticku“ časovače jsou hodnoty z daných částí těla ukládány ve dříve zmíněné struktuře.
- při druhém stisku:
 - zastavení časovače
 - zjištění názvů uložených cvičení
 - zobrazení dialogu pro zadání názvu cvičení s tlačítky pro uložení a pro zrušení
 - při zvolení uložení je provedena kontrola, zda daný název cvičení neexistuje a zda název není prázdný:
 - pokud je vše v pořádku, dojde k uložení do XML souboru se zadaným názvem doplněným o předponu „Cviceni“ a příponu „.xml“
 - pokud ne, je zobrazena informační chybová zpráva
 - pokud se uživatel rozhodne ukládání zrušit, je vše navráceno do původního stavu bez uložení

Základní komunikaci se senzory zajišťuje aplikace tvořená frameworkem, jíž je tato úloha součástí, proto zde nebude podrobně popsána. Při příjmu dat z měřících modulů je volána metoda OnDataToTaskCoplex() kopírující data přijatá z každého senzoru do pole instancí třídy „DataMovement“ (vysvětleno dále) a poté již volá funkci „OnDataChanged()“ ve zde vytvářené úloze. Tyto metody a třída DataMovement budou popsány dále.

třída „DataMovement“:

- EverageSet() – nastavení parametrů průměrování dle globálního nastavení aplikace
- Everage() – vykoná jeden krok průměrování (je potřeba volat vždy s novými daty)
- Angles() – výpočet úhlů z přijatých, popřípadě průměrovaných hodnot

Po aplikaci uvedených funkcí jsou v DataMovement přístupné tyto hodnoty (pro použití tedy DataMovement.*):

- Ax, Ay, Az – surová data z měřících modulů
- outAx, outAy, outAz – zprůměrované hodnoty z měřících modulů
- AngleX, AngleY a AngleZ – hodnoty úhlů, které jsou v kapitole 7.2 označeny jako α_x , α_y a α_z . Jsou také totožné s hodnotami Pitch, Roll a Heading

Třída „PredcivicWork“:

- „List“ values – seznam úhlů s časy, ve kterých se má postava nacházet
- „int“ index – index konce právě vykonávaného pohybu
- „double“ AngleActualX, AngleActualY a AngleActualZ – aktuální úhel
- „double“ AngleStepX, AngleStepY a AngleStepZ – vypočtený krok pohybu

SegmentNew (Values) – Zajišťuje zobrazení polohy postavy dle aktuálních úhlů ze seznamu a stanovuje výpočet kroků pro přechod do následujících úhlů v seznamu:

- nastaví vstupní hodnoty (Values) na hodnoty začátku daného segmentu pohybu (segment = pohyb od jedné polohy v seznamu ke druhé)
- zjistí délku trvání aktuálního segmentu pohybu podle času segmentu následujícího (pokud aktuální není poslední)
- vypočítají se aktuální kroky ve všech třech osách (AngleStepX, AngleStepY a AngleStepZ):
 - rozdíl úhlu aktuálního a následujícího se rozdělí podle délky trvání a rychlosti čítače

$$step = \frac{\alpha_{x_{i+1}} - \alpha_{x_i}}{\frac{t}{T}}$$

step – krok [°/Tick]

α_x – pole úhlů [°]

t – délka trvání segmentu [ms]

T – rychlost čítače (doba mezi “Ticky”) [ms]

Step (Values) – Zajišťuje vykonání jednoho kroku vypočteného v předchozí funkci:

- připočte k hodnotám AngleActualX, AngleActualY a AngleActualZ hodnoty příslušných kroků AngleStepX, AngleStepY a AngleStepZ (změna se provede jen pokud AngleStep nejsou hodnoty „NaN“ – tato situace nastane, pokud se daná část postavy v dané ose nepohybuje)
- nové aktuální hodnoty vloží do vstupních „Values“
- vrátí hodnotu „true“ pokud byli nějaké hodnoty změněny, jinak vrátí „false“

Třída „Predcivicovani“:

- „DispatcherTimer“ timerPredcvivovani – časovač pro vytvoření pohybu předcvičování
- „int“ TimerInterval – interval v milisekundách mezi „tikama“ časovače
- „DataValueInfoList“ PredcvicList – struktura obsahující stejná data jako již uvedený, XML soubor; tedy částí těla, se kterými se cvičí a u každé části těla její úhly s časy
- „Dictionary“ predcvicPartsWorks – zpracovaná data z PredcvicListu rozdělená dle části těla
- „DateTime“ predcvicovaniStartTime – čas začátku předcvičování,
- „bool“ predcvicOpakovat – „true“ pro opakování cvičení stále dokola

Start() – Spouští předcvičování po stisku tlačítka „Cvičení Start“ :

- vyčistí predcvicPartsWorks
- přidá do predcvicPartsWorks data z PredcvicListu
- uloží do predcvicovaniStartTime aktuální čas
- spustí časovač timerPredcvivovani

Stop() – Po stisku tlačítka „Cvičení Stop“ zastaví časovač timerPredcvivovani, čímž ukončí předcvičování.

TimerPredcvivovaniOnTick(...) – EventListener vyvolávaný „Tickem“ časovače timerPredcvivovani, při kterém se prochází list s hodnotami pro předcvičování.

- ověří, zda je v PredcvicListu nějaké cvičení, popřípadě, zda toto cvičení není u konce
- pokud je cvičení u konce a bool predcvicOpakovat je „true“, zavolá se znovu metoda Start()
- pokud není u konce, ověří se, zda je aktuální čas cvičení menší než čas další položky v predcvicPartWorks:
 - pokud ano (je aktuální čas menší) vykoná se již popsany Step(...), neboli u dat dané části postavy (instance třídy Clovek) dojde k aktualizaci hodnot a poté se provede Update(), tudíž se aktualizuje grafika postavy
 - pokud ne (aktuální čas je větší) dojde k vytvoření nové části pohybu pomocí již zmíněné funkce SegmentNew(...)

7.1.2 Základní zobrazení 3D postavy

Struktura grafického rozhraní pro zobrazení 3D objektů se skládá z těchto elementů:

```

Viewport3D           // je pouze jeden a obsahuje celou 3D scénu
  ContainerUIElement3D // tím je „obalena“ každá postava (možnost klikání)
    ModelUIElement3D  // tím je „obalena“ každá část těla (možnost klikání)
      Model3DGroup     // část těla (na této úrovni se nastavuje osvětlení)
        GeometryModel3D // trojúhelníky, ze kterých je stvořena každá část těla
  
```

Tento popis struktury znamená, že pro zobrazení 3D scény ve WPF je potřeba nejprve použít položku Viewport3D. Do ní se vkládají „kontejnery“ (ContainerUIElement3D), které obsahují grafiku celé postavy. Dále se skládají z elementů (ModelUIElement3D), jež v sobě obsahují model (Model3DGroup) s grafikou jednotlivých částí postavy. Každá tato část postavy se stejně jako všechna 3D grafika ve WPF skládá z barevných trojúhelníků, jež jsou instancí WPF třídy GeometryModel3D.

Základní třídy potřebné pro vytvoření postavy, pro její přidání do viewportu a pro základní operace s ní, jsou:

Třída „Postava“ – Sdružuje funkce potřebné k vytváření postav a k jejich přidání do viewportu3D (viz dříve)

Třída „Clovek“ – Její instancí je každá vytvořená postava. Obsahuje všechny potřebné parametry pro ovládání postavy a pro definování jejího 3D modelu.

Třída „TransXYZ“ – Přesně definuje každou transformaci, včetně toho, kterého modelu (části těla) se daná transformace týká. Toho je využito všude, kde k nějakým transformacím dochází.

Enum „EnumTransXYZ“ – Obsahuje názvy všech instancí třídy TransXYZ (neboli všech transformací), potřebných pro zobrazení a pohyb jedné postavy.

Třída „Values“ – Určena pro uchovávání úhlů natočení okolo 3 os. Data jsou tedy ve formátu double (AngleX, AngleY a AngleZ).

Zde jsou tyto třídy podrobně popsány a vysvětleny:

Třída „Postava“:

CreatePostavy() – Zde jsou vytvářeny postavy, a jsou zde uskutečněny všechny kroky potřebné k jejich zobrazení ve 3D scéně:

- přiřazuje pro Viewport3D kameru
- jsou zde definovány obě postavy (instance třídy Clovek – viz dále) a jsou přidány do Viewportu3D
- pro postavu ovládanou uživatelem je zde volána funkce ElementsToStackPanel(...), která vytvoří a zobrazí selektor pro povolení pohyblivosti vybraných částí těla

CreatePostava() – Při vytvoření instance třídy Clovek (viz dále), vytvoří potřebnou grafiku všech částí těla tímto postupem:

- vytvoří všechny „Point3D“, ve kterých bude docházet k ohybům (klouby)
- pro každou část těla zavolá funkci CreateCube(...) – která vytvoří kvádr s jedním, či dvěma jehlany na koncích kvádrů. Její funkce je tedy:
 - k zadanému Model3DGroup zobrazí stěny ve tvaru kvádrů o zadané velikosti a o zadaných barvách
 - pokud je vytvářena konečná část postavy (například předloktí nebo lýtko), je vytvořen jehlan na jednom konci kvádrů, pokud nejde o konečnou část těla, jsou jehlany na obou koncích kvádrů
 - v této funkci je také každému modelu přiřazeno světlo, které je definováno v rámci celé třídy
- pro každou část těla rovněž nastaví příslušnou TransXYZ, tedy posunutí a bod, okolo kterého bude probíhat rotace – ty se zde inicializují s nulovými hodnotami
- nakonec je zavolána funkce Rozloz(...), která propojí další potřebné transformace s částmi těla. Tato problematika bude podrobněji vysvětlena v kapitole 7.1.3

SetDist () – Pro přehledné zobrazení, zvětšuje, či zmenšuje mezeru mezi postavami.

setSize () – Zvětšuje, či zmenšuje zobrazené postavy, respektive posouvá kameru blíže nebo dále.

Třída „Clovek“:

- „ContainerUIElement3D“ allparts – grafická komponenta obsahující grafiku všech částí těla, jak zobrazuje úvod této kapitoly
- „Dictionary <EnumTransXYZ, TransXYZ>“ transforms – slovník přiřazující každému prvku z EnumTransXYZ vlastní instanci TransXYZ (viz dále)
- „Dictionary <EnumTransXYZ, Values>“ defaultValues – přiřazující každé části těla počáteční hodnoty
- „Dictionary<EnumTransXYZ, Values>“ data – aktuální data pro každou část těla
- „Dictionary<Model3DGroup, EnumTransXYZ>“ modelToEnum – slovník přiřazující 3D modelu hodnotu z EnumTransXYZ (využito pro rozpoznání zdroje události při kliknutí na část těla)

Clovek(...) – Konstruktor, využívaný při vytváření nové postavy, jehož parametry jsou:

- barva zepředu a zezadu
- barva z boků
- název

CopyData() – Vrací hloubkovou kopii slovníku s aktuálními daty

CopyDefaultValue(...) – Vrací hloubkovou kopii počátečních „Values“ z vybrané části postavy.

setDefaultValues() – Nastaví všechny transformace na počáteční / výchozí hodnoty.

Update() – Aktualizuje transformace postavy, podle obsahu jejího slovníku „data“.

Třída „TransXYZ“:

- „RotateTransform3D“ RotateX, RotateY a RotateZ – rotační transformace postupně ve všech osách
- „Point3D“ Center – střed rotací – bod, kolem kterého se daná část postavy otáčí
- „TranslateTransform3D“ translate – posuvná (translační) transformace, která označuje umístění dané části těla
- „string“ Name – název instance
- „Prumer“ Prumer – instance třídy, která uchovává 10 hodnot zpět a počítá z nich průměr; obsahuje metody:
 - Add(Values) – přidání nové hodnoty, kdy se nejstarší hodnota smaže
 - Prum() – metoda, která vrací hodnotu aritmetického průměru z posledních 10 hodnot
 - sameValues(Values) – metoda, která zaplní všechny hodnoty v řadě hodnotou v argumentu – průměr je pak právě tato hodnota
- „Model3DGroup“ Model – model (část těla), ke kterému se dané transformace vážou

Tyto uvedené třídy postačují k základnímu zobrazení úlohy. Poté pro pohyb postavy již stačí aktualizovat hodnoty ve slovníku „postava.Data[EnumTransXYZ]“ a u dané postavy „zavolat“ funkci Update() ze třídy Clovek (tedy postava.Update()). Pokud budou nové hodnoty odlišné od těch předchozích, tak se postava vždy pohne do nově předané polohy. Pohyb postav je ovládán / řízen z metod popsanych v kapitole 7.1.1.

7.1.3 Transformace částí těla

Vzhledem k tomu, že každá část těla obsahuje vlastní transformaci, je potřeba vždy nastavovat transformace i všech částí těla, které jsou za ní připojeny. Jinak by došlo k rozpojení postavy. Jelikož tyto transformace musí být přidány ve vhodném pořadí, bude zde toto pořadí vysvětleno. Jelikož je postava symetrická (i z pohledu transformací), budou zde popsány transformace pouze jedné částí postavy bez ohledu na to, zda jde o její pravou nebo levou část.

Jak již bylo řečeno, ke každé části těla se váže instance třídy TransXYZ. Aby bylo možné se k této instanci dostat, je ve třídě člověk vytvořena funkce GetTransXYZ, jejímž vstupem je EnumTransXYZ. V něm jsou obsaženy názvy všech instancí TransXYZ v rámci jedné postavy. Každá část těla má jednu svou „hlavní“ TransXYZ, ve které má svou grafickou reprezentaci a úhly, jež ji ovládají. Také obsahuje translační (posuvnou) transformaci. Tyto „hlavní“ TransXYZ mají jednoduché názvy jako Body, Head, ArmRight, atd. Dále má každá část postavy několik „vedlejších“ TransXYZ, jejichž model je pouze ukazatelem na model těch „hlavních“ a z nichž se dále využívají pouze rotační transformace a bod, okolo kterého se otáčí. EnumTransXYZ a třída TransXYZ byly vysvětleny dříve.

Každá část postavy musí projít stejnými rotačními transformacemi, jakými prošly předchozí částí postavy, na níž tato další část postavy navazuje. Pokud má však tato část postavy zůstat v původní poloze (například kolmo k zemi), musí také projít inverzními transformacemi k již zmíněným transformacím. Tyto inverzní transformace mají opačnou polaritu (znaménko) oproti neinverzním a také musí být v opačném pořadí podle os. Pokud však pro danou část postavy není přiřazen měřicí modul (nebo pokud je v selektoru daná část), neprovádí se inverzní transformace vůči částem s měřicím modulem. To znamená, že pokud na předloktí nebude měřicí modul, ale na paži bude, bude se ruka chovat jako celek. Pokud nebude měřicí modul ani na paži, bude celá ruka v původní pozici vůči celému tělu. Konkrétně to tedy vypadá takto:

Tělo (Body) – Je vykresleno od středu souřadnic, proto není potřeba translační transformace. Tělo se otáčí kolem středu souřadnic, kde se nachází oblast pánve. Protože od těla se teprve odvíjejí ostatní části těla, je zde pouze transformace složená z úhlů natočení těla. Tedy:

- rotační transformace v pořadí RotateX, RotateY, RotateZ (dáno tím, že střed otáčení je dole)

Hlava (Head) – Navazuje na tělo a není umístěna ve středu souřadnic. Tím jsou transformace již poněkud složitější. Je tedy potřeba uskutečnit tyto transformace:

- translační transformace, posouvající tuto část těla do oblasti krku
- rotační transformace v pořadí RotateX, RotateY, RotateZ (střed otáčení je dole)
- inverzní rotační transformace k transformaci těla, pouze se středem v krku (tato transformace je v pořadí RotateZ, RotateY, RotateX, s opačnými hodnotami než má

transformace těla – hodnoty uloženy v TransXYZ, která odpovídá EnumTransXYZ.inverseBodyTransForHead),

- tato transformace je aplikována pouze v případě, že je hlava zatržena v selektoru (viz Obrázek 17)
- stejná transformace jako má tělo

Paže (Arm) – Navazuje na tělo a není umístěna ve středu souřadnic. Transformace jsou tedy podobné jako u hlavy:

- translační transformace, posouvající tuto část těla do oblasti lokte
- rotační transformace v pořadí RotateZ, RotateY, RotateX (střed otáčení je nahoře)
- inverzní rotační transformace k transformaci těla (jako v předchozím bodě), ale se středem v rameni
- stejná transformace jako má tělo

Předloktí (ForeArm) – Navazuje na paži, která navazuje na tělo. Proto je zde transformací nejvíce:

- translační transformace, posouvající tuto část těla do oblasti zápěstí
- rotační transformace v pořadí RotateZ, RotateY, RotateX (střed otáčení je nahoře)
- inverzní rotační transformace k transformaci paže (pořadí X, Y a Z, znaménka opět opačná a střed otáčení v lokti - hodnoty uloženy v TransXYZ, která odpovídá EnumTransXYZ.inverseLeftArmTrans, respektive EnumTransXYZ.inverseRightArmTrans)
- rotační transformace paže (viz předchozí bod)
- inverzní rotace těla (viz předchozí bod)
- rotace těla

Stehno (Thigh) – Navazuje na tělo, avšak v místě, které se nehýbe. Tudíž nejsou potřebné transformace týkající se těla. Tedy:

- translační transformace, posouvající tuto část těla do oblasti kolene
- rotační transformace stehna v pořadí rotací podle os Z, Y, X

Lýtka (Calf) – Jelikož navazuje na stehno, tak:

- translační transformace, posouvající tuto část těla do oblasti kotníku
- rotační transformace lýtky v pořadí rotací podle os Z, Y, X
- inverzní transformace k transformaci stehna (opět opačná pořadí os a opačná znaménka u hodnot rotací)
- rotace stehna

Hodnoty inverzních rotací jsou v režii již popsané metody Update(), rozlišující části těla na ty, které jsou v selektoru (viz Obrázek 17) zatrženy a na ty, co nikoliv.

7.2 Přepočty naměřených hodnot

Cílem této práce není vytvářet algoritmy nebo jiný matematický aparát pro zpracování dat z pohybových senzorů a tedy přesný výpočet orientace měřícího modulu (umístěného na člověku) v prostoru. To by zcela jistě vydalo na celou samostatnou práci. Z tohoto důvodu pro ověření (sou)činnosti měřících senzorů a zobrazovací aplikace byl využit ten nejjednodušší postup pro výpočet uhlů aktuálního natočení měřících modulů z dat poskytnutých pohybovými senzory.

Vzhledem k základnímu principu akcelerometru, který byl popsán již v bakalářské práci [9], neměří akcelerometr přímo otočení (náklon), ale měří zrychlení (směr gravitačního zrychlení nebo posuvné zrychlení). V tomto případě, kdy uvažujeme pouze pomalé pohyby s malým zrychlením, lze za změřené zrychlení považovat to gravitační, které se v případě tříosého akcelerometru rozkládá do jeho tří os. Pro zjištění náklonu je tedy potřeba převést gravitační zrychlení na úhel náklonu. Nejdříve se hodnoty z akcelerometru musí normovat. Normování v tomto případě znamená jejich převedení na bezrozměrné číslo v intervalu $\langle -1,1 \rangle$, představující podíl celkového změřeného zrychlení. Při předpokladu, že všechna zrychlení, kromě zrychlení gravitačního, jsou blízká nule, lze říci, že součet druhých mocnin změřených zrychlení ve všech třech osách pod odmocninou (Pythagorova věta), bude hodnota pouze právě měřeného gravitačního zrychlení (hodnota měřená akcelerometrem).

$$g = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

g – hodnota měřená senzorem pro gravitační zrychlení

a_x, a_y, a_z – hodnoty měřené pro zrychlení v osách x, y a z

Pokud se tímto číslem podělí zrychlení měřené v libovolné ose, vyjde číslo mezi nulou a jedničkou, odpovídající části změřeného gravitačního zrychlení v konkrétní ose.

$$n_x = \frac{a_x}{g} \quad n_y = \frac{a_y}{g} \quad n_z = \frac{a_z}{g}$$

n_x, n_y, n_z – normovaný podíl gravitačního zrychlení

g – hodnota měřená senzorem pro gravitační zrychlení

a_x, a_y, a_z – hodnoty měřené pro zrychlení v osách x, y a z

Výpočet pro odklon od vodorovné osy je poté již vcelku jednoduchý. Pokud se úhel mezi vodorovnou osou a změřeným zrychlením označí „ α “ (jak ukazuje Obrázek 18), tak platí:

$$\alpha_x = \sin^{-1} n_x \quad \alpha_y = \sin^{-1} n_y \quad \alpha_z = \sin^{-1} n_z$$

$\alpha_x, \alpha_y, \alpha_z$ – odklon os x, y a z od vodorovné osy [rad]

n_x, n_y, n_z – normovaný podíl gravitačního zrychlení

Vzhledem k tomu, že funkce „asin“ z matematické knihovny .NETu vrací hodnoty v radiánech, je potřeba tyto hodnoty převést na stupně.

$$y = \frac{x}{\pi \cdot 180}$$

y – hodnota ve stupních

x – hodnota v radiánech

Dají-li se všechny vztahy dohromady, vznikne z toho kompletní vztah pro výpočet náklonu:

$$\alpha_x = \frac{\sin^{-1} \frac{a_x}{\sqrt{a_x^2 + a_y^2 + a_z^2}}}{\pi \cdot 180}$$

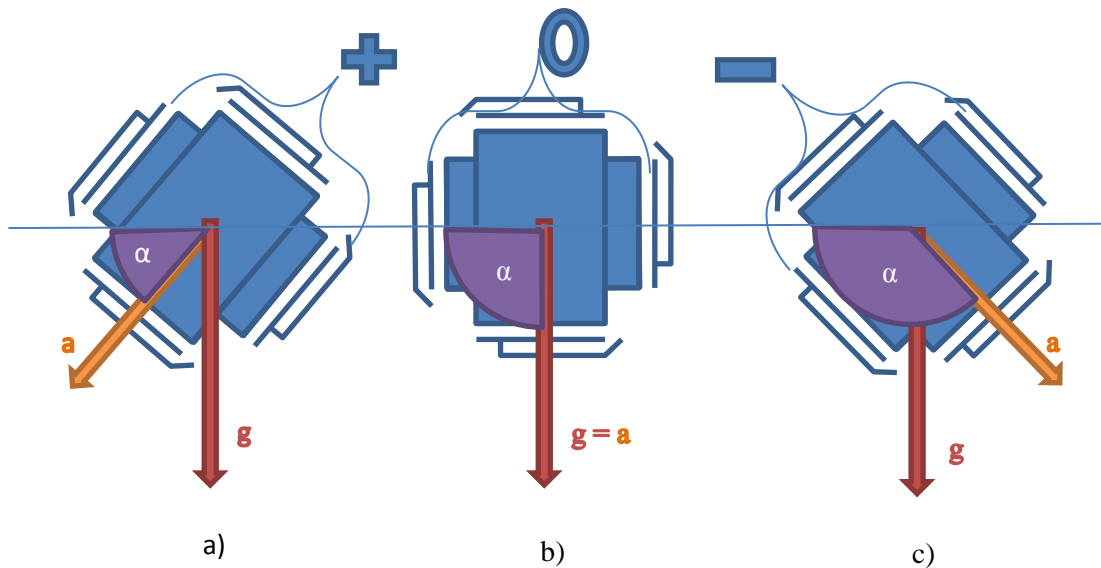
$$\alpha_y = \frac{\sin^{-1} \frac{a_y}{\sqrt{a_x^2 + a_y^2 + a_z^2}}}{\pi \cdot 180}$$

$$\alpha_z = \frac{\sin^{-1} \frac{a_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}}}{\pi \cdot 180}$$

$\alpha_x, \alpha_y, \alpha_z$ – úhly natočení ve stupních

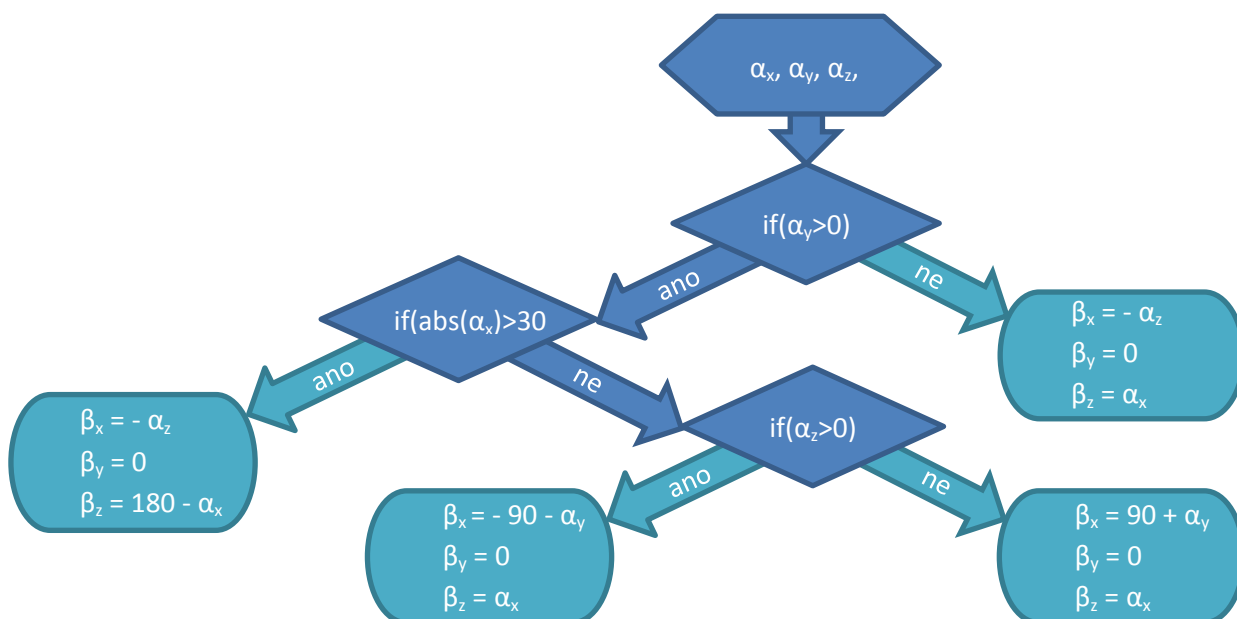
a_x, a_y, a_z – hodnoty změřené akcelerometrem

Jak však uvádí Obrázek 18, při otočení o více než 90 stupňů není možné z hodnot pouze jedné osy rozlišit, na jakou stranu je senzor pootočen, protože při pootočení oběma směry klesá změřená amplituda shodně. Proto je potřeba k rozlišení směru natočení využívat i dalších dvou os tohoto akcelerometru. Na obrázku lze vidět příklad, kdy při úhlu menším než 90° (viz Obrázek 18 – a) je hodnota druhé osy kladná, zatímco při úhlu větším než 90° (viz Obrázek 18 – c) je hodnota druhé osy záporná. Pokud se rozliší otočení senzoru o více než 90 stupňů, je nutné vykonat přepočty, kdy $\alpha = 180 - \alpha_z$ (pro $\alpha > 90^\circ$), kde α je skutečný úhel a α_z je změřená hodnota. V „horní“ části, tedy při otočení o 90 a více stupňů opačným směrem než jak zobrazuje Obrázek 18, je situace obdobná, akorát znaménko opačné. Tedy $\alpha = -180 - \alpha_z$ (pro $\alpha < 90^\circ$). Avšak různé senzory jsou na člověku umístěny v rozdílných polohách, tudíž i tyto přepočty jsou rozdílné. Stejně tak jako rozlišení strany, na kterou se senzor otáčí.



Obrázek 18 – Princip rozlišení směru náklonu akcelerometru.

Přepočít například pro tělo tedy vychází následovně. Za vstupní hodnoty jsou zde dříve zmíněné α_x , α_y a α_z a za výstupní hodnoty β_x , β_y a β_z .



Obrázek 19 – Přepočít hodnot akcelerometru pro měřící modul umístěný na těle.

Tuto techniku však nelze použít pro otočení kolem osy kolmé k zemi. Takové otočení samostatným akcelerometrem změřit nelze. Použitý senzor však obsahuje rovněž i tříosý gyroskop,

který je takovéto natočení schopen měřit. Zatím není využit, nicméně je na něj vše připraveno (data z gyroskopu se samozřejmě vyzvedávají a do úlohy tedy přenášejí). Bude však nutné doladit nenulový „drift“ gyroskopického senzoru. Tento drift vzniká zejména integrací „zašuměných“ dat z gyroskopu.

7.3 Druhá postava a předcvičování

Při dříve zmíněném přepracování aplikace a zřehlednění kódu došlo k vytvoření třídy „Clovek“, která obsahuje string „jmeno“ sloužící k rozlišení mezi předcvičujícím vzorem a postavou, kterou pohybují připojené senzory. Kdyby později došlo k přidání dalších postav do scény, mohlo by být využití rozsáhlejší. Dále tato třída obsahuje slovník (Dictionary) výchozích hodnot, slovník s částmi těla, kontejner (ContainerUIElement3D), který je potřeba pro „actionListener“ při obsluze kliknutí na některou část těla, slovník s naměřenými daty k částem těla a „Prumer“, což je proměnná, která uchovává deset posledních naměřených hodnot a průměruje je. Z konstruktoru této třídy jsou také volány všechny funkce, potřebné pro vytvoření grafické postavy. Přidání další postavy již tedy není žádný problém. Pro předcvičování byla vytvořena nová třída, která obsahuje „List“ s daty o pohybujících se částech těla. Každá část těla obsahuje string udávající, o jakou část těla se jedná, dále hodnoty natočení dané části těla kolem osy x, y, z a údaj samozřejmě o čase. Samotné předcvičování se spouští tlačítkem „Start“. Při stisku tohoto tlačítka se spustí časovač. Při každém „tiku“ tohoto časovače se zkontroluje, zda je aktuální čas nižší než čas pro následující pozici části postavy. Při překročení tohoto času dojde k uskutečnění dalšího „kroku“ v předcvičování. Pro každý krok v předcvičování je vždy vypočteno, za jaký čas se má daná část těla dostat do cílové pozice tohoto kroku. Díky této struktuře je možno přehrávat pohyby s libovolnou rychlostí bez ohledu na původní vzorkovací frekvenci.

7.4 Detekce připojených senzorů

Aby nedocházelo k „žádostem“ o data z nepřipojených senzorů (na tyto data by se zbytečně čekalo), je potřeba nejprve detekovat, se kterými měřicími moduly lze skutečně komunikovat, neboli které jsou opravdu připojeny. Pro tento účel je v úloze tlačítko „Detekce“. Po stisku tohoto tlačítka dochází postupně k vysílání „žádostí“ o informační paket na všechny měřicí moduly, tedy postupně se všemi přípustnými adresami (1-15). Při obdržení „odpovědi“ je modul označen za připojený. V tomto případě to znamená, že v poli booleanů „Sensor.Channels“ je na místě dané adresy, tedy „Sensor.Channels[adresa]“, nastavena hodnota „true“ (měřicí modul je dosažitelný a aktivní). Na informační paket se čeká 200 ms, a pokud do této doby nepřijde, označí se modul za nedosažitelný / nepřipojený. Na tomto místě v poli je tedy ponechána hodnota „false“. Postup detekce připojených měřících modulů lze popsat následovně:

- A) stisk tlačítka „Detekce“
- B) nastavení první možné adresy (I = 1)
- C) z aplikace vyslán požadavek na informační paket na měřicí modul s nastavenou adresou
- D) čekání na informační paket (maximálně 200 ms)
 - E1) pokud informační paket obdrženo, nastaví se adresa v poli na „true“
 - E2) pokud do 200 ms nepřijde, zůstane jeho adresa v poli rovna „false“

- F) opakuje se postup od bodu C s adresou o jedna vyšší (do adresy 15)
- G) navrácení GUI do původního stavu a ukončení detekce

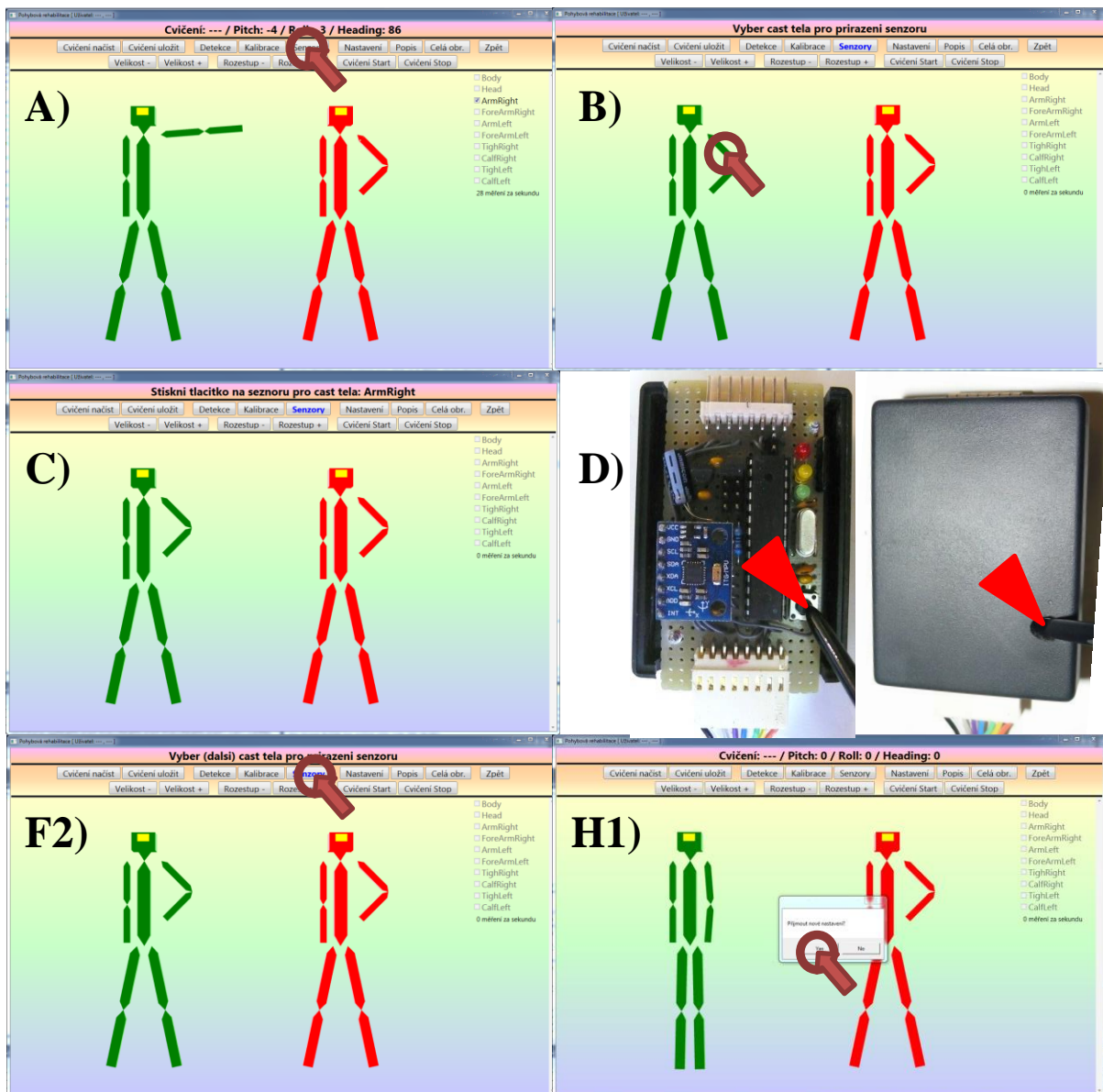
7.5 Přiřazení měřících modulů k částem těla

Tato část úlohy umožňuje uživateli pouze s několika málo měřícími moduly cvičit postupně celé tělo. Například pokud uživatel využije pouze čtyři měřící moduly a centrální modul, může uskutečnit jak cviky na nohy, tak cviky na ruce nebo například cviky na tělo a hlavu. Nelze však v takovém případě cvičit všechny části současně. Postup při přiřazování měřících modulů částem těla zobrazuje Obrázek 20 (A-H1):

- A) Stisk tlačítka „Senzory“
- B) aplikace přestane vysílat pakety na měřící moduly a uživatel je vyzván ke kliknutí na nějakou část postavy
- C) po kliknutí na libovolnou část postavy je uživatel vyzván ke stisku tlačítka na přiřazovaném modulu
- D) aplikace čeká na příchozí paket s adresou z libovolného / přiřazovaného modulu
- E) při příjmu takového paketu aplikace přiřadí danou adresu k části těla (v poli)
 - F1) pokračování v přiřazování – opakování od bodu B
 - F2) další stisk tlačítka „Senzory“
- G) uživatel je vyzván k potvrzení nového nastavení
 - H1) potvrdí – je ponecháno nové nastavení (je zároveň uloženo do příslušného XML souboru)
 - H2) nepotvrdí – je navrženo původní nastavení

Struktura XML souboru určeného pro ukládání nastavení přiřazených senzorů („SensorsToAddress.xml“):

```
<!--pole enumů značících část těla a čísla, které značí adresu senzoru-->
<ArrayOfKeyValuePairGenerOfEnumTransXYZInt32>
  <!--sloupec přiřazených částí: první řádek značí přiřazení hlavy k měřícímu modulu s adresou 3 -->
  <KeyValuePairGenerOfEnumTransXYZInt32 Key="Head" Value="3" />
  .
  .
  .
</ArrayOfKeyValuePairGenerOfEnumTransXYZInt32>
```



Obrázek 20 – Zobrazení postupu přiřazování měřících modulů k částem postavy.

7.6 Výběr aktivních částí

Pokud uživatel připojí senzor, ze kterého nechce data využívat, a tento senzor je propojen s nějakou částí zobrazené postavy, lze data z tohoto senzoru snadno zablokovat, respektive uzamknout danou část zobrazené postavy v původní poloze. K tomuto slouží oblast „selektor“, která se nachází v pravé části zobrazovaného okna úlohy (viz Obrázek 17). V něm každé části postavy přísluší jedno zatrhávací políčko. Tato políčka jsou v základním stavu nezatržena a zablokována (nelze je tedy zatrhnout / aktivovat). Pokud se však s danou částí těla propojí některý měřící modul, dojde k automatickému zatržení políčka a také k jeho povolení (lze jej pak libovolně odznačit, či znovu zatrhnout). Zatržené políčko přitom znázorňuje pohyblivou část postavy (nezatržené nepohyblivou).

7.7 Kalibrace

Protože při každém „oblékání / nasazování“ senzorů dojde k nějakým nepřesnostem (senzory se pokaždé umístí i nepatrně jinak), je nutné počáteční hodnoty (nulové úhly) nastavit. K tomu slouží tlačítko „kalibrace“. Po jeho stisku má uživatel pět vteřin na postavení se do „pozoru“, o čemž je informován v horním informačním panelu. Tato doba je zde proto, že v okamžiku stisku tlačítka na PC člověk fyzicky nemůže stát v „pozoru“. Po uběhnutí pěti vteřin je uživatel informován o dokončení kalibrace. Od této chvíle se od všech „aktivních“ hodnot odečítají hodnoty naměřené při této kalibraci. Aktivními hodnotami jsou zde myšleny úhly naměřené na částech těla, ke kterým je připojen měřicí modul a jejich pohyb je povolen.

7.8 Nahrávání, ukládání a přehrávání pohybů

Pro nahrávání neboli záznam pohybů slouží tlačítko „Cvičení uložit“. Po stisku tohoto tlačítka běží pětivteřinový odpočet. Je zde proto, aby se uživatel stihl připravit do výchozí pozice. Po uplynutí tohoto času se vytvoří nový „List“ určený pro ukládání hodnot. Poté se v něm vytvoří sekce pro části těla, na kterých jsou připojeny měřicí senzory. Do každé sekce se následně začnou ukládat hodnoty daných částí těla rychlostí pětikrát za sekundu. Toto číslo lze snadno změnit, ale ukazuje se, že je dostatečné a uložené cvičení tedy nezabírá příliš místa na disku. Jedna minuta nahrávky z jednoho senzoru zabere přibližně 17 kB místa na disku. Což znamená, že hodina cvičení se všemi (deseti) senzory zabere necelých 10 MB. Toto v dnešní době není žádný problém. Během nahrávání je v informační liště (v programu nahoře) zobrazena informace o nahrávání a tlačítko k ukládání je označené jako stisknuté / aktivní. Při jeho dalším stisku je nahrávání ukončeno a je zobrazen dialog s dotazem na název ukládaného cvičení. Po vyplnění a po potvrzení, že chce člověk záznam cvičení skutečně uložit je provedena kontrola, zda cvičení s daným názvem náhodou již neexistuje nebo zda není text prázdný a pokud je vše v pořádku, je vytvořen nový soubor ve formátu XML s nahranými daty. Pokud je někde nedostatek, je zobrazena příslušná chybová hláška.

Pro přehrávání / předcvičování pohybů se musí uložený pohyb nejdříve načíst. K tomuto slouží tlačítko „Cvičení načíst“. Při stisku tohoto tlačítka je zobrazen seznam dostupných cviků, tedy nahraných již uložených záznamů pohybů. Po dvojkliku na vybrané cvičení je v informačním panelu zobrazen název takto zvoleného cvičení. Cvičení se spustí tlačítkem „Cvičení start“. Další pokračování je popsáno v dřívější kapitole.

Postup při nahrávání tedy vypadá následovně (viz Obrázek 21):

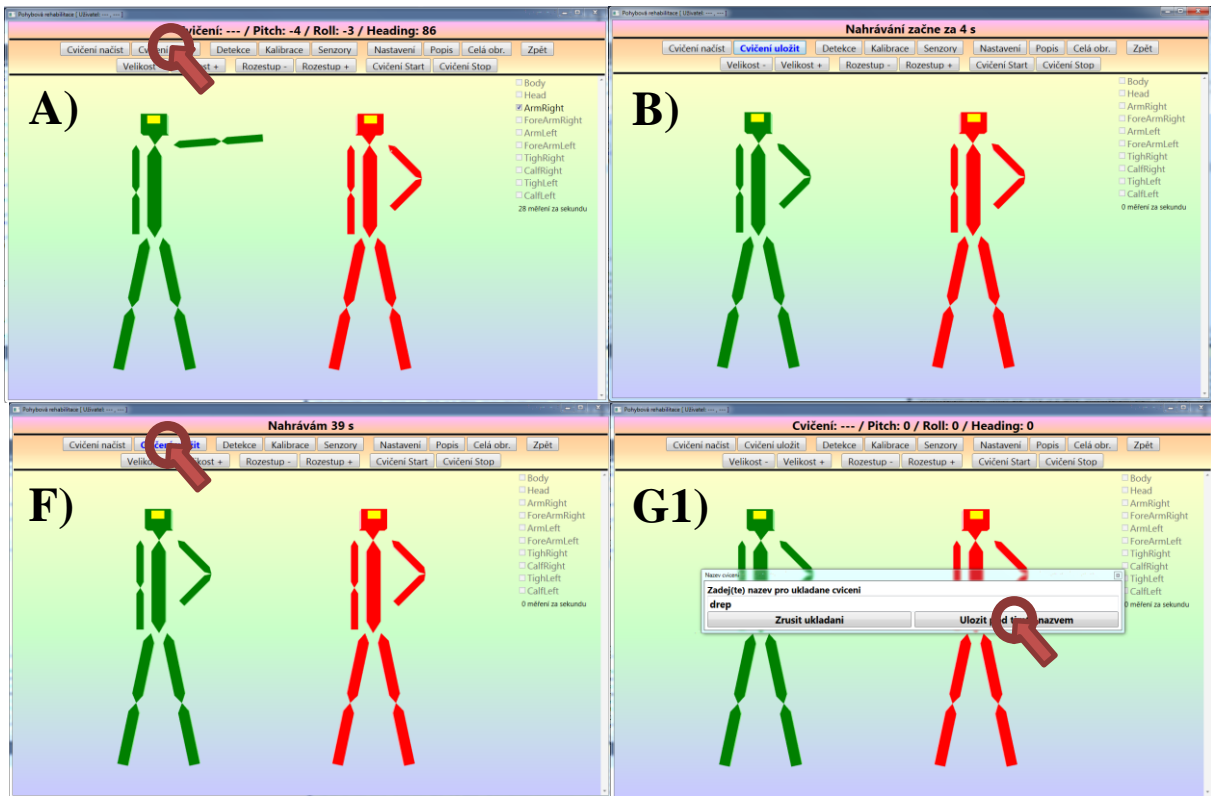
- A) Stisk tlačítka „Cvičení uložit“
- B) 5 vteřin odpočet do začátku nahrávání
- C) vytvoření nového „Listu“
- D) v listu vytvořena struktura pro části těla, ke kterým je připojen měřicí modul
- E) začátek nahrávání a vkládání hodnot do listu
- F) při dalším stisku tlačítka „Cvičení uložit“ nahrávání ukončeno a zobrazen dialog s dotazem na název cvičení
 - G1) potvrzení uložení – je zkontrolován název a pokud je v pořádku, cvičení je uloženo v novém XML souboru s názvem „Cviceni + zadaný text + .xml“
 - G2) zamítnutí uložení – uložení neproběhne a aplikace pokračuje dál

Postup pro přehrávání nahraného cvičení je následující (viz Obrázek 22):

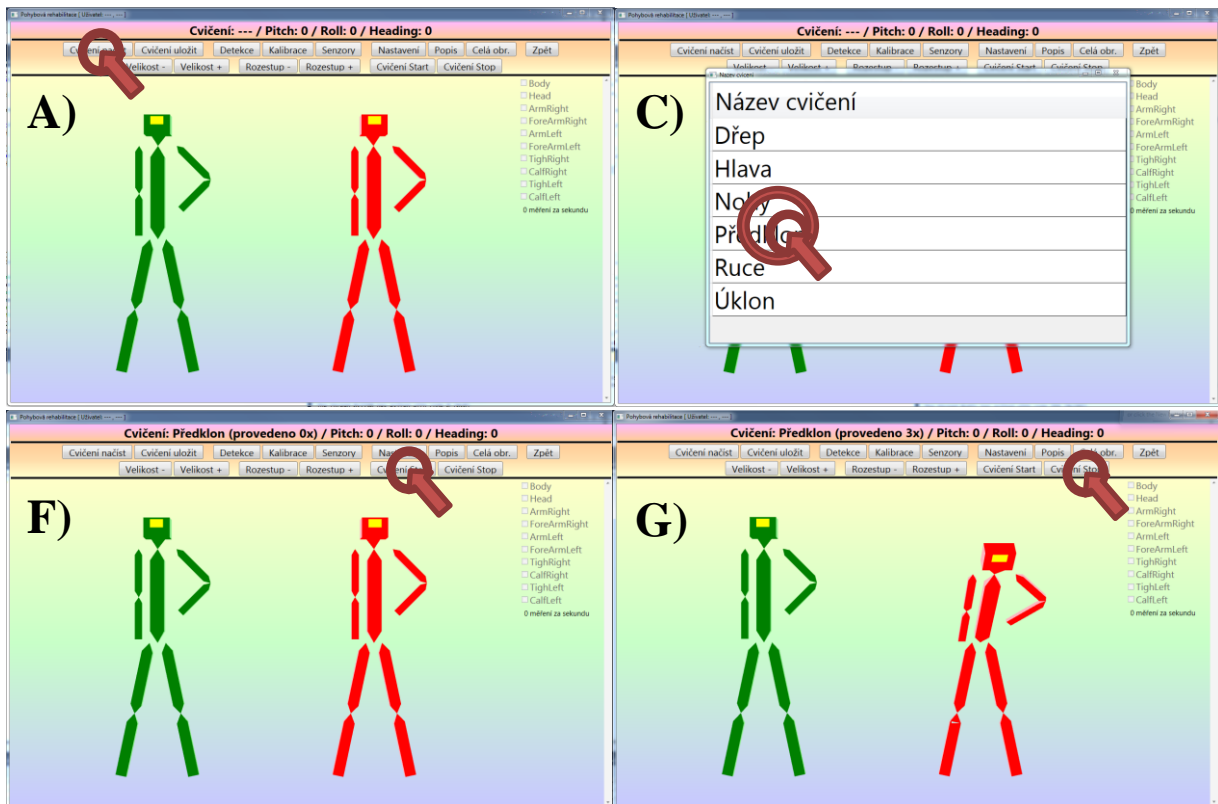
- A) stisk tlačítka „Cvičení načíst“
- B) zobrazen seznam všech dostupných záznamů cvičení
- C) výběr cvičení probíhá dvojklikem na požadované
- D) po výběru dojde k vytvoření nového „Listu“ v PC, do kterého se překopírují data z XML souboru
- E) v informačním panelu zobrazen název vybraného cvičení
- F) při stisku tlačítka „Cvičení start“ se začne procházet načtený „List“ hodnoty v listu jsou aplikovány na předcvičující postavu
- G) „Cvičení Stop“ předcvičování zastaví

Struktura XML souboru pro uložení cvičení:

```
<!-- infolist jmenující se Cviceni-->
<DVInfoList Name=„Cviceni“>
  <VS>
    <!-- list časů a 3D bodů / úhlů patřících k hlavě -->
    <DVItemsTimeSpanPnt3DInt32 Name="Head">
      <VS>
        <!-- list hodnot (každý řádek značí jednu polohu vazanou na čas) -->
        <!-- 1. řádek značí, že v case T=0 budou hodnoty otočení Vx=0, Vy=0 a Vz=0 -->
        <Pnt3DI32 T="0" Vx="0" Vy="0" Vz="0" /> .
        .
        .
      </VS>
    </DVItemsTimeSpanPnt3DInt32>
  </VS>
</DVInfoList>
```



Obrázek 21 – Zobrazení postupu pro nahrávání pohybu.



Obrázek 22 – Zobrazení postupu pro přehrání pohybu, respektive pro spuštění předcvičování.

8 Hodnocení

Výsledek práce představuje experimentální zařízení, které zatím nikterak nemůže konkurovat řešením uvedeným v úvodu, je však realizovatelné za zlomek ceny a má zajisté velký potenciál do budoucna. Je to však otázka dlouhodobějšího časového horizontu.

8.1 Rychlost měření

Vzhledem k tomu, že v současné verzi (platné k 04/2015) se SW v PC dotazuje postupně všech připojených senzorů, je rychlost měření dána jejich počtem. V případě jednoho připojeného senzoru je rychlost přibližně 62 vzorků za vteřinu. Počet měření za vteřinu je nepřímo úměrný počtu senzorů.

$$r = \frac{62}{n}$$

r – počet vzorků za vteřinu

n – počet připojených senzorů

To znamená, že při připojení všech deseti senzorů je rychlost měření přibližně 6 vzorků za vteřinu. Toto číslo však bude zvýšeno s dokončením modulu pro sběr dat druhé generace, kde se budou data z měřících modulů předávat paralelně do centrálního modulu, jak je popsáno v kapitole 6.5.2. Avšak i současných zhruba 5 vzorků za vteřinu při použití všech senzorů je dostačující pro zamýšlené předcvičování a kontrolu pohybu uživatele.

8.2 Stabilita výstupu

Protože výstup ze senzorů je z různých důvodů rozkmitaný, jsou v aplikaci v PC zařazeny dva filtry klouzavých průměrů. Poprvé se data filtrují již při vstupu z měřícího modulu do programu. V nastavení aplikace se dá zadat počet hodnot do tohoto filtru (délka filtru) a také, zda mají být hodnoty „váženy“, či nikoli. „Vážení“ v tomto případě znamená, že nejstarší hodnota je násobena „vahou“ o velikosti jedna a každá novější je násobena „vahou“ o jedna vyšší. Poté je z hodnot vytvořen klasický vážený průměr, tedy:

$$x = \frac{\sum_1^N (x_i \cdot i)}{\sum_1^N i}$$

x – výsledná hodnota

N – počet hodnot pro průměrování (délka filtru)

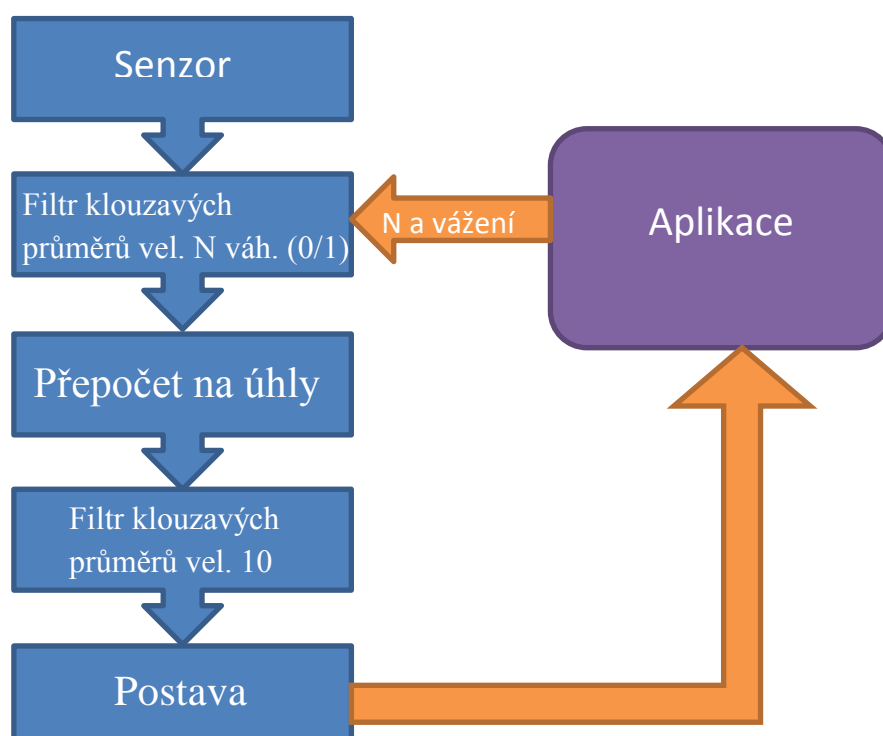
x_i – jedna z naměřených hodnot (x_N – nejnovější hodnota)

i – index (a v tomto případě i „váha“)

Klasický filtr, tedy bez váhování, je prostý aritmetický průměr z posledních N hodnot. N představuje délku filtru. Další filtr je poté zařazen za přepočtení hodnot. Ten je zde zapojen z důvodu malých

nepřesností při přetočení modulů přes úhlovou mez, kdy je od změřeného úhlu odečítáno 180° , jak je popsáno v kapitole 7.2. Bez tohoto filtru dochází k nespojitostem v pohybu. Zde se jedná opět o filtr typu klouzavý průměr, jehož předem daná délka má hodnotu 10.

Při použití současně prvního filtru bez váhování o délce deseti hodnot a zároveň druhého neměnného filtru, nejsou znatelné žádné zákmity, pokud je zařízení v klidu. Hodnoty jsou tedy velmi stabilní. Průběh filtrace ukazuje Obrázek 23.



Obrázek 23 – Zobrazení průběhu filtrace dat.

8.3 Přesnost získaných dat

Již na počátku vývoje bylo stanoveno, že zařízení nebude určené pro přesné měření, ale pouze pro orientační kontrolu rehabilitace. Z toho důvodu je plně dostačující, pokud se získané hodnoty pohybují s přesností $\pm 5^\circ$. Jelikož však nemáme prostředky ani potřebu pro kontrolu této přesnosti, byla přesnost (v současné době) pouze odhadnuta z pozorování.

Odhadujeme tedy, že data se skutečně získávají s přesností zhruba $\pm 5^\circ$, avšak při pohybu pouze ve dvou osách. Otočení vůči ose směřující kolmo k zemi akcelerometrem měřit nelze. Proto při takovém pohybu dochází k velkým nepřesnostem. Hardware je však navrhnout a zrealizován tak, aby poskytoval i hodnoty z gyroskopu, který takový pohyb změřit dokáže. Díky tomu může být brzy tento nedostatek odstraněn. Dosažená přesnost je také dána rychlostí pohybu. Při rychlejších pohybech budou hodnoty z akcelerometru ovlivněny i jinými složkami zrychlení (pohybové zrychlení), nejen gravitačním zrychlením. Pro jednoduché cviky, které nejsou příliš dynamické, dostačuje stávající řešení. Pro složitější cviky bude muset být software doplněn o výpočet natočení i z hodnot

z gyroskopu. Hardware bude i pro složitější cviky dostačující, protože už nyní jsou z něj vysílány hodnoty jak z akcelerometru, tak z gyroskopu. Již v současné verzi lze měřící modul osadit senzorem obsahující i magnetometr a jeho data jsou rovněž přenášena do aplikace a tedy i úlohy. Jejich využití je však mnohem složitější.

8.4 Cena

Zde se budou údaje s postupujícím časem výrazně měnit. Údaje zde uvedené jsou platné v dubnu roku 2015 a jsou vázány na současnou verzi vytvořeného zařízení.

Měřící modul (cena za jeden kus v 04/2015):

- procesor AtMega8 v DIP pouzdře – 30 Kč [eBay.com], 72 Kč [GMe.cz]
- senzor:
 - MPU 6050 – 50 Kč [eBay.com], 443 Kč [farnell.cz]
 - MPU 9150 – 100 Kč [eBay.com], 220 Kč [aliexpress.com]
 - MPU 9250 – 250 Kč [eBay.com], 220 Kč [aliexpress.com]
- Ostatní součástky: 20-50 Kč [eBay.com], 50-100 Kč [GMe.cz]

Centrální modul (první generace):

- převodník FTDI 232RL – 50 Kč [eBay.com], 114 Kč [farnell.cz]
- ostatní součástky: 50 Kč [eBay.com], 100 Kč [GMe.cz]

Cena za jeden měřící modul s MPU 6050 je tedy zhruba 100 až 615 Kč, za jeden centrální modul 100 až 214 Kč. Z tohoto vychází celková cena při použití 6 měřících modulů na 814 až 2600 Kč, tedy zhruba 1500 Kč. Verze s jinými senzory se liší pouze o cenu senzorů.

Rozpočet na celé zařízení je tedy 1500 Kč. K této ceně je však u některých součástí nutně připočítat poštovné, protože nalezené ceny z velké části nejsou v rámci České republiky. Obvykle se však dá najít varianta bez poštovného. V rámci České republiky jsou obvykle součástky výrazně dražší, nebo nejsou dostupné vůbec. Uvedené ceny jsou přepočteny dle aktuálního kurzu dolaru, který činí 25.366 Kč za jeden americký dolar k 25.4.2015.

Pokud by se zařízení vyrábělo sériově, cena by se sice snížila, avšak některé potřebné certifikace by jí opět zvýšily.

8.5 Možnosti rehabilitace

Toto zařízení je navrhováno pro dva celkem odlišné účely:

- předcvičování – kdy si pacient může přehrávat cviky od lékaře doma na obrazovce a tzv. „cvičit si podle nich“
- orientační kontrola – možnost pro lékaře dohlížet na pacientovu rehabilitaci a hodnotit její výsledky

8.5.1 Předcvičování

Možnost předcvičování je nyní již ve zcela využitelné fázi. Je zobrazena předcvičující postava, vedle které je názorně zobrazena postava cvičícího člověka. V nejbližší době dokonce přibude možnost čekání předcvičující postavy na dokončení zadaného pohybu postavou cvičící. Pro větší názornost bude také doděláno barevné rozlišení odchylky, kdy každá část těla bude mít barvu podle toho, jak se její poloha odchyluje / liší od polohy stejné části předcvičující postavy.

8.5.2 Orientační kontrola

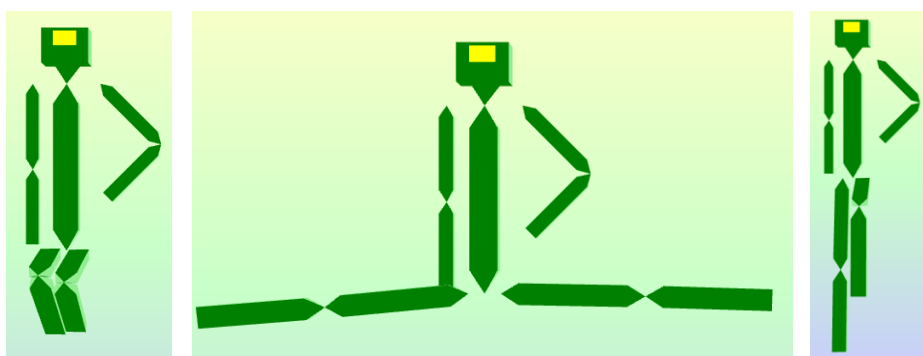
Dohlížení na rehabilitaci pacienta je v současné době reálné. Pacient by však musel využít možnosti nahrávání pohybů a lékař by tyto nahrané pohyby musel shlédnout. Pro tento účel ale nahrávání vytvořeno nebylo (pro lékaře by bylo zhlédnutí takového záznamu dlouhou a nezajímavou činností). Zařízení však k tomuto účelu stále směřuje. Následujícím krokem by bylo vytvoření další části úlohy, která by nad uloženými daty vytvořila základní statistické zpracování užitečné pro lékaře. Například maximální dosažené úhly pro jednotlivé části těla nebo průměrné dosahované úhly při určitém cviku a tak dále. U modulu pro sběr dat druhé generace také nebude nutné data ukládat na PC, ale budou uložena na SD kartě přímo na modulu. Dohlížejícímu pracovníkovi poté bude stačit pouze tato karta.

8.6 Cviky, pro které lze v současné době zařízení využít

Jak již bylo napsáno, zařízení není v současné době vhodné pro všechny typy cviků. Především ne pro ty, při kterých dochází k rotaci vůči ose kolmé k zemi. Vhodné cviky jsou nyní (se současnou verzí zařízení) například tyto:

Cviky pro nohy (4 měřící moduly – 2 na stehnech a 2 na lýtkách), viz Obrázek 24:

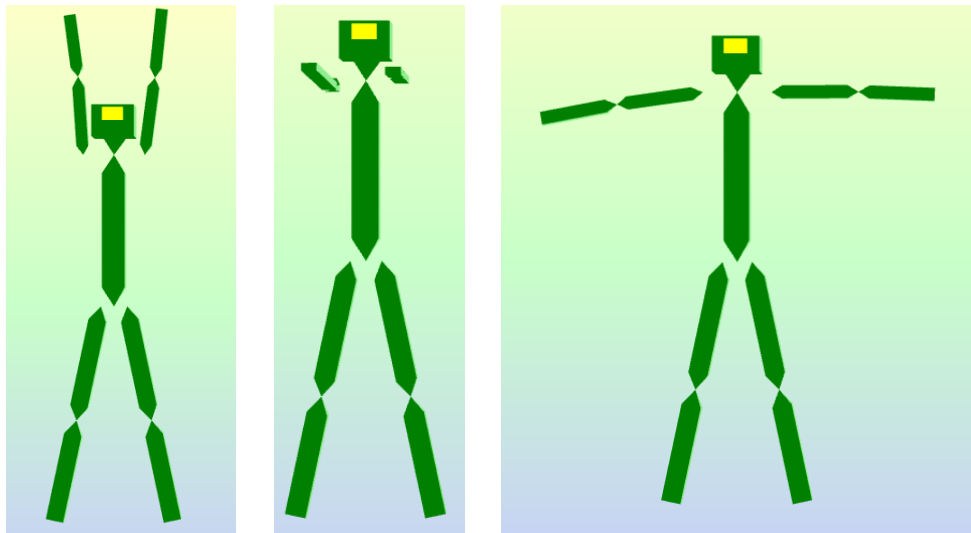
- dřep
- roznožení
- přednožení
- (další principem podobné)



Obrázek 24 – Postava se senzory na stehnech a lýtkách, zobrazená při vykonávání cviků: dřep, roznožení a přednožení.

Cviky pro ruce (4 měřící moduly – 2 na pažích a 2 na předloktích), viz Obrázek 25:

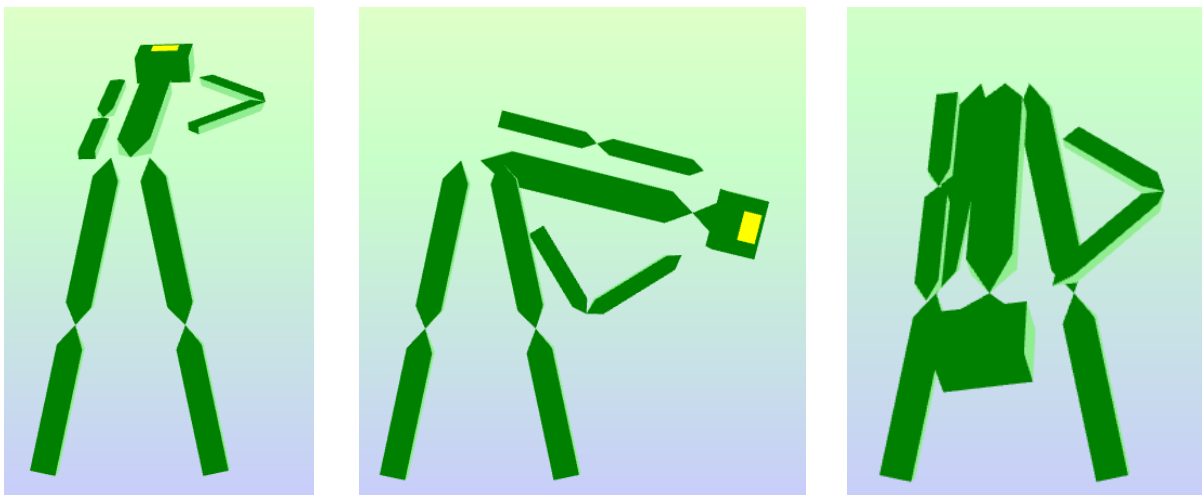
- vzpažení
- předpažení
- rozpažení
- kruhy celýma rukama
- kruhy předloktími
- (další principem podobné)



Obrázek 25 – Postava se senzory na pažích a předloktích, zobrazená při vykonávání cviků vzpažení, předpažení a upažení.

Cviky pro trup (1 senzor na těle), viz Obrázek 26:

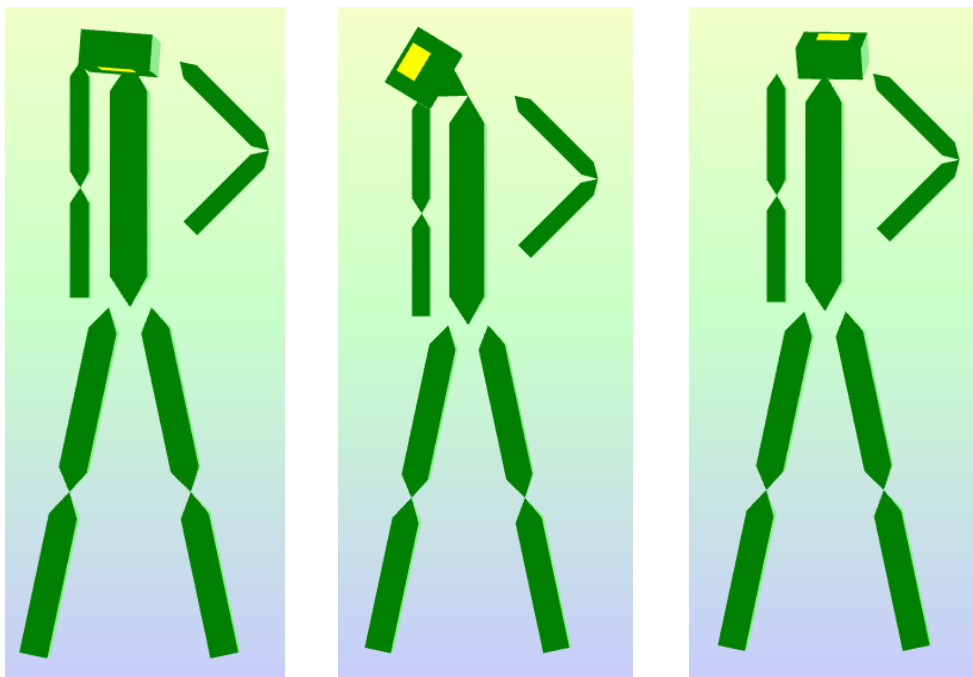
- předklon
- úklon
- záklon
- kruhy celým trupem
- (další principem podobné)



Obrázek 26 – Postava se senzorem na těle, zobrazená při vykonávání cviků: záklon, úklon a předklon.

Cviky pro hlavu (1 senzor na hlavě), viz Obrázek 27:

- předklon
- úklon
- záklon
- kruhy hlavou
- (další principem podobné)



Obrázek 27 – Postava se senzorem na hlavě, zobrazená při vykonávání cviků: předklon, úklon a záklon.

Kombinované cviky (měřící moduly různě):

- skákání „panáka“ – 8 měřících modulů: párově na stehnech, lýtkách, pažích a předloktích
- sedy / lehy – 5 měřících modulů: párově na stehnech a lýtkách a 1 na těle
- cviky trupu s nataženýma rukama – 5 modulů, párově na pažích a lýtkách a na těle
- (další podobné)

8.7 Možnosti využití zařízení

Návrh zařízení byl koncipován jako dostatečně univerzální. Jeho nejvíce zamýšlené využití bylo v domácích podmínkách. Za tímto účelem byly vybírány komponenty (senzory a další součástky) pro minimalizaci celkové ceny tohoto zařízení. Minimalizace ceny sebou samozřejmě přináší snížení přesnosti a tedy často i využitelnosti. Pro hodnocení domácí rehabilitace se jedná zejména o orientační měření pohybu člověka. K tomuto hlavnímu záměru byl tedy samozřejmě přizpůsoben tzv. poměr cena / výkon tohoto zařízení. Stanovené cíle jako 5 až 10 vzorků za vteřinu z připojených senzorů a přesnost určení úhlu ze senzoru na 5 až 10 stupňů, je pro toto použití zcela dostačující.

Vzhledem ke skutečnosti, že použité měřicí moduly obsahují vcelku přesné senzory, lze vhodným zpracováním dat (není součástí této práce) dosáhnout mnohem větší přesnosti a tím úhlové rozlišitelnosti. V tomto případě může zařízení velmi dobře posloužit i pro základní měření pohybu při celkovém vyšetření pohybových schopností člověka.

Navržené zařízení a poté vytvořený prototyp je tedy vhodný nejen pro domácí použití, ale i pro základní pohybovou diagnostiku přímo v ordinaci lékaře.

9 Návrhy na další vývoj

Protože bylo toto zařízení vyvíjeno jako prototyp, tedy jako vývojový mezistupeň, bylo předem počítáno s jeho dalším vývojem a rozšiřováním. Z tohoto důvodu má samozřejmě zařízení možnosti, které jsou zatím zcela nevyužity. Je však nyní možno na tuto práci dále navázat a již zkonstruované zařízení tedy dále rozvinout.

Zde jsou tedy uvedeny návrhy na další možná rozšíření tohoto projektu. Je však nutno upozornit, že většina těchto návrhů velmi snadno vydá na celou samostatnou práci. Z tohoto důvodu jsou zde uvedené návrhy popsány pouze velmi stručně.

Hardware:

- realizace zbývajících čtyř měřících modulů na plošný spoj (do celkového počtu 10)
- dokončení realizace centrálního modulu druhé generace dle vytvořeného návrhu
- zhodnocení možnosti využití umělého magnetického pole při použití magnetometru

Software:

- zpracování dat z gyroskopu a jejich využití při výpočtu úhlů natočení
- prozkoumání možností lepšího zpracování dat z magnetometru
- statistické zpracování naměřených dat (co nejvhodněji pro lékaře)

Firmware:

- ověření možnosti využití signálového procesoru umístěného přímo v senzorech
- vytvoření nového firmwaru pro centrální modul druhé generace

10 Závěr

Obsahem diplomové práce bylo navrhnout a co nejvíce realizovat prototyp zařízení pro dohled nad rehabilitačním cvičením zejména v domácích podmínkách. Rovněž zhodnotit jeho možnosti, tedy na jaké cvičební úkony jej bude možno hlavně využít.

Návrh a poté i realizace navazuje přímo na předchozí bakalářskou práci. Návrh zařízení byl však zcela aktualizován podle, v současné době, dostupných součástí a zejména pohybových senzorů. Jsou tedy využity nejnovější, avšak samozřejmě snadno dostupné, senzory. I když se v současné době využívají data pouze z akcelerometrů, tak senzory na měřících modulech obsahují i gyroskop, některé dokonce i magnetometr. Z tohoto pohledu je vytvořené zařízení velmi univerzální a tím určeno pro široké experimentální použití. V průběhu práce byly vytvořeny v podstatě dvě verze zařízení. První jednodušší pro pouhé ověření činnosti měřících modulů a použitých senzorů a druhá pro skutečné praktické využití s bezdrátovým přenosem dat.

Ovládací aplikace rovněž staví na základech z bakalářské práce, avšak byla znatelně rozšířena. Toto rozšíření spočívá zejména ve vytvoření dvou postav, kdy jedna předcvičuje zadanou sekvenci pohybů a druhá zobrazuje aktuální stav uživatele. Aplikace je rovněž schopna zaznamenat a uchovat nahraný pohyb a ten poté využít jako vzor při vlastním cvičení. Vytvořená aplikace však zachází dále a poskytuje možnosti nejen automatické detekce aktuálně připojených senzorů (pro jednoduchost použití), ale i možnost sestavit si v podstatě libovou konfiguraci senzorů na těle člověka a ty přiřadit určitým částem těla.

V průběhu práce se rovněž vyskytlo velké množství problémů, které bylo nutno vyřešit, nebo byly pro jejich velkou složitost prostě odloženy pro pozdější pokračovatele tohoto projektu. Nemalé úsilí bylo věnováno vytvoření 3D postavy, která je schopna ohýbat své jednotlivé části podle dat přicházejících z pohybových senzorů. Právě vytvoření těchto lokálních souřadnicových systémů a jejich transformací představuje velké know-how této aplikace, i když se na první pohled jedná o celkem malou část programového kódu. Naopak například původní snaha pro jednoduché využití magnetometru pro detekci orientace v prostoru však byla pro velkou obtížnost zcela opuštěna.

Rovněž byl stanoven seznam doporučených cviků, které jsou pro současný vytvořený prototyp zařízení vhodné i přes jeho aktuální omezení a případné nedokonalosti.

Výstupem práce tedy není ucelené a ukončené zařízení, ale prototyp několika (šesti) měřících modulů a centrálního modulu pro experimenty, zabývající se detekcí a měřením pohybu člověka. Hlavním cílem bylo vytvořit komplexní návrh a, i když omezený, prototyp zařízení vhodný pro domácí rehabilitační cvičení za minimální cenu, ale s dostačující přesností. V tomto směru tato práce splnila svůj cíl.

11 Zdroje

- [1] META MOTION. Motion Capture: Meta Motion sells Motion Capture Hardware and Software [online]. [cit. 2015-05-05]. Dostupné z: <http://metamotion.com/>
- [2] Princip a.s.: Děláme chytré věci [online]. 2012. [cit. 2015-05-05]. Dostupné z: <http://www.princip.cz/>
- [3] Xsens 3D motion tracking [online]. [cit. 2015-05-05]. Dostupné z: <https://www.xsens.com/>
- [4] INVENSENSE INC. 2013. MPU-6000 and MPU-6050 Product Specification. Revision 3.4. 52 s. Dostupné také z: <http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf>
- [5] INVENSENSE INC. 2013. MPU-9150 Product Specification. Revision 4.3. 50 s. Dostupné také z: http://www.invensense.com/mems/gyro/documents/PS-MPU-9150A-00v4_3.pdf
- [6] INVENSENSE INC. 2014. MPU-9250 Product Specification. Revision 1.0. 42 s. Dostupné také z: <http://www.invensense.com/mems/gyro/documents/PS-MPU-9250A-01.pdf>
- [7] MPU-6050 Triple-axis Accelerometer [online]. 2013. [cit. 2015-05-05]. Dostupné z: <http://www.digibay.in/mpu-6050-triple-axis-accelerometer-gyroscope-6-dof-module>
- [8] RTIMULib-Arduino now supports the InvenSense MPU-9250 9-dof IMU: richards-tech notes [online]. [cit. 2015-05-05]. Dostupné z: <https://richardstechnotes.wordpress.com/2014/11/08/rtimulib-arduino-now-supports-the-invensense-mpu-9250-9-dof-imu/>
- [9] VILDMAN, Jiří. 2013. Prototyp zařízení pro základní diagnostiku a rehabilitaci pohybového aparátu člověka. Praha. Bakalářská práce. České vysoké učení technické v Praze.
- [10] 2013. 9 DOF IMU MPU-9150 Nine-axis Attitude sensor [online]. [cit. 2015-05-05]. Dostupné z: http://digipak.org/zencart/index.php?main_page=product_info&products_id=227

12 Obsah příloženého CD

Příložené CD obsahuje:

- text diplomové práce – DP_Vildman_Jiří.pdf
- schémata modulů a plošný spoj měřícího modulu - složka eagle (vytvořeno v programu Eagle 7.2.0)
- dokumentace k sensorům – složka dokumentace_k_senzorum
- fotky zařízení – složka foto
- aplikaci, jíž je úloha vytvořená v rámci této práce součástí – složka aplikace (aplikace se spouští souborem „WindowsRehabMovement.exe“, jenž se nachází v podsložce „Program“)