

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND
BIOMECHANICS

DATOVÁ RUKAVICE: VIZUALIZACE VE VRML A NÁVRH ELEKTRONIKY

DATA GLOVE: VRML VISUALISATION AND DESIGN OF ELECTRONIC CIRCUITS

BAKALÁŘSKÁ PRÁCE
BACHELOR THESIS

AUTOR PRÁCE
AUTHOR

JAN ŠTĚPÁNEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ROBERT GREPL, Ph.D.

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky

Akademický rok: 2008/09

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Štěpánek Jan

který/která studuje v **bakalářském studijním programu**

obor: **Mechatronika (3906R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Datová rukavice: vizualizace ve VRML a návrh elektroniky

v anglickém jazyce:

Data glove: VRML visualization and design of electronic circuits

Stručná charakteristika problematiky úkolu:

Práce se bude zabývat návrhem a realizací rukavice s integrovanými snímači pohybu prstů.

Cíle bakalářské práce:

- 1) Seznamte se s programováním mikrokontroleru Atmel AVR a realizujte komunikaci mikrokontroleru a PC přes BlueTooth.
- 2) Navrhněte DPS pro autonomní provoz rukavice s následujícími vlastnostmi: a) bateriové napájení; b) minimální rozměry a hmotnost; c) mikrokontroler snímá analogové hodnoty ohybových snímačů a MEMS akcelerometru; d) komunikace přes BlueTooth s PC. Proved'te osazení DPS.
- 3) Vytvořte 3D model ruky v SW SolidWorks a převed'te jej do hierarchicky uspořádaného vizualizačního modelu VRML.
- 4) Vytvořte GUI na PC s těmito vlastnostmi: a) implementováno v Matlabu; b) komunikace s mikrokontrolerem přes sériovou linku realizovanou BlueTooth; c) vykreslování stavu ruky v reálném čase na základě dat z ohybových snímačů a MEMS akcelerometru (jako sensoru náklonu).

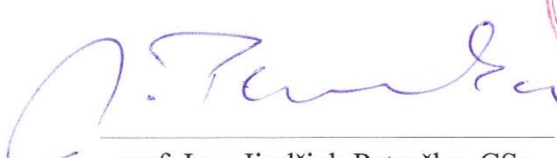
Seznam odborné literatury:

1. Dušek, F.: Matlab a Simulink, skriptum ČVUT
2. Gergelitsová Š.: VRML v příkladech, BEN, 2004
3. Váňa, V.: Mikrokontroléry ATMEL AVR - assembler, Nakladatelství BEN, 2003
4. Herout, P.: Učebnice jazyka C
5. Noskievič: Modelování a identifikace systémů

Vedoucí bakalářské práce: Ing. Robert Grepl, Ph.D.


Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2008/09.

V Brně, dne 5.11.2008



prof. Ing. Jindřich Petruška, CSc.
Ředitel ústavu





doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Anotace

Tato práce se zabývá návrhem a výrobou elektroniky datové rukavice a tvorbou vizualizace ruky, která v reálném čase vykresluje stav ruky uživatele. Realizace je týmovou prací dvou studentů. Rozdělení úkolů v týmu je popsáno v kapitole Stanovení cílů práce.

Začátek práce je věnován rešeršní studii v oblasti datových rukavic a seznámení se s důležitými pojmy ve světě virtuální reality, část rešeršní studie se zabývá také jinými technickými provedeními datových rukavic. Další kapitoly práce popisují tvorbu virtuálního modelu ruky, jehož úkolem je vykreslování stavu uživatele ruky v reálném čase na základě dat z ohybových senzorů a MEMS akcelerometru. Komunikace s PC je provedena přes sériovou linku realizovanou Bluetooth. Pomocí počítačového programu Matlab byl vytvořen matematický model, který zpracovává data přijímaná z rukavice a vypočítává hodnoty polohy jednotlivých částí ruky. Program Matlab také sloužil ke tvorbě grafického uživatelského rozhraní, které uživateli umožňuje rukavici používat.

Klíčová slova

Datová rukavice, VRML, Matlab, Bluetooth komunikace, Akcelerometr, Ohybové senzory

Annotation

This work deals with designing and assembly of electronics of a data glove and creation of a hand visualization which projects a real-time state of user's hand. Realization was a team effort of two students. The division of tasks is described in the chapter called Stanovení cílů práce.

The beginning of the work is dedicated to a background research in the field of data gloves. Next chapters describe creation of the model of virtual hand. This model's task is to project real-time state of user's hand based on data received from the flexure sensors and MEMS accelerometer. Communication with PC is realized through BlueTooth serial port virtualization. Mathematical model created in Matlab processes data received from the glove and calculates the values of the positions of the individual parts of hand. Matlab also served for development of a graphical user interface which enables the user to utilize the glove.

Keywords

Data glove, VRML, Matlab, BlueTooth communication, Accelerometer, Bend sensors

Bibliografická citace mé práce:

ŠTĚPÁNEK, J. Datová rukavice: vizualizace ve VRML a návrh elektroniky. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2009. 46 s. Vedoucí bakalářské práce Ing. Robert Grepl, Ph.D.

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně na základě rad a pokynů vedoucího práce, a že jsem veškeré literární zdroje uvedl v seznamu použité literatury.

V Brně dne:

.....
Jan Štěpánek

Poděkování

Na tomto místě je mojí milou povinností poděkovat všem, kteří mi pomáhali při vypracování této bakalářské práce. Především bych rád poděkoval vedoucímu práce Ing. Robertu Greplovi, Ph.D. za cenné rady a odbornou pomoc a také mé přítelkyni za její podporu ve chvílích, kdy byly snahy o vyřešení problému neúspěšné.

Obsah

1. ÚVOD	15
2. STANOVENÍ CÍLŮ PRÁCE	17
3. OBECNĚ O DATOVÝCH RUKAVICÍCH	19
3.1 Datová rukavice DGTech DG5 VHand 2.0	19
3.2 Datová rukavice Immersion CyberGrasp	20
4. VIZUALIZACE V JAZYCE VRML	21
4.1 Tvorba modelu ruky v SW Solidworks.....	22
4.1.1 Postup při modelování jednotlivých dílů ruky.....	22
4.1.2 Postup při tvorbě sestavy modelu ruky.....	24
4.2 Tvorba VRML modelu ruky v SW V-Realm Builder 2.0	25
4.3 Implementace VRML modelu do programu Simulink	26
5. ELEKTRONIKA	27
5.1 Popis hlavních elektronických součástí.....	27
5.1.1 Mikrokotroler Atmel Atmega 8	27
5.1.2 Akcelerometr Freescale MMA7260QR2.....	28
5.1.3 Multiplexer CD4052B	28
5.1.4 Ohybový senzor Flexpoint.....	29
5.1.5 BlueTooth modul Free2move F2M03C1	30
6. SESTAVENÍ MODELU V PROGRAMU SIMULINK A TVORBA ALGORITMU KALIBRACE	33
6.1 Kalibrace ohybových senzorů.....	33
6.2 Tvorba modelu v programu Simulink	35
6.3 Převod hodnot z akcelerometru na úhel natočení VRML modelu.....	37
7. TVORBA GRAFICKÉHO UŽIVATELSKÉHO ROZHRANÍ	39
8. ZÁVĚR	41
9. SEZNAM POUŽITÉ LITERATURY	43
10. PŘÍLOHY	45

1. ÚVOD

Počátky virtuální reality spadají do konce 60. a začátku 70. let, kdy byly předvedeny první prostředky pro interaktivní grafiku. První virtuální helma je připisována Ivanu Sutherlandovi (viz. [12]). Virtuální helma však není zdaleka jediným prostředkem umožňujícím vstoupit do virtuální reality. Existuje několik dalších zařízení, zkvalitňujících uživateli procházení virtuálními světy. Důležitým prvkem je datová rukavice.

Datová rukavice je nedílným technickým prostředkem virtuální reality, protože slouží k interakci uživatele a virtuálního prostředí. Jedná se o rukavici osazenou snímači ohybu prstů, případně dalšími zařízeními, které převádí pohyb uživatelské ruky na analogový signál. Signál je dále zpracováván mikroprocesorem a převáděn na signál digitální, který je snadno dále zpracovatelný počítačem.

Spojením výhod virtuální helmy a datové rukavice se otevírá virtuální svět, který nabízí takřka neomezené možnosti využití. V dnešní době se s výhodou virtuální reality využívá například v lékařství, kdy si chirurgové mohou složitou nebo zřídka prováděnou operaci nejdříve vyzkoušet na virtuálním pacientovi, mohou simulovat určité komplikace, jež mohou při skutečné operaci nastat a natrénovat si jejich řešení, aniž by ohrozili život pacienta. Virtuální realita je také často využívána ve vojenství. Snem nejednoho dítěte je vyzkoušet si pilotování nadzvukového bojového letounu. Existuje spousta dnes již technologicky velmi vyspělých simulátorů, které si sice vzhledem k cenové nedostupnosti není možné pořídit domů, bojoví piloti však mají možnost učit se na těchto simulátorech zvládat různorodé situace, aniž by opět riskovali život svůj nebo kopilotův a zničení mnohonásobně dražšího letounu.

Značnou nevýhodou virtuální reality je i v dnešní době stále její cenová nedostupnost pro veřejnost. Faktem však je, že rozvoj v oblasti informačních technologií je neskutečně rychlý a dalo by se očekávat, že v blízké budoucnosti by se mohla virtuální realita rozšířit do každé domácnosti podobně jako osobní počítače. V ostatních částech práce se budeme zabývat výrobou cenově dostupné datové rukavice.

2. STANOVENÍ CÍLŮ PRÁCE

Cílem týmové práce je osazení rukavice ohybovými senzory a akcelerometrem, návrh a realizace elektroniky rukavice, komunikující s PC přes rozhraní BlueTooth, vytvoření virtuálního modelu lidské ruky v programu Solidworks a jeho následná konverze do formátu VRML (Virtual Reality Modeling Language), tvorba modelu v programu Matlab – Simulink, který převádí přijímaná data z elektroniky rukavice na takový formát, který umožní vykreslovat stav ruky uživatele v reálném čase a tvorba grafického rozhraní, jež umožňuje uživateli rukavici používat.

Tato práce se bude zabývat mojí činností v týmu, tedy:

- Tvorbou virtuálního modelu lidské ruky v jazyce VRML
- Rozšířením již hotové elektroniky rukavice o bezdrátovou komunikaci s PC přes rozhraní BlueTooth
- Tvorbou modelu v programu Simulink
- Tvorbou grafického rozhraní pro snadné užívání rukavice

Práce mého kolegy spočívala v:

- Osazení rukavice ohybovými senzory a MEMS akcelerometrem
- Návrhu a výrobě desky plošného spoje elektroniky rukavice
- Sestavení programu pro mikrokontroler Atmega 8
- Realizaci komunikace elektroniky rukavice s PC přes sériovou linku, snímání hodnot z ohybových senzorů a MEMS akcelerometru

3. OBECNĚ O DATOVÝCH RUKAVICÍCH

Existuje velké množství možností, jak z datové rukavice přenést pohyb uživatelské ruky do počítače, který tento pohyb vyhodnocuje a dále zpracovává. Právě technologická úroveň datové rukavice velkou měrou ovlivňuje realistický dojem z užívání virtuální reality. Můžeme říci, že základním požadavkem na datovou rukavici je snímání a přenos všech možných pohybů lidské ruky do počítače. Je tedy třeba znát relativně dobře fyziologickou stavbu ruky, aby bylo možné rukavici osadit senzory ohybu na správných místech.

Také počet ohybových senzorů je důležitým ukazatelem, zda je rukavice schopna zpracovávat reálné pohyby uživatelské ruky. Aby byl snímáný pohyb realistický, nesmíme zapomenout zpracovávat také náklon ruky. Nejčastěji bývá náklon snímán pomocí akcelerometru.

Velmi důležitým faktorem, ovlivňující realistický pocit z užívání datové rukavice, je možnost rukavice poskytovat hmatovou odezvu uživateli. Úroveň zážitku uživatele nepochybně vzroste, pokud při použití virtuální reality uchopí nějaký předmět a opravdu cítí, že jej drží v ruce. Tato technologie je však relativně technologicky náročná, a proto většina firem tuto technologii do svých rukavic neimplementuje.

3.1 Datová rukavice DGTech DG5 VHand 2.0

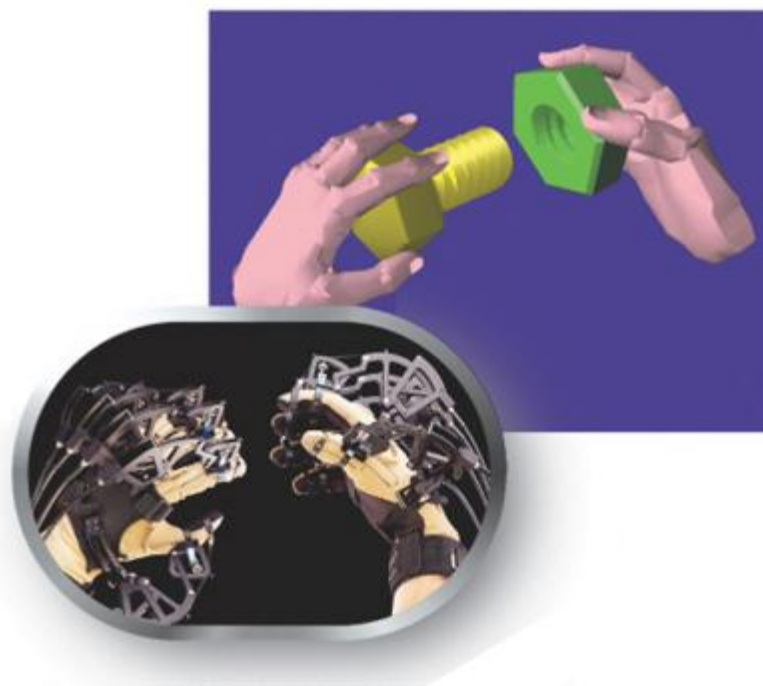
Jedním z profesionálních řešení je datová rukavice firmy DGTech [2]. Tato rukavice je osazena pěti ohybovými senzory, na každém prstu jedním, a akcelerometrem jako senzorem náklonu. Firma neuvádí princip činnosti ohybových senzorů, osazených na rukavici, dá se však předpokládat, že stejně jako v případě našeho řešení se jedná o ohybové tenzometry – zařízení, jež mění velikost elektrického odporu v závislosti na úhlu ohybu. Tento typ rukavice však nedisponuje technologií poskytnutí hmatové zpětné vazby.



Obr. 3.1: Datová rukavice DGTech DG5 VHand 2.0 (převzato z [2])

3.2 Datová rukavice Immersion CyberGrasp

Poněkud pokročilejší řešení datové rukavice nabízí firma Inition [6]. CyberGrasp je nástavbou datové rukavice CyberGlove II, vyráběné také firmou Inition, rozšiřující její funkce o poskytnutí uživateli hmatové zpětné vazby. Datová rukavice CyberGrasp tedy nedovolí uživateli „promáčknout“ virtuální tuhý předmět, který drží v ruce. Zpětná vazba je realizována pěticí stejnsměrných motorků, pro každý prst jedním. Výrobce udává maximální sílu zpětné vazby 12 N (Newtonů) pro každý prst.



Obr. 3.2: Datová rukavice Immersion CyberGlove II se systémem zpětné vazby CyberGrasp (převzato z [6])

4. VIZUALIZACE V JAZYCE VRML

Počátky jazyka Virtual Reality Modeling Language, dále VRML, sahají do konce 80. let minulého století. Vyvinuli jej programátoři firmy Silicon Graphics, Inc. pod názvem Inventor. Jednalo se o knihovnu umožňující práci s prostorovými objekty. Začátkem 90. let spolu se vznikem dnes už známé grafické knihovny OpenGL vznikla nová aplikační knihovna OpenInventor. Během několika dalších let prodělal jazyk mnoho změn, až byl v roce 1995 zveřejněn formát VRML 1.0, který byl oproti OpenInventoru rozšířen o možnosti využívání prostorových dat ze sítě internet. Současně se vznikem VRML 1.0 vznikla neformální skupina nezávislých programátorů, označovaná zkratkou VAG (VRML Architecture Group), která se později stala formální skupinou, nesoucí název VRML Consortium, Inc. a velkým dílem se podílela na vzniku aktuální verze VRML 97. V dalších částech práce budeme pod pojmem VRML vždy mínit VRML 97.

Díky skutečnosti, že je jazyk VRML blízce spjat se sítí internet, otevírají se nové možnosti, příjemnější např. nakupování přes internet. Zákazník by se mohl procházet po virtuálním obchodě a prohlížet si prostorové modely nabízeného zboží. Postupně se také rozvíjejí virtuální města, kde se člověk snadno může seznamovat s lidmi, které potká na virtuální ulici. V zásadě můžeme říci, že v jazyce VRML můžeme vymodelovat libovolné těleso v prostoru, pokrýt ho libovolnou texturou a můžeme ovlivnit i jeho okolí např. světly nebo zvuky.

V dalších podkapitolách práce se budeme zabývat jedním z přístupů, jak můžeme vytvořit virtuální model ruky.

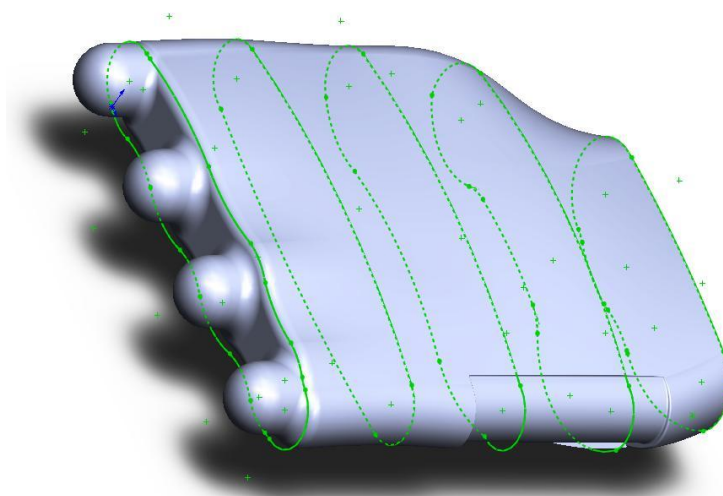
4.1 Tvorba modelu ruky v SW Solidworks

Sestavení virtuálního modelu v samotném jazyce VRML by bylo velice náročné, proto byl s výhodou pro tvorbu jednotlivých dílů modelu využit program Solidworks. Solidworks je 3D CAD (Computer Aided Design) program, používaný ve strojírenství, umožňující tvorbu prostorového modelu a jeho konverzi do jazyka VRML. Prostředí programu Solidworks můžeme rozdělit na dvě části. První část umožňuje modelování jednoho prvku (dílu) celého modelu, pro který platí, že vzdálenost všech jeho bodů je konstantní (díl je pohyblivý pouze jako celek). Druhá část potom umožňuje tyto prvky spojit v sestavu pohyblivými vazbami.

Při tvorbě modelu budeme za díly považovat jednotlivé články prstů a samotnou dlaň, sestavou budeme rozumět dlaň spojenou s prsty s důrazem na funkčnost všech kloubů.

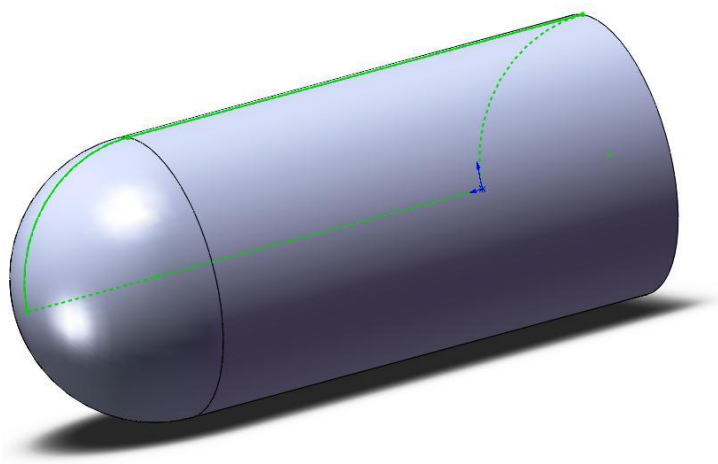
4.1.1 Postup při modelování jednotlivých dílů ruky

Dlaň by mohla být považována za nejpodstatnější část modelu ruky, protože věrohodnost jejího provedení zásadně ovlivňuje věrohodnost celého modelu. Při modelování tvaru dlaně byl zvolen postup načrtnutí pěti skic příčného průřezu lidskou dlaní, znázorněných v Obr. 4.1 zelenými křivkami. Ke spojení skic v prostorový celek sloužila v programu Solidworks funkce *Přidání spojením profilů*. Dále bylo potřeba dlaň osadit klouby, které umožňují rotační pohyb prstů. Toto bylo realizováno polokulovými vrchlíky v přední části dlaně. Hotový model dlaně je vidět na Obr. 4.1.



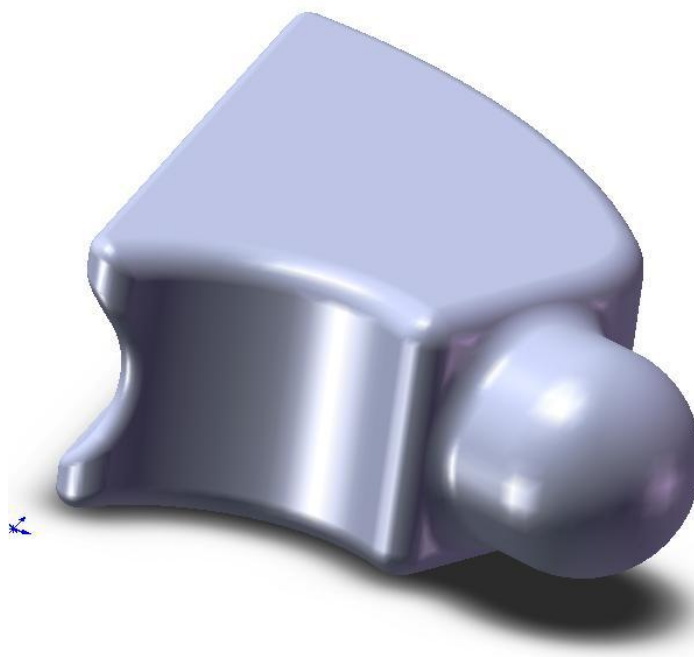
Obr. 4.1: Model dlaně vytvořený v programu Solidworks

Dalšími prvky modelu ruky byly jednotlivé články prstů (Obr. 4.2). Každý z článků vycházel ze skici stejného tvaru, rozdílná byla pouze jejich délka. K vytvoření prostorového modelu článku byla použita funkce *Přidání rotací*. Jeho zadní část je protikusem k polokulovému vrchlíku na přední části dlaně a na přední části jiného článku prstu. Tím byla zaručena vzájemná propojitelnost vazbami.



Obr. 4.2: Model článku prstu vytvořený v programu Solidworks

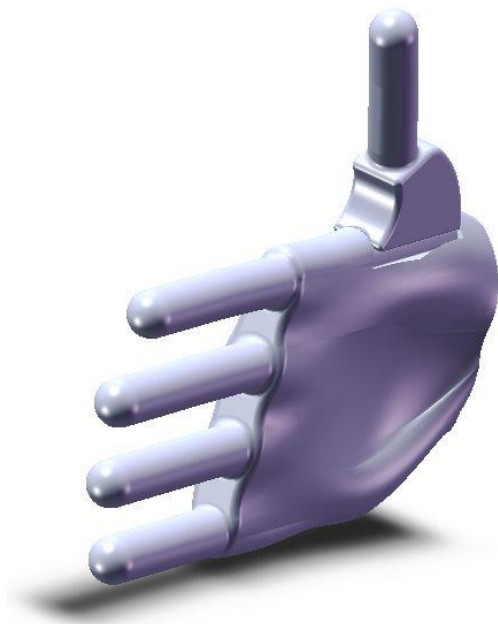
Posledním dílem potřebným pro kompletní model ruky byla část palce, kterou byly články palce připojené k dlani. Tento díl je vyobrazen na Obr. 4.3.



Obr. 4.3: Model části palce vytvořený v programu Solidworks

4.1.2 Postup při tvorbě sestavy modelu ruky

Z uvedených tří základních dílů bylo možno sestavit model ruky. Sestavování jednotlivých komponent je v programu Solidworks věnována celá samostatná část. K propojení jednotlivých pohyblivých dílů slouží funkce *Vazby*. Pro připojení jednotlivých článků prstů k dlani byl použit typ vazby *Soustředná*, jež umožňuje rotaci článků okolo všech tří os. Na Obr. 4.4 je zobrazen sestava hotového modelu ruky.



Obr. 4.4: Hotová sestava modelu ruky

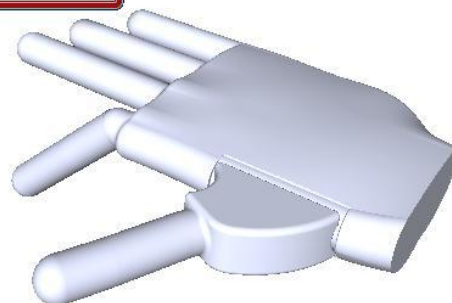
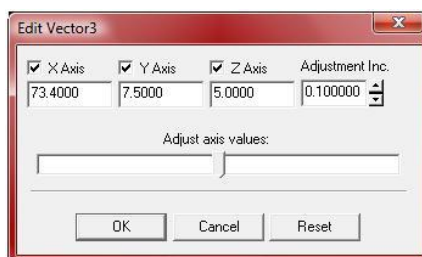
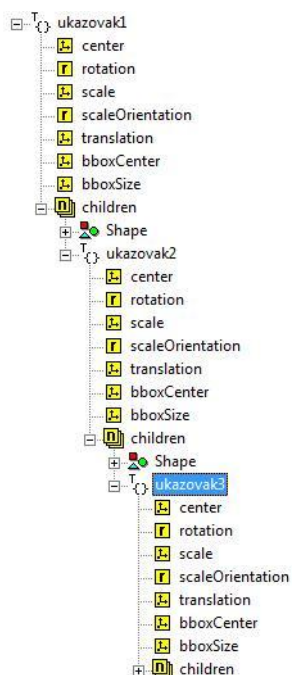
Nyní bylo potřeba provést konverzi sestavy z formátu programu Solidworks (přípona .SLDASM) na formát jazyka VRML (přípona .wrl). Při ukládání bylo nutno v záložce Možnosti vybrat verzi výstupního formátu novější VRML 97, základní jednotky milimetry a zaškrtnout volbu „Uložit všechny součásti sestavy do jednoho souboru“. Takto uložený model byl připraven k dalšímu zpracování.

4.2 Tvorba VRML modelu ruky v SW V-Realm Builder 2.0

Po zhotovení geometrie modelu bylo nyní potřeba definovat osy rotací jednotlivých dílů sestavy, protože ty jsou defaultně nastaveny ke globálnímu počátku souřadného systému modelu (zvolen při tvorbě sestavy). Dále bylo nutno modelu určit, jestliže se pohybuje jeden článek prstu, které další články téhož prstu se musí pohybovat taktéž. K tomu byl použit program V – Realm Builder 2.0, který je dodáván zároveň s programem Matlab.

S výhodou byl využit jednoduchý princip, který jazyk VRML využívá. Jedná se o hierarchicky uspořádaný seznam jednotlivých dílů sestavy, fungující na principu rodič/potomek (Obr.: 4.5). Ten jasně definuje, že pohybuje-li se rodič, pohybuje se i potomek. Toto však neplatí naopak. Tím je zaručena soudržnost modelu za jakýchkoli podmínek. Veškeré prvky, které bylo nutno uspořádat do hierarchicky uspořádaného seznamu, byly automaticky převedeny z prvků sestavy v programu Solidworks. Do hotového seznamu však bylo nutné vřadit jeden transformační uzel, jenž byl rodičem celého modelu ruky. Tento nadřazený uzel byl poté použit ke vstupu hodnot z akcelerometru pro náklon ruky ve směru jedné osy (Dlan1), k náklonu ve směru druhé osy byl využit stávající transformační uzel (Dlan2). Smysl rotací jednotlivých uzlů byl zadán v kolonce *rotation*.

Nastavení os rotací jednotlivých dílů sestavy vycházelo ze znalosti geometrické stavby modelu a jeho rozměrů (Obr. 4.6). Do tabulky bylo nutno vložit vektory posunutí os rotací všech článků prstů od počátku globálního souřadného systému.

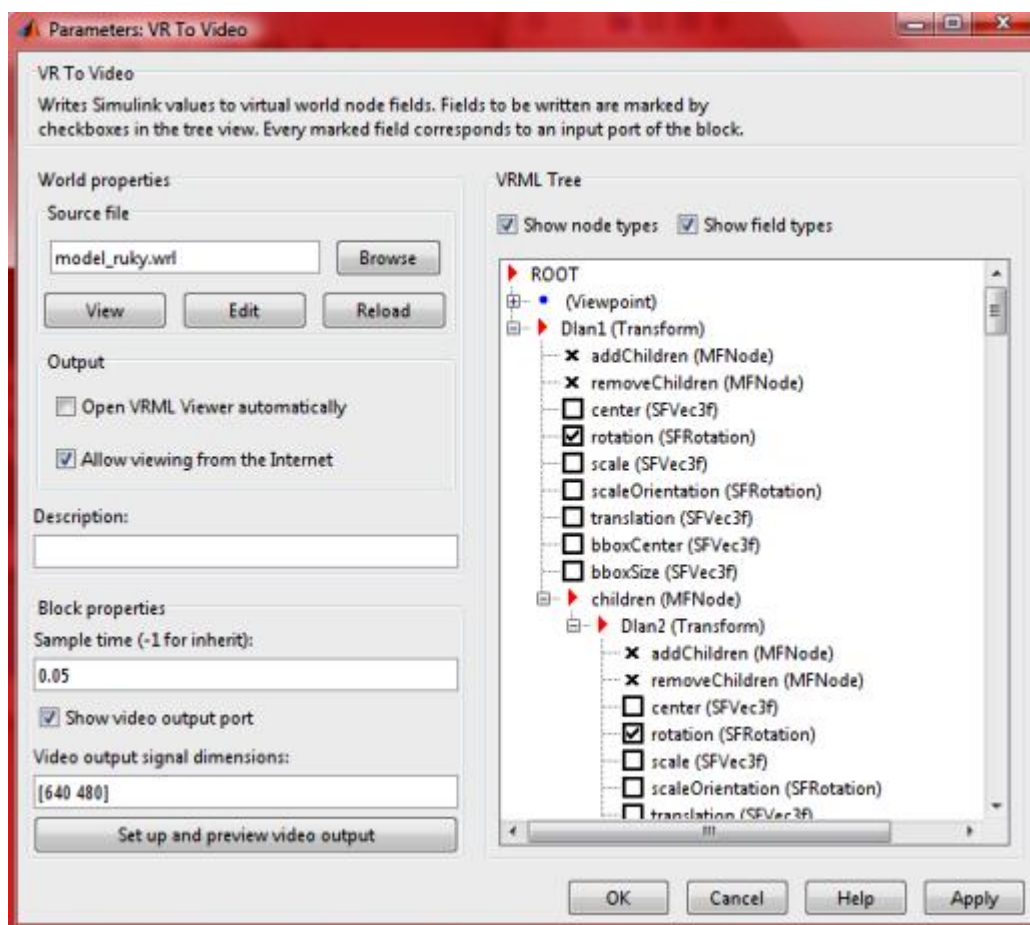


Obr. 4.6: Stanovení osy rotace článku prstu

Obr. 4.5: Hierarchicky uspořádaný model

4.3 Implementace VRML modelu do programu Simulink

Nyní bylo třeba VRML model implementovat do programu Simulink, ve kterém budou probíhat výpočty dat, přijatých od elektroniky rukavice. K implementaci byl využit Virtual Reality Toolbox, konkrétně blok VRSink. Ten umožňuje VRML model vložit do modelu Simulinku a ovládat jej.



Obr. 4.7: Nastavení parametrů bloku VRSink, znázornění vloženého transformačního uzlu (Dlan1) a stávajícího transformačního uzlu (Dlan2)

Na Obr. 4.7 je vpravo vidět již známá hierarchicky uspořádaná struktura VRML modelu. Nyní bylo u každého dílu modelu třeba vybrat funkci, která se má provádět. V našem případě každý díl modelu koná pouze rotační pohyb, zvolena tedy byla funkce *Rotation*.

Blok VRSink byl nyní správně nastaven a připraven na vložení do modelu v programu Simulink. Tvorba modelu je popsána v kapitole 6.2.

5. ELEKTRONIKA

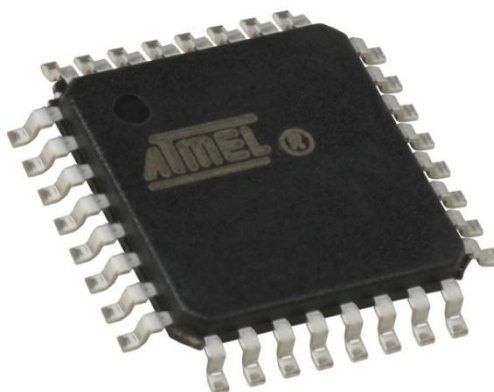
V této kapitole se budeme zabývat návrhem elektroniky rukavice. Při výběru použitých součástí bylo nutné respektovat tyto požadavky:

- Bateriové napájení
- Minimální rozměry a hmotnost
- Bezdrátová komunikace s PC přes Bluetooth
- Mikrokontroler snímá analogové hodnoty ohybových snímačů a MEMS akcelerometru

5.1 Popis hlavních elektronických součástek

5.1.1 Mikrokontroler Atmel Atmega 8

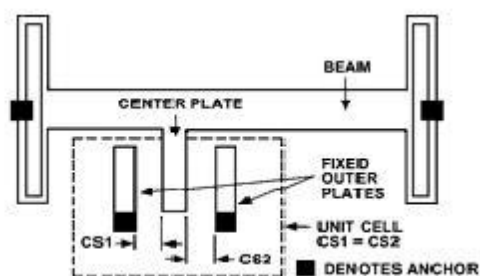
Uvedeným požadavkům zvolený mikrokontroler vyhovuje. Disponuje 10 bitovým A/D (Analogově/Digitálním) převodníkem, který s dostatečnou přesností převádí analogové hodnoty z ohybových snímačů a akcelerometru na hodnoty digitální. S ohledem na požadavek malých rozměrů byla zvolena verze v SMD (Surface Mount Device) pouzdře, které má, proti standardnímu DIL (Dual In Line) pouzdru, znatelně menší rozměry. Veškeré parametry mikrokontroleru jsou uvedeny v [1].



Obr. 5.1: Mikrokontroler Atmega8 (převzato z [13])

5.1.2 Akcelerometr Freescale MMA7260QR2

Uvedený trojosý kapacitní polovodičový akcelerometr funguje na principu změny kapacity kondenzátoru, integrovaného uvnitř akcelerometru, vlivem působící síly, která vzniká zrychlením pouzdra akcelerometru (Obr. 5.2). Umožňuje měřit jak statické zrychlení (náklon), tak zrychlení dynamické (translační pohyb). V našem případě byl akcelerometr využit pouze k měření náklonu ruky, tedy zrychlení statického. Podrobné specifikace akcelerometru jsou uvedeny v [4].



Obr. 5.2 : Princip činnosti kapacitního MEMS akcelerometru, zobrazení klidového stavu – bez působení zrychlení (převzato z [9])

5.1.3 Multiplexer CD4052B

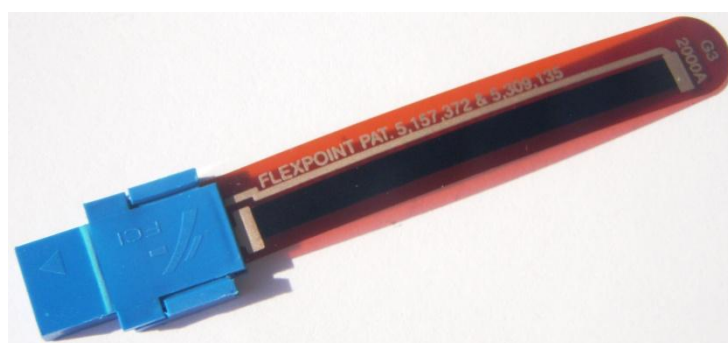
Mikrokontroler Atmega8 je vybaven 8mi kanálovým A/D převodníkem, což je pro obsluhu všech ohybových senzorů a akcelerometru nedostačující. Byly proto použity dva multiplexery, které jsou také 8mi kanálové. Fungují na principu přepínání jednotlivých vstupů, přičemž v každém ze čtyř možných stavů jsou aktivní vždy dva vstupy. Přepínání vstupů je realizováno pomocí pinů A a B, do nichž jsou posílány podle logické tabulky (Obr. 5.3) hodnoty logické 0 a 1. Pomocí multiplexerů bylo možno připojit celkem 16 senzorů, přičemž na mikrokontroleru byly využity pouze čtyři kanály převodníku. Podrobné specifikace multiplexeru jsou uvedeny v [10].

INPUT STATES			"ON" CHANNEL(S)
INHIBIT	B	A	
0	0	0	0x, 0y
0	0	1	1x, 1y
0	1	0	2x, 2y
0	1	1	3x, 3y

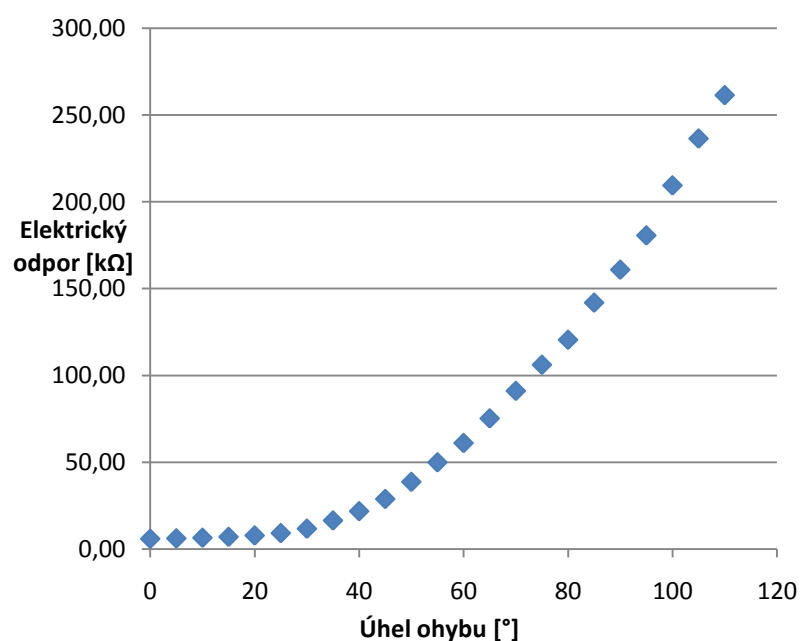
Obr. 5.3: Logická tabulka pro přepínání vstupů multiplexeru (převzato z [10])

5.1.4 Ohybový senzor Flexpoint

Ohybové senzory byly použity především z důvodu jejich cenové dostupnosti. Jedná se o tenzometry, jejich funkce spočívá ve změně elektrického odporu v závislosti na úhlu ohybu. Nevýhodou je jejich nelineární změna elektrického odporu v závislosti na úhlu ohybu (Obr. 5.5) a odlišná změna elektrického odporu při ohýbání senzoru v různých místech po jeho délce.



Obr. 5.4: Ohybový senzor Flexpoint



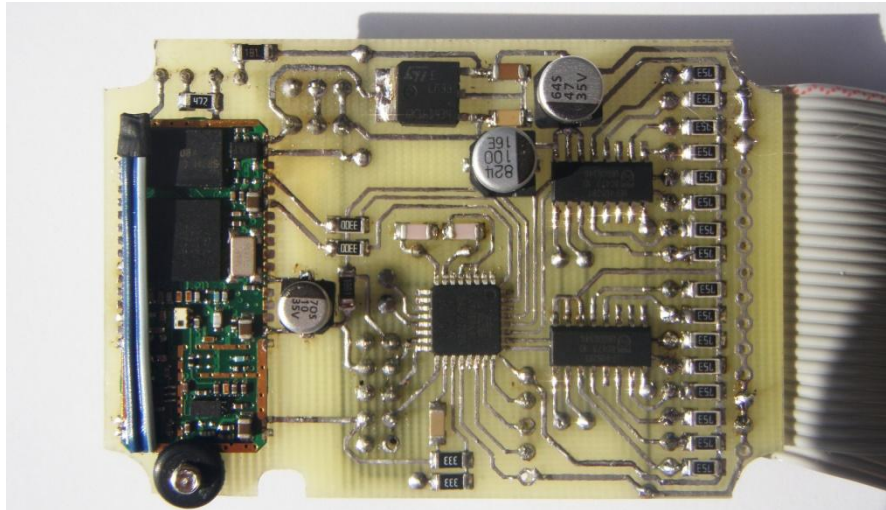
Obr. 5.5: Graf závislosti elektrického odporu na úhlu ohybu senzoru

5.1.5 BlueTooth modul Free2move F2M03C1

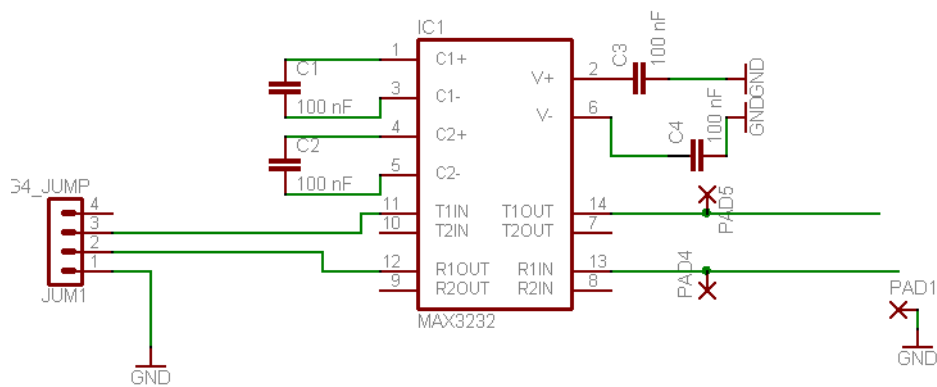
Jedním z požadavků na rukavici je bezdrátový přenos dat pomocí technologie BlueTooth. To je podstatné z hlediska pohodlí uživatele, který při používání rukavice nebude vázán délkou kabelů. Teoretická vzdálenost, na kterou by komunikace s PC měla fungovat, je asi 10 metrů. Daní za bezdrátový přenos dat je nutnost bateriového napájení. Uvedený modul (Obr. 5.6) komunikuje s mikrokontrolerem pomocí sběrnice UART (Universal Asynchronous receiver/transmitter). Komunikace s PC probíhá přes virtuální sériový (COM) port. Specifikace modulu jsou uvedeny v [3]. Modul bylo nutno nastavit pomocí konfigurační utility Wireless Uart Configuration Software [3]. Propojení s PC pro účel konfigurace bylo realizováno převodníkem napěťových úrovní MAX323 [8]. Schéma zapojení převodníku je na Obr. 5.8. V konfigurační utilitě bylo nutné nastavit rychlost komunikace (Baud rate) shodně s rychlostí nastavenou programem mikrokontroleru, v našem případě tedy 115200 baud. Elektrické schéma elektroniky rukavice a návrh desky plošného spoje je zobrazeno v přílohách na Obr. 10.1 a 10.2.



Obr. 5.6: Deska plošného spoje BlueTooth modulu (převzato z [3])



Obr. 5.7: Hotová deska plošného spoje osazená BlueTooth modulem



Obr. 5.8: Elektrické schéma převodníku MAX3232

6. SESTAVENÍ MODELU V PROGRAMU SIMULINK A TVORBA ALGORITMU KALIBRACE

6.1 Kalibrace ohybových senzorů

V kap. 5.1.4 byla zmíněna nevýhoda ohybových senzorů, spočívající v různých změnách elektrického odporu snímačů při ohýbání v různých místech senzoru. Experimentálně bylo zjištěno, že senzory dokonce velmi citlivě reagují na tlak působící shora nebo zdola. Problém by pak vznikl tehdy, pokud by uživatel měl kratší nebo delší prsty, většího, či menšího průměru. Senzory by se ohýbaly v různých bodech, elektrický odpor měnily nekorektně a vizualizace by poté zobrazovala zkreslené výsledky.

Z tohoto důvodu bylo nutné přistoupit k řešení, jež vyžaduje při každém spuštění vizualizace nejdříve provést kalibraci. Při sestavování algoritmu kalibrace je nutno splnit následující požadavky:

- Kalibrační body musí být voleny v takových polohách natočení článků prstů, kde jsou známy úhly jejich natočení
- V okamžiku akčního zásahu uživatele je třeba mít k dispozici aktuální data ze senzorů
- Nelineární nárůst elektrického odporu při zvětšování úhlu ohybu článku prstu je třeba převést na lineární nárůst úhlu ohybu článku prstu v modelu

Prvotním úmyslem bylo použít dva kalibrační body, a to při stavech natažených prstů a zatnuté ruce v pěst. Od tohoto záměru muselo být upuštěno hned ze dvou důvodů. Prvním důvodem byla již zmiňovaná nelinearita ohybových senzorů. Aby bylo docíleno lineárního nárůstu úhlu natočení článku prstu v modelu (eliminování nelinearity ohybových senzorů), bylo nutné provést interpolaci kalibračních hodnot křivkou. Pro interpolaci křivkou jsou však zapotřebí alespoň tři kalibrační body, z tohoto důvodu pouze dva kalibrační body nestačí. Druhý důvod vylučuje volit kalibrační bod při zatnuté ruce v pěst. Po několika experimentech bylo zjištěno, že klouby prvních článků ruky v této poloze vyvolávaly tlakovou sílu na ohybové senzory. Senzory pak vykazovaly nesprávnou hodnotu elektrického odporu a kalibrace byla provedena nekorektně.

Program mikrokontroleru využívá komunikaci typu dotaz/odpověď mezi elektronikou rukavice a programem Matlab. Důvodem použití dozajista pomalejší komunikace tohoto typu je nutnost mít při kalibraci k dispozici aktuální data z ohybových senzorů. Prvotní verze programu pro mikrokontroler používala pouze jednostrannou

komunikaci, neustále tedy zasílala data. Při kalibraci pak nastal problém. Program Matlab totiž data, která byla zaslána za dobu od spuštění komunikace po stisknutí kalibračního tlačítka, ukládal do zásobníku (bufferu), kde čekala ve frontě na vyzvednutí (po přetečení bufferu další data zahazoval). Po stisknutí tlačítka tedy algoritmus kalibrace načtl z bufferu staré hodnoty. Při komunikaci typu dotaz/odpověď tento problém nenastává. Po stisku kalibračního tlačítka je zaslán dotazovací znak, v mikrokontroleru proběhne jeden cyklus, který změří aktuální hodnoty všech senzorů a pošle je do PC, kde je dále zpracován.

Kalibrace tedy probíhala ve třech stavech uživatelské ruky. Prvním stavem byly natažené prsty, k dosažení druhého stavu bylo třeba uchopit předmět kruhového průřezu, při jehož držení můžeme změřit úhel natočení článků prstů a ten přiřadit ke kalibračnímu bodu. Třetí stav je analogický ke stavu druhému, uchopený předmět však musí mít menší průměr, než předmět předchozí.

Jak již bylo řečeno, kalibrace probíhala ve třech krocích. V prvním kroku kalibrace byla spuštěna komunikace rukavice s programem Matlab pomocí programu XSerial (zdůvodněno v kapitole 6.2), zaslán dotazovací znak a přijata data prvního kalibračního kroku.

```
com2 = xserial('Port', 'COM2');    % vytvoření seriového portu
[result, msg] = fopen(com2);        % otevření seriového portu
fprintf(com2, 'S');                 % zaslání dotazovacího znaku
a1=fscanf(com2, '%i');              % přijata data uložena do proměnné a1
```

Komunikace zůstala aktivní pro druhý krok kalibrace, ve kterém byl opět zaslán dotazovací znak, data byla uložena do jiné proměnné. Třetí kalibrační krok byl analogický s krokem druhým, byl však rozšířen o vytvoření vektorů, které datům všech tří kalibračních kroků přiřazovaly úhly natočení. Komunikace byla poté ukončena.

```
u11 = [a1(5) a2(5) a3(5)]; % vytvoření vektoru kalibračních kroků pro
5.kanal převodníku
phi1 = [0 pi/6 pi/2];      % vektor natočení článku prstu v kalibračních
bodech [rad]
fclose(com2);              % uzavření portu - ukončení komunikace
```

Tyto dva vektory postačovaly ke kalibraci jednoho článku prstu modelu. Již zmiňovaná interpolace probíhala přímo v modelu programu Simulink (funkce *Matlab*

Function, Obr. 6.1) a prováděla převod hodnot z elektroniky rukavice na hodnoty úhlu natočení článku prstu v modelu.

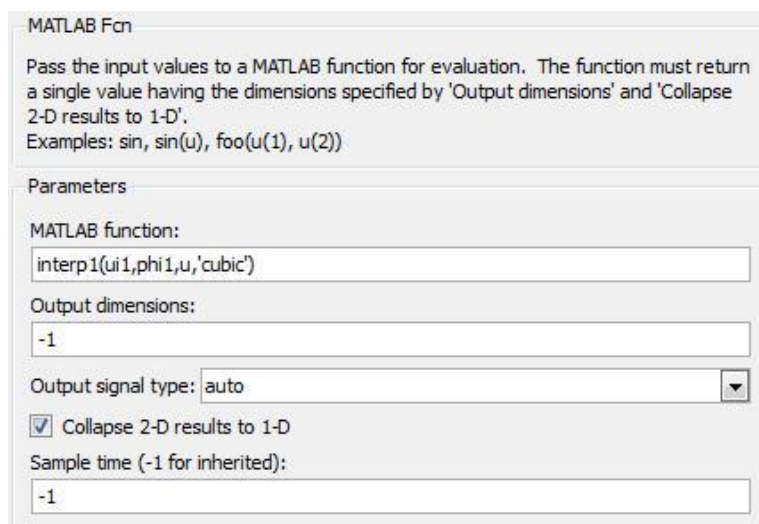
6.2 Tvorba modelu v programu Simulink

Prvním problémem bylo úspěšné spuštění komunikace mezi programem Simulink a elektronikou datové rukavice. Interní funkce programu Matlab pro sériovou komunikaci se neosvědčila, jelikož veškeré snahy o otevření virtuálního sériového portu byly neúspěšné. O poznání lepších výsledků bylo dosaženo použitím toolboxu XSerial, který vyvinula firma Humusoft [5]. Virtuální sériový port lze pomocí tohoto toolboxu otevřít jak v programu Simulink (blok Serial In), tak pomocí skriptu v programu Matlab (kapitola 6.1 – Kalibrace ohybových senzorů).

Aby fungovala sériová komunikace správně, bylo nutné nastavit v bloku Serial In Baud Rate opět na hodnotu 115200 baud. Důležité také bylo do kolonky Data request string vložit dotazovací znak (v našem případě 'S').

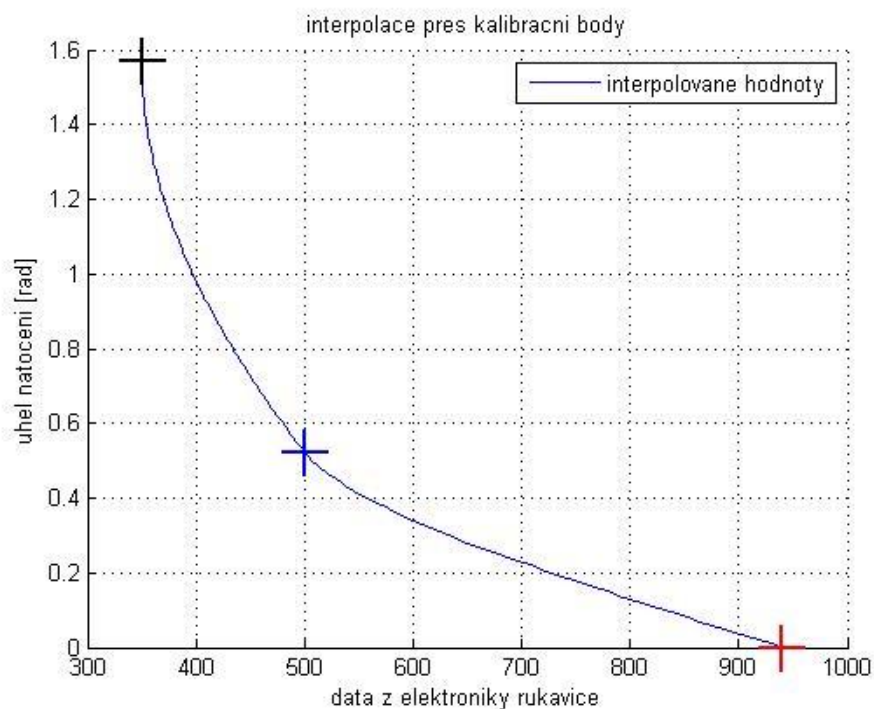
Do modelu bylo také nutno vložit blok VRSink s VRML modelem, jehož tvorba byla popsána v kapitole 4.3.

K ovládní rotačního pohybu každého článku prstů bylo nutné do bloku VRSink přivést dva signály. První signál definuje vektor, okolo kterého rotace probíhá, druhý signál pak přivádí velikost úhlu natočení článku v radiánech. Tento úhel byl vypočítáván blokem *Matlab Function* na základě dat přijímaných z rukavice a předem provedené kalibrace. Přepočet byl prováděn interpolací, pomocí funkce *interp1*.



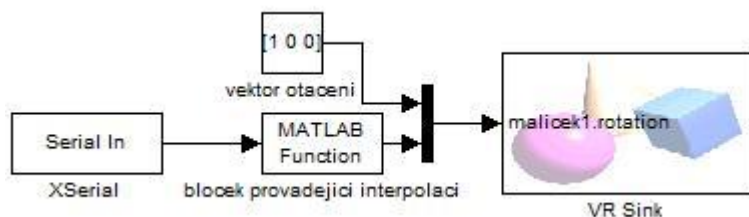
Obr. 6.1: Otevřený blok Matlab Function, přepočítávající data z rukavice na velikost úhlu natočení jednoho článku prstu

Na Obr. 6.2 je znázorněn příklad interpolace. Vektor natočení $\phi_1 = [0 \pi/6 \pi/2]$, vektor kalibračních kroků (získaný třemi kroky kalibrace) $u_{i1} = [940 \ 500 \ 350]$. Modrá křivka znázorňuje závislost interpolovaných hodnot úhlů natočení článku prstů modelu na hodnotách přijímaných z elektroniky rukavice. Vyznačené body označují kalibrační body.



Obr. 6.2: Grafické znázornění interpolace přes 3 kalibrační body

Na Obr. 6.3 je znázorněn model v programu Simulink, zjednodušený na pohyb jednoho článku prstu modelu ruky. Vektor otáčení $[1 \ 0 \ 0]$ říká, že článek rotuje okolo osy x o jednonásobek velikosti úhlu, vypočítaného blokem *Matlab Function*.

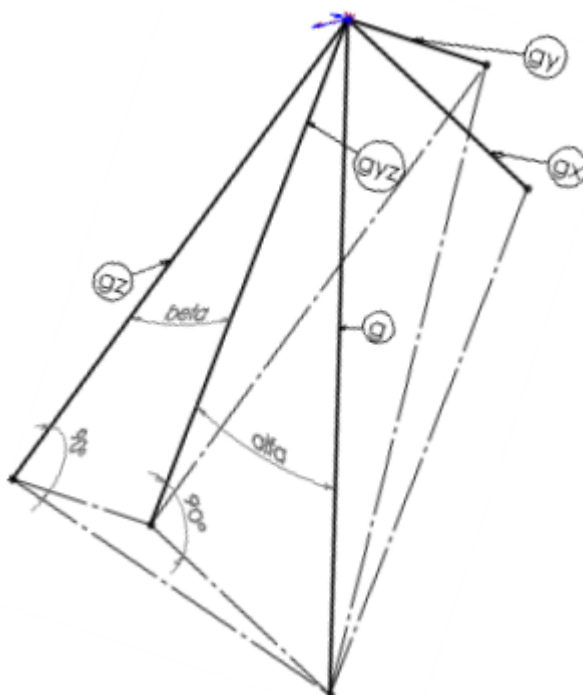


Obr. 6.3: Zjednodušený model v programu Simulink

Tímto způsobem byly řízeny všechny články prstů. Dále bylo třeba převést hodnoty z akcelerometru do tvaru potřebného k realizaci zobrazování náklonu rukavice ve VRML vizualizaci.

6.3 Převod hodnot z akcelerometru na úhel natočení VRML modelu

Posledním problémem byl převod hodnot z akcelerometru na natočení VRML modelu ruky. Pro názornost byl vytvořen náčrt situace v programu Solidworks (Obr. 6.4).



Obr. 6.4: Náčrt vytvořený pro grafické znázornění výpočtu natočení VRML modelu na základě dat z akcelerometru

Akcelerometr měří v jednotlivých osách průměty vektoru gravitace g do těchto os. Vektor g tedy bylo nutné rozložit do složek g_x , g_y , g_z . Velikosti g_x , g_y , g_z odpovídají hodnotě signálů přijímaných z akcelerometru. Naším záměrem bylo vypočítat z těchto složek velikosti úhlů alfa (odpovídá natočení v uzlu Dlan1) a beta (odpovídá natočení v uzlu Dlan2). Jako první bylo třeba vypočítat pomocí Pythagorovy věty velikost vektoru g_{yz} , který vznikl ze složek g_y a g_z . Výpočet byl proveden podle rovnice (1).

$$g_{yz} = \sqrt{g_y^2 + g_z^2} \quad (1)$$

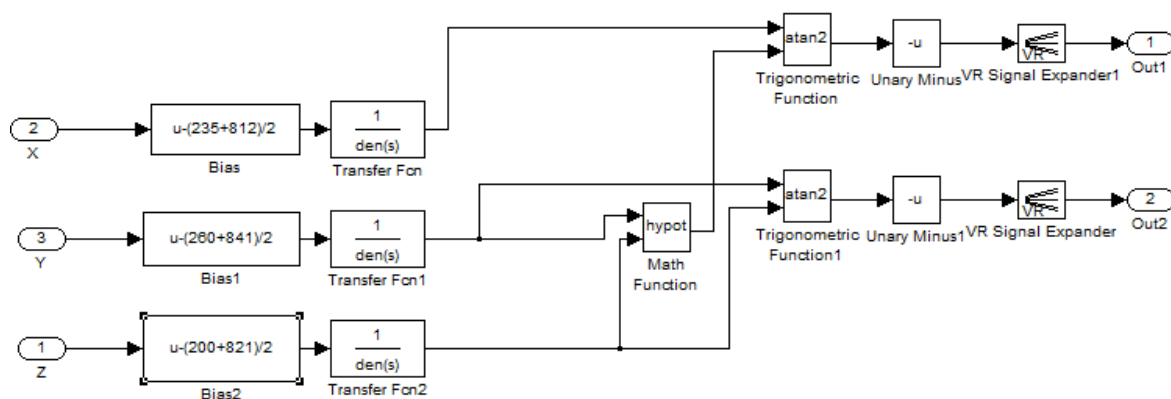
Vektory g_{yz} , g a g_x tvoří pravoúhlý trojúhelník, ve kterém se nachází požadovaný úhel alfa. K výpočtu úhlu alfa byla použita v programu Simulink použita funkce `atan2`. Jedná se o vylepšenou funkci arkus tangens, která uvažuje znaménka argumentů. Z toho důvodu ji lze použít pro všechny 4 kvadranty roviny. Funkce vrací úhel $\in (-\pi, \pi)$. Výpočet úhlu alfa byl proveden podle rovnice (2).

$$\alpha = \arctg\left(\frac{g_x}{g_y}\right) \quad (2)$$

Dalším krokem bylo určit úhel beta. K jeho výpočtu byl použit pravoúhlý trojúhelník, který tvoří vektory g_yz , g_z a g_y . K výpočtu úhlu beta byla opět využita funkce atan2 , která zajišťuje funkci ruky v plném rozsahu. Výpočet úhlu beta byl proveden podle rovnice (3).

$$\beta = \arctg\left(\frac{g_y}{g_z}\right) \quad (3)$$

Nyní bylo možné vypočítané úhly odesílat do bloku VRSink s VRML modelem. Na Obr. 6.5 je znázorněn model algoritmu v programu Simulink, který slouží k přepočítávání dat z akcelerometru na úhly alfa a beta. Jelikož data, která přicházela z akcelerometru, nenabývala hodnot „od 0“, bylo nutné provést „jednorázovou kalibraci“. Ta byla provedena sečtením maximálních a minimálních hodnot pro každou osu, podělených dvěma (tzv. offset). V programu Simulink byl pro výpočet offsetu použit blok *bias*.



Obr. 6.5: Algoritmus přepočtu dat z akcelerometru na úhly alfa a beta zhotovený v programu Simulink

7. TVORBA GRAFICKÉHO UŽIVATELSKÉHO ROZHRAŇÍ

Aby bylo možno rukavici používat bez nutnosti otevírání modelu a ručního provádění kalibrace, bylo nutno vytvořit jednoduché grafické prostředí, dále GUI (Graphical User Interface), které uživateli zpřijemní používání datové rukavice. Jednalo se o okno, které veškeré nutné funkce koncentrovalo na jedno místo. V zásadě bylo potřeba, aby bylo možné přes GUI provést kalibraci, spustit vizualizaci, zastavit vizualizaci a monitorovat stav baterie datové rukavice.

V programu Matlab slouží ke spuštění průvodce tvorby grafického rozhraní příkaz *guide*. Tvorba GUI byla po vizuální stránce banální, bylo třeba z nabídky vybrat objekt, který měl být do GUI vložen, a umístit jej na požadované místo. Bylo potřeba použít tlačítka, která umožňovala uživateli datovou rukavici kalibrovat a také spouštět a zastavovat vizualizaci, dále textová pole, ve kterých byl vypisován stav baterie a případné chyby, a okno, ve kterém byl zobrazován při kalibraci obrázek s předmětem, který měl uživatel v danou chvíli uchopit.

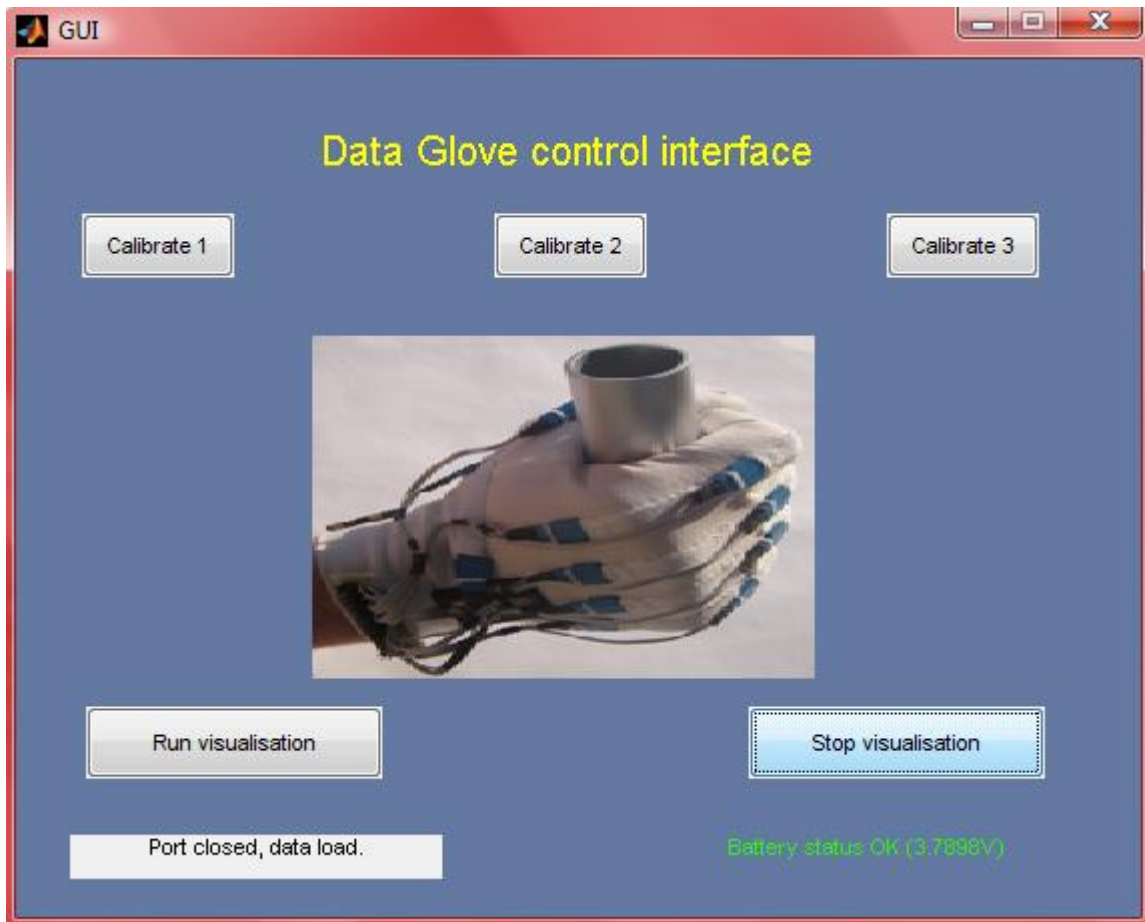
Po kliknutí na tlačítko *M-File* editor se otevřel M-File, ve kterém se nacházel inicializační kód, do kterého není doporučeno zasahovat, dále pak prázdné funkce akcí (callbacků) objektů, které byly zvoleny při tvorbě vizuální podoby GUI. Těmto funkcím lze přiřazovat v podstatě libovolné akce.

V našem GUI byly k jednotlivým kalibračním tlačítkům přiřazeny části algoritmu kalibrace. Spuštění vizualizace bylo provedeno příkazem:

```
set_param('jmeno_modelu', 'SimulationCommand', 'start');
```

K zastavení vizualizace sloužil stejný příkaz, funkci *SimulationCommand* byla však přiřazena akce 'stop'.

Podoba grafického rozhraní je znázorněna na Obr. 7.1.



Obr. 7.1: GUI pro ovládání datové rukavice

8. ZÁVĚR

Cílem bakalářské práce bylo rozšíření již hotové elektroniky rukavice o bezdrátovou komunikaci přes rozhraní Bluetooth, vytvoření virtuálního modelu ruky v jazyce VRML, sestavení modelu v programu Simulink, jehož úkolem bylo přepočítávání data přijatá z elektroniky rukavice na data, kterými bylo možné ovládat VRML model, a vytvořit grafické uživatelské rozhraní, jež usnadňovalo uživateli ovládání rukavice.

Úvod práce byl věnován základním informacím o virtuální realitě jako takové, o jejich výhodách a nevýhodách a řešerši v oblasti datových rukavic, která se zejména zabývala profesionálními provedeními.

V dalších kapitolách práce byla popsána řešení jednotlivých problémů. Cíle stanovené v první kapitole se podařilo splnit, při jejich zpracování jsem však narazil na několik problémů, které v určité míře komplikovaly navrhované řešení.

Prvním problémem byla nelinearita ohybových senzorů. Právě kvůli nelineární změně elektrického odporu v závislosti na úhlu ohybu bylo nutné navrhnout vcelku netradiční způsob kalibrace, navíc ve třech kalibračních bodech. Řešením tohoto problému by mohlo být použití ohybových senzorů s lineární závislostí změny elektrického odporu na úhlu ohybu senzoru. Při použití takového senzoru by při kalibraci postačovaly dva kalibrační body (natažené prsty/zatnutá ruka v pěst).

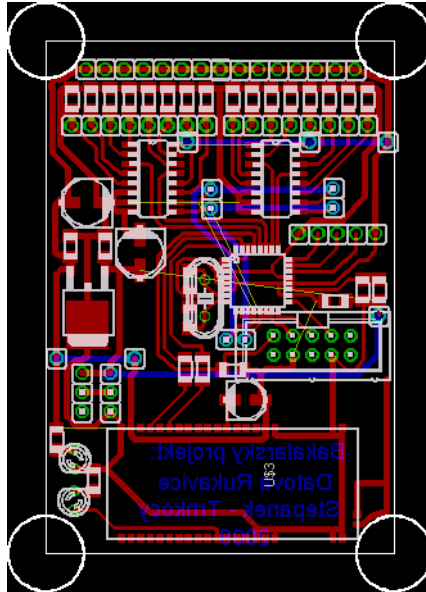
Druhým problémem byla relativně vysoká hardwarová náročnost. Ta byla způsobena programem Simulink, který musí v reálném čase přepočítávat příchozí data z elektroniky rukavice na data, kterými ovládá VRML model. Tento problém by mohl být zčásti vyřešen použitím menšího počtu ohybových senzorů, což by vedlo ke snížení počtu prováděných operací, zhoršila by se tím však věrnost vykreslovaných pohybů modelu ruky.

Způsobů využití datových rukavic je obrovské množství, námi vyrobená datová rukavice však příliš praktického využití nenabízí. Jedná se však o cenově dostupný prvek, umožňující nahlédnout uživateli pod pokličku virtuální reality a vyzkoušet si do ní proniknout na vlastní kůži.

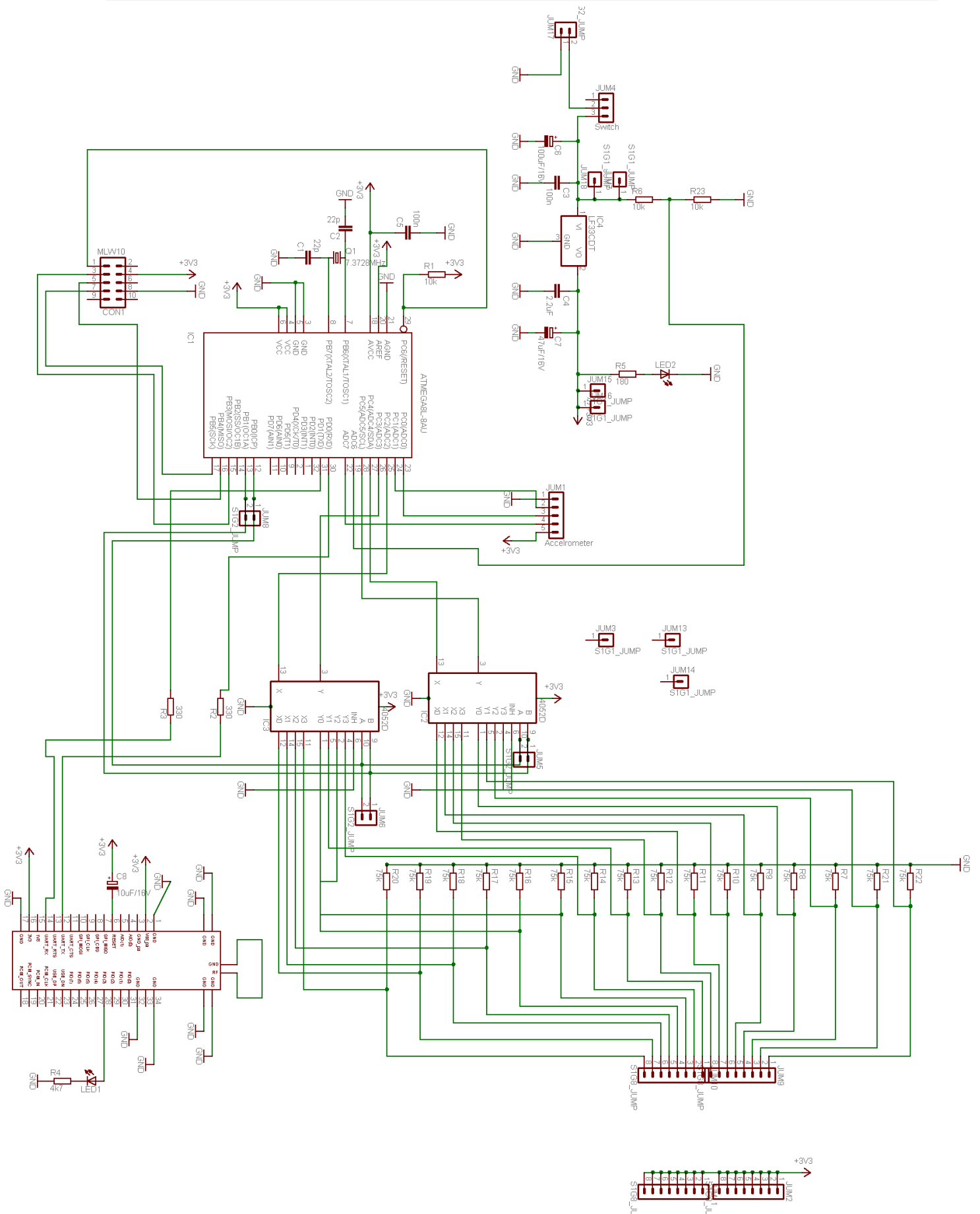
9. SEZNAM POUŽITÉ LITERATURY

- [1] Atmel: Popis mikrokontroleru Atmel AVR Atmega 8L, www.atmel.com/dyn/resources/prod_documents/doc2486.pdf [online], 2009-2
- [2] DGTech: DG5 VHand 2.0, <http://www.dg-tech.it/vhand/eng/> [online], 2009-5
- [3] Free2Move: Popis BlueTooth modulu F2M03C1, <http://www.free2move.se/index.php?type=view&pk=28> [online], 2009-5
- [4] Freescale: Popis akcelerometru MMA7260QR2, <http://sigma.octopart.com/133945/datasheet/Freescale-Semiconductor-MMA7260QR2.pdf>
- [5] Humusoft XSerial: Toolbox program Simulink pro sériovou komunikaci, <http://www.umt.fme.vutbr.cz/Plone/teamwork/matlab/serial-communication/xserial-od-humusoftu/xserial.zip/view> [online], 2009-4
- [6] Inition: Rukavice Immersion CyberGrasp, http://www.inition.co.uk/inition/product.php?URL=_product_glove_vti_grasp&SubCatID_=26 [online], 2009-5
- [7] Mathworks:, <http://www.mathworks.com/matlabcentral/> [online], 2009-3
- [8] Maxim: popis převodníku napět'ových úrovní, <http://datasheets.maxim-ic.com/en/ds/MAX3222-MAX3241.pdf> [online], 2009-1
- [9] SZTWIERTNIA, V. *Využití MEMS akcelerometru pro stabilizaci chodícího robotu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2008. 88 s. Vedoucí diplomové práce Ing. Robert Grepl, Ph.D.
- [10] Texas Instruments: popis multiplexeru CD4052B, http://www.datasheetcatalog.org/datasheets/120/109150_DS.pdf [online], 2009-2
- [11] Žára, Jiří: *Laskavý průvodce virtuálními světy*, <http://www.cgg.cvut.cz/LaskavyPruvodce/> [online], 2009-4
- [12] <http://virtualni-realita.molbud.cz/historie-virtualni-reality> [online], 2009-3
- [13] <http://media.digikey.com/photos/Atmel%20Photos/313-32-TQFP.jpg> [online], 2009-5

10. PŘÍLOHY



Obr. 10.1 : Deska plošného spoje elektroniky rukavice zhotovená v programu Eagle



Obr. 10.2 : Schéma zapojení elektroniky rukavice zhotovené v programu Eagle