

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF BIOMEDICAL ENGINEERING

SYSTEM ŘÍZENÍ ROBOTICKÉ RUKY Z OPERAČNÍHO ROBOTA DA  
VINCI

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JURAJ JAKUBÍK

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF BIOMEDICAL ENGINEERING

# SYSTÉM ŘÍZENÍ ROBOTICKÉ RUKY Z OPERAČNÍHO ROBOTA DA VINCI

THE DA VINCI ROBOTIC SURGEON HAND CONTROL SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JURAJ JAKUBÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ SEKORA

BRNO 2014



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav biomedicínského inženýrství

# Bakalářská práce

bakalářský studijní obor  
Biomedicínská technika a bioinformatika

**Student:** Juraj Jakubík

**ID:** 146191

**Ročník:** 3

**Akademický rok:** 2013/2014

## NÁZEV TÉMATU:

**Systém řízení robotické ruky z operačního robota Da Vinci**

## POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s funkcí robotické ruky operačního robota Da Vinci a s programem v LabVIEW pro generování řídicího bitového slova, dostupným na ÚBMI. 2) Prostudujte stupně volnosti robotické ruky a navrhnete přípravek pro ovládání mechanických prvků ruky. 3) Navažte na komunikační protokol z LabVIEW a navrhnete systém řízení zvolených motorů pomocí mikrokontroléru. 4) Realizujte navržený přípravek. 5) Uvažujte i systém kontroly a prokluzu hraničních poloh.

## DOPORUČENÁ LITERATURA:

[1] Servo motor controller using labVIEW [online]. Dostupný z <<http://forums.ni.com/>>.

[2] Intuitive Surgical, The da Vinci Surgical System [online]. Dostupný z <<http://www.intuitivesurgical.com/>>

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 30.5.2014

**Vedoucí práce:** Ing. Jiří Sekora

**Konzultanti bakalářské práce:**

**prof. Ing. Ivo Provazník, Ph.D.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Práca popisuje návrh systému riadenia nástroja ruky operačného robota Da Vinci amerického výrobcu Intuitive Surgical. Prvotný návrh spočíva v nadviazaní na program pre generovanie riadiaceho bitového slova v prostredí LabVIEW a následnom vykonaní pohybu nástroja pomocou krokových motorov prostredníctvom mikrokontroléra Arduino. Výsledne implementovaný riadiaci systém využíva servomotory ovládané analógovými manipulátormi pripojenými priamo na mikrokontrolér. Práca uvažuje ošetrovanie kontroly a preklzu hraničných polôh u oboch riadiacich systémov.

## **KLÚČOVÉ SLOVÁ**

Da Vinci, robot, mikrokontrolér, LabView, Arduino IDE, krokový motor, servomotor

## **ABSTRACT**

This bachelor thesis deals with the design of the control system of the Da Vinci robotic hand made by american company Intuitive Surgical. The initial draft consists of linking with the program for control bitword generation in the LabVIEW environment and subsequent execution of movement using stepper motors with the Arduino microcontroller. The resulting design is using servos controlled by analog manipulators connected directly to the microcontroller. Treatment of border position control and slip is considered within both control systems.

## **KEYWORDS**

Da Vinci, robot, microcontroller, LabView, Arduino IDE, stepper motor, servo

JAKUBÍK, Juraj *Systém riadenia robotickej ruky z operačného robota da Vinci*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínskeho inžinierstva, 2014. II s. Vedúci práce bol Ing. Jiří Sekora

## PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Systém riadenia robotickej ruky z operačného robota da Vinci“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisejúcich s právom autorským a o zmene niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Jiřímu Sekorovi za odborné vedenie, podnetné pripomienky, motivujúci a trpezlivý prístup.

Brno .....

.....

(podpis autora)

# OBSAH

Úvod	10
<b>1 Robotické systémy v medicíne</b>	<b>11</b>
1.1 Historický vývoj operačných robotov	11
1.2 Robotika a české zdravotníctvo	13
1.3 Operačný robot Da Vinci	13
1.3.1 Funkčné súčasti operačného robota	14
1.3.2 Anatómia nástroja robotickej ruky operačného robota	16
1.3.3 Možnosti zakončenia robotickej ruky	17
1.3.4 Pohyby nástroja ruky	19
1.4 Stupne voľnosti a ich určovanie	20
<b>2 Návrh systému riadenia nástroja ruky robota Da Vinci</b>	<b>21</b>
2.1 Systémový návrh	21
2.2 Voľba vhodnej pohonnej jednotky	22
2.3 Východiskový softvér	24
2.4 Hardvérové riešenie	26
2.5 Softvérové riešenie	28
<b>3 Realizácia</b>	<b>30</b>
3.1 Sériová komunikácia LabVIEW - Arduino	30
3.1.1 Komunikácia prostredníctvom NI-VISA	30
3.1.2 Simulácia dátového toku	32
3.1.3 LabVIEW rozhranie pre Arduino	34
3.2 Hardvér	34
3.2.1 Arduino hardvér	35
3.2.2 Arduino Uno	35
3.2.3 Arduino softvér	36
3.2.4 Motor Shield	37
3.2.5 Pohon	39
3.3 Softvér	41
<b>4 Diskusia výsledkov</b>	<b>43</b>
4.1 Kontrola a preklz hraničných polôh	43
4.2 Komunikácia	44
4.3 Ovládacie módy	44
<b>5 Záver</b>	<b>46</b>

Literatúra	47
Zoznam symbolov, veličín a skratiek	49
Zoznam príloh	50
A Schéma dosky plošného spoja	I
B Návrh dosky plošného spoja	II
C Zoznam použitých súčiastok	II



# ZOZNAM OBRÁZKOV

1.1	Hlasom riadený systém AESOP [4] . . . . .	12
1.2	Systém Da Vinci Si (Zľava doprava: riadiace konzoly, operačná konzola, videovozík)[4] . . . . .	14
1.3	Operačná konzola robota Da Vinci [7] . . . . .	15
1.4	Premena pohybu manipulátorov (obrázok dole) na identický pohyb nástrojov (obrázok hore)[4] . . . . .	16
1.5	Pracovný nástroj a jeho stavba[12] . . . . .	17
1.6	Variabilita zakončenia operačného nástroja[8] . . . . .	18
1.7	Znázornenie pohybu ramena okolo statického rotačného bodu T . . . . .	19
1.8	Možné pohyby nástrojovej hlavy . . . . .	20
2.1	Bloková schéma riadiaceho systému . . . . .	21
2.2	Gamepad Logitech Dual Action[10] . . . . .	22
2.3	Krokový motorček ROHS[11] . . . . .	23
2.4	Užívateľské rozhranie programu na generovanie bitového slova [12] . . . . .	24
2.5	Komunikačný protokol . . . . .	25
2.6	Príklad podoby komunikačného protokolu vrátane výpočtu kontrolného súčtu . . . . .	26
2.7	Spodná strana tela nástroja ruky operačného robota Da Vinci [12] . . . . .	27
2.8	Vývojový diagram programu . . . . .	28
3.1	Schéma zapojenia blokov VISA za účelom dátového prenosu na sériový port . . . . .	31
3.2	Simulácia dátového toku medzi prostredím LabVIEW a mikrokontrolérom Arduino . . . . .	33
3.3	Schéma blokov pre ovládanie serva pomocou Adruina naprogramovaného v prostredí LabVIEW . . . . .	34
3.4	Vývojová platforma Arduino Uno . . . . .	36
3.5	Grafické vývojové prostredie platformy Arduino . . . . .	37
3.6	Schématický návrh prvku Motor Shield . . . . .	38
3.7	Digitálne servo Hitec [19] . . . . .	40
3.8	Vývojový diagram vytvoreného softvéru . . . . .	41
4.1	Mechanizmus šnekového súkolia [18] . . . . .	43

## ZOZNAM TABULIEK

2.1	Možné bitové dvojice pre riadenie jedného motorčeka . . . . .	25
2.2	Pravdivostná tabuľka funkcie XOR . . . . .	26
2.3	Špecifikácia vlastností ovládacích prvkov nástroja robotického ruky . .	27
3.1	Technické parametre serva Hitec [19] . . . . .	40

# ÚVOD

Odjakživa sa ľudia snažia používať zariadenia určené na uľahčovanie práce a vykonávanie činností, ktoré si vyžadujú značné fyzické úsilie. Tento pretrvávajúci trend neustále vedie k vývoju nových a sofistikovanejších „otrokov“, robotov.

V modernej histórii bolo slovo robot zavedené českým spisovateľom Karlom Čapkovi v jeho divadelnej hre R.U.R. v roku 1923. Východiskom sa stalo slovo robota, pomenovanie pre ťažkú fyzickú prácu. [1]

Isaac Asimov bol prvý, kto použil slovo robotika, a to vo svojej poviedke Roundabout (Hra na naháňačku) v roku 1942. V tomto diele taktiež sformuloval 3 zákony robotiky:

1. Robot nesmie ublížiť človeku alebo svojou nečinnosťou dopustiť, aby bolo človeku ublížené.
2. Robot musí poslúchnuť človeka, okrem prípadov, keď je to v rozpore s prvým zákonom.
3. Robot sa musí chrániť pred poškodením, okrem prípadov, keď je to v rozpore s prvým alebo druhým zákonom. [2]

Dnešné funkčné roboty zaznamenávajú neustály pokrok. Už nie sú ďalej schopné len samostatného pohybu. Vedia držať predmety, hrať na hudobné nástroje, zbierať úlomky na Marse, ale tiež spolupracovať pri záchranných akciách. Stávajú sa súčasťou nielen všetkých odvetví priemyslu a vedy, ale aj každodenného života.

Roboty vplyvom inovácií nadobúdajú nové rozmery, tvary, funkcie, schopnosti, a preto je veľmi ťažké vytvoriť exaktnú definíciu tohoto slova. Používa sa niekoľko variant, avšak univerzálna stále neexistuje. Joseph F. Engelberger, tvorca prvého priemyselného robota prezývaný tiež otec robotiky, raz vtipne poznamenal:

*„Neviem definovať robota, ale keď ho uvidím, spoznám ho.“* [3]

Robotizované technológie sa závratným tempom integrujú aj v medicínskej sfére, kde nachádzajú široké uplatnenie. Pomáhajú starším alebo postihnutým pacientom pri pohybe, stravovaní, môžu dokonca nahradiť pri lôžku lekára, ktorý s pacientom komunikuje z kancelárie prostredníctvom internetového spojenia.

Vyšší level využitia spočíva v spolupráci pri operáciách alebo v ich samotnom vykonávaní. Príkladom takéhoto zariadenia je systém Da Vinci od americkej spoločnosti Intuitive Surgical Inc., operačný robot, prostredníctvom ktorého chirurg vykonáva zákrok s maximálnou presnosťou na pacientovi ležiacom vo vedľajšej miestnosti. Meno dostal po renesančnom umelcovi Leonardovi da Vincim, veľkom vedcovi, umelcovi a inovátorovi, ktorého odvážne myšlienky a vynálezy sa častokrát v priebehu storočí stali skutočnosťou. Práca sa bude zaoberať návrhom riadiaceho systému nástroja ruky tohoto robota.

# 1 ROBOTICKÉ SYSTÉMY V MEDICÍNE

Roboty sú dnes bežnou súčasťou operačných sál, kde plnia viaceré funkcie. Podľa nich možno chirurgickú robotiku rozdeliť na dve hlavné kategórie.

Prvú tvoria chirurgické pracoviská, kde roboty priamo vykonávajú chirurgické zákroky. Buď prevádzajú zákrok dopredu naplánovaný chirurgom, ktorý následne na priebeh len dohliada, alebo sú riadené na diaľku pomocou ovládacích prvkov v reálnom čase. Vtedy hovoríme o tzv. teleprezenčnej robotike.

Druhou hlavnou kategóriou chirurgickej robotiky sú zákroky asistované robotmi. Tu roboty neprichádzajú priamo do kontaktu s ľudským telom, ale vykonávajú iné s operáciou súvisiace činnosti, ako je napríklad podávanie nástrojov.

Primárne však robotické systémy predstavujú prostriedok, prostredníctvom ktorého vie chirurg operovať oveľa presnejšie, bez zbytočného poškodzovania okolitého tkaniva. Takého miniinvazívne zákroky prinášajú pacientom nespočetné výhody, akými sú rýchla rekonvalescencia a návrat do bežného života. Operačné roboty sú zatiaľ len sofistikované nástroje bez vlastnej inteligencie, ktoré nie sú schopné diagnostiky a následného plánovania operačného postupu. Napriek tomu posunuli chirurgiu o obrovský kus dopredu a ďalší vývoj bude smerovať práve ku úplnému nahradeniu človeka, chirurga, samostatne mysliacim operačným robotom.

## 1.1 Historický vývoj operačných robotov

Prvé funkčné roboty sa objavili v polovici 20. storočia. V roku 1954 George Devol a Joseph Engelberger prišli s objavom robotickej ruky riadenej elektronickým regulátorom. Pohyb bol naprogramovaný a prevádzal sa pomocou hydraulického systému. Zariadenie pomenované Unimate bolo súčasťou výrobnéj linky v koncerne General Motors. Značné vylepšenie priniesol v roku 1978 objav Victora Scheinmana, ktorý predstavil svoj vynález PUMA (*Programmable Universal Manipulation Arm*). Hlavnou odlišnosťou oproti predošlému modelu bolo zväčšenie počtu stupňov voľnosti a schopnosť výkonu zložitých technických požiadaviek. PUMA, vybavená elektrickým pohonom, bola vďaka svojim prednostiam po mnoho rokov používaná ako štandardný robotický systém vo viacerých odvetviach priemyslu.[1]

História chirurgických robotických systémov začala v roku 1985 používaním zariadenia PUMA-560. V neurochirurgii slúžil ako nástroj pre veľmi presnú manipuláciu pri biopsii mozgu pod CT kontrolou. [4]

V roku 1988 bol vyrobený robotický systém Probot pre účel transuterálnej resekcie prostaty, ktorá bola ešte daného roku aj vykonaná. V predoperačnej fáze bol vytvorený 3D model prostaty, kde chirurg vyznačil krajné polohy oblasti, ktorá má

byť odstránená. Trajektóriu pohybu si následne vypočítal systém sám a pohyby vykonal. [1]

Vznik chirurgických robotov mal tri hlavné príčiny: operácia na diaľku, vojenskú medicínu a snaha o maximálnu miniinvazívnosť výkonov. V 70. a 80. rokoch prebiehal vojenský program, ktorého cieľom bolo vytvorenie chirurgického systému určeného na armádne účely. Zámerom bola schopnosť výkonu chirurgického zákroku priamo na bojisku, pričom obsluhujúci lekár by bol v bezpečí. K realizácii však nedošlo a prototyp bol neskôr odkúpený spoločnosťou Intuitive Surgical, ktorá na konci 90. rokov aplikovala technológiu do civilnej sféry. [4]

RoboDoc od firmy IBM sa stal prvým robotickým systémom s aplikáciou v ortopédii pri náhrade bedrového kĺbu, kde umožňoval vyvrtanie presnej diery do stehennej kosti. Vývojovo dokonalejšie Casper system a Acrobat boli schopné vykonať niektoré pasáže v operačnom postupe pri výmene kĺbu v automatickom režime. [1]

Prvý robot-chirurg bol v roku 1994 predstavený americkou spoločnosťou US Computer Motion Co. Jednalo sa o systém AESOP (*Automated Endoscopic System for Optical Positioning*). Základnou a funkčnou súčasťou prístroja bola robotická ruka so 7 stupňami voľnosti, ktorá menila polohu endoskopu. Poskytoval stabilné pole videnia a o 2 roky neskôr bol systém schopný rozpoznávať hlasové príkazy chirurga.[1]



Obr. 1.1: Hlasom riadený systém AESOP [4]

Na konci 90. rokov boli predstavené dva systémy. Systém Da Vinci v roku 1997,

ktorý bol produktom práve spoločnosti Intuitive Surgical. Príbuzným a koncepcne podobným systémom sa stal robot ZEUS, ktorý vznikol o rok neskôr a slúžil taktiež k účelom endoskopickéj chirurgie. V súčasnosti však už nie je dostupný pre klinické účely a jednoznačným lídrom na trhu v oblasti robotickéj chirurgie je firma Intuitive Surgical Inc. [4]

## 1.2 Robotika a české zdravotníctvo

Prvý robotický systém sa na územie Českej republiky dostal na konci októbra 2005. Jednalo sa o operačného robota Da Vinci od amerického výrobcu. V súčasnej dobe je používaných celkovo 9 robotických systémov v 7 nemocniciach, a to v Nemocnici na Homolce, Ústrednej vojenskej nemocnici v Prahe, Fakultnej nemocnici u sv. Anny v Brne, nemocnici sv. Zdislavy v Mostištích, Masarykovej nemocnici v Ústí nad Labem, Nemocnici s poliklinikou v Novom Jičíne a Fakultnej nemocnici Olomouc, kde bol systém inštalovaný v roku 2009. Nemocnica na Homolce využíva 2 robotické systémy, jeden umiestnený na centrálnej operačnej sále a druhý na sále kardiochirurgickej. Taktiež dvoma systémami disponuje Masarykova nemocnica v Ústí nad labem, kde jeden systém využíva multioborová sála a druhý je umiestnený v školiacom centre. Spoločnosť Hospimed, strategický partner amerického výrobcu Intuitive Surgical, zastrešuje nielen český trh, ale aj Slovensko, Maďarsko, Slovinsko, Ruskú Federáciu a ďalšie krajiny Východnej a Strednej Európy.[5]

K máju 2012 bolo inštalovaných viac ako 1840 systémov da Vinci v približne 1450 nemocniciach po celom svete. [4]

## 1.3 Operačný robot Da Vinci

Operačný robot Da Vinci, vytvorený v roku 1997 americkou spoločnosťou Intuitive Surgical Inc., je jediným zariadením svojho druhu dostupným na trhu. Aplikácia tohoto systému siaha do mnohých oborov medicíny. Používa sa najmä v urológii (radikálna operácia prostaty, parciálna resekcia obličky, plastika obličkovej panvičky a nefrektómia), brušnej chirurgii (resekcia žalúdka, čriev, konečníka a pečene), cievnej chirurgii (rekonštrukcia brušnej aorty a panvových tepien) a kardiochirurgii. [6]

Robotický systém Da Vinci prináša výhody ako na strane chirurga, tak aj pacienta. Únava operátora počas náročných výkonov nie je tak veľká, systém navyše eliminuje prípadné nežiadúce javy, ako vyčerpaním vznikajúci tras rúk. Zvýšenie bezpečnosti zo sebou prináša aj použitie nástrojov, ktoré sú presné, spoľahlivé, s veľkým rozsahom pohybu aj v obmedzenom priestore. Celkový výkon je ku pacientovi šetrnejší, spôsobuje menšie jazvy, minimalizuje krvné straty a možnosť výskytu

infekcie.



Obr. 1.2: Systém Da Vinci Si (Zľava doprava: riadiace konzoly, operačná konzola, videovozík)[4]

Systém Da Vinci má dnes 3 vývojové generácie odlišujúce sa konštrukčným riešením jednotlivých častí. Systém Standard bol prvý typom aktívne používaným v medicínskej praxi, bol inštalovaný v objeme približne 500 kusov a dnes sa už nevyrába. Druhá generácia, modely Da Vinci S a S HD, sa na prvý pohľad od predošlého modelu neodlišujú. Podstatnou konštrukčnou zmenou je presunutie výkonových ovládacích modulov ramien do operačnej konzoly. Taktiež je tu použitý nový stereoskopický zobrazovací modul, ktorý umožňuje využívať vyššie rozlíženie obrazu. Najnovším modelom v rade Da Vinci je Da Vinci Si, ktorý má kompletne prepracovanú konštrukciu videovozíka a ovládacej konzoly. Model taktiež umožňuje pripojenie druhej ovládacej konzoly. Tú je možné použiť pre spoluprácu dvoch chirurgov, ktorí tak súčasne ovládajú všetky nástroje ramien. Veľmi vhodné je využitie druhej konzoly ako tréningového modulu pre zaškolovanie nových chirurgov.

### 1.3.1 Funkčné súčasti operačného robota

Celý robot sa skladá z troch základných častí (viď obrázok 1.2). Prvou je ovládacia konzola (podľa modelovej rady môže byť 1 alebo 2), pomocou ktorej chirurg ovláda ďalšiu časť, a to operačnú konzolu s nástrojmi. Poslednú časť tvorí videovozík, ktorý obsahuje kamerové jednotky a ďalšie moduly pre spracovanie videosignálu. [7]

Operačná konzola umiestnená nad lôžkom operovaného pacienta tvorí výkonnú jednotku a priamo pohybuje nástrojmi. Skladá sa zo štyroch pohyblivých ramien

(viď obrázok 1.3). Jedno rameno je zakončené kamerou, ktorá sníma dianie v ope-rovanej oblasti. Ostatné ramená sú zakončené špeciálnymi nástrojmi, ktoré presne odpovedajú chirurgovým aktuálnym požiadavkám a ich typ je počas operácie pravi-delne obmieňaný. Všetky časti komplexu musia byť sterilné, sterilizácia sa prevádza pred každým zákrokom.

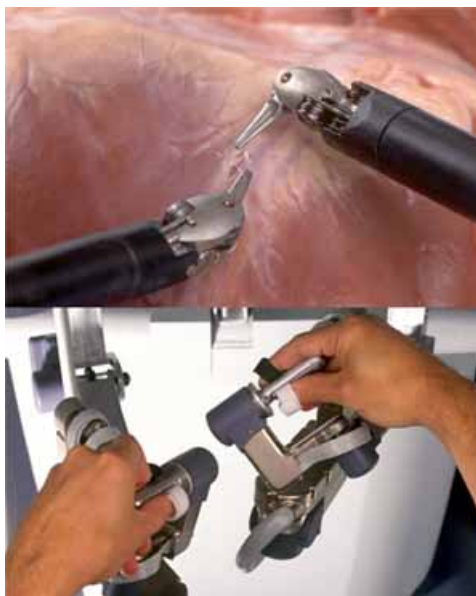


Obr. 1.3: Operačná konzola robota Da Vinci [7]

Základnou funkciou systému je prevod pohybov rúk operátora na pohyb pracovných nástrojov. Na prvý pohľad jednoduchá činnosť je realizovaná prepracovaným technickým riešením, ktorým je práve systém Da Vinci jedinečný. Pohyb rúk sa sníma pomocou takzvaných manipulátorov-joystikov (viď obrázok 1.3.1). Jedná sa o pákový mechanizmus s ôsmimi pohyblivými kĺbmi – osami. V každej z týchto osí sa priebežne sníma pozícia. Počítač následne určuje polohu prstov v reálnom čase a transformuje ju na dáta pre riadenie pohybu ovládaných nástrojov. Celok je doplnený motorom pre každú os, ktorý eliminuje pôsobenie gravitácie na mechanizmus a simuláciu spätnej väzby v prípade prekážky v pohybe nástroja, napr. kolízii ramien.

Podobne, avšak v obrátenom zmysle, funguje nástrojové rameno. Obsahuje sedem pohonných elektromotorických jednotiek, ktoré nastavením pozície jednotlivých osí určujú polohu nástroja. Oproti ovládaciemu manipulátoru (snímaču pohybu), je tu o jednu os menej. Z konštrukčných dôvodov sú motory umiestnené pri päte ramena, tam, kde je rameno upevnené. Prevod pohybu ku nástroju je riešený ocelovými





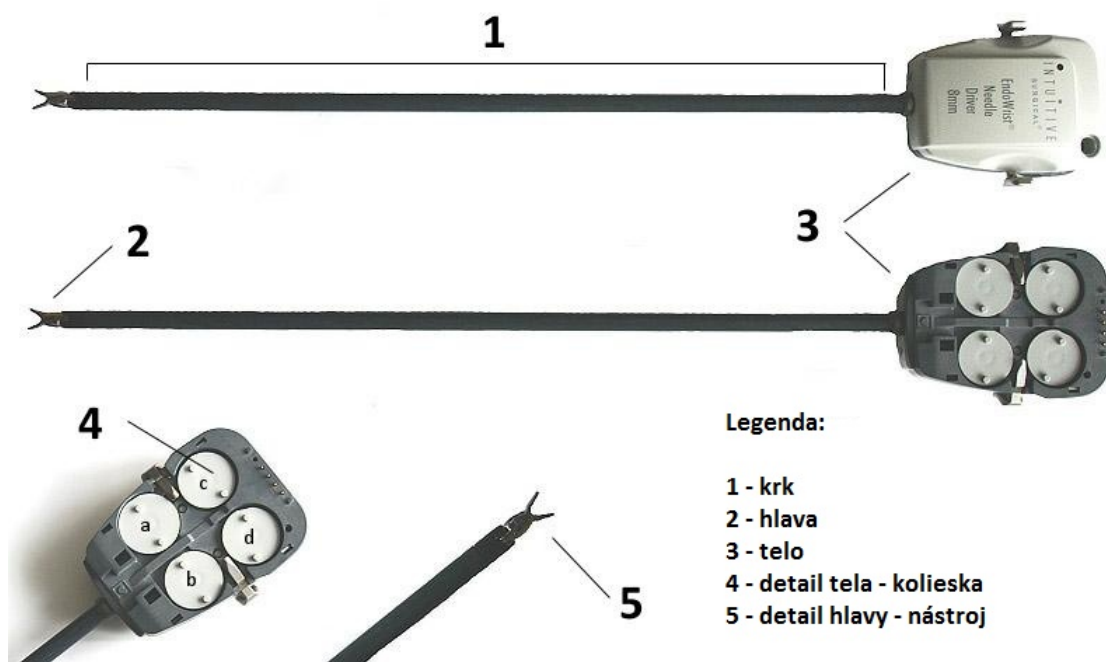
Obr. 1.4: Premena pohybu manipulátorov (obrázok dole) na identický pohyb nástrojov (obrázok hore)[4]

lankami. Operatér neovláda robota len prostredníctvom rúk. Pomocou nášlapných pedálov je pri použití odpovedajúceho nástroja možné aktivovať koagulačný proces, prípadne prepnúť z režimu pohybu nástrojov, na režim kamery a zamerať sa tak na ovládanie optického systému. Tento systém pozostáva z dvoch paralelných kamier. Každá je vybavená svetelným zdrojom a má vlastný optický kanál. Kľúčovou súčasťou systému je synchronizátor, ktorý zlučuje obrazy z oboch kamier a vytvára tak výsledný reálny trojdimenzionálny obraz. Chirurg tak dostáva dokonalý obraz o priestorovom rozložení orgánov a tkanív v mieste záujmu.

### 1.3.2 Anatómia nástroja robotického ruky operačného robota

Nástroj ruky operačného robota Da Vinci sa skladá z troch hlavných častí: hlava, krk a telo. Každá časť má špecifickú konštrukciu, spolu tvoria dokonalý funkčný celok. Nástrojová rada je nazvaná *EndoWrist*, keďže hlavným zámerom bolo priblížiť sa pohyblivosťou ľudskému zápästiu.

Najzložitejšou časťou nástroja je telo, krabička s rozmermi 120 x 60 mm, ktorá fixuje nástroj na rameno operačného robota. Spodná strana obsahuje štyri otočné kolieska, každé s dvomi výstupkami, ktoré sú spojené s protikusom nachádzajúcim sa na ramene robota. Vytvorí sa tak spoj, prostredníctvom ktorého sa prenáša mechanický pohyb cez krk na hlavu nástroja. Pre vykonanie rôznych druhov pohybu je nutná rotácia príslušného kolieska. Koliesko *a* rotuje hlavou ruky v rozmedzí viac



Obr. 1.5: Pracovný nástroj a jeho stavba[12]

ako 360 stupňov. Koliesko *b* nakláňa hlavu v priestore. Kolieskami *c* a *d* sa zatvárajú a otvárajú jednotlivé krídla hlavy. Každé krídlo je ovládané samostatne, čo výrazne zvyšuje variabilitu pohybu pri uchopovaní alebo strihaní. Každé koliesko určuje jeden druh pohybu, hlava má teda celkovo 4 stupne voľnosti. Každý pohyb má svoj limitovaný rozsah určený mechanickou konštrukciou. Napriek tomu je pohyblivosť oproti ľudského zápästiu neporovnateľne vyššia.

Krk zabezpečuje prepojenie medzi telom a hlavou nástroja. Konštrukčne sa jedná o 395 mm dlhú dutú trubičku z ohybných kompozitných materiálov. Vnútro sú vedené ocelové lanká, ktoré ohýbajú hlavu nástroja v závislosti na rotácii koliesok. Lanká svojím skracovaním a predlžovaním simulujú činnosť ľudských svalov a šliach. Výhodou tohto mechanického prevodného systému je extrémna presnosť.

Dve vyššie spomenuté časti sú u všetkých nástrojov *EndoWrist* rovnaké, líšia sa len zakončením – hlavou. Hlava nástroja má mnoho prevedení, ktoré sú bližšie popísané v nasledujúcej časti.

### 1.3.3 Možnosti zakončenia robotickej ruky

Spoločnosť Intuitive Surgical ponúka *EndoWrist* nástroje s obrovským spektrom zakončení. Každá hlava má špecifický dizajn a je dimenzovaná na výkon konkrétneho operačného postupu. Najpoužívanjšie sú nožnice, skalpely a rôzne uchopovače

používané napríklad pri zošívaní.

Pred každým operačným zákrokom sa určí spektrum nástrojov, ktoré bude nutné počas zákroku pripevniť na ramená operačného robota. Pri požiadavke na zmenu nástrojovej hlavy je nutné vymeniť celý nástroj. Rameno nesúce zobrazovací systém sa počas zákroku nijako nemodifikuje, zabezpečuje neustály prenos obrazu do ovládačnej konzoly, prostredníctvom ktorej chirurg vizuálne kontroluje priebeh operačného zákroku a pohyby nástrojov. Konštrukčne sa telá nástrojov neodlišujú, môžu byť teda upevnené na ktorékoľvek rameno robota.



Obr. 1.6: Variabilita zakončenia operačného nástroja[8]

Nástroje sú buď čisto mechanické (kliešte, uchopovače), alebo navyše využívajú elektrický prúd za účelom koagulácie, prípadne rezu a resekcie tkaniva (elektrokautery). Na obrázku 1.6 je možné vidieť širokú škálu rôznych nástrojov, ktoré sú dostupné na trhu a možno ich zakúpiť:

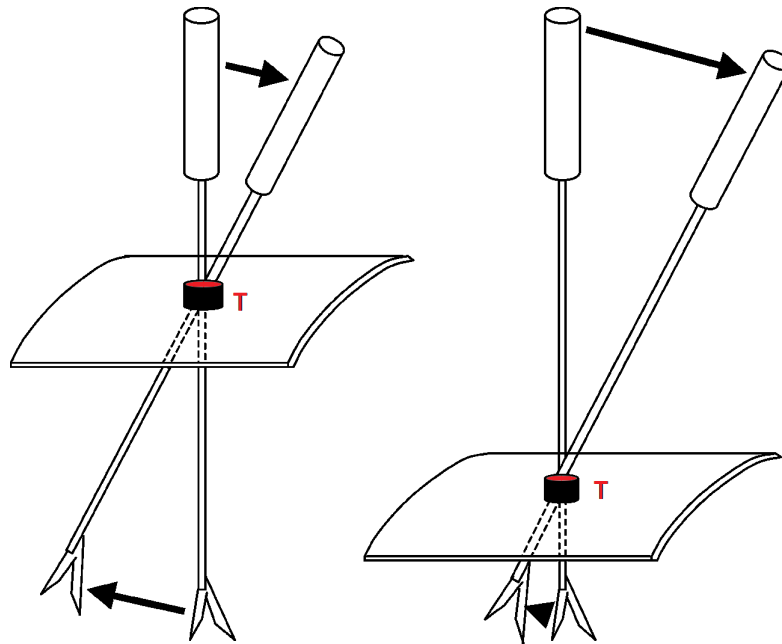
1. *EndoWrist Black Diamond Micro Forceps 8mm* - úchopové mikro kliešte
2. *EndoWrist HotShears 8mm* - monopolárne zahnuté nožnice
3. *EndoWrist Maryland Bipolar Forceps 8mm* - bipolarárne disekčné kliešte
4. *EndoWrist Cobra Grasper 8mm* - uchopovač hustého tkaniva
5. *EndoWrist Stabilizer 12mm* - nástroj pre prepojenie dvoch ciev počas endoskopických operácií bijúceho srdca[8]

Každý nástroj má na vrchej strane tela uvedené základné popisné informácie, ktoré uľahčujú orientáciu a zrýchľujú tak priebeh celého výkonu. Jedná sa o typ použitia

a veľkosť funkčnej pracovnej plochy. Napríklad *EndoWrist Scissors 8 mm* – nástroj EndoWrist zakončený nožnicami s veľkosťou funkčnej pracovnej plochy 8 mm. Štandardne dostupné sú aj nástroje v veľkosťou funkčnej pracovnej plochy 5 mm. Tie sú uprednostňované niektorými chirurgami najmä v pediatrii a brušnej chirurgii.[8]

### 1.3.4 Pohyby nástroja ruky

Celková pohyblivosť operačného robota Da Vinci počas operačného výkonu je daná zlúčením možných pohybov hlavy nástroja a robotického ramena. Rameno ako celok zabezpečuje všeobecnú orientáciu v priestore a riadi hĺbku vniku nástrojovej hlavy do pacientovho tela. Robotická ruka sa pohybuje okolo pevného rotačného bodu označeného *T* na obrázku 1.7. Bod odpovedá miestu vniku nástroja do tela pacienta, čím sa eliminuje poškodzovanie tkanív spôsobené zmenou polohy nástrojovej hlavy v rámci operovanej oblasti.



Obr. 1.7: Znáznornenie pohybu ramena okolo statického rotačného bodu T

Pohyby na obrázku 1.8 označené číslami 1,2 a 3 vykonáva rameno robota, ktoré v tomto prípade nie je predmetom záujmu. Pre potreby tejto práce je k dispozícii nástroj *EndoWrist Needle Driver 8mm*, na ktorého pohyblivosť hlavy sa práca zameriava. Samotná hlava nástroja disponuje 4 stupňami voľnosti v priestore (problematika je podrobnejšie rozobraná v podkapitole 1.4). Rotácia nástroja je sprostredkovaná rotáciou celého krku, čomu odpovedá číslo 4. Ohyb nástrojovej hlavy na strany je možný v rozmedzí približne 180 stupňov a reprezentuje ho číslo 5.



Obr. 1.8: Možné pohyby nástrojovej hlavy

## 1.4 Stupne voľnosti a ich určovanie

Stupeň voľnosti pohybu objektu je počet ľubovoľne voliteľných nezávislých číselných údajov, ktoré určujú polohu daného objektu vzhľadom na vytýčený priestor. Zjednodušene možno povedať, že stupne voľnosti udávajú pohybové možnosti daného telesa. Objektom môže byť bod, sústava bodov (teleso), alebo sústava telies.

Pre **bod** platí:

- V priestore má bod 3 stupne voľnosti, pretože je možné určiť jeho tri nezávislé súradnice  $(x, y, z)$ .
- Na ploche (špeciálny prípad v rovine) má bod 2 stupne voľnosti, pretože je možné určiť len dve nezávislé súradnice, tretia je určená funkciou, ktorá definuje plochu.
- Na krivke (špeciálny prípad na priamke) má bod 1 stupeň voľnosti, pretože je možné určiť len jednu nezávislú súradnicu.

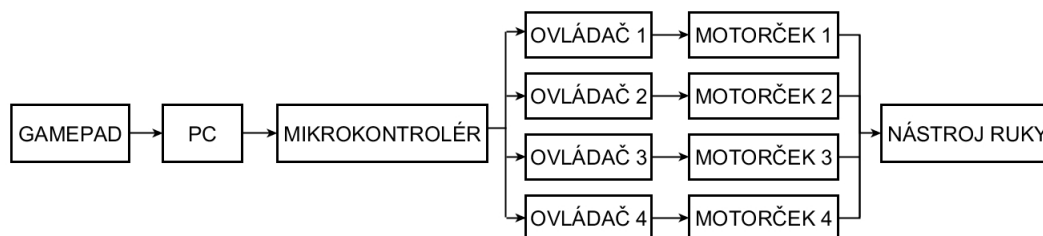
**Teleso** má v priestore 6 stupňov voľnosti, pretože je možné určiť tri nezávislé súradnice  $(x, y, z)$  jeho ťažiska a môže sa natočiť okolo troch nezávislých osí. [9]

## 2 NÁVRH SYSTÉMU RIADENIA NÁSTROJA RUKY ROBOTA DA VINCI

Pri návrhu systému riadenia nástroja ruky robota Da Vinci je kladený dôraz najmä na jednoduchosť prevedenia, dostupnosť použitých komponentov a intuitívne ovládanie požadovaných pohybov.

### 2.1 Systémový návrh

Hardvérové riešenie je realizované prostredníctvom niekoľkých základných komponentov, viď obrázok 2.1. Nosný prvok predstavuje systémová jednotka počítača (ďalej len PC), ktorá je doplnená o ďalšie periférne zariadenia.



Obr. 2.1: Bloková schéma riadiaceho systému

Pre jednoduché ovládanie robotického ruky je nutné zvoliť vhodné zariadenie, ktoré by bolo užívateľsky prívetivé, a zároveň spĺňalo technické požiadavky. Ako optimálne sa ukázalo herné zariadenie gamepad (viď obrázok 2.2). Tento ovládač je intuitívny, ergonomický, človek s nimi môže pracovať bez nutnosti zdĺhavej inštrukcie. Hlavnými ovládacími prvkami sú 2 analógové manipulátory (joisticky), prostredníctvom ktorých je možné komplexne ovládať nástroj ruky disponujúci 4 stupňami voľnosti. Rozsah joystickov je od 0, kedy sú v danom smere v strednej, kludovej polohe až po hodnotu  $\pm 32768$ . Tento uvedený celočíselný rozsah je možné definovať vo všetkých dimenziách (os  $x$ ,  $y$ ,  $z$  a rotácia  $z$ ). Veľkosť intervalu zabezpečuje veľmi vysokú presnosť riadenia pohybu. Pri návrhu východiskového softvérového riešenia v prostredí LabVIEW bola ako spodná hranica rozsahu zvolená hodnota  $\pm 4096$ . Počíta sa tu s opotrebovaním zariadenia, kedy je prakticky nemožné nastaviť východiskové polohy oboch manipulátorov tak, aby boli počiatkové hodnoty nulové. Program tak generuje nenulové 16bitové slovo len v prípade, že došlo k výraznejšej zmene polohy joysticku na znak, že pohyb nástroja ruky v danom smere je žiadúci. Gamepad je prepojený s počítačom prostredníctvom USB portu.



Obr. 2.2: Gamepad Logitech Dual Action[10]

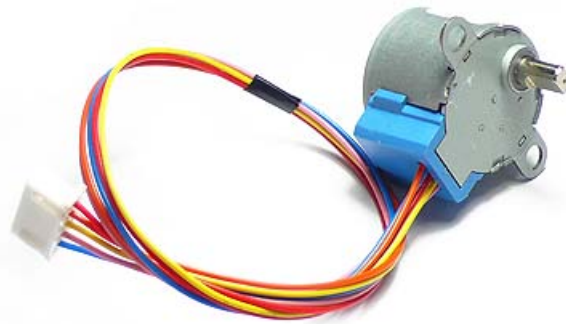
PC zostava pomocou programu pre tvorbu 16bitového slova v prostredí *LabVIEW* zaznamenáva pohyby oboch manipulátorov a získava potrebné informácie o smere a rotácii žiadaného pohybu hlavy nástroja. Na základe týchto informácií zostaví sériový komunikačný protokol (viď obrázok 2.5), prostredníctvom ktorého počítač komunikuje s externým zariadením – mikrokontrolérom, ktorý ďalej riadi pohyb motorčekov.

Pre potreby tejto práce je k dispozícii mikrokontrolér Arduino Uno. Podrobnejšie informácie o tejto platforme sú uvedené v podkapitole 3.2. Pre ovládanie 4 krokových motorčekov je celkovo potrebných 16 digitálnych pinov (každý motorček má 4 digitálne piny určené na riadenie), čo naznačuje potrebu použitia 2 vývojových platforiem talianskeho výrobcu, ktoré budú vzájomne prepojené. Pri realizácii pohonu prostredníctvom servo motorčekov postačuje jedna platforma Arduino Uno rozšírená o shield, ktorý je dimenzovaný na separabilné ovládanie až štyroch servo pohonných jednotiek.

## 2.2 Voľba vhodnej pohonnej jednotky

Pohon nástroja robotickej ruky je možné prevádzať pomocou komerčne dostupných elektromotorov. K dispozícii je niekoľko koncepčne odlišných systémov, pričom pre tento účel sa ako najvhodnejšie javia modelárske servá, prípadne krokové motorčeky.

Ovládač krokového motorčeka, v tomto prípade *Sepper Motor Driver BTE13-008*, je súčiastka, špeciálny elektronický obvod, ktorý na základe riadiaceho signálu obdržaného z mikrokontroléru generuje impulzy, ktoré následne cez výkonovú časť budia v presnom poradí vinutia rotoru. Impulzy z riadiaceho obvodu majú variabilnú frekvenciu, poradie aj dĺžku. Na základe týchto signálových špecifikácií je riadený



Obr. 2.3: Krokový motorček ROHS[11]

počet, zmysel otáčok rotora a tiež točivý moment. Ovládač je prepojený s mikrokontrolérom prostredníctvom 6 pinov - 4 ovládacie (IN1-IN4), 1 napájací (VCC), 1 zemniaci (GND). Na výstupe ovládača je celkovo 5 pinov (4 ovládacie, 1 napájací), ktoré napájajú priamo krokový motorček. Pohonnú jednotku predstavuje striedavý synchronný elektromotor, ktorý prevádza digitálne pulzy generované ovládačom na mechanické otáčanie hriadeľa. Rotácia hriadeľa je, ako názov napovedá, kroková. Počet krokov potrebný na jednu otáčku hriadeľa o  $360^\circ$  sa líši v závislosti od konkrétneho modelu. Vykonanie jedného kroku iniciuje samostatný impulz generovaný ovládačom. Pre návrh systému bol zvolený motorček *ROHS Step Motor 28BYJ-48* (obrázok 2.3). Zariadenie je napájané jednosmerným napätím 5 V. Na jednu otáčku motorčeka o  $360^\circ$  pripadá celkovo 64 krokov ( $5,625^\circ$  na každý krok). Motorček zabezpečuje výsledný pohyb nástroja ruky operačného robota prostredníctvom rotácie koliesok na tele nástroja. [11]

Modelárske servá predstavujú lacné a dostupné polohové jednotky natočenia. Štandardné typy poskytujú krútiaci moment okolo 30 Ncm pri napájaní zo zdroja 4,8 V, prípadne až 40 Ncm / 6 V. Servo obsahuje jednosmerný motorček, prevodovku a s výstupným hriadelom sprážený miniatúrny potenciometer slúžiaci ako senzor uhla natočenia pre regulačný obvod polohy. Potenciometer je pripojený buď priamo na výstupný hriadeľ, alebo u serv vyšších kategórií tzv. nepriamym náhonom cez zvláštny prevod, ktorý slúži ako veľmi účinná ochrana pred prenosom vibrácií. Z hľadiska rozsahu pohybu, jeho presnosti a celkových charakteristík týkajúcich sa aj programovania jeho pohybu sa servo pohon javí ako výrazne lepšia varianta pre

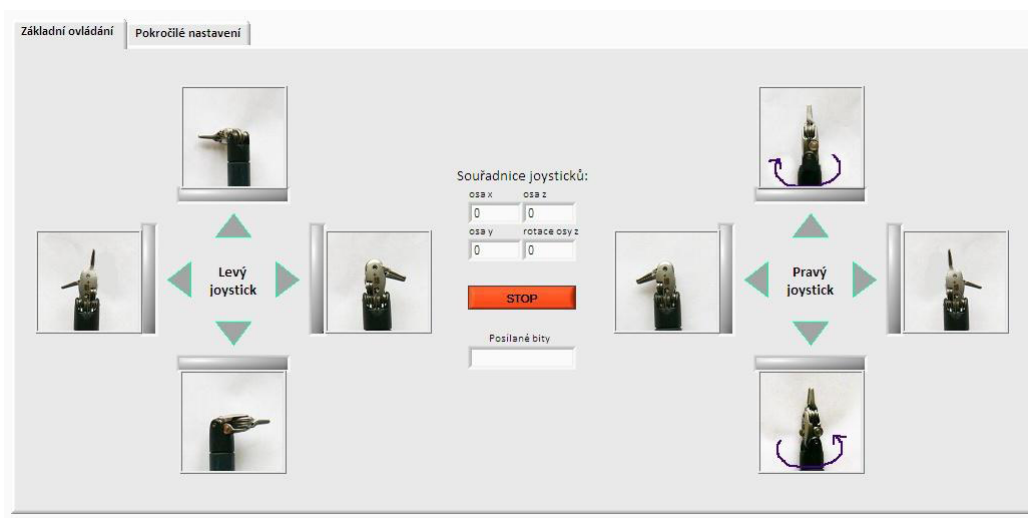


aplikáciu v riadiacom systéme. Preto je vo finálnej realizácii počítané práve s týmto typom pohonnej jednotky, ktorej bližšia špecifikácia je uvedená v podsekcii 3.2.5.

## 2.3 Východiskový softvér

Východiskom pre návrh systému riadenia robotической ruky operačního robota Da Vinci sa stal program zostrojený vo vývojovom prostredí LabVIEW, verzia 8.2 . Program je dimenzovaný pre použitie ovládacieho prvku gamepad.

Užívateľské prostredie je jednoduché a prehľadné (viď obrázok 2.4). Hlavnou úlohou je jednoznačná signalizácia pohybu manipulátorov, ktorá je realizovaná pomocou zelených kontroliek. Ich poloha na paneli odpovedá reálnemu smeru pohybu manipulátorov na gamepade. Ku každej kontrolke je priradený obrázok hlavy nástroja zachycujúci vždy jej krajné polohy. Pohyb páčiek je zobrazovaný taktiež v textových poliach. Má podobu aktuálnych súradníc páčky. Pre vizuálnu kontrolu je pod tlačítkom STOP zobrazovaný aj výsledný 16bitový komunikačný protokol, ktorý je posielaný na výstup sériového portu.



Obr. 2.4: Užívateľské rozhranie programu na generovanie bitového slova [12]

Výstupom programu je komunikačný protokol, ktorý slúži na komunikáciu medzi počítačom a externým zariadením. V tomto prípade je použitý na ovládanie krokového motorčeka. Možnosti pre jeden motorček sú uvedené v tabuľke 2.1 a v protokole obsadia celkom 2 bity.

Môžu nastať celkovo 4 číselné kombinácie, ktorý ovplyvňujú pohyby krokového motorčeka:

- 00 zastavenie motorčeka, je s ním ale možné voľne otáčať

Tab. 2.1: Možné bitové dvojice pre riadenie jedného motorčeka

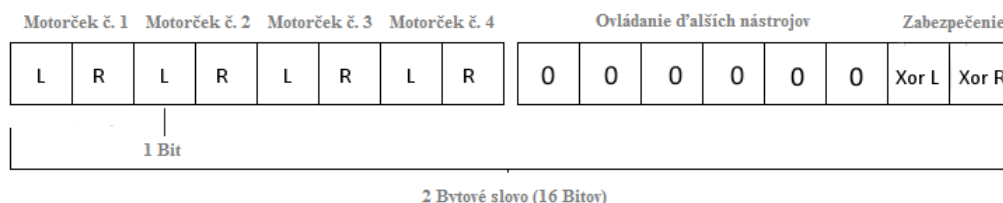
L	0	0	1	1
R	0	1	0	1

- 01 motorček sa otáča doprava
- 10 motorček sa otáča doľava
- 11 zastavenie motorčeka, nie je s ním možné otáčať

Výsledný protokol je zobrazený na obrázku 2.5. Pozostáva z dvoch 8bitových slov, celková informácia teda predstavuje 2 bajty. [12]

Prvých 8 bitov patrí porovnaniu smeru. Tieto bity sú určené pre zoradené 4 dvojice logických hodnôt, ktoré reprezentujú aktuálne pozície pákových ovládačov gamepadu.

Druhé slovo slúži primárne k zabezpečeniu komunikačného protokolu. Prvých 6 bitov je voľných, rezerva je ponechaná pre budúce potreby aplikácie, napríklad pre ovládanie ďalších nástrojov. Posledné dva bity sú určené na samotné zabezpečenie.



Obr. 2.5: Komunikačný protokol

Pre zabezpečenie komunikačného protokolu bola použitá funkcia XOR – exkluzívna disjunkcia. Keďže má užívateľ nad pohybmi nástroja vizuálnu kontrolu, ďalšie zabezpečenie nie je nutné.

V programe je XOR realizovaný kaskádou XORov, ktoré vzájomne porovnávajú vždy párne a nepárne, resp. pravé (R) a ľavé (L) bity. Dochádza aj k zabezpečeniu 6 bitov naplnených nulami, preto je XOR L a XOR R radení až na konci komunikačného protokolu. Z tabuľky 2.2 je zrejmé, že logická 1 je na výstupe len v prípade nezahody (nonekvivalencie) medzi vstupmi. Všetky ostatné prípady sú hodnotené ako logická 0.

Na obrázku 2.6 vidieť príklad možného komunikačného protokolu. Postupnosť výpočtu XORov je znázornená šípkami. Červená logická hodnota predstavuje pravdivostnú hodnotu kontrolného súčtu dvoch predošlých hodnôt spojených šípkami, a

zároveň je jedným zo vstupov pre výpočet ďalšieho čiastkového kontrolného súčtu. Výslednú hodnotu kontrolného súčtu predstavuje dvojica binárnych hodnôt umiestnená na konci komunikačného protokolu.

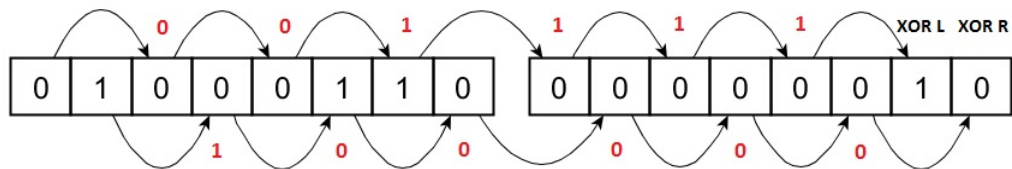
Konkrétna binárna konfigurácia odpovedá na základe obrázku 2.6 následovným pohybom:

- motorček č.1 sa otáča doprava
- motorček č.2 stojí
- motorček č.3 sa otáča doprava
- motorček č.4 sa otáča dolava

Pokiaľ by bol tento konkrétny komunikačný protokol zasielaný do mikrokontroléru, výsledný pohyb chirurgického robota by na základe tabuľky 2.3 predstavoval rotáciu krku nástroja okolo osi  $z$  za súčasného roztvárania oboch nástrojových krídel.

Tab. 2.2: Pravdivostná tabuľka funkcie XOR

x	y	x XOR y
0	0	0
0	1	1
1	0	1
1	1	0



Obr. 2.6: Príklad podoby komunikačného protokolu vrátane výpočtu kontrolného súčtu

## 2.4 Hardvérové riešenie

Predmetom hardvérového riešenia je prevod signálu generovaného ovládacím zariadením (gamepadom) na výsledný mechanický pohyb nástroja robotической ruky.

Pohyb je sprostredkovaný motorčkami, ktoré zabezpečujú rotáciu ovládacích prvkov nástroja - koliesok umiestnených na spodnej strane tela nástroja (viď obrázok

2.7). Jednotlivé kolieska sú vzájomne odlišné vo viacerých aspektoch. Funkčné aj dispozičné odlišnosti sumarizuje tabuľka 2.3.

Tab. 2.3: Špecifikácia vlastností ovládacích prvkov nástroja robotického ruky

koliesko	rotačná pohyblivosť	generovaný pohyb	väzba
1	$\pm 170^\circ$	rotácia okolo osi $z$	žiadna
2	$\pm 85^\circ$	náklon nástrojovej hlavy	3, 4
3	$\pm 95^\circ$	pohyb ľavého krídla nástr. hlavy	4
4	$\pm 95^\circ$	pohyb pravého krídla nástr. hlavy	3



Obr. 2.7: Spodná strana tela nástroja ruky operačného robota Da Vinci [12]

Vzájomnou súčasnou rotáciou jednotlivých koliesok dostávame komplexný výsledný pohyb nástroja ruky, ktorý svojou variabilitou mnohonásobne prevyšuje fyziologický pohyb ľudskej končatiny.

Koliesko č.1 svojou rotáciou vykonáva rotáciu krku nástroja. Tento pohyb je nezávislý na polohe, resp. pohybe zvyšných ovládacích koliesok nástroja.

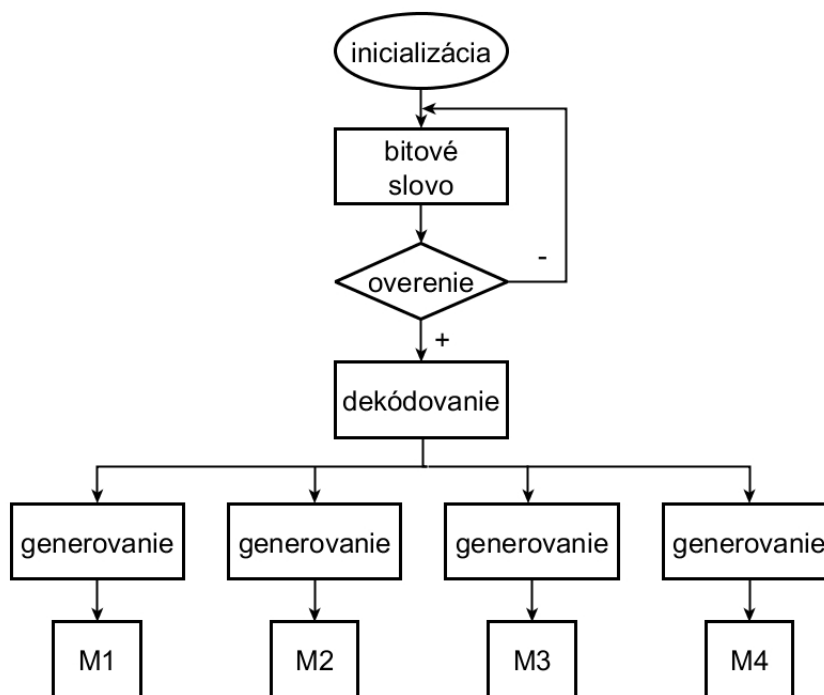
Koliesko č.2 zabezpečuje náklon nástrojovej hlavy na obe strany. Tento pohyb je pomerne zložitý vzhľadom na konštrukciu mechanického prevodného systému. Ten je tvorený sústavou oceľových laniek, ktoré zabezpečujú prenos mechanického pohybu z motorčeka na konkrétnu časť hlavy nástroja. Systém je veľmi presný (simulácia

šliach ľudskej ruky), avšak spôsobuje vznik väzieb medzi jednotlivými kolieskami pri jednom type pohybu, čo celkovo komplikuje návrh riadenia nástroja. V tomto prípade je pohyb kolieska č.2 je viazaný na pohyb koliesok č.3 a 4. Pri náklone hlavy je nutný minimálny kompenzačný pohyb oboch koliesok ovládajúcich ramená hlavy. V opačnom prípade prípadne vykonávajú ramená nástroja počas náklonu nežiadúci súhlasný pohyb na jednu alebo druhú stranu.

Kolieska č.3 a 4 svojou rotáciou ovládajú rozovieranie a zvieranie krídel nástrojovej hlavy. Ovládanie každého z krídel je samostatné, čo si vyžaduje najmä vizuálnu kontrolu človeka, ktorý nástroj pomocou manipulátorov ovláda. Najčastejšie pohyby, ktoré krídla hlavy vykonávajú (strihanie, uchopovanie), sú viazané na prácu oboch krídel, ktoré je nutné v rámci možností synchronizovať. V opačnom prípade bude jedno krídlo brániť v pohybe druhému, prípadne budú pôsobiť tlakom proti sebe.

## 2.5 Softvérové riešenie

Návrh softvérového riešenia zobrazuje vývojový diagram na obrázku 2.8.



Obr. 2.8: Vývojový diagram programu

Pre riešenie problematiky riadenia nástroja robotického ruky som zvolil vývojovú platformu Arduino Uno od talianskeho výrobcu. Značka Arduino ponúka pre programovanie svojich mikrokontrolérov vlastný softvér Arduino, v mojom prípade verzia

1.0.5. Tento softvér je takzvané IDE (Integrated Development Environment). Ide o vývojové prostredie umožňujúce písanie zdrojového kódu v jazyku Wiring, ktorý je podobný jazyku C. Programovanie mikrokontroléra je nosnou časťou nadväzujúcej bakalárskej práce.

Po spustení sa ako prvá prevedie inicializácia nástroja ruky operačného robota. Tento úkon je kľúčový z hľadiska zachovania miniinvazivity operačného zákroku. Cieľom je dosiahnutie východiskovej polohy nástrojovej hlavy - hlava je vzpriamená (bez náklonu) a obe krídla sú orientované rovnobežne s krkom nástroja. Pri takejto polohe sa otvor potrebný pre vnik nástroja do tela pacienta minimalizuje na veľkosť plochy prierezu krku nástroja. Rotáciu nástroja v ose  $z$  v tomto prípade neuvažujeme. Inicializácia sa týka motorčekov ovládajúcich zvislé pohyby hlavy nástroja, konkrétne koliesko č.2, 3 a 4 (viď tabuľku 2.3). Tabuľka obsahuje tiež údaje týkajúce sa rozsahu pohybu jednotlivých koliesok. Princípom inicializácie je rotácia kolieska o plný rozsah na jednu stranu bez ohľadu na východiskovú pozíciu a následná spätná rotácia o polovičný rozsah. Na základe technických údajov krokového motorčeka uvedených v podkapitole 2.1 dojde k prepočtu uhlového rozsahu rotácie konkrétneho kolieska na presný počet krokov, ktoré musí motorček vykonať. Dôležité je ošetrovanie preklzu v hraničných polohách, aby nedošlo k preťaženiu elektronických súčiastok systému.

Po ukončení inicializácie je možné začať prenos dát z gamepadu do počítača. Komunikácia prebieha po sériovej linke rýchlosťou 9600 Bd (1 Baud datového komunikačného systému je počet symbolov prenesených za 1 sekundu). Program pre tvorbu bitového slova na základe dát z gamepadu vytvorí komunikačný protokol, ktorý je zobrazený na obrázku 2.5 a odošle ho po sériovej linke do mikrokontroléra.[12]

Aby sa predišlo nežiadúcim pohybom nástroja ruky je nutné skontrolovať či nedošlo ku prenosovej chybe medzi počítačom a mikrokontrolérom. Verifikácia prebieha pomocou výpočtu kontrolného súčtu (XOR). Mikrokontrolér opäť spočíta hodnoty XOR L a XOR R z obdržaných binárnych dát (viď obrázok 2.6) a skontroluje ich z hodnotami XOR, ktoré boli protokolu pridelené počítačom. V prípade nezahody sa dané bitové slovo neberie do úvahy a čaká sa na nové.

Pokiaľ nastane zhoda kontrolných súčtov sú dáta ďalej spracúvané. Ďalšia kontrola XOR nie je nutná, preto sa pôvodný komunikačný protokol obmedzí na prvé 8 bitové slovo, ktoré nesie informáciu pre výsledný pohyb 4 serv (jedna dvojica binárnych bitov pre každú pohonnú jednotku). Mikrokontrolér na základe pozície dvojíc binárnych bitov dekoduje informáciu v zmysle príslušnosti k jednému zo štyroch serv (viď prvé 8bitové slovo na obrázku 2.5) a túto informáciu na ne prostredníctvom signálnych káblov odošle.

## 3 REALIZÁCIA

Finálne realizovaný systém riadenia robotickej ruky operačného robota Da Vinci sa od pôvodného návrhu (viď kapitolu 2) odlišuje pomerne zásadným spôsobom. Reštrukturalizácia bola prevedená v softvérovej aj hardvérovej sfére. Nosná informácia pre pohyb nástroja je získavaná z joystickov umiestnených priamo na rozširujúcej doske platformy Arduino (viď podkapitolu 3.2.4). Snímanie polohy joystickov gamepadu a jej následné vyhodnotenie programom pre tvorbu bitového slova v prostredí LabVIEW v novom návrhu nefiguruje z dôvodu výrazných skreslení dátového toku objasnených v nasledujúcich statiach kapitoly 3.1.

### 3.1 Sériová komunikácia LabVIEW - Arduino

Táto sekcia je dôležitá z hľadiska ako pôvodného, tak aj výsledného návrhu riadiaceho systému. Závery a kľúčové zistenia sú preto zhrnuté na konci dokumentu v diskusii výsledkov (viď kapitolu 4.2).

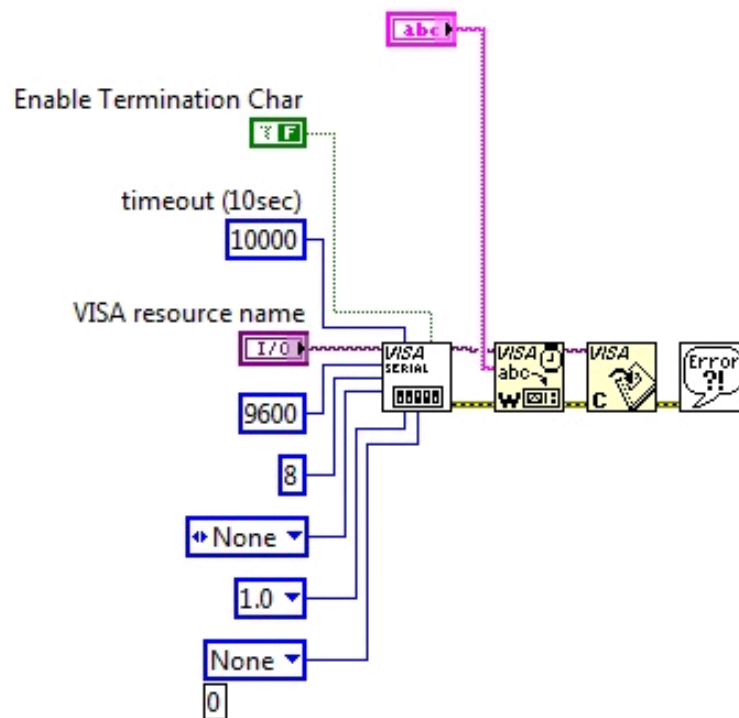
#### 3.1.1 Komunikácia prostredníctvom NI-VISA

Softvérová architektúra virtuálneho nástroja (VISA) je často používaný vstup alebo výstup (I/O API) pre komunikáciu medzi nástrojmi a počítačom. Dokáže komunikovať cez štandardné počítačové protokoly, ako je napríklad sériový port alebo USB.[13]

V navrhovanom systéme riadenia robotickej ruky bola VISA použitá pre nastavenie parametrov komunikačného protokolu posiadaného na sériový port. Schéma blokov realizujúca dátový prenos v LabVIEW zobrazuje obrázok 3.1.

Na základe týchto parametrov mal byť mikrokontrolér schopný čítať dáta poslané na sériový port, spracúvať ich a na základe vyhodnotenia iniciovať pohyb serv, a tým ruky nástroja. Kľúčové parametre sériovej komunikácie sú:

- **Prenosová rýchlosť** - defaultne nastavená na 9 600 Bd, pričom 1 Baud dátového komunikačného systému definuje počet symbolov prenesených za 1 sekundu
- **Parita** - bit parity umožňuje kontrolu vzniknutých chýb, defaultne je však vybraná hodnota none - žiadna
- **Počet bitov** - udáva množstvo prenesených hodnôt, akonáhle je odoslaný štartovací bit - v tomto prípade 8
- **Počet stop bitov** - umožňuje nastaviť dĺžku stop bitu v rámci násobkov 1, 1,5 alebo 2 časových intervalov klasického bitového prenosu



Obr. 3.1: Schéma zapojenia blokov VISA za účelom dátového prenosu na sériový port

Prenosová rýchlosť pri sériovej komunikácii sa u platformy Arduino pomocou príkazu *Serial.begin()* štandardne nastavuje taktiež na 9 600 Bd.

Jednoduchý test na začiatku implementácie navrhovaného systému riadenia spočíval v overení synchronity dátového toku z LabVIEW do Arduino, t.j. či Arduino dokáže v reálnom čase prijímať rovnaké množstvo dát ako LabVIEW odošle na sériový port. Splnenie tejto podmienky je pre prevádzanie presných pohybov robota v medicínskej sfére kľúčové. Experiment bol prevedený na nepájivom poli. Ak Arduino pomocou funkcie *Serial.available()* zistilo prítomnosť dát na sériovom porte, prečítalo dáta pomocou funkcie *Serial.read()* a rozblikalo indikačnú diódu. Ak by spracúvanie dát prebiehalo v poriadku, pri spustení programu pre tvorbu bitového slova by sa dióda rozblikala, pri vypnutí by naopak okamžite zhasla. V tomto prípade však dióda blikala aj po ukončení programu v LabVIEW, čo znamená, že platforma spracúva určité dáta napriek tomu, že žiadne nie sú odosielané na sériový port. Tento fakt poukazuje na istú komunikačnú nekompatibilitu pri interakcii odlišných softvérových architektúr LabVIEW a Arduino IDE. Je možné, že vplyvom náporu dát dochádza v arduine k pretekaniu tzv. bufferu. Jedná sa o dočasné úložisko dát, najčastejšie v pamäti RAM, ktoré umožňuje procesoru manipulovať s



dátami predtým, ako sú odosielané na iné zariadenie. Problém tohoto charakteru môže výrazným spôsobom skresľovať posielané dáta.

Keďže Arduino a prostredie LabVIEW zdieľajú jeden sériový port, nie je možné, aby tak robili súčasne. Pri modifikácii zdrojového kódu v IDE je nutné zavrieť okno programu LabVIEW a opačne. Preto nie je možné posielať dáta na mikrokontrolér, a zároveň ich zobrazovať v sériovom monitore za účelom overenia validity dátového prenosu.

Riešenie v tomto prípade predstavuje softvérový nástroj Proteus, ktorý je schopný simulovať dátový prenos medzi prostredím LabVIEW a Arduino. Postup simulácie dátového toku ako aj jeho výsledky sú uvedené v nasledujúcej podkapitole 3.1.2.

### 3.1.2 Simulácia dátového toku

Proteus je aplikácia určená pre modelovanie virtuálnych systémov a simuláciu elektronických obvodov vytvorená spoločnosťou Labcenter Electronics. Program je tiež schopný simulovať interakciu medzi softvérom bežiacom na mikroprocesore a akoukoľvek analógovou alebo digitálnou elektronikou, ktorá je ku nemu pripojená. Proteus simuluje vstupné a výstupné porty, prerušenia, časovače a ďalšie periférne zariadenia pripojené k procesoru, ktoré sú programom podporované.

Aby bolo možné previesť simuláciu, je nutná inštalácia niekoľkých programov voľne dostupných na internete:

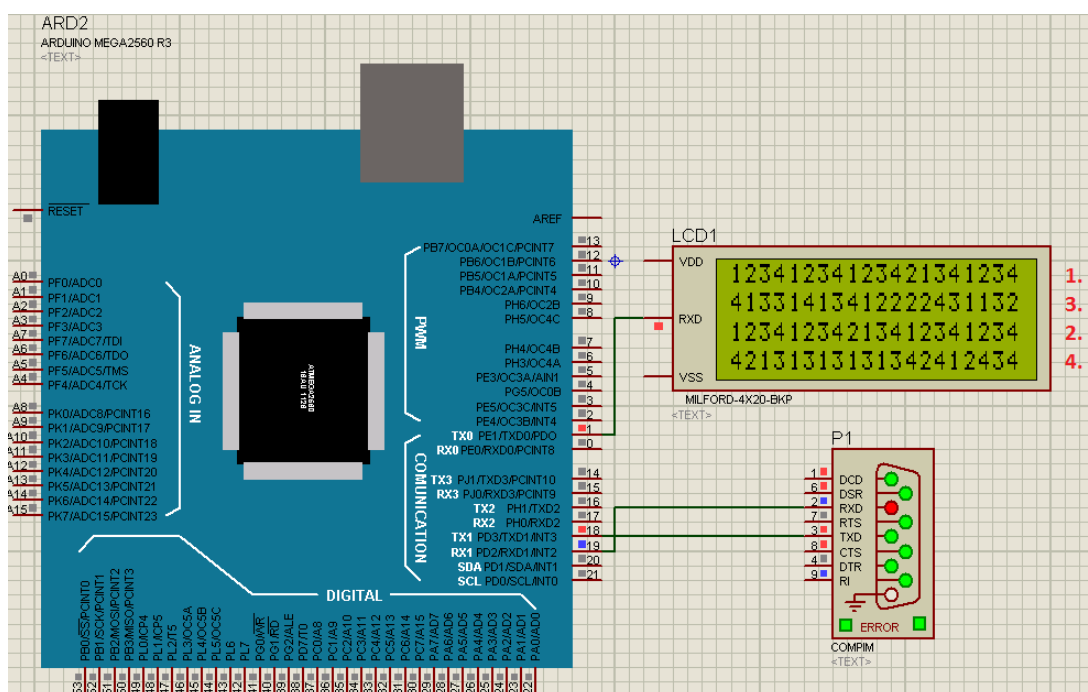
- **Proteus 7 Professional** - simulačný nástroj
- **Virtual Serial Port Driver 7.2** - program pre tvorbu virtuálnych portov dostupný na internete ako voľná verzia na obmedzenú časovú dobu
- **Proteus library for Arduino** - voľne dostupná knižnica umožňujúca simuláciu mikrokontrolérov Arduino

Pre simuláciu bude použitá platforma Arduino Mega2560, ktorá obsahuje viac ako jeden hardvérový sériový port, tzv. UART. Postup simulácie možno zhrnúť do niekoľkých bodov:

1. Vloženie blokov Arduino - Mega, Milford Serial LCD a COMPIM z knižnice do simulačného poľa a ich prepojenie podľa obrázka 3.2.
2. Po potvrdení položky COMPIM nastavenie jej fyzického portu na COM2.
3. Napísanie zdrojového kódu podľa obrázka 3.5 do prostredia IDE a jeho následná kompilácia. Keďže platforma nie je fyzicky pripojená k PC, chybovú hlášku treba ignorovať.
4. Spárovanie virtuálnych portov COM2 (pripojený k portu COMPIM v programe Proteus a COM3 (pripojený k LabVIEW) pomocou Virtual Serial Port Driver 7.2)
5. Realizácia identického blokového diagramu v prostredí LabVIEW (viď 3.1).

6. Nastavenie Visa resource name na COM3 a voľba reťazca znakov, ktorý bude posielaný cez sériový port na Arduino.
7. Spustenie programu Proteus a LabVIEW, posielaný reťazec sa zobrazuje na položke Milford Serial LCD.

Program Proteus používa na simuláciu zdrojového kódu nahraného do mikrokontroléra súbor s koncovkou *.hex*. Odkedy Arduino IDE začalo používať formát súborov *.pde*, respektíve *.ino*, nie je možné Arduino zdrojový kód Proteom simulovať priamo. Vloženie súboru do programu Proteus je možné po potvrdení položky Arduino v simulačnom poli. Hľadaný súbor sa s koncovkou *.cpp.hex* nachádza v počítači na adrese: Jednotka C » Užívateľ » Názov počítača » AppData » Local » Temp. Jedná sa o lokalizáciu v systémových priečiňkoch, preto je najskôr nutné v PC zobrazit jeho skryté súbory. [14]

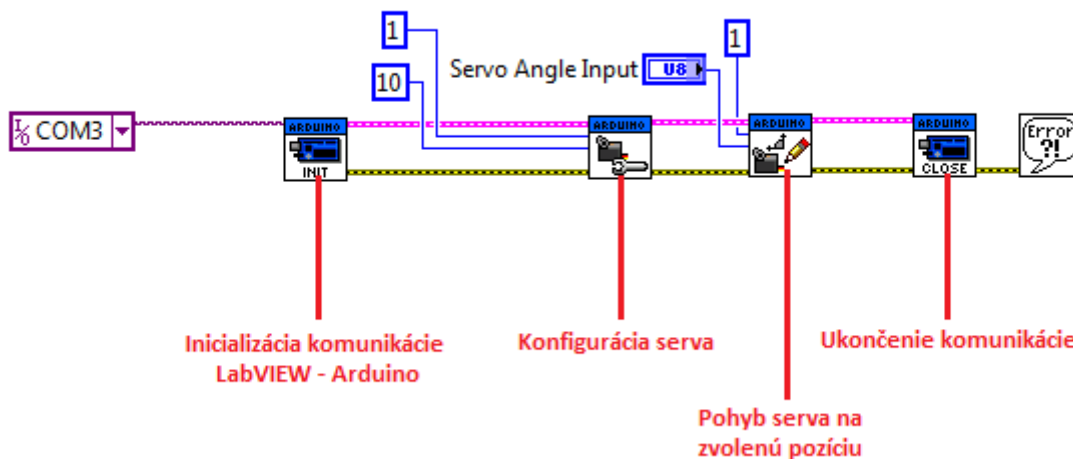


Obr. 3.2: Simulácia dátového toku medzi prostredím LabVIEW a mikrokontrolérom Arduino

Návrh komunikačného protokolu pozostával z 2 osembitových slov, celkovo 2 bajtov. Nosná informácia pre pohyb bola ukrytá v 4 dvojiciach binárnych bitov. Pre lepšiu vizualizáciu prípadného skresľovania dát som použil 16 znakový testovací protokol v podobe reťazca 1234123412341234. Dáta prijaté mikrokontrolérom sú vizualizované na LCD prvku obrázka 3.2. Spočiatku sú badateľné len drobné zámery, s postupom času je však skreslenie dát zjavné.

### 3.1.3 LabVIEW rozhranie pre Arduino

LabVIEW rozhranie pre Arduino predstavuje možnosť ako sa vyhnúť vzájomnej interakcii dvoch odlišných vývojových prostredí spôsobujúcej skreslenie dát. S týmto nástrojom a LabVIEW je možné ovládať, prípadne prijímať dáta z mikrokontroléru Arduino. Zdrojový kód pre mikrokontrolér sa správa ako I/O prostriedok, ktorý umožňuje prepojenie LabVIEW prostredníctvom sériovej komunikácie. Týmto spôsobom sa zabezpečí rýchly prenos z pinov Arduina do prostredia LabVIEW bez nastavovania parametrov komunikácie, synchronizácie alebo programovania zariadenia v prostredí IDE. Použitím blokov Open, Read / Write, Close (viď obrázok 3.3) je možné získať prístup k digitálnym, analógovým, PWM pinom, prípadne SPI (*Serial Peripheral Interface*) signálom mikrokontroléra a tie následne modifikovať, prípadne spracúvať. [15]



Obr. 3.3: Schéma blokov pre ovládanie serva pomocou Arduina naprogramovaného v prostredí LabVIEW

## 3.2 Hardvér

Nosným prvkom hardvérového riešenia je open-source platforma Arduino založená na 8bitovom mikrokontroléri ATmega a grafickom vývojovom prostredí, ktoré vychádza z prostredia Wiring (podobné jazyku C) a Processing. Arduino je schopné snímať okolité prostredie pomocou vstupov z rozličných senzorov a prepínačov. Veľkou výhodou projektov založených na Arduine je jednoduchá komunikácia so softvérom ako stolných, tak aj prenosných počítačov.

### 3.2.1 Arduino hardvér

Dosky Arduino obsahujú 8bitové mikrokontroléry z rodiny AVR od firmy Atmel s komplementárnymi komponentami pre uľahčenie programovania a začlenenie do ďalších obvodov. Umiestnenie konektorov na doskách je štandardizované, čím umožňujú jednoduché pripojenie prídavných modulov, tzv. shieldov. Jedná sa napríklad o Ethernet Shield, Wifi Shield a iné. Tieto moduly komunikujú s Arduino priamo prostredníctvom I/O pinov. Shieldy môžu s platformou potenciálne komunikovať aj ako individuálne adresovateľné cez  $I^2C$  sériovú zbernicu. Tento spôsob umožňuje paralelné prepojenie viacerých shieldov. Dosky obsahujú štandardne niekoľko diód, konektory pre ICSP programovanie, resetovacie tlačidlo, konektor externého napájania, 5 V lineárny regulátor napätia, 16 MHz kryštálový oscilátor a obvod zabezpečujúci komunikáciu cez USB. Napriek tomu, že je Arduino pripojené k PC pomocou rozhrania USB, je softvérovo simulovaná sériová komunikácia cez linku RS-232. V modeli Uno túto možnosť obstaráva ATmega8U2 s predprogramovaným firmware ako USB-to-serial konvertor. Hlavný mikrokontrolér, ktorý je programovateľný, defaultne obsahuje bootloader (kód, ktorý sa po spustení postará o základné nastavenie mikrokontroléru, ako sú interné časovače, nastavenie rozhrania UART a ďalšie) a nastavené potrebné fuses bajty, ktoré nízkoúrovňovo nastavujú niektoré vlastnosti čipu. [16]

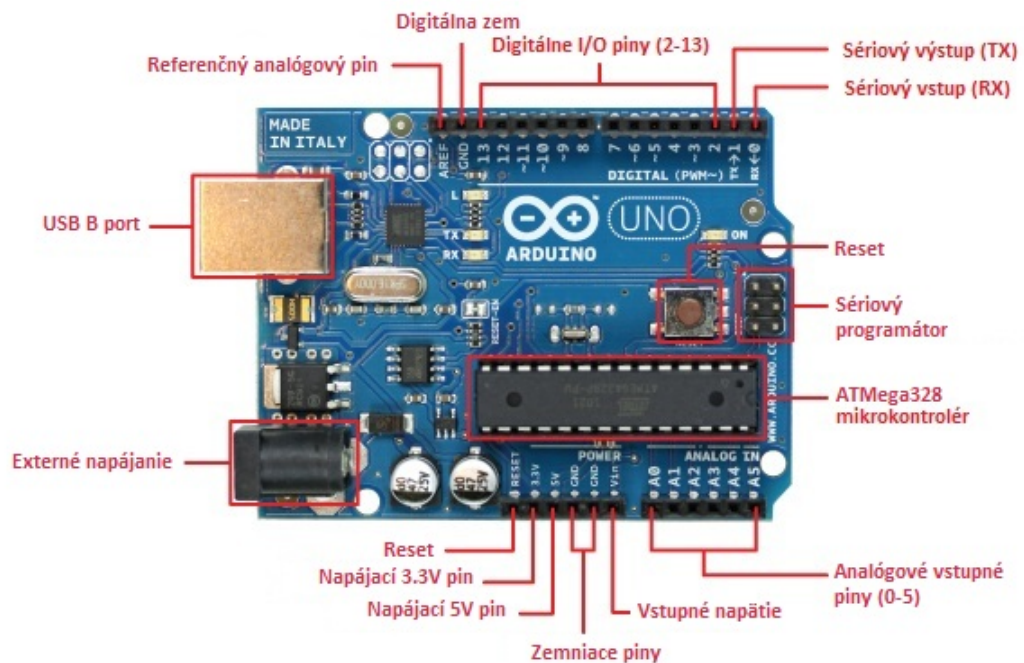
Platformy Arduino sa vyrábajú v niekoľkých prevedeniach, ktorých parametre sa líšia podľa aplikácia, pre ktorú sú primárne určené. Pre ovládanie modelárskych serv plne postačuje základný model Arduino Uno v spojení so shieldom, ktorý je vyvinutý priamo pre ovládanie nástroja robotického ruky operačného robota Da Vinci.

### 3.2.2 Arduino Uno

Arduino Uno je platforma založená na mikrokontroléry ATmega328. Obsahuje 14 digitálnych vstupne-výstupných pinov. Keďže výstupné periodické signály zo 6 digitálnych pinov je možné modulovať zmenou striedy (*Pulse Width Modulation*), môžu byť tieto piny použité ako zdroj riadiaceho signálu pre ovládanie servo pohonných jednotiek. Platforma ďalej obsahuje 6 analógových pinov, keramický rezonátor, USB konektor a ďalšie podporné a regulačné prvky (viď obrázok 3.4).

Prevádzkové napätie platformy je 5 V, pričom maximálne prúdové zaťaženie každého z I/O pinov predstavuje 40 mA. Arduino Uno môže byť napájané buď priamo cez USB konektor, alebo prostredníctvom externého zdroja.

Externé napájanie môže byť realizované AC-to-DC adaptérom alebo batériou. Pri voľbe batérového napájania je nutné použiť piny Gnd a Vin. Adaptér je k doske pripojený pomocou 2,1 mm napájacieho konektoru. Doporučené rozmedzie externého napájacieho zdroja je 7 – 12 V.



Obr. 3.4: Vývojová platforma Arduino Uno

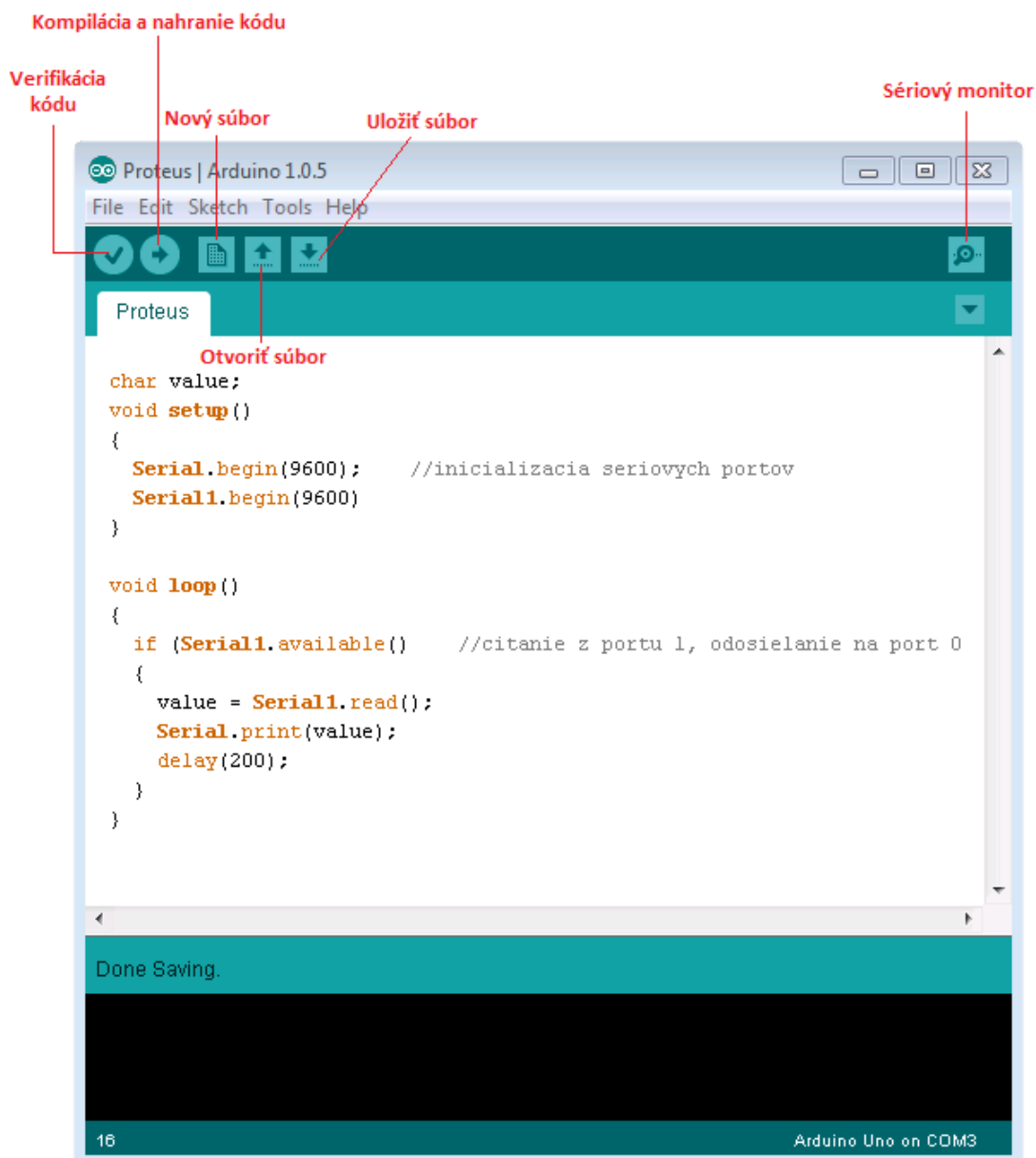
Arduino Uno obsahuje niekoľko prvkov zabezpečujúcich komunikáciu s počítačom, prípadne inými zariadeniami. ATmega328 ponúka hardvérové UART pre TTL (5 V) sériovú komunikáciu dostupnú na pinoch 0 (RX) a 1 (TX). Obojsmerný dátový tok je možné monitorovať pomocou monitoru sériového portu, ktorý je súčasťou softvéru. Príjem, resp. odosielanie dát sprevádza blikanie diód RX a TX. [17]

### 3.2.3 Arduino softvér

Grafické vývojové prostredie (IDE) platformy Arduino je aplikácia vytvorená pomocou programovacieho jazyka Java. Je odvodené z IDE pre Processing programovanie a prostredie Wiring.

Jadro programu tvoria dve hlavné funkcie, ktoré musí užívateľ definovať, a to konkrétne:

- **setup()** - inicializačná funkcia, ktoré prebehne vždy raz, a to na začiatku programu po jeho spustení
- **loop()** - funkcia, ktorá má charakter nekonečného cyklu, ktorý je možné ukončiť len odpojením napájania dosky Arduino



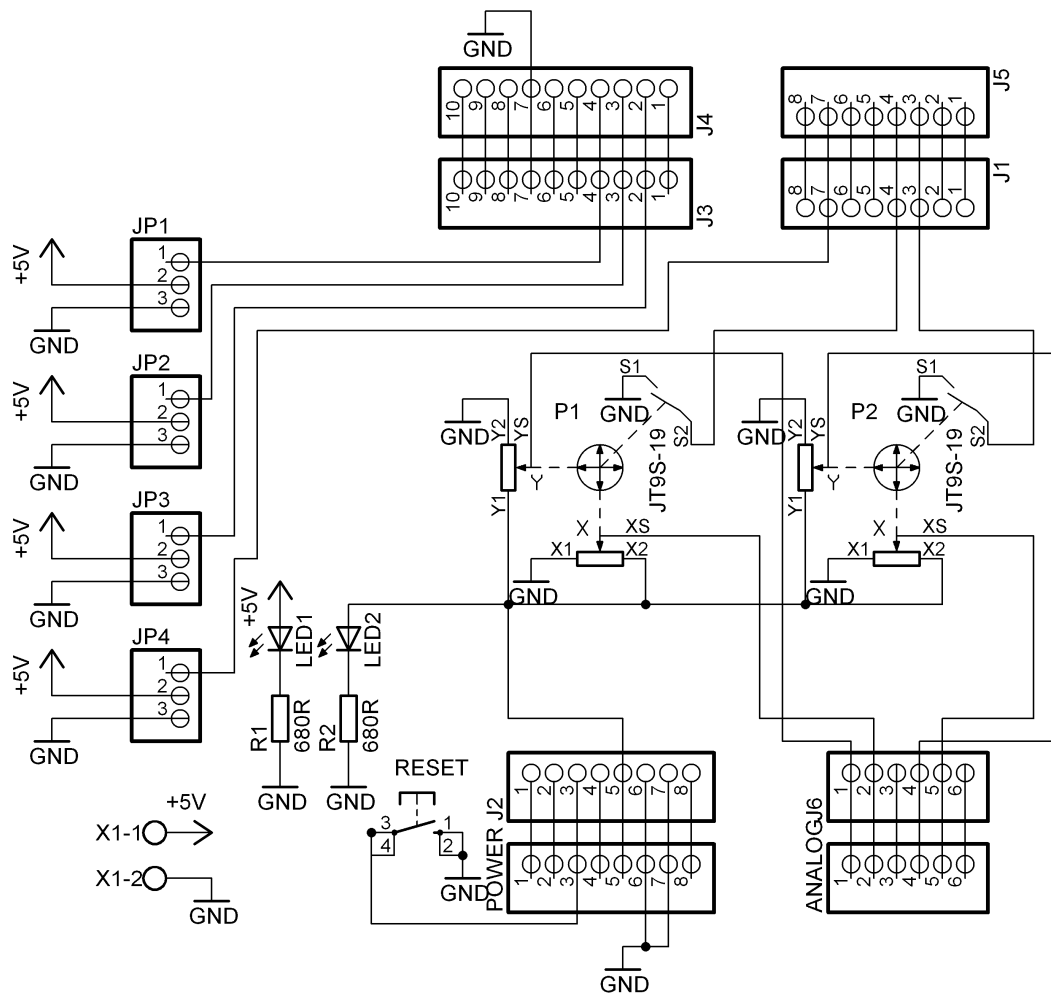
Obr. 3.5: Grafické vývojové prostredie platformy Arduino

### 3.2.4 Motor Shield

Za účelom ovládania modelárskych serv pomocou Arduina je nutné platformu rozšíriť o nastavbový prvok - shield. Shield pre ovládanie širokej škály pohonných jednotiek je komerčne dostupný. Motor Shield v1.2 disponuje pinmi pre pripojenie 2 modelárskych serv, a ďalej je schopný ovládať až 4 jednosmerné, resp. 2 krokové motory. Keďže jednosmerné ani krokové motory nedisponujú vlastnými riadiacimi jednotkami, je majoritná časť prvkov tohoto rozširujúceho zariadenia (časovače, mik-

roprocesor) určená práve na tento účel. Z konštrukčného a funkčného hľadiska je teda shield v1.2 obmedzený na ovládanie 2 serv napriek tomu, že kapacitne je Arduino Uno limitované len počtom PWM výstupov. Je teda schopné ovládať až 6 pohonných jednotiek tohoto typu. Za týmto účelom bola vytvorená vlastná rozširujúca platforma, ktorá odpovedá špecifickým požiadavkám tejto práce.

Motor Shield je realizovaný formou PCB (Printed Circuit Board) vyrobenej technikou fotopozitívneho vyvýjania a následného leptania. Za účelom schématického návrhu prvku bolo použitý vývojový softvér Eagle firmy CadSoft. Schématický návrh je znázornený na obrázku 3.6.



Obr. 3.6: Schématický návrh prvku Motor Shield

Shield obsahuje celkovo 12 pinov pre pripojenie pohonných jednotiek umiestnených na ľavom okraji dosky pod uhlom 90°. Na doske sa ďalej nachádzajú 2 indikačné diódy, resetovacie tlačidlo, dvojitá pružinová svorkovnica pre pripojenie externého

zdroja napätia a 2 analógové manipulátory. Tieto joysticky sú určené primárne pre ovládače hracích konzol, majú odpor  $10k\Omega$  a elektronickú zaťažiteľnosť 60 W.

Kvôli konštrukčnej pevnosti je shield k platforme pripojený zo spodnej strany pomocou všetkých dostupných pinov. Aktívnym prepojením však disponujú len niektoré, a to konk.:

- **digitálne piny 6, 9, 10, 11** slúžia pre ovládanie serv
- 2 piny **Gnd** zabezpečujúce uzemnenie
- **analógové piny 0, 1, 3, 4** slúžia na snímanie aktuálnej polohy joystickov
- **reset pin**
- **5 V pin** napája potenciometre joystickov

Použitie joysticky je možné využiť aj ako tlačidlo. V tomto prípade sa jedná o implementáciu rozširujúcej funkcie, preto v návrhu dosky plošného spoja ani jej schéme nasledujúce riešenie nefiguruje.

Táto funkcia vyžaduje 4 voľné digitálne piny, konk. 2, 3, 4, 5. Prostredné piny 3 a 4 (naprogramované na vstup) sú spojené každý s jedným tlačidlom umiestneným na joysticku. Technické riešenie v podobe vodivej cesty na doske plošného spoja nebolo v dôsledku nedostatku miesta možné. Alternatívny spôsob spočíva v prepojení pomocou vodivého drátu. Vonkajšie piny 2 a 5 (naprogramované na výstup) napájajú vnútorné piny napätím 5 V cez odpory  $10k\Omega$ . Pri stlačení tlačidla dojde k uzemneniu príslušného vnútorného pinu. Rozširujúca funkcia je spúšťaná na základe detekovaného nulového stavu.

Shield obsahuje 2 izolované nezávisle napájané okruhy. Joysticky pre svoju činnosť využívajú napätie 5 V zo sériovej zbernice USB, prostredníctvom ktorej je Arduino pripojené k PC. Prítomnosť napätia v tomto okruhu signalizuje červená dióda. Pohonné jednotky sú napájané pomocou dvojitej pružinovej svorkovnice z externého zdroja napätia, ktorý je dimenzovaný na 5 V a prúdový odber 2 A, pri maximálnom zaťažení všetkých 4 serv. Prítomnosť prúdu v tomto okruhu signalizuje modrá dióda.

### 3.2.5 Pohon

Pohonná jednotka Hitec HS-5485 HB (viď 3.7) je precízne digitálne servo štandardných rozmerov disponujúce dostatočným ťahom pre danú aplikáciu. Obsahuje karbonitové prevody a výstupný hriadeľ uložený v guľčikovom ložisku. Servo sa pripája pomocou trojžilového kábla pričom žltý, resp. oranžový kábel je signálny, čierny, resp. hnedý napájací (-) a červený napájací (+). Pracovné napätie je v rozmedzí 4,8 - 6,0 V.

Dôležité technické parametre (udané vždy pre hornú aj dolnú hranicu pracovného napätia) sú obsiahnuté v nasledujúcej tabuľke (viď tab. 3.1).



Tab. 3.1: Technické parametre serva Hitec [19]

<b>Napájacie napätie [V]</b>	<b>4,8 - 6,0</b>
<b>Ťah [kg/cm]</b>	5,2 - 6,4
<b>Rýchlosť [s/60°]</b>	0,20 - 0,17
<b>Rozmery [mm]</b>	39,8 x 19,8 x 38
<b>Hmotnosť [g]</b>	45
<b>Defaultný rozsah pohybu</b>	20° - 165°
<b>Maximálny rozsah pohybu</b>	0° - 180°

Rozsah pohybu použitých serv je vzhľadom na pohybové možnosti nástrojovej hlavy (viď tabuľka 2.3) pomerne obmedzený. Pre kľúčové pohyby nástroja (pohyb krídel, náklon nástrojovej hlavy) však plne dostačuje. Výrazné obmedzenie nastáva pri rotácii nástrojovej hlavy, kde je využitých menej ako 50 % pohybového potenciálu.

Digitálne servá majú oproti analógovým vo všeobecnosti lepšiu špecifikáciu. Majú síce vyššiu spotrebu energie, avšak nespornou výhodou je vyšší ťah, prídržná sila a programovateľnosť. Programátorom digitálnych serv Hitec HFP-10\*, HFP-20, HPP-21 alebo HPP-21 Plus je možné naprogramovať množstvo parametrov, napr.:

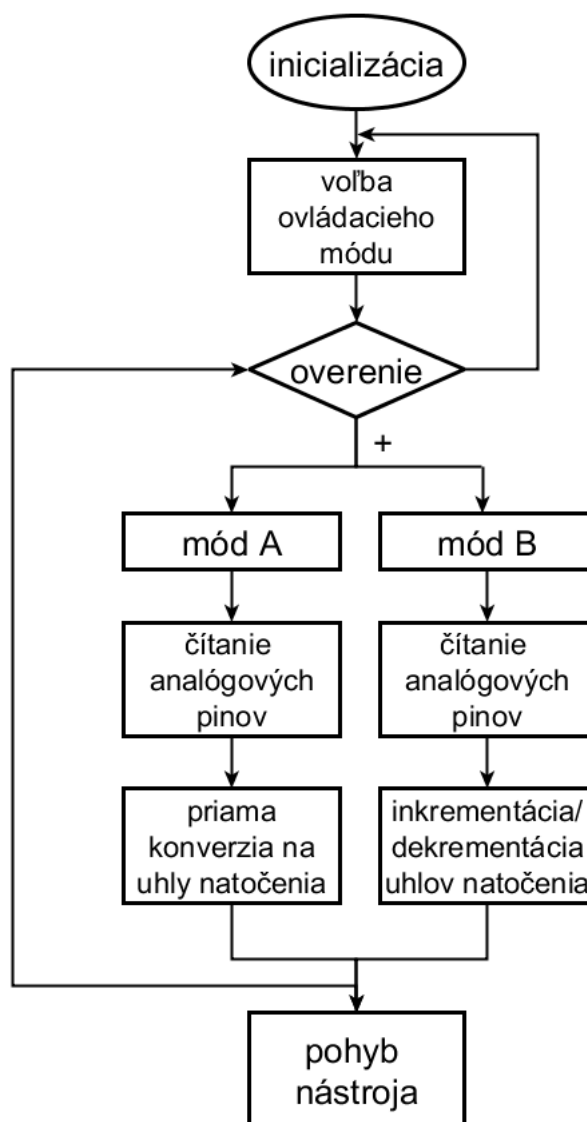
- rýchlosť a zmysel otáčania
- veľkosť výchylky
- ochranu proti preťaženiu (obmedzenie výkonu o 0-50% pri preťažení)
- šírku pásma necitlivosti
- reset do východiskovej továrnej hodnoty
- neutrálna poloha



Obr. 3.7: Digitálne servo Hitec [19]

### 3.3 Softvér

Výsledný softvér je vyvinutý v prostredí Arduino IDE. Jeho činnosť štruktúralne popisuje nasledujúci vývojový diagram (viď obr. 3.8).



Obr. 3.8: Vývojový diagram vytvoreného softvéru

Program v prostredí IDE, tzv. sketch, je po verifikácii možné nahráť do mikrokontroléra, čím dojde k jeho automatickému spusteniu. Nasleduje fáza inicializácie, v ktorej program spracuje preddefinované vstupné premenné a nastaví servá do ich východiskovej polohy. Tá odpovedá stredu medzi hodnotami maximálnych krajných výchyliek serv.

Servá je možné riadiť v dvoch odlišných módoch. Voľba jednotlivého módu je prevádzaná cez sériový monitor. Užívateľ je vyzvaný k zadaniu a potvrdeniu znaku, ktorý iniciuje spustenie príslušného módu. Na zadanie nekorektného znaku je užívateľ upozornený chybovou hláškou. Korektná voľba ovládacieho módu je taktiež potvrdená výpisom hlásenia na sériovom monitore. Zobrazenie informatívnych hlásení je jednorázové, v ďalšom priebehu sa očakáva pochopenie princípu voľby módu užívateľom.

Módy riadenia sa odlišujú charakterom výsledného pohybu. V móde A sú v rámci nekonečného cyklu snímané aktuálne polohy analógových joystickov. Keďže rozsahy hodnôt potenciometrov (0 - 1023) sú definované v odlišnom rozmedzí ako rozsahy serv (v tomto prípade 20 - 165) je nutné transformovať dáta tak, aby si tieto rozsahy vzájomne odpovedali. Konverzia je prevedená pomocou príkazu *map()*, do ktorého vstupujú 4 parametre, konk. minimálne a maximálne hodnoty rozsahov potenciometrov a serv. Vo výsledku pohyb serva kopíruje pohyb potenciometra, resp. joysticku v reálnom čase. Charakter pohybu serv v móde B je koncepčne odlišný. Cieľom snímania aktuálnej polohy analógových joystickov je v tomto prípade zisk informácie o smere pohybu vzhľadom na východiskovú polohu potenciometra. Riešenie v zdrojovom kóde je sprostredkované pomocou rozhodovacieho príkazu vo forme podmienky *if*. Na základe smeru pohybu potenciometra je parameter predstavujúci polohu serva inkrementovaný, resp. dekrementovaný. Implementáciu v zdrojovom kóde predstavuje vnorená podmienka správajúca sa ako cyklus *while*. Charakter pohybu je principiálne krokový, spracovanie dát mikrokontrolérom je však dostatočne rýchle, aby sa pohyb vo výsledku javil ako spojitý. Vychýlenie joysticku z východiskovej polohy iniciuje pohyb príslušného serva doprava, resp. doľava. Pri návrate joysticku do strednej polohy ostáva servo v aktuálne nadobudnutej polohe, t.j. nevracia sa do polohy východiskovej. Pre tento účel slúži dodatočne realizovaná rozširujúca funkcia opísaná v podkapitole 3.2.4, ktorá je prevedená stlačením tlačidla joysticku. Každé tlačidlo zabezpečuje reset konkrétnych dvoch serv do ich východiskovej polohy.

Oba módy zabezpečujú cielený pohyb hlavy nástroja robotického ruky operačného robota. V spustenom programe je možné módy meniť pomocou jednoduchého zadania znakov A / a, resp. B / b do sériového monitoru. Pri každej zmene módu sú servá najskôr nastavené do svojich východiskových polôh.

## 4 DISKUSIA VÝSLEDKOV

### 4.1 Kontrola a preklz hraničných polôh

Dôležitou súčasťou riadiaceho systému nástroja operačného robota je ošetrovanie preklzu hraničných polôh nástroja.

V pôvodnom návrhu, ktorý počíta s použitím krokových motorov, je jednou z možností použitie šnekového súkolia. Mechanizmus sa skladá zo vzájomne prepojeného šnekového kolesa a šneka (viď obrázok 4.1). Požadovaný efekt by bol docielený, pokiaľ by bol hriadeľ krokového motora pevne spojený so šnekom a šnekové koleso s ovládacím kolieskom tela nástroja. Pokiaľ by došlo k ošetrovaniu všetkých ovládacích koliesok nástroja týmto mechanizmom, nebolo by možné vonkajším mechanickým pôsobením na nástrojovú hlavu za žiadnych okolností zmeniť jej aktuálnu polohu.



Obr. 4.1: Mechanizmus šnekového súkolia [18]

Praktická implementácia mechanizmu je však možná len v prípade rotácie nástroja okolo osi  $z$ . Medzi ostatnými pohybmi nástroja existujú väzby (viď tabuľka 2.3), kde si jeden pohyb vyžaduje súčasnú rotáciu viacerých koliesok. Pri použití šnekového súkolia by v tomto prípade dochádzalo k nežiadúcim pohybom (samovolný náklon krídel počas náklonu hlavy na strany), prípadne by si zavádzali v pohybe krídla nástroja navzájom pri žiadúcom pohybe len jedného z krídel. Kontrolu hraničných polôh nástroja by bolo možné snímať na základe zvýšeného prúdového odberu motora pri jej nadobudnutí.

Výsledný implementovaný návrh využíva pre pohon koliesok nástroja modelárske servá. Tieto pohonné jednotky majú obmedzený rozsah pohybu v dôsledku vstavaného potenciometra, na základe ktorého je snímaný aktuálny ohol natočenia hriadeľa. Aditívne ošetrovanie kontroly krajných polôh je teda bezpredmetné. Vysoká prídržná sila serv je taktiež postačujúca pre daný účel. Použitie pohonných jednotiek

tohoto typu výrazne zjednodušuje konečný hardvérový návrh a je možné ich priame prepojenie z kolieskami zabezpečujúcimi pohyb operačného nástroja.

## 4.2 Komunikácia

Nosnou časťou implementácie navrhovaného riadiaceho systému je sériový prenos informácie o polohe analógových manipulátorov na platformu Arduino, ktorá zabezpečí ich spracovanie a následný odpovedajúci pohyb serv. Existuje niekoľko možností ako dátový prenos sprostredkovať.

Pôvodný návrh uvažoval použitie herného ovládacieho zariadenia - gamepadu. Informácia o polohe jeho joystickov bola analyzovaná príslušným programom v prostredí LabVIEW a prostredníctvom architektúry VISA odosielaná na sériový port. Tok dát medzi prostredím LabVIEW a mikrokontrolérom sa nepodarilo zosynchronizovať a dochádzalo k výraznému skresľovaniu informácie obsiahnutej v komunikačnom protokole (viď podkapitolu 3.1.2).

Potenciálne riešenie predstavuje použitie LabVIEW rozhrania pre Arduino. Systém riadenia by počítal s použitím len jedného softvérového nástroja - LabVIEW. Náhradou blokov VISA (viď obr. 3.1) za bloky Arduino (viď obr. 3.3) je možné zabezpečiť korektnú a bezproblémovú komunikáciu medzi softvérom a mikrokontrolérom Arduino. Nadviazať však týmto spôsobom na program pre tvorbu bitového slova v prostredí LabVIEW nie je možné. Implementácia daného riešenia by vyžadovala vytvorenie úplne nového riadiaceho softvéru.

Výsledný návrh riadiaceho systému kladie dôraz na jednoduchosť a spoľahlivosť. Snahou je ako obmedzenie počtu použitých softvérových nástrojov, tak minimalizácia množstva použitých hardvérových zariadení. Jadro tvorí mikrokontrolér naprogramovaný v prostredí IDE, ktorý je priamo prepojený s rozširujúcim zariadením *Motor Shield* vytvoreným pre túto konkrétnu aplikáciu. USB prepojenie Arduina s PC zabezpečuje len napájanie analógových joystickov napätím 5 V a ponecháva možnosť modifikácie zdrojového kódu.

## 4.3 Ovládacie módy

Odlíšne prístupy ovládania nástroja robotickej ruky boli zvolené s ohľadom na potenciálne použitie riadiaceho systému v praxi. Výkon operačného zákroku si vyžaduje ako razantné, tak aj pomalé sekvenčné pohyby pre jemnú manipuláciu s tkanivom. Ovládací mód A zabezpečuje prevod pohybu joystickov na presne odpovedajúci pohyb serv a tým aj operačného nástroja. Uplatnenie nájde pri resekcii tkaniva, orgánov, prípadne pri rezaní, strihaní alebo umiestňovaní rôznych svoriek na cievy a

podobne. Použitie módu B zabezpečuje plynulý krokový pohyb umožňujúci zotrvanie v statickej polohe mimo východiskový stav nástroja. Mód je vhodný napríklad pri procese šitia, kedy je ihla fixovaná v nástroji a cielený pohyb vykonáva celé rameno operačného robota. Rýchlosť pohybu serv v móde A je konštantná bez ohľadu na počet serv, ktorých činnosť je nutná pre výkon daného komplexného pohybu nástroja. Mód B je viazaný na výpočetnú kapacitu použitého mikrokontroléra. S narastajúcim počtom serv pohybujúcich sa naraz v jednom okamihu stúpa výpočetná náročnosť celého procesu. Pohyb sa výrazne spomaľuje, prípadne vykazuje drobné abnormality a nežiadúce zákmity. S týmto obmedzením je nutné počítat.

## 5 ZÁVER

Výstupom bakalárskej práce je návrh systému riadenia robotickej ruky operačného robota Da Vinci. Prvotným cieľom bolo nadviazanie na program pre tvorbu riadiaceho bitového slova vytvoreného v prostredí LabVIEW a doplnenie tohoto riešenia o hardvérovú časť. Systém bol navrhnutý s nosnými prvkami v podobe herného ovládacieho zariadenia gamepad, mikrokontroléra Arduino a krokových motorov. Program, na ktorý má práca nadväzovať, uvažuje použitie virtuálnej softvérovej architektúry (VISA) ako nosného prvku zabezpečujúceho dátovú komunikáciu medzi zariadením gamepad a mikrokontrolérom Arduino. Použitím nástroja VISA dochádza k výraznému skreslovaniu transferovaných dát a nie je teda možné ho v implementácii navrhovaného riadiaceho systému použiť. Problematika dátového prenosu je podrobne riešená v kapitole 3.1. Možné riešenie spočíva v použití nástroja LabVIEW rozhranie pre Arduino. Pri tejto variante by bolo nutné vytvoriť úplne nový riadiaci softvér. Nadviazať na program pre tvorbu bitového slova nie je možné.

Výsledný návrh riadiaceho systému kladie dôraz na jednoduchosť a spoľahlivosť. Snahou je obmedzenie počtu použitých softvérových nástrojov, a tým obmedzenie možného skreslenia nosnej informácie pre pohyb nástroja na minimum. Jadro tvorí rozširujúce zariadenie Motor Shield obsahujúce ovládacie joysticky, ktoré bolo fyzicky vytvorené pre potreby implementácie riešena navrhnutého v tejto bakalárskej práci. Shield je priamo prepojený s mikrokontrolérom Arduino Uno. Riadiaci softvér je vytvorený v prostredí IDE.

Cieľom návrhu riadiaceho systému je taktiež ošetrovanie kontroly a preklzu hraničných polôh zvolenej pohonnej jednotky. V prípade pôvodného návrhu, ktorý uvažuje použitie krokových motorov, je možné ošetriť preklz použitím šnekového súkolia. Kontrolu hraničných polôh nástroja by bolo možné snímať na základe zvýšeného prúdového odberu motora pri jej nadobudnutí. V prípade výsledného návrhu rieši túto problematiku samotná konštrukcia zvolenej pohonnej jednotky - serva.

Predmetom nadväzujúcej práce by mohlo byť technické riešenie mechanického prepojenia hriadelov servomotorov s kolieskami, ktorých rotácia iniciuje pohyb operačného nástroja.

V dôsledku neočakávaných abnormalít v dátovom prenose sa výsledný návrh riadiaceho systému odlišuje od pôvodného, a tým aj časti zadania bakalárskej práce, pomerne zásadným spôsobom. Navrhnutý riadiaci systém je implementovaný v súlade s hlavnou myšlienkou zadania práce. Maximálny dôraz je kladený na principiálnu funkčnosť, a preto som presvedčený, že bolo zadanie práce splnené v plnom rozsahu.

## LITERATÚRA

- [1] PUSHKAR', D. Yu. Robots in medicine. Herald of the Russian Academy of Sciences [online]. 2012, vol. 82, issue 6, s. 414-422 [cit. 2013-11-25]. DOI: 10.1134/S1019331612060056. Dostupné z: <http://link.springer.com/10.1134/S1019331612060056>
- [2] ASIMOV, Isaac. Já, robot. Vyd. 2. V Praze: Triton, 2004, s. 323. Trifid (Triton). ISBN 80-7254-560-4.
- [3] All on Robots. What is a robot? [online] [cit. 2013-11-25]. Dostupné z: <http://www.allonrobots.com>
- [4] All About Robotic Surgery. A brief history. [online] [cit. 2013-11-25]. Dostupné z: <http://allaboutroboticsurgery.com/surgicalrobots.html>
- [5] NOSKA, Martin. Computerworld. Robotická chirurgie přinesla revoluci i do českého zdravotnictví [online]. Vydané 16. 5. 2009, [cit. 2013-11-28]. Dostupné z URL: <http://computerworld.cz/aktuality/Roboticka-chirurgie-prinesla-revoluci-i-do-ceskeho-zdravotnictvi-4036>>, CZ-ISSN 1210-9924
- [6] Nemocnice sv. Zdislavy. Centrum robotické chirurgie Vysočina [online]. [cit. 2013-11-28]. Dostupné z URL: <http://www.nemocnice-mostiste.cz/rubrika/349-Zdravotnicke-sluzby-Specialni-sluzby-Centrum-roboticke-chirurgie-Vysocina/index.htm>
- [7] Intuitive Surgical. The da Vinci Surgical System. [online]. [cit. 2013-11-28]. Dostupné z URL: <http://www.intuitivesurgical.com/products/>
- [8] Intuitive Surgical. EndoWrist Instruments. [online]. [cit. 2013-12-02]. Dostupné z URL: <http://www.intuitivesurgical.com/products/instruments/>
- [9] Learning Engineering. Degrees of Freedom|Mechanics [online]. 2011 [cit. 2013-12-02]. Dostupné z URL: <http://www.learnengineering.org/2013/04/degrees-of-freedom-mechanics.html>
- [10] Heureka.cz Logitech Dual Action [online, upraveno] [cit. 2014-01-03]. Dostupný z URL: <http://gamepady.heureka.cz/logitech-dual-action>
- [11] Emartee.com. Product|Motor and Servo [online]. 2013 [cit. 2014-01-03]. Dostupné z URL: <http://www.emartee.com/product/41886/28BYJ>



- [12] VÍTOVÁ, H. Systém řízení nástroje robotické ruky z operačního robota Da Vinci v LabVIEW. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií [online], 2010 [cit. 2014-01-03] 42 s. Dostupné z URL: [https://www.vutbr.cz/studium/zaverecne-prace?zp\\_id=30691](https://www.vutbr.cz/studium/zaverecne-prace?zp_id=30691)
- [13] National Instruments. National Instrument VISA [online]. 2012 [cit. 2014-05-14]. Dostupné z URL: <http://www.ni.com/visa/>
- [14] Microcontroller Made Easy. Proteus Library for Arduino [online]. 2013 [cit. 2014-05-20]. Dostupné z URL: <http://microcontroller-made-easy.blogspot.cz/2013/09/proteus-library-for-arduino.html>
- [15] National Instruments. NI LabVIEW Interface for Arduino Toolkit [online]. 2014 [cit. 2014-05-21]. Dostupné z URL: <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/209835>
- [16] MCROBERTS, Michael. Beginning Arduino. New York: Apress, 2010, xxi, 433 s. technology in action. ISBN 978-1-4302-3240-7.
- [17] Arduino. Arduino Uno [online]. 2014 [cit. 2014-05-21]. Dostupné z URL: <http://arduino.cc/en/Main/arduinoBoardUno>
- [18] T.E.A. TECHNIK s.r.o. Šnekové súkolia [online]. 2007 [cit. 2014-01-05]. Dostupné z URL: <http://www.teatechnik.sk/typ-1/>
- [19] ServoDatabase.com. RC servo Specifications and Reviews. [online]. 2014 [cit. 2014-05-21]. Dostupné z URL: <http://www.servodatabase.com/servo/hitec/hs-5485hb>

# ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

PUMA Programmable Universal Manipulation Arm

R. U. R. Rossum's Universal Robots

3D Tri dimenzie

CT Computed Tomography

AESOP Automated Endoscopic System for Optical Positioning

PC Personal computer

USB Universal Serial Bus

XOR Exclusive OR

IDE Integrated Development Environment

NI-VISA National Instruments - Virtual Instrument Software Architecture

I/O API Input/Output Applications Programming Interface

RAM Random Access Memory

UART Universal Asynchronous Receiver / Transmitter

PWM Power Width Modulation

RS-232 Recommende Standard 232

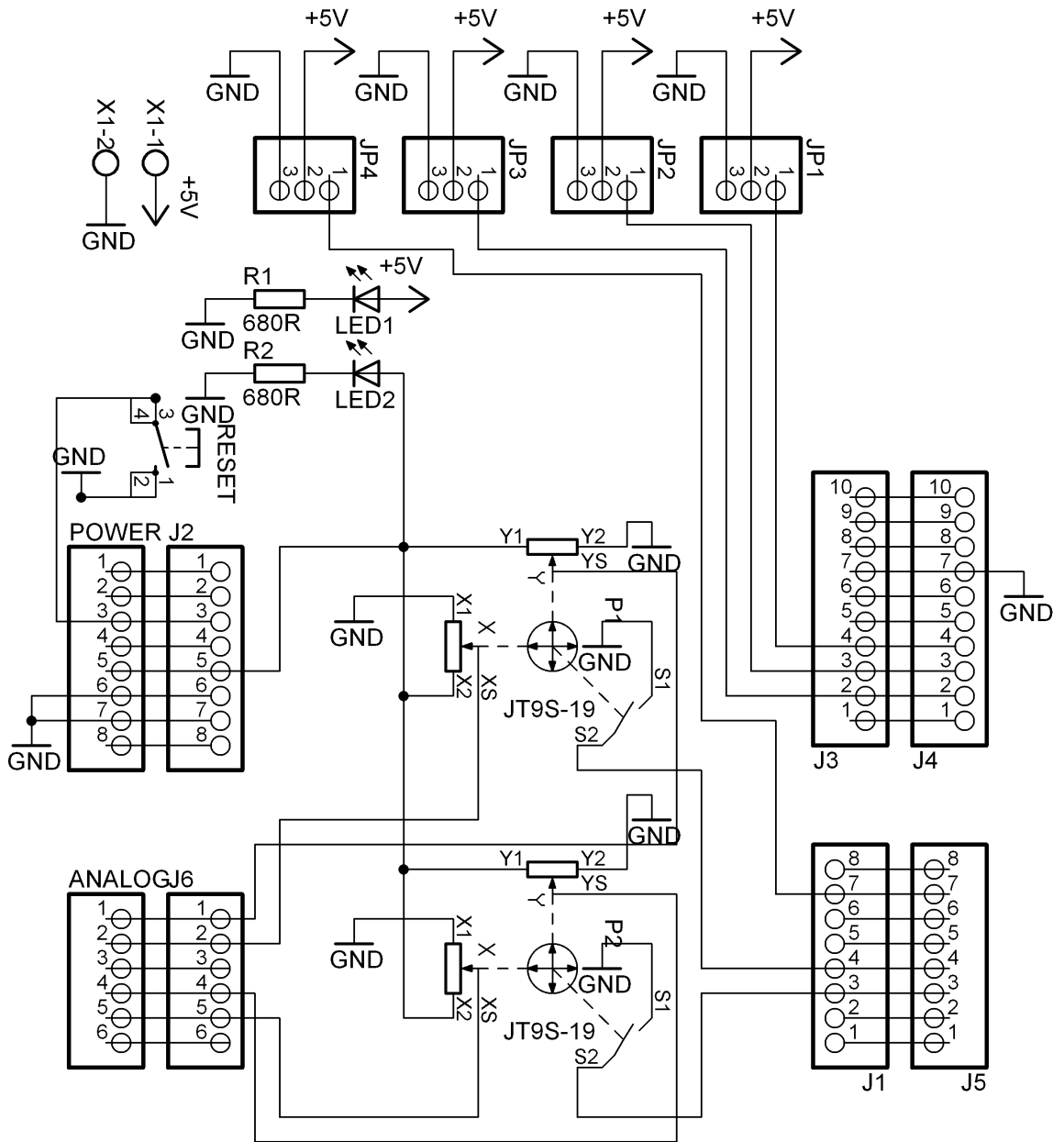
PCB Printed Circuit Board

PWM Power Width Modulation

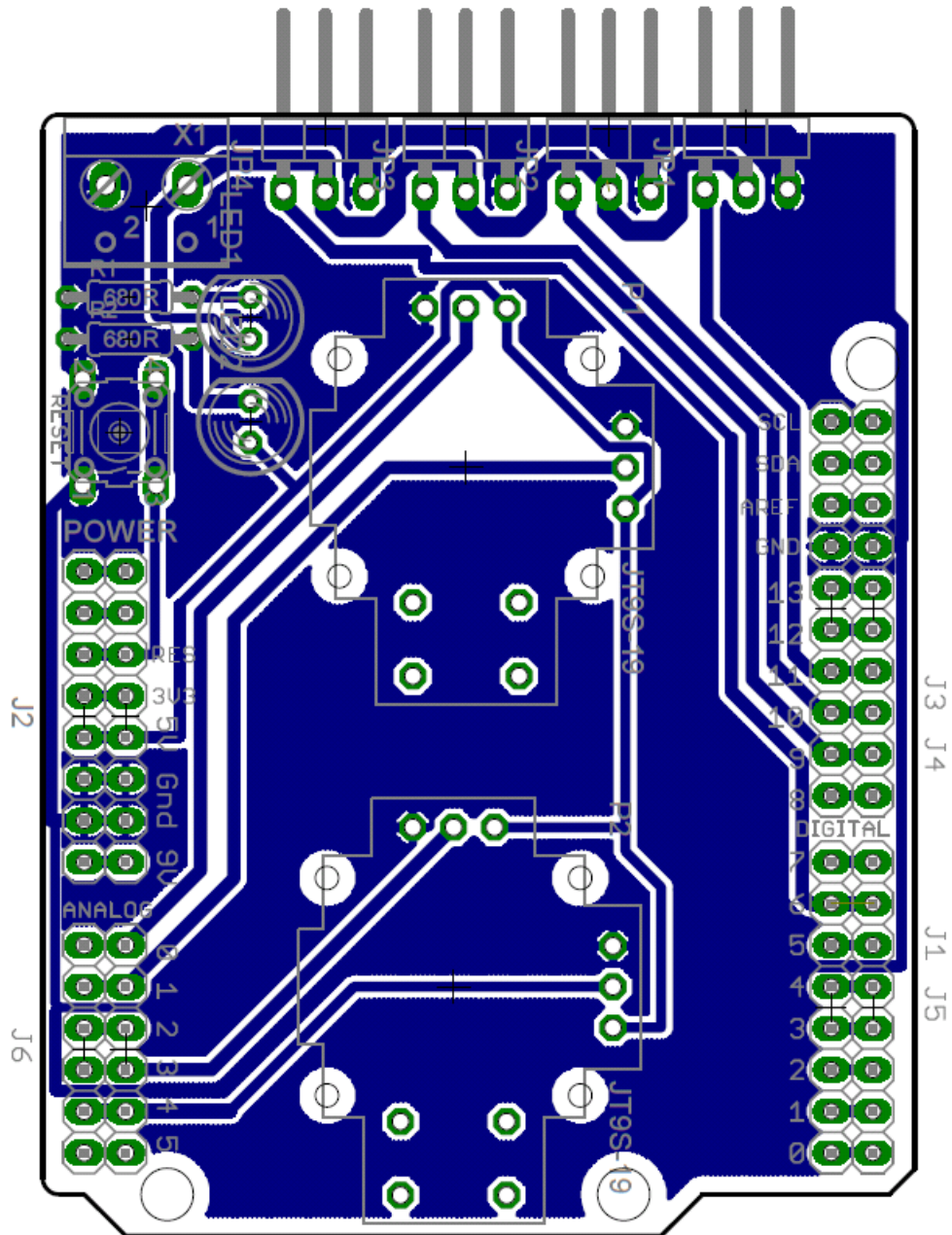
## ZOZNAM PRÍLOH

A Schéma dosky plošného spoja	I
B Návrh dosky plošného spoja	II
C Zoznam použitých súčiastok	II

# A SCHÉMA DOSKY PLOŠNÉHO SPOJA



## B NÁVRH DOSKY PLOŠNÉHO SPOJA



## C ZOZNAM POUŽITÝCH SÚČIASTOK

počet kusov	označenie	charakteristika zariadenia
2	P1, P2	3-D joystick s vypínačom 60 W; 10 k $\Omega$
2	LED1, LED2	LED 3 mm; 35°; 150 mcd; 1,8 V; 20 mA
2	R1, R2	rezistor 680 $\Omega$ ; 1/4 W
12	JP1, JP2, JP3, JP4	obojsstranný kolík XINYA 2,54 mm
1	X1	pružinová svorkovnica; počet kontaktov = 2
1	RESET	tlačidlo B3F-1000 Omron