

# Západočeská univerzita v Plzni

FAKULTA PEDAGOGICKÁ

KATEDRA MATEMATIKY, FYZIKY A TECHNICKÉ VÝCHOVY

## VYUŽITÍ ROBOTICKÉHO MANIPULÁTORU PRO VÝUKU NA ZŠ DIPLOMOVÁ PRÁCE

*Bc. Aleš Křivánek*

*Učitelství pro základní školy, obor INF-Te  
léta studia (2010 - 2012)*

Vedoucí práce: *Mgr. Jan Krotký*

Plzeň, 29. Červen 2012

Prohlašuji, že jsem diplomovou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

Plzeň, 29. Červen 2012

.....  
vlastnoruční podpis

## OBSAH

1	ÚVOD .....	1
2	ANALÝZA RVP ZV – KDY A JAK NASADIT ROBOTIKU NA ZV .....	3
2.1	VÝBĚR VHODNÝCH VZDĚLÁVACÍCH OBLASTÍ.....	3
2.1.1	Člověk a svět práce .....	3
2.1.2	Informační a komunikační technologie .....	6
2.1.3	Člověk a příroda.....	6
2.2	JAK A ZDA VŮBEC SE VYUČUJE ROBOTIKA U NÁS .....	8
2.2.1	Roboti od Lynxmotion .....	8
2.2.2	Jiné druhy robotických stavebnic .....	8
2.2.3	Soutěže s roboty .....	9
2.3	JAK SE VYUČUJE ROBOTIKA VE SVĚTĚ.....	10
2.3.1	Mezinárodní projekty a soutěže .....	10
3	ROBOTICKÉ STAVEBNICE OD FIRMY LYNXMOTION .....	11
3.1	O FIRMĚ LYNXMOTION .....	11
3.1.1	Distributor v ČR.....	11
3.2	PŘEDSTAVENÍ STAVEBNIC .....	12
3.2.1	Robotické ruce .....	12
3.2.2	„Chodící“ roboti .....	13
3.2.3	Podvozkové platformy.....	14
3.2.4	Řídící elektronika .....	15
4	SESTAVENÍ A OŽIVENÍ ROBOTICKÉ RUKY AL5D .....	17
4.1	TIPY PRO SESTAVENÍ .....	18
4.1.1	Správné vedení kabeláže .....	19
4.2	UPEVNĚNÍ A ZAPOJENÍ SENZORŮ A KAMERY .....	21
4.2.1	Infračervený senzor Sharp GP2Y0A41 .....	21
4.2.2	Bezdrátová kamera.....	24
4.3	SPOJENÍ S POČÍTAČEM .....	25
4.3.1	Redukce od FTDI .....	25
4.3.2	Redukce od Omron.....	27
4.3.3	Test spojení pomocí aplikace LynxTerm.....	28
4.4	KONSTRUKCE BLUETOOTH BEZDRÁTOVÉ REDUKCE .....	30
4.4.1	Nastavení ovladačů pro USB Bluetooth adaptér .....	33
5	SEZNÁMENÍ S UŽIVATELSKÝM PROSTŘEDÍM OVLÁDACÍHO SOFTWARE RIOS .....	35
5.1	HLAVNÍ OBRAZOVKA PROGRAMU .....	35
5.2	ČERVENÝ CELEK – KOMUNIKACE .....	36
5.3	MODRÝ CELEK – KONFIGURACE A KALIBRACE .....	36
5.3.1	Tlačítko All=1,5mS .....	36
5.3.2	Tlačítko SSC-32 .....	37
5.3.3	Tlačítko ARM.....	41
5.3.4	Tlačítko Project .....	42
5.3.5	Tlačítko XYZ sys.....	43
5.4	ZELENÝ CELEK – OVLÁDACÍ ČÁST .....	43
5.4.1	Moves (Pohyby).....	43
5.4.2	Play (přehrávání projektů).....	51
5.5	3D SCAN .....	56
5.6	SOCKET SERVER A SOCKET CLIENT .....	58

---

6	PROBLEMATIKA PŘÍMÉ A INVERZNÍ KINEMATIKY .....	60
6.1	PŘÍMÁ KINEMATIKA .....	60
6.2	INVERZNÍ KINEMATIKA .....	61
7	ALTERNATIVNÍ SOFTWARE PRO OVLÁDÁNÍ ROBOTICKÉ RUKY .....	62
7.1	FLOWSTONE .....	62
7.2	FLOWBOTICS STUDIO .....	64
7.3	LYNX SERVO SEQUENCER .....	65
8	PRAKTICKÉ PŘÍKLADY PRO VYUŽITÍ VE VÝUCE NA ZŠ .....	68
8.1	SEZNÁMENÍ S ROBOTICKOU RUKOU – ZAMÁVÁNÍ .....	68
8.2	PŘENESENÍ PŘEDMĚTU .....	69
8.3	PŘENESENÍ PŘEDMĚTU S VYUŽITÍM JOYSTICKU .....	70
8.4	SIMULACE PYROTECHNICKÉHO ROBOTA .....	72
8.5	VYKONÁNÍ POSLOUPNOSTI PŘÍKAZŮ S VYUŽITÍM CYKLŮ .....	74
8.6	PŘENESENÍ PŘEDMĚTU S VYUŽITÍM PODMÍNKY IF .....	75
9	ZÁVĚR .....	78
10	SEZNAM OBRÁZKŮ .....	79
11	SEZNAM LITERATURY .....	81
12	RESUMÉ .....	82
13	PŘÍLOHY .....	I

## 1 ÚVOD

Rozvoj počítačové techniky v několika posledních desetiletích přinesl do té doby nevídané možnosti v mnoha převážně technických odvětvích. Mezi ně můžeme určitě počítat i automatizaci. A s ní jde ruku v ruce i robotika.

Dnes si můžeme již jen těžko představit jakoukoli továrnu, kde se nepoužívají alespoň v omezené míře roboti. Když se podíváme do moderních automobilek, naskytne se nám úžasný pohled na roboty, kteří automobily sestavují, sváří, lakují a dokonce i kontrolují. V posledních několika letech se můžeme setkat s robotickými rukama, které provádí operace, které by člověk nikdy nebyl schopen realizovat. A takových příkladů si můžeme vyjmenovat celou řadu. Ale vědí vlastně lidé, jak roboti pracují a co je všechno potřeba? Většinou asi ne. Ale od toho je tu škola, aby rozšiřovala obzory a podněcovala zájem o vědu i techniku.

V dnešní době můžeme najít mnoho robotických stavebnic, vyjmenujme např. Merkur, Lynxmotion, LEGO a další. V této práci se budeme podrobněji zabývat jedinou, a to sice robotickou rukou od Lynxmotion. Důvodů, proč jsme si vybrali zrovna tento typ, je několik a zmíníme je dále v textu.

Jelikož jsme na pedagogické fakultě, předpokládá se, že práce bude koncipována na výuku. Tato diplomová práce by měla sloužit jako základní příručka a průvodce pro učitele, který by chtěl začlenit robotiku do svojí výuky. A ač se to nezdá, příležitostí je mnoho.

V první kapitole se budeme věnovat analýze RVP ZŠ. Budeme chtít zjistit, zda současné RVP obsahuje vzdělávací oblasti, která by s robotikou souvisela nebo kam by šla zařadit. Budeme se také snažit zjistit, zda je už někde v ČR robotika začleněna do výuky.

V další kapitole se budeme věnovat samotným robotům od firmy Lynxmotion. Ačkoli se budeme dále věnovat pouze robotické ruce, sluší se zmínit i další výrobky této firmy, neboť všechny stavebnice jsou vyráběny modulárně, tudíž není problém v budoucnu stavebnici dále rozšiřovat. V této kapitole je také popsána stavba a hlavně problémy, které při ní mohou nastat s důrazem na oživení ruky a propojení s PC.

V následující kapitole se budeme podrobněji zabývat softwarem, který byl spolu s touto stavebnicí zakoupen. Jedná se o aplikaci vyvinutou přímo výrobcem - RIOS. Popíšeme si uživatelské prostředí, základní ovládací prvky a možnosti. Důraz bude kladen na pochopení logiky práce s tímto programem.

V další kapitole se podíváme na některé vybrané alternativní aplikace, které se dnes používají pro ovládání robotů. Musíme samozřejmě přihlídnout na to, že práce je orientována na výuku na ZŠ, tudíž jedním z hlavních kritérií pro výběr softwaru byla uživatelská přívětivost a snadnost porozumění i nezkušeným uživatelem.

V poslední části si ukážeme několik praktických příkladů od zadání až po samotnou realizaci v programu RIOS.

## 2 ANALÝZA RVP ZV – KDY A JAK NASADIT ROBOTIKU NA ZV

Ještě, než se začneme věnovat samotné „ruce“, měli bychom napřed zjistit, zda můžeme zahrnout robotiku do výuky podle dnes platného RVP pro ZŠ.

Zaměření RVP pro základní školy vybíráme z důvodu mé aprobace, která je technická výchova pro 2. stupeň základní školy. Možnosti těchto stavebnic zdaleka přesahují možnosti výuky nejen na SŠ, natož pak ZŠ. Roboty od Lynxmotion dokonce používá pro výuku robotiky i plzeňská FEL. Z toho si můžeme odvodit, že se jedná o skutečně komplexní řešení a nikoliv žádnou „hračku.“

### 2.1 VÝBĚR VHODNÝCH VZDĚLÁVACÍCH OBLASTÍ

Když se podíváme na jednotlivé vzdělávací oblasti RVP v obecné rovině, nejvíce se nám budou nabízet tyto tři:

- Člověk a svět práce
- Informační a komunikační technologie
- Člověk a příroda – vzdělávací obor Fyzika

Vidíme, že všechny tyto oblasti se svými obory k sobě mají velice blízko a podle RVP by mezi sebou měly mít mezipředmětové vazby. Pojdme se nyní podívat na jednotlivé obory podrobněji.

#### 2.1.1 ČLOVĚK A SVĚT PRÁCE

Jsme členy katedry matematiky, fyziky a technické výchovy, tudíž máme k této oblasti nejbližše a budeme se jí také nejvíce věnovat.

Nespornou výhodou kurikulární reformy je to, že si učitel může pro výuku oboru celé ŠVP postavit tak, jak mu to vyhovuje. Povinný v plném rozsahu je pouze obor Svět práce, začlenění ostatních je na nás a na vedení školy, takže když se nám povede prosadit nakoupení robotů a současnou změnu ŠVP, můžeme pro naši výuku získat celkem dobrou hodinovou dotaci.

Uveďme si můj případ. Moje aprobace jsou Informatika a Technická výchova. Předpokládá se, že oba předměty budu učit jednu třídu. Jak zjistíme dále, oba dva předměty se navzájem prolínají, takže ve výsledku nebude problém využít například pro

projektovou výuku robotiky jak hodiny informatiky, tak i technické výchovy. Rázem nám vznikají nové možnosti a mezipředmětové vazby. Ideální bude, když se do projektu zapojí ještě např. matika a fyzika. Vznikne nám najednou vazba mezi 4 předměty a žáky to bude navíc ke všemu ještě bavit.

Z RVP víme, že je vzdělávací obsah oblasti realizován na 1. i 2. stupni. Na první pohled by se mohlo zdát, že robotika je vhodná pouze na 2. stupeň, kdy už mají děti jisté znalosti i dovednosti. Ale jak jsme mohli ověřit v praxi na dnech vědy a techniky, kdy byla naše ruka v celodenní permanenci, i pro děti na 1. stupni může mít veliký přínos do výuky jinak „nudných“ předmětů.

### **1. STUPEŇ**

Zde můžeme najít pouze jeden tematický okruh. Tím je **Konstrukční činnosti**. Na prvním stupni je tento okruh orientován na elementární dovednosti a činnosti při práci se stavebnicemi. Ve výstupech můžeme najít, že žák provádí montáž a demontáž, pracuje podle návodu a předlohy apod.

Nemůžeme čekat, že budou žáci vymýšlet složité programy či algoritmy, ale již v takto malém věku se seznámí s roboty, hlavně s tím, jak pracují a jak se pohybují. Z našich získaných zkušeností na dnech vědy a techniky víme, že i šestileté dítě nemá po chvíli problém pochopit princip ovládní robotické ruky pomocí joysticku. Dítě zde naprosto přirozenou cestou pomocí hry získá povědomí o tom, jak se v prostoru předměty pohybují (osy x, y a z). A právě aplikace RIOS tomu jen pomáhá. Obsahuje v sobě totiž simulaci pohybu ruky na všech třech osách. Jak se dozvíme později, máme na výběr z několika druhů pohybu ruky v prostoru (samostatné klouby, pohyb v jednotlivých osách,...). Hned zde můžeme vidět mezipředmětovou vazbu do matematiky, respektive hlavně do geometrie a prostorové představivosti.

Hlavní způsob využití na 1. stupni bychom viděli v jednoduchém přesouvání předmětů pomocí ruky s joystickem. Software RIOS totiž umí ovládat ruku i pomocí periférií, nejčastěji to bude joystick či joypad.

Jak už bylo zmíněno, ovládní robota pomocí joysticku jsme si prakticky vyzkoušeli na dnech vědy a techniky. Zájem měli nejvíce děti od 6 do 12 let a i ti nejmenší neměli s ovládním nejmenší problémy. To přisuzuji tomu, že dnes již skoro každý hraje



počítačové hry, tudíž má s těmito ovladači již zkušenosti. Děti si tak vůbec nebudou uvědomovat, že nehrají počítačovou hru, ale učí se.

## 2. STUPEŇ

Na 2. stupni již můžeme v RVP najít podstatně více oborů, do kterých můžeme robotiku zakomponovat. Navíc žáci jsou v tomto věku již mnohem více vyspělí a mají mnohem více znalostí a dovedností. Můžeme proto využít více možností, které nám robotické stavebnice nabízí. Od jednoduchých úloh s ovládáním pomocí periférií můžeme sklouznout k samotnému programování robota.

Robotiku můžeme zahrnout do těchto 4 tematických okruhů:

- **Design a konstruování** – obsahem je hlavně to, aby uměl žák sestavit podle náčrtu či návodu stavebnici, uměl sám zkonstruovat a ozkoušet funkčnost vlastních konstrukčních prvků atd.  
Jelikož je stavebnice modulární, žáci si mohou sami navrhovat vylepšení či nové konstrukční prvky.
- **Práce s laboratorní technikou** – zde by měl žák umět správně po všech stránkách provést experiment a formulovat závěry, ke kterým došel. Žák by také měl mít znalosti o obsluze laboratorní techniky atd. Těchto všech znalostí je potřeba pro splnění komplexních úloh, které nám robotika přináší. Vyžaduje po žácích myšlení v souvislostech a hlavně dochází k uvedení naučených znalostí do praxe
- **Využití digitálních technologií** – Všechny stavebnice od Lynxmotion jsou konstruovány tak, že vyžadují ke svému fungování počítač. V tomto oboru se jedná konkrétně o digitální technologie typu Bluetooth, periferní zařízení, počítačové porty, kamery atd. Opět tedy dochází k využití teoretických znalostí do praxe (propojení pomocí Bluetooth, infračervená kamera apod.

### 2.1.2 INFORMAČNÍ A KOMUNIKAČNÍ TECHNOLOGIE

Obsah tohoto oboru poslouží jako výborné doplnění znalostí a dovedností, které budou žáci potřebovat pro práci s robotickou rukou. Je členěn na 1. a 2. stupeň.

Na prvním stupni nás bude zajímat pouze **Základy práce s počítačem**. Žáci by se tedy měli naučit základní obsluhu počítače, včetně práce s periferiemi (v našem případě joystick). Měli by být schopni spustit ovládací program RIOS a provést potřebné úkony, které povedou ke spuštění robota.

Druhý stupeň rozšiřuje již nabyté znalosti. Za zmínku stojí **Zpracování a využití informací**. Žáci by měli ze získaných údajů (např. z infračervené kamery) vyvodit závěry a ty zpracovat například pomocí tabulkového editoru do grafu. Opět zde můžeme najít mezipředmětovou vazbu na fyzikální veličiny (infračervené barevné spektrum) i matematiku.

### 2.1.3 ČLOVĚK A PŘÍRODA

V této oblasti jsou 3 obory: Fyzika, Chemie a Přírodopis. Nás bude zajímat převážně Fyzika, neboť ke správnému pochopení toho, jak roboti fungují, je zapotřebí osvojení si základních fyzikálních zákonů a jevů.

Nejdůležitější pro nás budou tyto 2 moduly:

- **Pohyb těles** – jelikož má ruka několik kloubů, které se pohybují v prostoru, je zapotřebí, aby žáci znali základní pojmy z kinematiky pevných těles, skládání sil, gravitační sílu a další. Ruka bude zvedat tělesa, žáci si mohou spočítat, jak těžké těleso ruka nese. A opět zde vidíme mezipředmětovou vazbu – napřed si žáci budou muset spočítat, zda ruka unese zvedané těleso, když známe její délku (páka). K řešení celého příkladu bude zapotřebí kombinovat znalosti z několika předmětů.
- **Elektromagnetické a světelné děje** – v tomto modulu je obsaženo všechno učivo, která se týká optiky a elektřiny. Když tuto látku aplikujeme na robotickou ruku, zjistíme, že se dá najít mnoho společného. Robotická ruka je velice komplexní zařízení, které v sobě sdružuje mnoho samostatně funkčních součástí. Uvedme si několik příkladů:

- **servomotory** – žáci se prakticky seznámí s tím, jak fungují a jak se ovládají – to spadá do učiva elektrického a magnetického pole.
- **infračervená kamera** – žáci uvidí, jak funguje v reálných podmínkách detekce vzdálenosti od robota, seznámí se také s principem funkce infračervené kamery – spadá do učiva vlastnosti světla.

Seznámili jsme se tedy s možnostmi nasazení výuky robotiky na ZŠ podle platného RVP. Zjistili jsme, že nám v podstatě nic nebrání tomu, abychom ji použili v hned v několika předmětech typu informatiky, fyziky, technické výchovy a podobně. K tomu vyhovuje hned několik vzdělávacích oblastí a oborů.

Problémů, na které můžeme v reálných podmínkách narazit, je několik. Jedním z těch největších může být pro většinu školy otázka financí. Pořízení stavebnice není sice velká investice (řádově tisíce korun), ale i to může být dnes pro školu problém.

Dalším problémem je nedostatečná hodinová dotace předmětů, kde se dá robotika vyučovat (informatika a technická výchova). Každý z předmětů má týdně pouze 1–2 hodiny, což je pro rozsáhlejší projekty málo. Na druhou stranu nic nebrání tomu, aby výuka probíhala jako projekt mezi několika předměty či jako dobrovolný kroužek nebo povinně volitelný předmět. A jak zjistíme dále, na školách to tak skutečně funguje.

Posledním problémem může být neochota ostatních vyučujících či vedení školy k začlenění robotiky do jejich výuky. To ale záleží pouze na nás, abychom o tom ostatní přesvědčili.

Nyní se podívejme, jak a zda vůbec se robotika vyučuje u nás a ve světě...

## 2.2 JAK A ZDA VŮBEC SE VYUČUJE ROBOTIKA U NÁS

Zaměříme se teď na samotnou výuku robotiky více prakticky. Pokusíme se zjistit, jak to vypadá se současným stavem výuky robotiky na českých školách.

### 2.2.1 ROBOTI OD LYNXMOTION

Jelikož je tato práce zaměřena na roboty firmy Lynxmotion, snažili jsme se nejprve dohledat, zda v ČR existuje nějaká škola, kde se robotika či mechatronika s těmito stavebnicemi vyučuje. Prošli jsme dostupné zdroje a bohužel s negativním výsledkem.

Na českých základních školách se v současné době nenachází žádná škola, která by roboty od Lynxmotion využívala pro aktivní výuku. A o moc lepší to není ani na středních školách. Podařilo se nám najít pouze SPŠ a VOŠ Písek.

Na vysokých školách je to přeci jen lepší. Když vezmeme jako příklad pouze ZČU, tak na elektrotechnické fakultě na katedře elektromechaniky a výkonové elektroniky existuje celá učebna pro výuku robotiky a právě v ní se používají stavební prvky a elektronika od Lynxmotion.

### 2.2.2 JINÉ DRUHY ROBOTICKÝCH STAVEBNIC

Doposud jsme se bavili pouze o využití robotických stavebnic Lynxmotion. Ale na trhu jich existuje mnohem více. Patrně tou nejznámější je Lego Dacta a její nástupce Lego NXT Mindstorms.

Stavebnice LEGO Mindstorms získává v poslední době velkou popularitu a když se pokusíme zapátrat na internetu, brzy zjistíme, že mnoho základních škol je využívá. Zmínit můžeme například [ZŠ Hustopeče na Břeclavsku](#), která dokonce získala od evropských fondů dotaci na nákup těchto stavebnic.

Dalším příkladem může být projekt od [Nadace ČEZ](#) pro základní školu v Písku u Jablunkova, kde byla těmito roboty vybavena multimediální učebna pro výuku fyzika. V Plzni běžně používá na výuku fyziky robotické stavebnice od [LEGA 28. ZŠ](#). Zkrátka těchto škol je hodně a neustále přibývají. A to je jedině dobře.

Zajímavě vypadá projekt pedagogické fakulty Univerzity Karlovy v Praze s názvem ICT školám. Zde je nabízeno několik modulů zaměřených na didaktiku a technologie. A právě jedním z nich je předmět [Dětské robotické stavebnice](#), který má za úkol přinést

osvětou o možnostech nasazení robotů ve výuce mezi současné či budoucí učitele. V současné době se také na FPE ZČU v Plzni rozbíhá ESF projekt „Podpora pro rozvoj a zavádění ICT ve školách v Plzeňském kraji“, který kromě mnoha zajímavých moderních prvků v ICT zahrnuje i tematické kurzy robotiky v prostředí Lego Mindstorms. Je nabízen zdarma pro všechny školy a učitele, které se přihlásí.

Když se podíváme krátce na střední školy, tak velice aktivní je [SPŠ a VOŠ v Písku](#), které už několik let běží projekt Aplikovaná robotika. Z něho se jim podařilo nakoupit několik desítek stavebnic, včetně robotické ruky Lynxmotion. Na škole probíhá pravidelně kroužek robotiky a žáci se pravidelně zúčastňují různých soutěží.

### 2.2.3 SOUTĚŽE S ROBOTY

Ač většina škol využívá výhradně stavebnice LEGO Mindstorms, pro nás to nemusí být překážka. Protože většina z nich nijak neomezuje použitý typ robota, takže když postavíme kompletně výuku pouze na robotech od Lynxmotion, neměl by to být problém. Soutěží probíhá celá řada, pokusíme se zmínit ty nejdůležitější.

- **Kyber robot** – tuto soutěž pořádá Technická univerzita v Liberci a je určena pro žáky ZŠ i SŠ. Předmětem je libovolná kreace malých výukových robotů podle volné invence. Koná se každoročně. Odkaz na minulý ročník: <http://www.fm.tul.cz/kyber-robot-2011>.
- **Robotour** – velice zajímavá soutěž pro kohokoliv, spočívá v autonomním pohybu robota do zadaného bodu. Podmínkou je vezení nákladu (sud piva). Koná se letos již 6. rokem. Odkaz na oficiální web: <http://robotika.cz/competitions/robotour/cs>.
- **First Lego League** – asi největší soutěž pouze pro robotické stavebnice od firmy LEGO. Účastní se jí statisíce dětí z celého světa – to je velkou výhodou a výzvou pro učitele. Studentské projekty soutěží napřed v lokálním kole. Když je projekt úspěšný, žáci postupně postupují výš, až se dostanou do finálního kola, kde je zastoupen doslova celý svět. Úspěšný tým má šanci podívat se do celého světa (Miláně Evropy). Letošní finálové kolo se bude konat v německém Mannheimu 7-9. června. Oficiální web je zde: <http://www.firstlegoleague.org/>

## 2.3 JAK SE VYUČUJE ROBOTIKA VE SVĚTĚ

Podle předchozí kapitoly jsme viděli, že na tom není ČR se začleňováním robotiky tak špatně. České týmy ze základních, ale hlavně středních škol, mají poměrně slušné úspěchy i na mezinárodních soutěžích.

Situace v cizině je dosti podobná té naší. Z toho, co se nám podařilo dohledat, tak na většině škol se vyučuje robotika na stavebnicích od LEGa. Například ve Španělsku či Itálii je robotika poměrně často využívána při výuce.

### 2.3.1 MEZINÁRODNÍ PROJEKTY A SOUTĚŽE

#### FIRST LEGO LEAGUE

O této soutěži jsme se zmínili již v předchozí kapitole. Celosvětově rozšířená soutěž pro mladé konstruktéry z LEGa. Zajímavostí je to, že ji sponzoruje americká NASA.

#### TERECOP

Mezinárodní projekt programu Comenius. Spolupracuje v něm 6 evropských institucí včetně ČR zaměřených na vzdělávání učitelů. Cílem je vývoj a ověření kurzu, pomocí něhož budou učitelé souvisle připravováni na implementaci robotiky do výuky a začlenění do kurikul škol. Hlavní stránka je zde: <http://www.terecop.eu/>. Projekt byl bohužel ukončen v roce 2009, a ačkoli jsou stránky přeloženy do 6 jazyků, informace se dají najít pouze na těch anglických. Projekt sliboval mnoho, ale většina z velkých plánů nebyla naplněna.

#### EUROPEAN OPEN DAY ON EDUCATIONAL ROBOTICS

Zajímavá akce, která navazuje na TERECoP. Jednalo se několik workshopů, přednášek a konferencí, které probíhaly na kampusu univerzity ve španělské Pamploně a účastnily se ho zájemci z řady škol z několika zemí EU.

#### EUROPEAN CONFERENCE ON EDUCATIONAL ROBOTICS

Tato soutěž proběhla letos v dubnu poprvé. Soutěžili zde roboti týmů z celé EU v takzvaném Bootballu, což je obdoba fotbalu, pouze ho hrají roboti. Tato akce vypadá slibně. Uvidíme, jak to s ní bude do budoucna. Oficiální web je zde: <http://ecer12.pria.at/index.php?page=home>.

### 3 ROBOTICKÉ STAVEBNICE OD FIRMY LYNXMOTION

Předtím, než se vrhneme na praktické řešení a popis naší robotické ruky, bychom si měli představit firmu Lynxmotion a hlavně produkty, které vyrábí. Tyto stavebnice jsou modulární, tudíž si můžeme sestavit nejen již od výrobce předem navrhnuté roboty, ale i zkonstruovat roboty zcela nové. V tomto směru není zákazník nijak svázán. Bude tedy vhodný, abychom získali určitý přehled o tom, jaké máme možnosti. Toto se zvláště hodí pro nové zájemce, kteří nemají doposud žádné zkušenosti.

#### 3.1 O FIRMĚ LYNXMOTION

Jedná se o soukromou firmu, založenou a doposud vlastněnou jedním člověkem. Firma se všemi sklady i elektronickým obchodem se nachází v USA a byla založena v roce 1995. Od začátku založila svoji činnost na výrobě a prodeji výukových robotů. Oficiální internetové stránky jsou na této adrese: <http://www.lynxmotion.com/>. Zde se nachází jak informace o produktech, tak i stránky podpory a samotný obchod.

Firma provozuje ještě jeden server s adresou <http://www.lynxmotion.net/>. Jedná se o oficiální internetové diskusní fórum podporované výrobcem. Dnes má již velikou uživatelskou základnu a můžeme zde najít opravdu mnoho užitečných informací o robotech od této firmy. Současně zde mnoho uživatelů zveřejňuje i průvodce pro další rozšíření robotů či nový software. Zvláště noví uživatelé zde budou častými hosty. Nevýhodou je, že je pouze v anglickém jazyce.

##### 3.1.1 DISTRIBUTOR V ČR

Poloha firmy v USA nám ale přináší problém. Tím je doprava, clo, daň a poštovné. K ceně samotného robota si musíme připočítat i clo a daň (10% + 20%). Poštovné se pohybuje kolem 600 Kč. V poslední době naštěstí vznikl v ČR neoficiální distributor, který má stavebnice na skladě, případně je umí dovézt. Odkaz na tento obchod je následující: <http://shop.snailinstruments.com/>.

## 3.2 PŘEDSTAVENÍ STAVEBNIC

Nyní se pokusíme demonstrovat možnosti robotických stavebnic tohoto výrobce. Firma produkuje nejen samotné roboty, ale vytváří i vlastní software pro jednotlivé druhy stavebnic. To je velkou výhodou pro méně zkušené uživatele, kteří se nechtějí pouštět do samostatných projektů. Současně firma produkuje i několik univerzálních řídicích jednotek pro ovládání. O těch si řekneme něco více dále v textu. Napřed ale ke stavebnicím robotům.

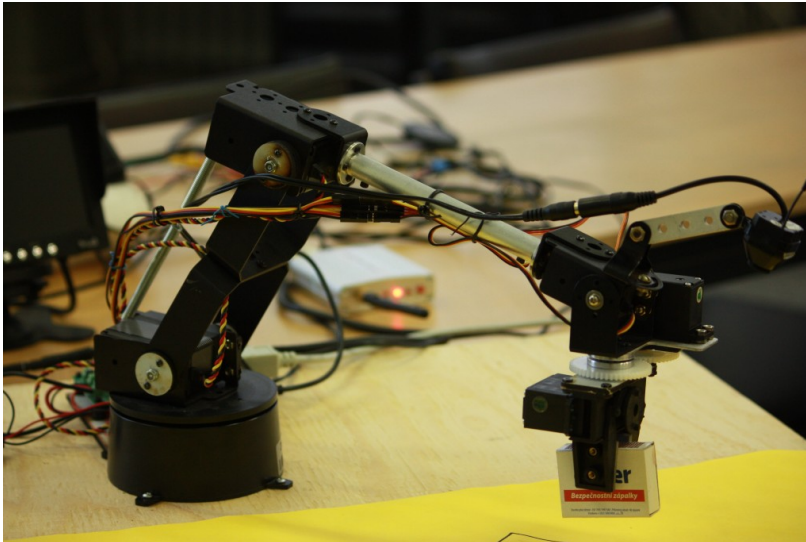
### 3.2.1 ROBOTICKÉ RUCE

Tato práce je věnována robotické ruce, konkrétně typu **AL5D**. Typů rukou je více a liší se délkou ramene (tedy jeho dosahem) a nosností – náš model má garantovanou nosnost cca 400g. Počet ovládaných kloubů je stejný – je jich 6. Říkáme, že má 6 stupňů volnosti. Každý typ ruky se prodává ve 3 verzích, uveďme si příklad na náš model.

- **AL5D s bezplatným softwarem a řídicí elektronikou SSC-32** – k chodu je nutný počítač se sériovým portem a spuštěným ovládacím softwarem, který je volně ke stažení od výrobce, cena je **340\$** (cca 7000 Kč)
- **AL5D se softwarem RIOS a řídicí elektronikou SSC-32** – k chodu také potřebujeme počítač se sériovým portem, ale v ceně je velice propracovaný software RIOS, který slouží ke komplexnímu programování robota včetně ovládání pomocí joypadu či joysticku, vhodné pro pokročilejší uživatele, cena je **390\$** (cca 8150 Kč) – toto je naše verze ruky
- **AL5D s řídicí elektronikou BotBoarduino** – verze pro ovládání robota bez počítače, obsahuje jinou řídicí elektroniku s BasicAtom interpreterem, kde jsou již nahráné programy pro základní práci. Umožňuje přímé připojení ovladače z konzole Sony Playstation 2. Obsahuje také rozhraní USB pro propojení s počítačem pro nahrání nových programů do procesoru, cena je **340\$** (cca 7000 Kč)

V našem případě jsme dokoupili odolný mechanismus pro rotaci zápěstí s cenou **45\$** (1000 Kč). Dále jsme robota doplnili o infračervený senzor na měření vzdálenosti a také bezdrátovou kameru pro přenos obrazu. Tu jsme upevnili na zápěstí (cca 1500 Kč).





Obrázek 1 Náš model již sestavené ruky

#### 3.2.2 „CHODÍCÍ“ ROBOTI

Zde výrobce nabízí velké množství robotů dělených podle počtu nohou, složitosti provedení a ceny. Začít můžeme u relativně levného a jednoduchého „chodícího“ robota jménem BRAT. Ten má pro pohon pouze 6 serv a cena u ovládání počítačem začíná na **5000Kč**. Máme možnost přikoupit stejného robota se složitější řídicí elektronikou, který se umí pohybovat po místnosti sám. Pomáhají mu k tomu infračervená a ultrazvuková čidla.

Postoupit můžeme k tzv. hexapodům, což jsou pavouci, kteří mají 6 nohou. Opět si můžeme vybrat z několika modelů podle složitosti, ceny a ovládání. Nejjednodušší sestává z 12 serv a stojí kolem **10000 Kč**. Ovládají se pomocí PS2 (Playstation2) ovladačů. Opravdu zajímavý je model, který se skládá z 25 servomotorů a stojí kolem **30000Kč**. Ovládání se děje opět pomocí PS2 ovladače. Za autonomní typ si musíme ještě připlatit.



Obrázek 2 Vlajková loď hexapodů<sup>1</sup>

---

<sup>1</sup> Převzato z: <http://www.lynxmotion.com/images/jpg/apod03.jpg>

### 3.2.3 PODVOZKOVÉ PLATFORMY

Můžeme si vybrat z několika odlišných podvozkových soustav. Nejlevnější podvozek je pásového typu. Samotný s ovládním pomocí RC soustavy stojí kolem **8000 Kč**. Ovládní probíhá jako u pásových vozidel, tj. každý pás má jeden motor. Podvozek je postaven tak, abychom na něj mohly připevnit další moduly od stejného výrobce. Tzn., že můžeme na tento podvozek přidat např. robotickou ruku a obojí ovládat buďto samostatně pomocí ovladačů, či případně počítačem přes jedinou řídicí elektroniku.

Dalším příkladem podvozku je klasický kolový s pohonem všech čtyřech kol. Zde si také můžeme vybrat, zde budeme chtít ovládní pomocí RC soupravy, počítačem či autonomní. Opět je postaven tak, že na něj můžeme připevnit robotickou ruku či jiné komponenty. Jelikož je řídicí elektronika stejná pro všechny produkty, můžeme na stejnou řídicí desku připojit jak podvozek, tak ruku. Budou si navzájem předávat údaje ze senzorů a podle toho se chovat. Záleží na programátorovi. Cena začíná na **10000 Kč**.



Obrázek 3 Ukázka vynikající modularity stavebnic Lynxmotion<sup>2</sup>

Zajímavostí může být hotové vozidlo pro robotické sumo zápasy, které jsou populární hlavně v Asii. Robot se pohybuje sám pomocí senzorů a paměti má již předem uložené programy pro tento účel napsané. Je konstruován na velkou pevnost. Cena začíná na **10000 Kč**.

---

<sup>2</sup> Převzato z: <http://www.lynxmotion.com/images/jpg/a4wd102.jpg>

### 3.2.4 ŘÍDÍCÍ ELEKTRONIKA

Máme 3 základní druhy řídicích obvodů:

- 1) Servo-kontroléry (SSC-32)
- 2) Interpretery jazyka Basic a Arduino<sup>3</sup> se servo-kontrolérem
- 3) Kontroléry pro řízení pohonů (elektromotorů)

#### **SERVO KONTROLÉR SSC-32**

Výrobce je přímo Lynxmotion a je založen na čipu Atmel ATmega 8. Umožňuje řízení 32 kanálů serv současně. Mimo to můžeme připojit až 4 vstupní senzory – digitální i analogové. V našem případě to bude infračervený senzor.

Jelikož neobsahuje žádný interpreter programovacího jazyka, vyžaduje ke svému chodu počítač, který se připojuje přes sériový port. Tzn., že robot je ovládán přes tento kontrolér pomocí softwaru v počítači. Je to současně nejlevnější a nejschůdnější cesta k řízení robotů.

Tento kontrolér je použit i ve stavebnici naší robotické ruky a jeho cena je cca **800 Kč**.

#### **BOTBOARDUINO**

Jedná se o jednodušší interpreter programovacího jazyka Arduino. Nejlepší poměr cena – výkon. Slouží pro autonomní řízení robotů. Obsahuje jak 18 kanálů pro regulaci serv a další kanály na vstupy ze senzorů. Je použit například v levnějších hexapodech. Výhodou je přítomnost PS2 portu – lze použít PS2 ovladače. Cena je cca **800 Kč**.

#### **BOT BOARD II + BASIC ATOM PRO 28 PIN**

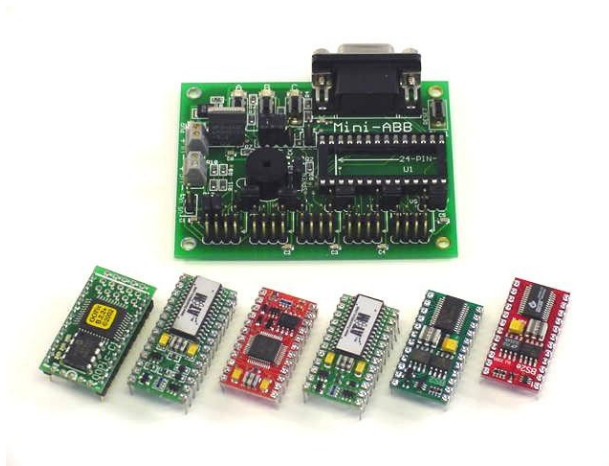
Nejkomplexnější řídicí jednotka od Lynxmotion. Obsahuje základní desku s 20 kanály pro přímé řízení serv, umožňuje připojení mnoha různých periférií, včetně PS2 ovladače. Umožňuje také propojení s kontrolérem SSC-32 – můžeme tak navýšit počet řízených serv. To je např. potřeba u nejsložitějšího hexapodu, který vyžaduje 25 kanálů.

---

<sup>3</sup> Open-source programovací jazyk podobný jazyku Basic, uzpůsoben pro programování robotiky a pohonů, odkaz na projekt zde: <http://www.arduino.cc/>

Do patice se osazují mikroprocesory BASIC Atom Pro 28. To jsou interpretery programovacího jazyka Basic. Zvládají až 100 000 příkazů za vteřinu a obsahují 32 KB uživatelské paměti.

Tyto procesory umožňují komplexní řízení všech autonomních robotů. Cena je **500 Kč** za desku + **1200 Kč** za procesor Basic Atom Pro.



Obrázek 4 Řídící jednotka s procesory Basic Atom<sup>4</sup>

#### **ŘÍZENÍ POHONŮ**

To jsou standardní regulátory proudu pro modelářské RC soupravy. Používají se pro pohony pásových a kolových platforem. Ceny se pohybují kolem **1200 Kč**.

#### **SENZORY**

Výrobce nabízí množství doplňkových senzorů. Díky nim se poté může robot svým způsobem chovat samostatně.

Za zmínku stojí určitě infračervený senzor Sharp GP2D12, kterým jsme vybavili i naši ruku. Ten umožňuje rozpoznat vzdálenost předmětů od senzoru ve vzdálenosti od 3 do 50 cm. Díky němu může ruka pomocí podmínek sama reagovat na kolize apod.

Za zmínku stojí také např. tlakové čidlo, případně také dvouosý akcelerometr, který detekuje orientaci robota v prostoru (osy x a y).

---

<sup>4</sup> Převzato z: <http://www.lynxmotion.com/images/jpg/abb03.jpg>

## 4 SESTAVENÍ A OŽIVENÍ ROBOTICKÉ RUKY AL5D

Nezáleží na tom, zda si objednáme stavebnici ruky z USA přímo od výrobce či z obchodu českého distributora, dorazí nám vždy v rozloženém stavu a je na nás, abychom ji složili. Samotné složení není pro průměrně šikovného jedince nic složitého, ale vyžaduje přeci jen jistou dávku zručnosti a zkušeností. Takže bychom nedoporučovali, aby ji sestavovali žáci úplně celou sami. Je to poměrně složité zařízení a mnoho součástek se může neopatrným zacházením zničit či poškodit a sehnání stejného dílu se může prodražit.

V pokročilejších ročnících na 2. stupni ZŠ by šlo robota předsestavit do určitých modulů, které by pak studenti sestavili dohromady, případně se podíleli na konstrukci některých rozšiřujících prvků. O úplném sestavení robota žáky můžeme uvažovat na SŠ, kde se předpokládá již jistá úroveň znalostí a zkušeností. Ale pravdou je, že to všechno záleží na učiteli a jeho studentech.

Když budu mluvit z vlastní zkušenosti, tak konkrétně náš model robotické ruky jsme sestavovali kolem **3 až 4 hodin**.



Obrázek 5 Všechny součásti robotické ruky po vybalení z krabice<sup>5</sup>

<sup>5</sup> Převzato z <http://www.josefnav.cz/images/LynxArm/01.JPG>

### 4.1 TIPY PRO SESTAVENÍ

Na obrázku č.5 nahoře vidíme všechny součásti robotické ruky. Každý dílčí celek je pro větší přehlednost zabalen v samostatném pytlíku.

Balení se skládá z:

- 5 kusů servomotorů Hitec
- CD-ROM se softwarem RIOS
- Stejnosměrný zdroj na pro silovou část robota (6V – 2A)
- Sériový kabel
- Balíčky s:
  - Elektronikou SSC-32
  - Samotné rameno a jeho součásti
  - Základna pro rotaci
  - Ostatní elektrické součásti a základna pro řídicí elektroniku
  - Odolné otáčecí zápěstí včetně 1 servomotoru

Toto jsou součástky pro stavbu celého robota. Budeme ještě potřebovat malý **plochý a křížový šroubovák** a sadu malých **imbusů** (2 mm). Jako základnu pro celou ruku jsme dle doporučení výrobce zvolili **MDF desku** o rozměrech **64 x 64 cm**. Ta je zvolena tak, aby z ní ruka ani při maximálním natažení nevyjela ven. Zároveň jsme na desku v pracovním dosahu ruky připevnili pás žluté čtvrtky. Ten jen výměnný a bude sloužit jako plocha, na kterou zakreslíme přesné umístění předmětů, se kterými bude ruka pracovat.

Řídicí elektronika je napájena 9V baterií. Toto se nám ale příliš neosvědčilo, protože se příliš brzy vybila. Proto jsme ještě dokoupili univerzální stejnosměrný napájecí zdroj pro **9V** – výstupní proud stačí cca **500mA**.

Tímto máme tedy vše potřebné pro správnou funkci robota a můžeme se pustit do samotné stavby. Tou se ale zabývat nebudeme, neboť to není v možnostech této práce. Navíc výrobce nabízí opravdu přehledný návod na sestavení na svých internetových stránkách, a ačkoli jsou v anglickém jazyce, nebude to vadit ani těm, kteří tento jazyk

neznají. Návodů jsou totiž opatřeny mnoha názornými ilustracemi a fotografiemi. Stačí postupovat dle návodu a nemůžeme narazit na problém. Uvedeme jen hypertextové odkazy na stránky s manuály přesně pro náš typ ruky:

- 1) **Sestavení rotační základny, zapojení řídicí elektroniky a její oživení:**  
<http://www.lynxmotion.com/images/html/build40d.htm>
- 2) **Sestavení všech částí ruky, její instalace na základnu, zapojení serv do řídicí desky, uvedení do chodu a testování funkčnosti komunikace s PC:**  
<http://www.lynxmotion.com/images/html/build143b.htm>
- 3) **Sestavení, instalace a zapojení odolného otočného zápěstí:**  
<http://www.lynxmotion.com/images/html/build145.htm>
- 4) **Základní uživatelský manuál v anglickém jazyce pro aplikaci RIOS:**  
<http://www.lynxmotion.com/images/data/rios106h.pdf>

### 4.1.1 SPRÁVNÉ VEDENÍ KABELÁŽE

#### HLAVNÍ SVAZEK

Správný způsob vedení kabeláže servomotorů a senzorů výrobce příliš nezmiňuje. To je jedna z mála věcí, která se dá na manuálu kritizovat. Kabely dodané ke stavebnici jsou rozměrově správně, některé by mohly být i delší. Proto je potřeba je řádně upevnit a vést je správným způsobem. Jinak nám hrozí, že se při určitých pohybech ruky v krajních polohách kabely vytrhnou a tím by mohly poškodit i desku s řídicí elektronikou.

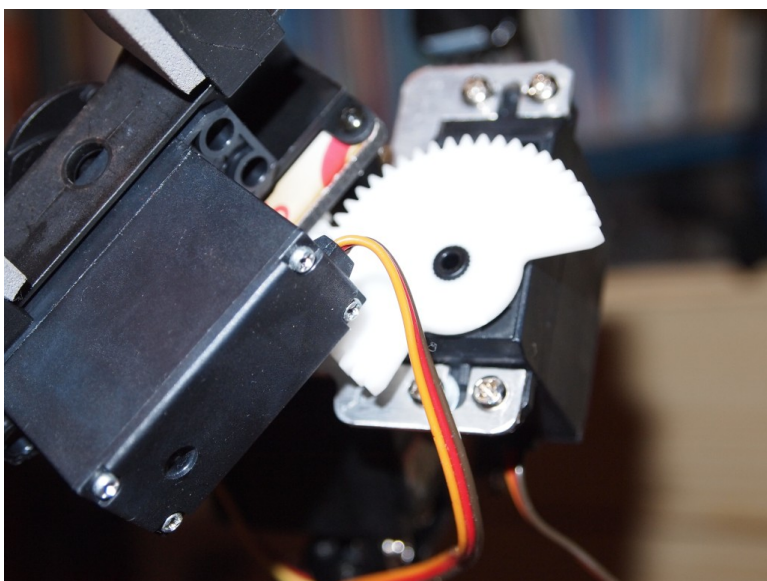
Chce to trochu experimentování a logické úvahy. Když si kabely roztřídíme podle délky a provedeme jejich správné zapojení, měli bychom je svázat do jednoho svazku. Po několika pokusech jsme zjistili, že bude nejlepší vést svazek po pravé straně ramene směrem od elektroniky. To je jediná poloha, kdy ruka o sebe kabely ani v krajních polohách nezatrhávala. Lépe to bude demonstrovat obrázek níže.

Na něm vidíme nejvíce problematické místo z hlediska možných zaseknutí. Toto místo je třeba hlídat. V krajních polohách mají kabely tendenci zaseknout se o hlavní servo ramene.



Obrázek 6 Správné vedení kabelů hlavního svazku

### **KABEL OTOČNÉHO ZÁPĚSTÍ**



Obrázek 7 Problematické místo na zápěstí

Na obrázku nahoře vidíme situaci, kdy se kabel serva od svírací rukojeti blíží převodům na otáčení zápěstí. Hrozí zde riziko toho, že když je kabel moc volný, zachytne se do převodů a zablokuje další pohyb. Hrozí tedy poškození serva, případně i kabelu od převodových kol. Museli jsme kabel napnout přesně na hranici, aby měl vůli na otáčení a zároveň aby se při otáčení nedostal do převodů. Poté jsme ho pevně upevnili k celé ruce. Toto mohl výrobce vyřešit lépe.



## 4.2 UPEVNĚNÍ A ZAPOJENÍ SENZORŮ A KAMERY

Jelikož jsme chtěli demonstrovat co možná nejlépe co nejvíce možností robotické ruky, spolu s ní jsme zakoupili bezdrátovou kameru a infračervený senzor pro měření vzdálenosti. Díky těmto senzorům se ruka mnohem více přiblíží reálným provozním podmínkám robota a bude do jisté míry i autonomní. Navíc z didaktického hlediska nabízí mnohem více přesahů do jiných předmětů – viz kapitola o RVP na straně 3.

Vznikl nám ale problém kam a jak senzory upevnit. Výrobce nabízí univerzální držák na senzory, ale ten nebyl zrovna dostupný. Proto jsme si vyrobili vlastní z běžně dostupných dílů stavebnice Merkur. Ten jsme upevnili na ruku ještě před zápěstím, aby se senzory nepohybovaly spolu s ním, ale aby snímaly stále pevný bod.

### 4.2.1 INFRAČERVENÝ SENZOR SHARP GP2Y0A41

Jednoduchý senzor pro určení vzdálenosti. Využití v našem případě je pro detekci překážek či jednoduché povely. Senzor je dle specifikace výrobce schopen rozpoznat vzdálenost od **4 do 30 cm**. Reálné hodnoty se ale liší, což jsme si i ověřili jednoduchým měřením, které bude dále v textu.

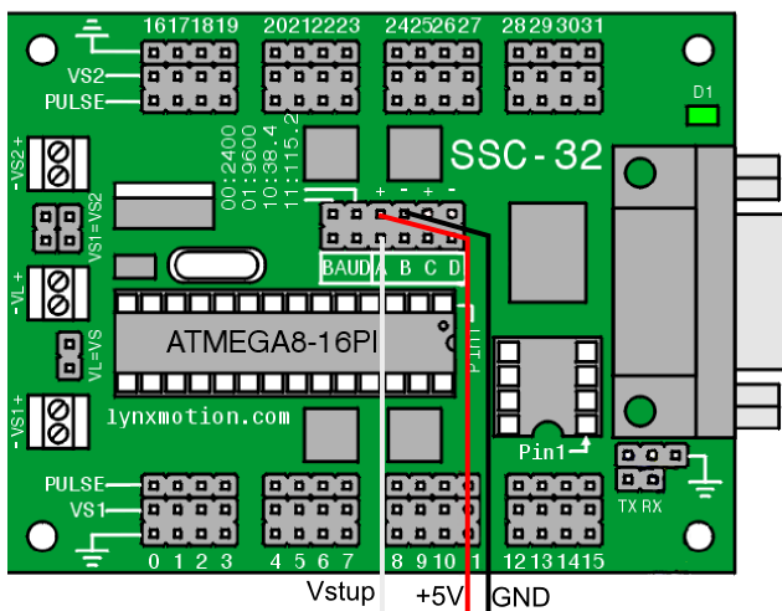
Tento typ senzoru nahradil původní **Sharp GP2D12**, který se už nevyrábí a v Evropě je problematické ho sehnat. Zapojení mají oba stejné, napájení také. Liší se ale ve vzdálenosti, kterou jsou schopny rozlišit. Původní senzor rozlišoval vzdálenost od **10 do 80 cm**.

Princip fungování je u obou jednoduchý. Senzor napájíme 5V a do řídicí elektroniky jde zpět analogová hodnota napětí úměrná vzdálenosti předmětu od senzoru. A právě tuto hodnotu použijeme pro vyhodnocení dalšího chování robota na základě podmínek (IF-ELSE a další). Řídicí deska SSC-32 nabízí použití 4 takovéto vstupy.

Pro naše potřeby bylo třeba vyrobit vlastní konektor pro připojení. Máme 3 kabely ze senzoru:

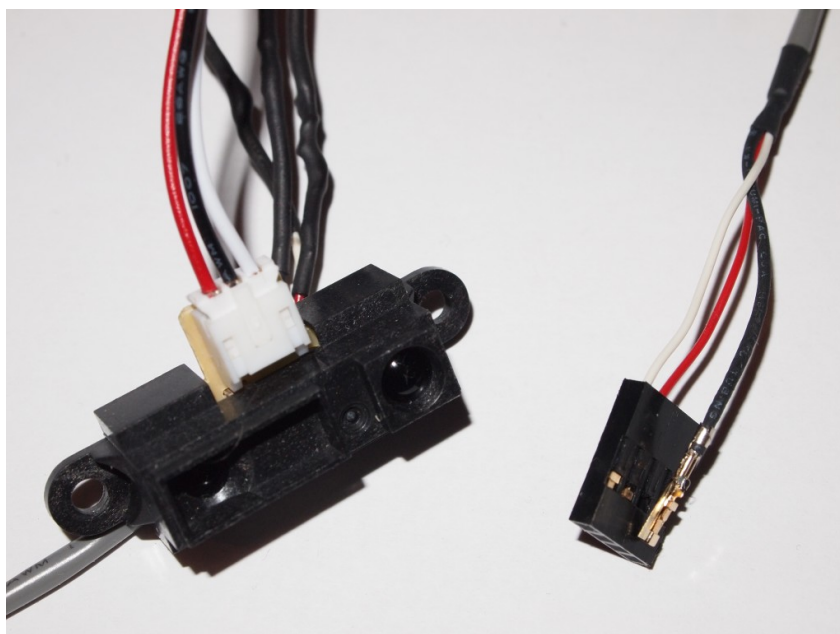
- Levý – **bílý** – signálový kabel
- Prostřední – **černý** – GND (zem)
- Pravý – **červený** - +5V stejnosměrné napětí

Pro úpravy jsme použili nevyužitý audio kabel pro zvukovou kartu z CD-ROM. Přehodily jsme červený a bílý pin na kraj a černý kabel je vyveden samostatně. Poté připojíme kabel na desku robota. Musíme dát pozor na správnou polaritu.



Obrázek 8 Schéma zapojení IR senzoru

Hotový kabel pro připojení čidla by měl vypadat nějak takto:

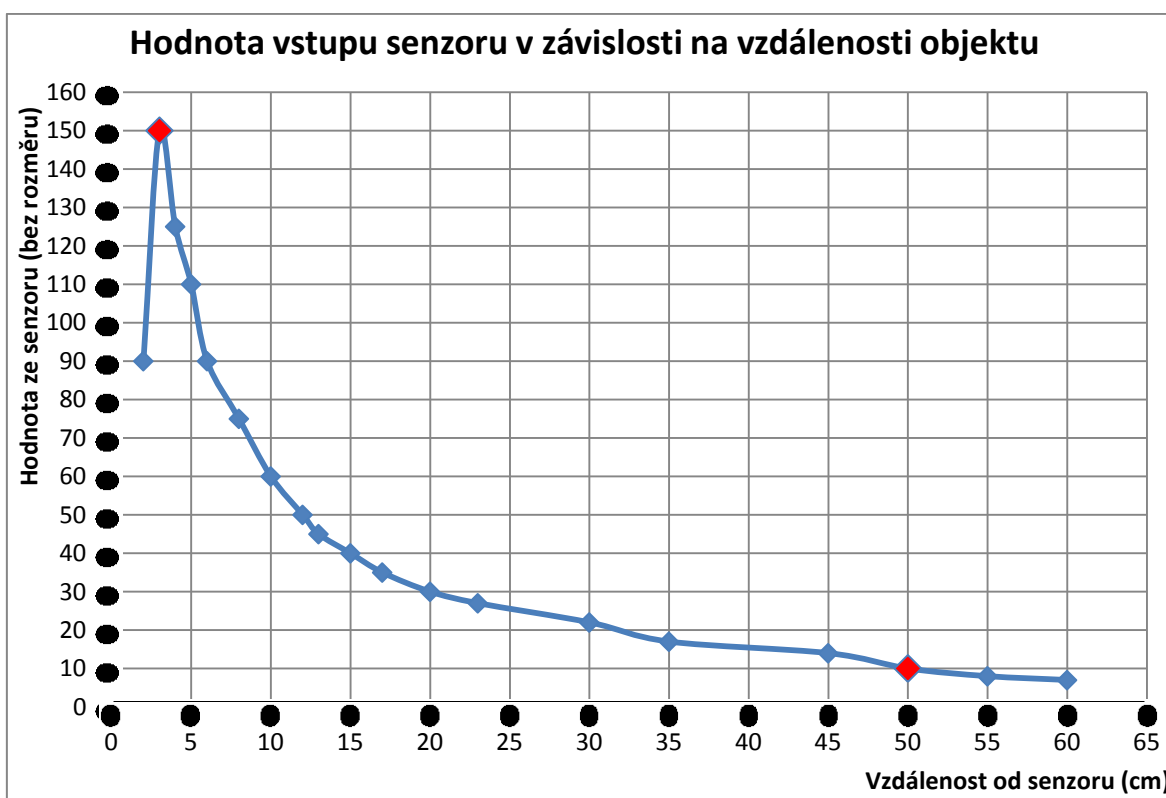


Obrázek 9 IR senzor a vyrobený kabel pro připojení

Samotné práci s IR senzorem se budeme věnovat v kapitole věnované softwaru RIOS.

### MĚŘENÍ SKUTEČNÝCH HODNOT IR SENZORU SHARP

Měření probíhalo jednoduše. Senzor jsme upevnili do stabilní polohy a přibližovali jsme k němu předmět tak velký, aby pokryl úhel vyřazování senzoru (postačila velikost A4). Měřili jsme vzdálenost objektu a současně jsme ze softwaru RIOS odečítali hodnoty, které čip do elektroniky posílá. Ty by měly být úměrné vzdálenosti a rozlišovací schopnost by měla odpovídat specifikacím, tj. 4 – 30 cm. Přesvědčíme se, jak tomu odpovídá skutečnost.



Obrázek 10 Naměřené hodnoty z IR senzoru

Z grafu vidíme, že naměřené hodnoty téměř odpovídají skutečnosti. Hraniční body jsou označeny červenými body. Nejmenší vzdálenost je **3 cm** při hodnotě **150**. Maximální použitelná vzdálenost měření je **50 cm**, kdy čidlo posílá hodnotu **10**. Znamená to, že celková reakční vzdálenost čidla je téměř 50 cm, což je pozitivní zjištění. Dále je již rozlišovací schopnost malá a nedoporučujeme se na ni spoléhat.

Z grafu také jasně vidíme, že charakteristika není lineární. Čím více se přibližujeme objektem k čidlu, tím více je charakteristika strmější. S tím musíme počítat.

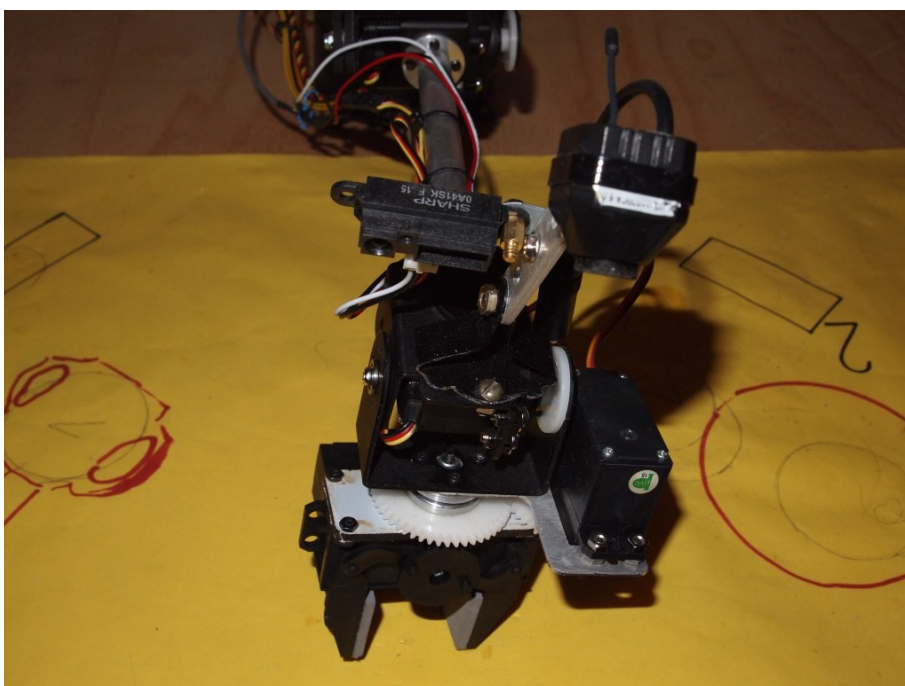
Problémy může způsobovat vzdálenost větší než 50 cm. Zde se na výstupu objevují cyklicky se střídající hodnoty od 1 do 10. To může způsobovat problémy, pokud budeme v programu počítat s hodnotami kolem 10. Proto s tím musíme počítat a zahrnout to do případné konstrukce programu. Vytvořený graf může sloužit pro přesný odečet hodnoty a vzdálenosti, kterou budeme potřebovat.

### 4.2.2 BEZDRÁTOVÁ KAMERA

Jedná se o miniaturní bezdrátovou barevnou kameru, která přenáší pouze obraz v pásmu 2,4 GHz. Výrobce neznáme. Je to klasický levný model z Číny s cenou okolo 400 Kč. Připevnili jsme ji přímo nad IR čidlo. Tato sestava nám bude tvořit „oči“ naší robotické ruky.

Kamera byla použita v parkovací soupravě pro automobily. Součástí sady byl tedy i přijímač signálu kamery a malý LCD televizor, do kterého se bude přenášet signál z přijímače. To celé se musí napájet z přiloženého 9V adaptéru. Takovéto sety se dají levně pořídit v čínských internetových obchodech s cenou kolem 1500 Kč.

Kameru využijeme na úlohy, kde budeme chtít simulovat policejního robota například na zneškodňování bomb. Robot bude umístěn jinde a my ho budeme muset ovládat na dálku pouze pomocí obrazu z kamery a IR senzoru.



Obrázek 11 Celá sestava senzoru a kamery upevněná na ruce

### 4.3 SPOJENÍ S POČÍTAČEM

Řídící deska SSC-32 používá ke spojení s počítačem výhradně sériový port RS232. To nám dnes přináší nemalé problémy. Moderní počítače totiž již sériový port nemají. Takže mohou nastat 2 situace. Buďto máme starý počítač s tímto rozhraním a potom bude vše bez problémů. Anebo máme moderní počítač a to musíme použít redukci USB na RS232. A právě tyto redukce občas způsobují výpadky či úplnou nefunkčnost.

Testovali jsme 3 redukce od 3 výrobců. Bez potíží fungovala pouze 1. Další fungovala s občasnými výpadky a poslední nekomunikovala s SSC-3 vůbec. Když jsme pátrali po příčině, zjistili jsme, že čipy těchto převodníků vyrábějí pouze 2 firmy – FTDI a OMRON. Většina běžně dostupných redukcí je vybavena čipy od FTDI. Zajímavé je to, že čipy od FTDI mají úplně jiné rozhraní ovladačů a připojení do systému než OMRON. A my se podíváme na 2 nastavení ovladačů, které nám fungovalo nejlépe.

#### 4.3.1 REDUKCE OD FTDI



Obrázek 12 Nefunkční redukce

To, co vidíme nahoře, je redukce s blíže neidentifikovaným čipem od FTDI, která nefungovala. V systému se přihlásila, vše jsme nastavili správně, ale SSC-32 vůbec s počítačem nekomunikovala. Chyba tedy bude v redukci. Doporučujeme vyhnout se tomuto produktu.

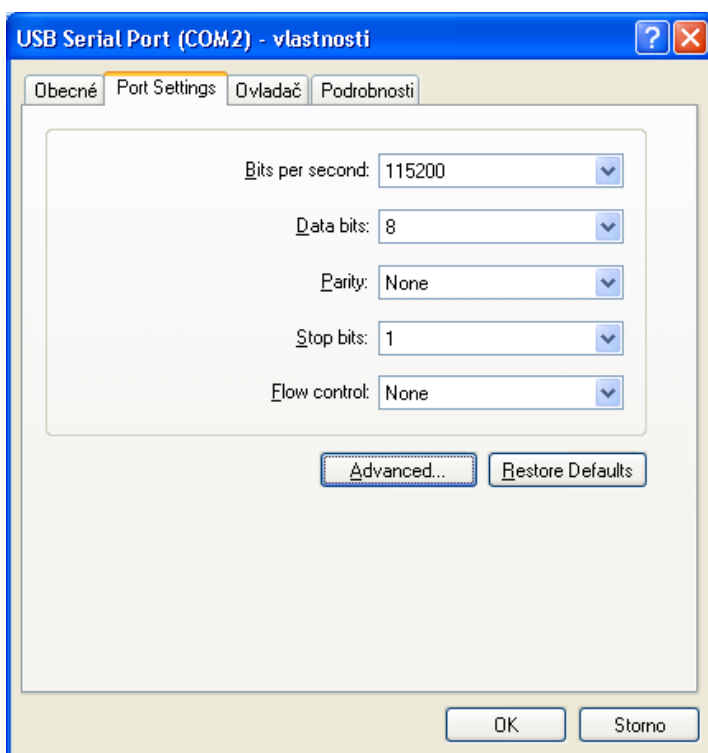


Obrázek 13 Tato redukce fungovala bez chyb

Zajímavé je, že 2 produkty s čipy stejného výrobce fungují odlišně. Toto byla asi nejlepší redukce, se kterou jsme se setkali. Po počátečním nastavení fungovala trvale bez chyb. Výrobce bohužel neznáme, neboť ji už používáme dlouho. Ale patřila mezi ty dražší, což může být také vodítko.

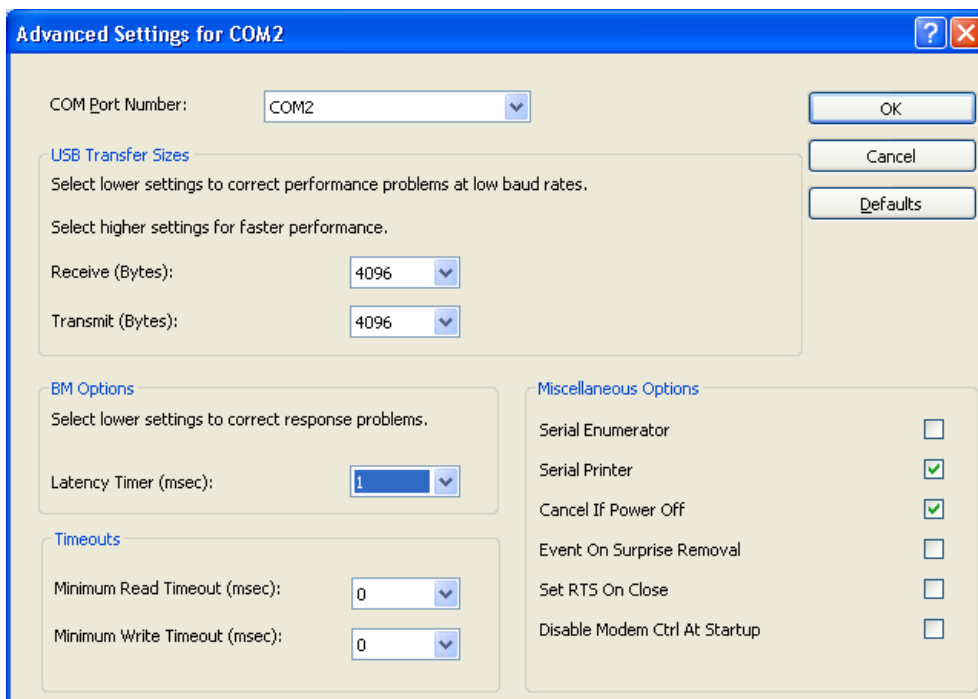
### SPRÁVNÉ NASTAVENÍ OVLADAČŮ

Když používáme operační systém Windows XP a novější, ovladače pro redukci se nainstalují samy. My ale musíme některé volby změnit. Přejdeme tedy do **Správce zařízení** a v něm vybereme. Ve složce **Porty (COM a LPT)** vybereme ovladač redukce (v našem případě **USB seriál port (COM2)**) a provedeme nastavení dle obrázků.



Obrázek 14 Nastavení přenosu pro FTDI

Nejdůležitější je **rychlost** zařízení v bitech za sekundu. Dle našich zkušeností nejlépe funguje rychlost pouze 115200 B/s. A ač výrobce píše v manuálu, že máme nastavit 9600, nám to takto nefungovalo. Dále musíme nastavit **Data bity na 8** a **Stop bit na 1**. Parita a řízení toku dat je vybráno na **none**. Poté klepneme na záložku **Advanced...** (Pokročilí).



Obrázek 15 Nastavení pokročilých možností ovladače

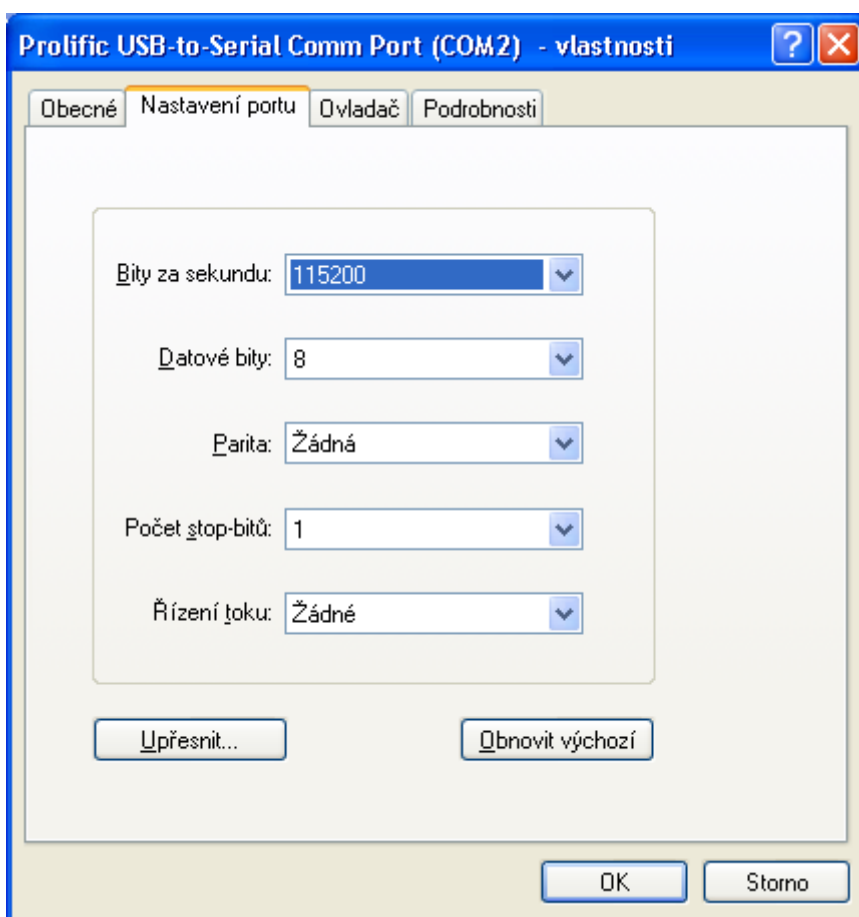
V pokročilých nastaveních je důležitá Latence (časování impulzů). Implicitně je nastavena velká a my ji snížíme na 1ms. Bez toho vznikaly chyby a prodlevy v komunikaci. Ostatní můžeme nechat tak jak je či jak vidíme na obrázku. Kdyby vznikaly ještě nějaké problémy, můžeme zkusit změnit velikost paketů – zmenšit je ze 4096 bytů. Po tomto nastavení by již vše mělo fungovat bez problémů. Když ne, doporučujeme zkusit jinou redukci.

#### 4.3.2 REDUKCE OD OMRON



Obrázek 16 Redukce od Omron electronics

Dle našich zkušeností doporučujeme tuto redukci. Patří mezi ty nejspolehlivější. Používá se dokonce pro komunikaci s routery od firmy Cisco, což je záruka kvality. V systému se navíc hlásí pouze jako jedno zařízení, v našem zařízení **Prolific USB-to-Serial Comm Port**. Mnohem lépe simuluje sériový port. Opět ale musíme provést důležité nastavení rychlosti.



Obrázek 17 Nastavení rychlosti přenosu

Tentokrát ale již nemusíme nastavovat další položky na kartě upřesnit. Spojení by teď mělo bez problémů fungovat. Zapojíme napájení robota, zapneme hlavní vypínače a otestujeme si spojení.

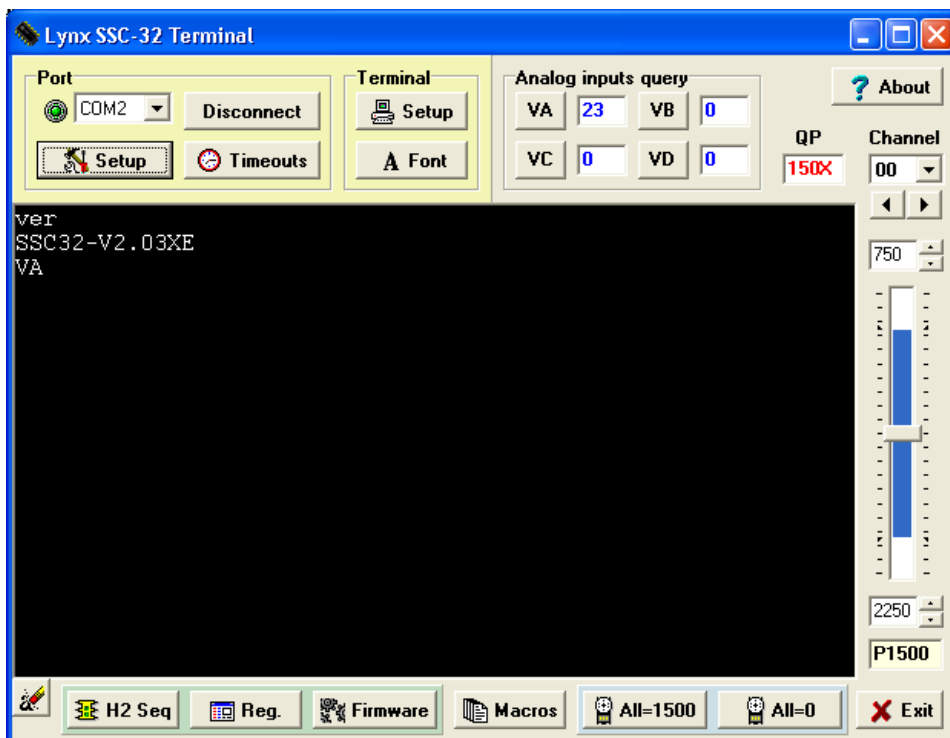
#### 4.3.3 TEST SPOJENÍ POMOCÍ APLIKACE LYNXTERM

Pokud jsme všechny parametry natavili správně a máme k dispozici tu správnou redukci či sériový port, měli bychom nejdříve ze všeho otestovat komunikaci. K tomu budeme potřebovat jednoduchý program **LynxTerm**, který nabízí výrobce zdarma ke



stažení na svých internetových stránkách. Odkaz je zde: <http://www.lynxmotion.com/p-567-free-download-lynxterm.aspx>.

Tento program slouží k otestování hlavních funkcí všech robotických stavebnic výrobce. Obsahuje jak konzoli pro zadávání textových příkazů pro komunikaci s jednotkou, tak i několik ovládacích prvků pro samostatná serva. Můžeme zde také aktualizovat software řídicí jednotky (SSC-32) či spouštět uložená makra.



Obrázek 18 Prostředí programu LynxTerm

Na obrázku výše vidíme hlavní okno programu LynxTerm. Žlutá část vlevo nahoře označuje nastavení portu pro komunikaci (v našem případě COM2). Pod položkou **Setup** se skrývá nastavení rychlosti komunikace a parity dat. Zde nastavíme stejné hodnoty, jako jsou u portu COM ve správci zařízení Windows (**115200 baudů**). Kliknutím na tlačítko **Connect** se program spojí s řídicí deskou.

Když zadáme do terminálu příkaz **VER**, měl by se nám zobrazit text, jaký můžeme vidět výše (SSC32-V2.03XE). Je to verze firmware na řídicí desce. Pokud se tento text nezobrazil, nastala chyba v komunikaci. Doporučujeme znovu projít nastavení portů, hlavně rychlost. Musí být shodná s ovladačem ve Správci zařízení!

Vedle žlutého pole nahoře vidíme hodnoty z analogových vstupů. Na položce VA (vstup A) vidíme po stisku tlačítka aktuální hodnotu z IR senzoru, který je připojen.

Napravo od terminálu je výběrové pole pro samostatné kanály serv. Pod ním se nachází posuvník, kterým můžeme v reálném čase měnit polohy serv.

Dole jsou tlačítka pro uvedení serv do nulové polohy ( $All=0$ ) či do polohy v  $90^\circ$  ( $All=1500$ ). Další volby nás zajímat nemusí, neboť je běžný uživatel nepotřebuje. Pro zkušené uživatele zmíním, že tlačítko *Reg* spouští program pro přímý přístup do registrů procesoru ATMEL na řídicí jednotce.

Nyní jsme si otestovali funkčnost komunikace počítače s robotem. Abychom si usnadnili práci s robotem a zároveň ho více přiblížili reálným podmínkám, rozhodli jsme se pro výrobu bezdrátové Bluetooth redukce na sériový port.

### 4.4 KONSTRUKCE BLUETOOTH BEZDRÁTOVÉ REDUKCE

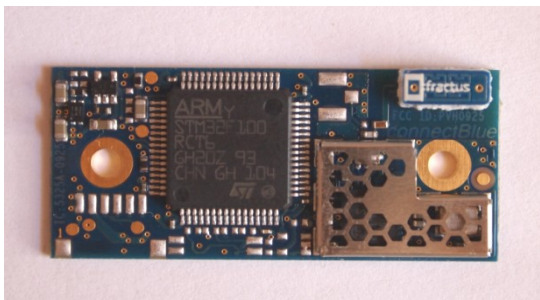
Spolu s příchodem bezdrátového rozhraní Bluetooth přišly i různé redukce, které ho využívají pro bezdrátový přenos dat i u rozhraní, kde to bylo dříve nemyslitelné. Mezi ně patří i sériový port. A právě u robotických stavebnic se toto použití přímo nabízí, neboť většina z nich využívá právě toto zařízení. A jelikož je poměrně staré a má i jistá HW omezení (např. délka kabelu), bezdrátový přenos je mnohem výhodnější a v mnoha případech i snazší.

K realizaci tohoto spojení potřebujeme:

- USB Bluetooth adaptér (cena je cca 200 Kč)
- Redukci Bluetooth na sériový port (cca 800 Kč)

S USB adaptérem není sebemenší problém. Seženeme ho v každém obchodě s elektronikou. S redukcí na sériový port už to bude horší. Zde máme v podstatě pouze 2 možnosti. Přímou na stránkách podpory Lynxmotion můžeme najít doporučení a návod na konkrétní typ adaptéru. Jedná se o **BlueSMiRF RP-SMA**. V USA je běžně dostupný v ceně kolem 1300 Kč. Konstrukce je velice snadná a stačí se řídit podle návodu, který je zde: <http://www.lynxmotion.com/images/html/build117.htm>. Problém je ale ten, že v Evropě je obtížné ho pořídit, musíme proto použít jiný.

V ČR jde zakoupit modul **connectBlue OBS410i** v ceně kolem 700 Kč. Prodává ho např. firma [Spezial electronic](#). Problém je ten, že modul přijde bez dokumentace a samotné zapojení je také mnohem složitější, než u dříve jmenovaného modulu, proto se stavbě budeme věnovat.



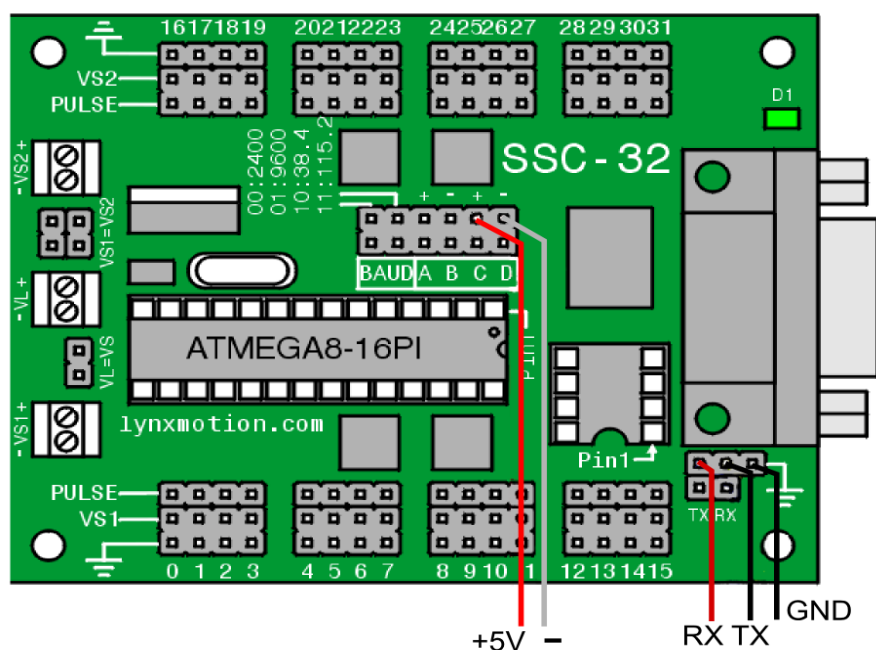
Obrázek 20 modul connectBlue OBS410i



Obrázek 19 USB Bluetooth adaptér

Máme tedy oba adaptéry. Podle dostupného [datasheetu](#) k modulu si musíme sestavit zapojení, které bude správně komunikovat s řídicí deskou SSC-32. Když se podíváme na kontakty na ní, zjistíme, že je dostupných několik pinů pro napájení dalších periférií, hlavně senzorů. Je dostupné napětí **5V DC**. Máme tedy modul čím napájet.

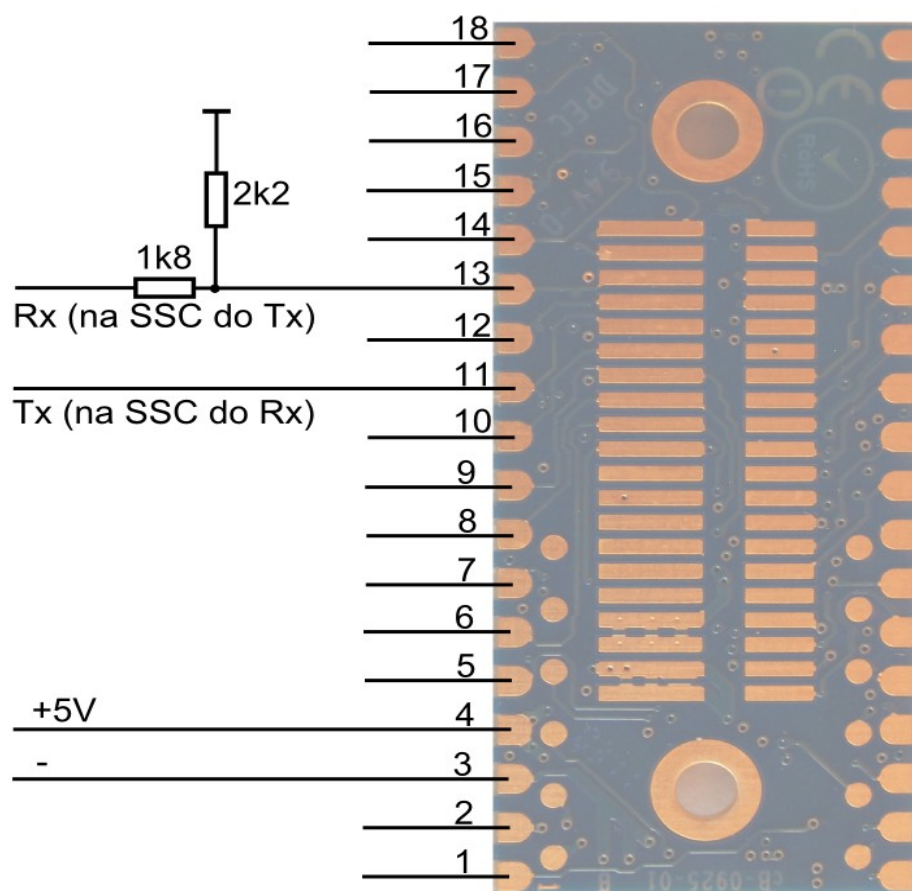
Dále vidíme na desce, že obsahuje další piny, které vedou ze sériového portu na desce, konkrétně jsou to piny **Rx** a **Tx**. To jsou základní komunikační kanály sériového portu. Žádné jiné nepotřebujeme.



Obrázek 21 Schéma připojení na řídicí desce SSC-32

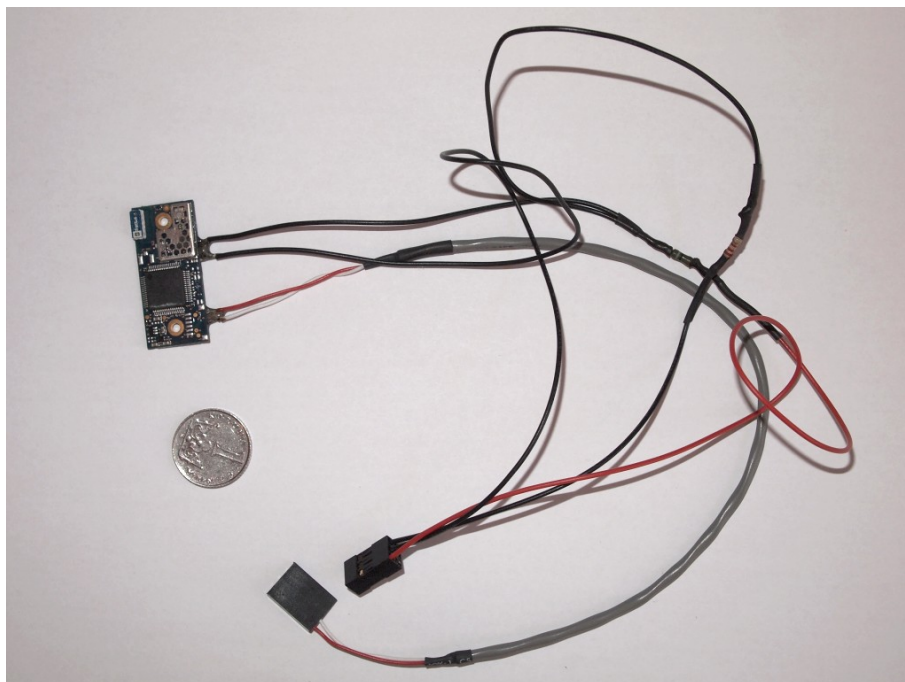
K celkovému sestavení funkčního zapojení budeme potřebovat:

- **2 rezistory** – 1k8, 2k2
- **2 mini DIN konektory** – v našem případě jsme použili staré kabely sloužící pro propojení zvukové karty s CD mechanikou
- Smršťovací bužírku
- Pájku, izolační pásku atd.




Obrázek 22 Schéma zapojení kabelů od modulu connectBlue

Správné konektory pro zapojení jsme si zjistili v dokumentaci. Použili jsme 2 kabely. Jeden **2-pinový** na napájení a jeden **3-pinový** pro data a společnou zem. Při pájení musíme postupovat rychle a za menší teploty, abychom nepoškodili obvody na modulu. Doporučujeme použít kvalitní mikropájku. Všechny spoje musíme důkladně zaizolovat. Po provedení konektory připojíme na desku SSC-32 a nastavíme samotnou komunikaci na počítači. Modul connectBlue by měl být již správně nastaven z výroby.



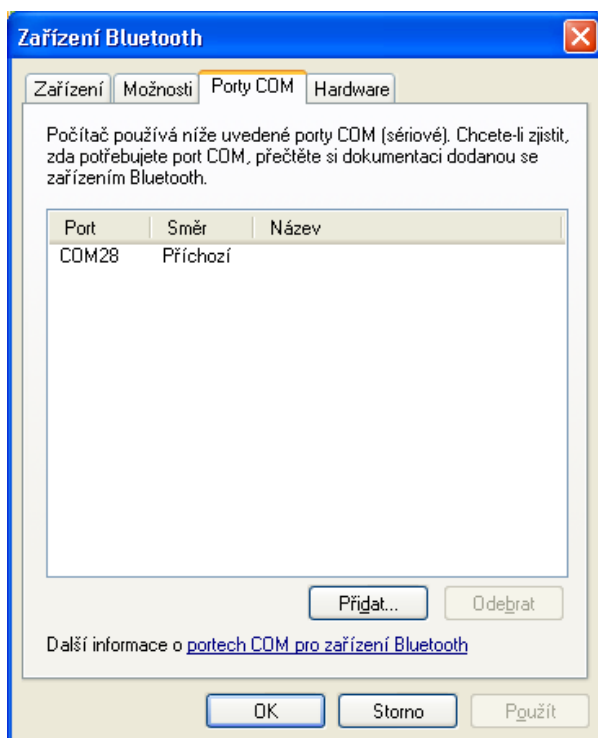
Obrázek 23 Kompletní modul připravený k zapojení

### 4.4.1 NASTAVENÍ OVLADAČŮ PRO USB BLUETOOTH ADAPTÉR

Když připojíme adaptér do portu USB, systémy Windows od verze XP by se měly ovladače najít samy. Po správné instalaci vidíme modrou ikonku bluetooth  v oznamovací oblasti systému vpravo dole. Když ji nevidíme, nastavení najdeme také v **Ovládacích panelech** pod položkou **Zařízení Bluetooth**.

1. Nejprve musíme zařízení spárovat. Klikneme na ikonku a vybereme **Přidat zařízení**.
2. Systém nyní prohledá okolí. Jestli je náš adaptér správně připojený a zapojený, vysílá signál a my bychom ho měli vidět.
3. Adaptér se bude jmenovat **connectBlue Serial Port Adapter**. Připojíme se tedy k němu. Systém nás vyzve ke spárování. Zde záleží na výrobci, jak je modul nastaven. Některé mají zabezpečení vypnuté. Tudíž ho nepoužijeme i my. Ale některé ho mají zapnuté a používají kód **1234**, který musíme zadat do políčka **Použít klíč nalezený v dokumentaci**.
4. Poté musíme zkontrolovat svázání Bluetooth spojení s COM portech, které vytváří sériovou linku. Klikneme tedy v ovládacích panelech na položku **Zařízení Bluetooth**. Poté zvolíme naše zařízení a vybereme kartu **Porty**

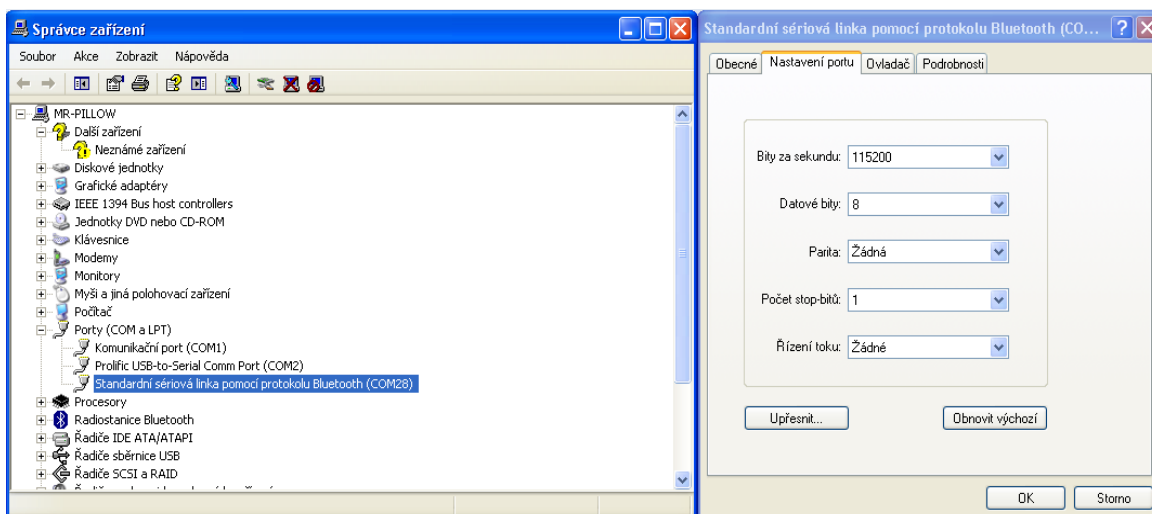
**COM.** Zde uvidíme příslušný port pro komunikaci. Ten musíme vybrat i v softwaru pro ovládání robotické ruky (např. LynxTerm či RIOS)



Obrázek 24 Zobrazení komunikačního portu pro adaptér

Když zde nic nevidíme, klikneme na tlačítko **Přidat** a přidáme komunikační **Odchozí port** podle našeho výběru.

Nyní máme obě zařízení spojená a spárovaná, včetně simulace sériového portu. Ve správci zařízení ještě musíme nakonfigurovat správně rychlost dle obrázku níže.

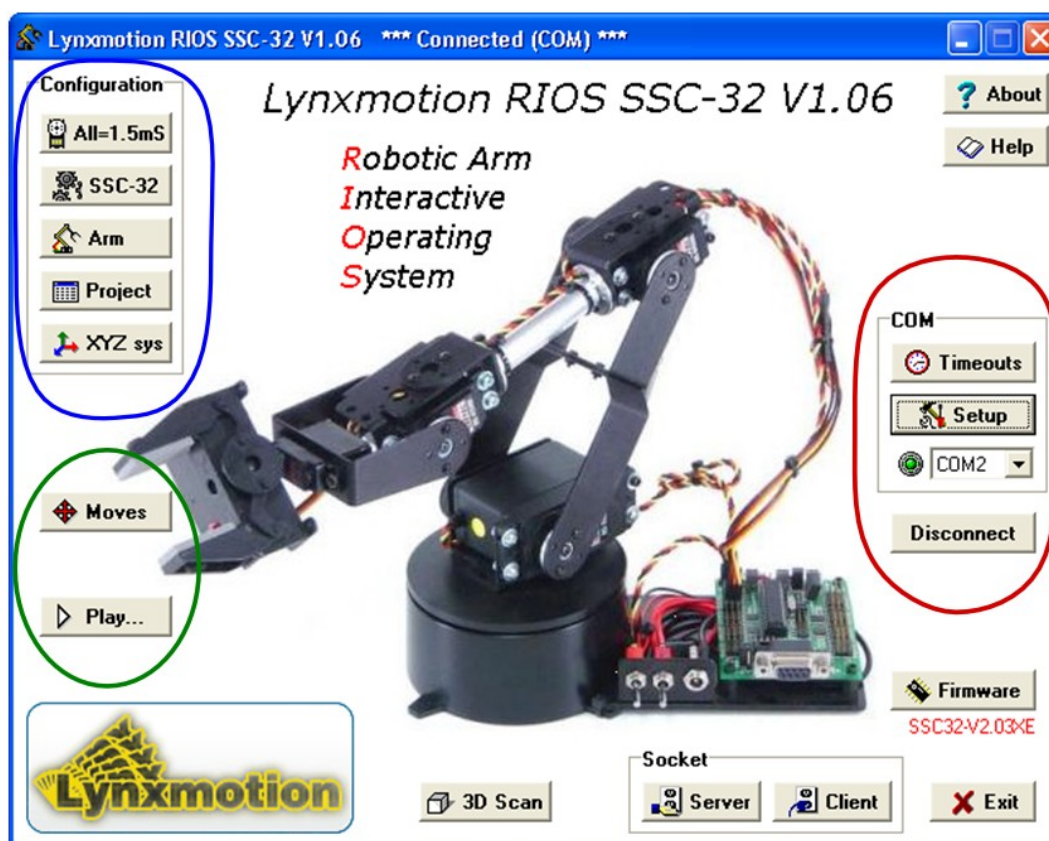


Obrázek 25 Nastavení Bluetooth COM portu

## 5 SEZNÁMENÍ S UŽIVATELSKÝM PROSTŘEDÍM OVLÁDACÍHO SOFTWARE RIOS

Nejvíce se budeme věnovat programu RIOS (Robotic arm interactive operating system<sup>6</sup>), neboť je dodáván s naším typem robotické ruky a navíc je napsán přesně pro potřeby stavebnic robotických rukou. Navíc dle našich zkušeností má opravdu velice široké možnosti použití.

### 5.1 HLAVNÍ OBRAZOVKA PROGRAMU



Obrázek 26 Úvodní obrazovka programu RIOS

Po spuštění se nám zobrazí takováto obrazovka. Pro lepší přehlednost jsme ji rozdělili na 3 logické celky:

- **Červený** - nabídka a nastavení komunikace
- **Zelený** – nabídka pro programování pohybů robotické ruky a spuštění již hotových projektů
- **Modrý** – nabídka pro celkovou konfiguraci a kalibraci robotické ruky

<sup>6</sup> Volně přeloženo jako: Systém pro interaktivní práci s robotickou rukou

## 5.2 ČERVENÝ CELEK – KOMUNIKACE

V této části programu se nastavuje komunikační rozhraní. Vidíme zde tlačítko **Connect** a **Disconnect**, který se připojíme či odpojíme k robotické ruce. Nad tímto tlačítkem vidíme rozbalovací nabídku, kde jsou čísla portů COM. Musíme se vybrat ten správný – v našem případě je to port COM2. Když se program úspěšně spojí s robotem, vidíme svítit indikátor zeleně.

Důležité je tlačítko **Setup**, které nastavuje volby komunikace. Hodnoty zde musejí být shodné, jako v ovladači sériového portu či redukce. Podstatná je především rychlost – **115200 baudů**. Ostatní volby upravovat nemusíme, neboť jsou nastaveny správně.

Tlačítko **Timeouts** spouští aplikace, která umí upravit TTL impulsy komunikace sériového portu. Ty se upravují pouze v případě, vyskytne-li se nějaký problém. Jinak je necháme tak, jak jsou. Upozornit můžeme na volbu VER, která zjistí verzi firmwaru jednotky. Jedná se stejný příkaz, jaký jsme používali hned při oživení ruky v programu LynxTerm.

## 5.3 MODRÝ CELEK – KONFIGURACE A KALIBRACE

Zde se v několika krocích provede prvotní nastavení a kalibrace robotické ruky. Jelikož se vytvoří kalibrační soubor, který budeme používat během práce v tomto softwaru, je opravdu důležité věnovat těmto bodům pozornost. Jakmile provedeme špatně nastavení bytí jedné osy, může to mít destruktivní následky na celého robota.

Samotná kalibrace není nijak složitá, chce to jen věnovat několika klíčovým bodům. Výrobce postupoval logicky a potřebné kroky se vykonávají postupně odshora dolů podle stupně důležitosti.

### 5.3.1 TLAČÍTKO ALL=1,5MS All=1.5mS

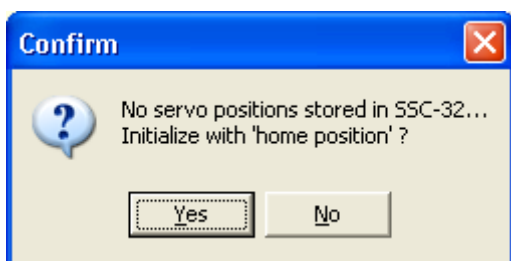
Tato volba slouží k ověření správného zapojení všech servomotorů a také funkčnosti celé stavebnice. Po kliknutí na tlačítko **Test** v následujícím okně se všechna serva uvedou do výchozí polohy, kterou vidíme na obrázku v tomto okně. Jestliže je všechno v pořádku a ruka je ve správné pozici, můžeme pokračovat v další kalibraci.



### 5.3.2 TLAČÍTKO SSC-32

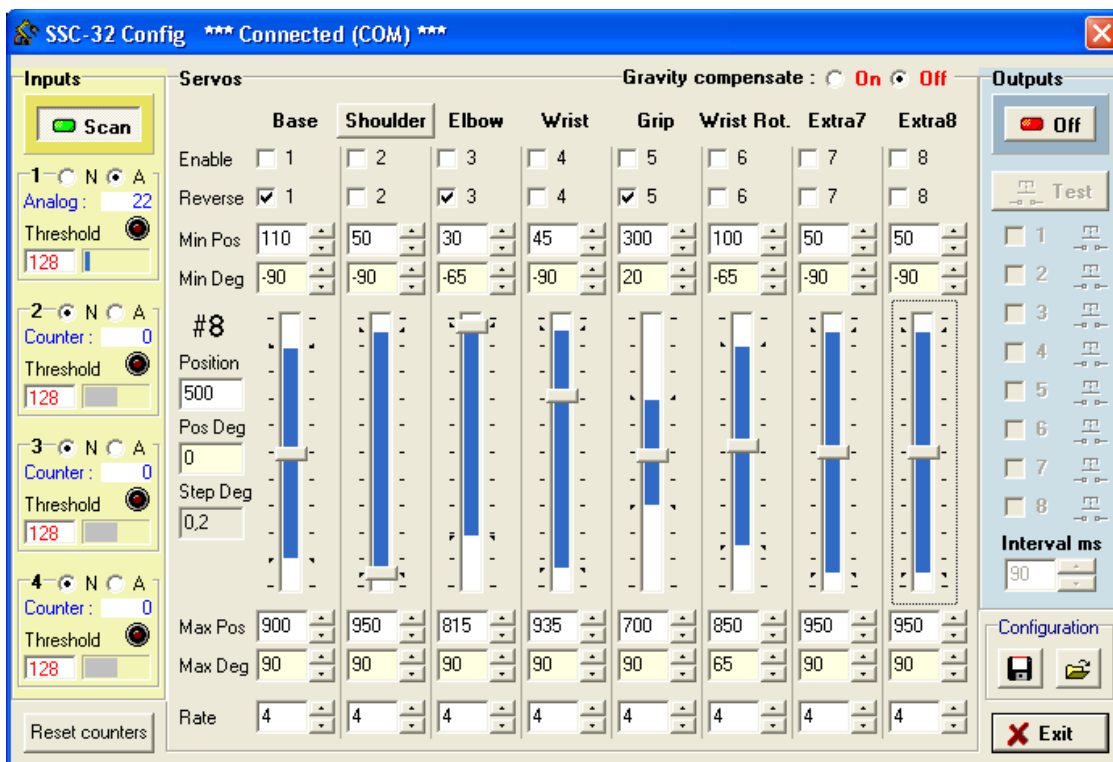
Důležitá volba, ve které se provádí prvotní kalibrace všech servomotorů na robotické ruce. Před započítím kalibrace doporučujeme vypnout napájení servomotorů a ponechat zapnuté pouze napájení elektroniky.

Když spouštíme aplikaci poprvé, tak se po klinutí objeví dotazovací pole, které se nás ptá, zda chceme uvést pozice serv do základní polohy, jelikož nebyl nalezen konfigurační soubor. Zvolíme **Yes** (ano).



Obrázek 27 Dotaz na inicializaci do základní polohy serv

V dalším okně se nás zeptá další dotazovací pole na to, jestli chceme **aktivovat všechna serva**. Je nutné vybrat **No** (ne)! Objeví se nám následující okno. Ověříme si, že jsou opravdu vypnutá a zapneme jejich napájení.



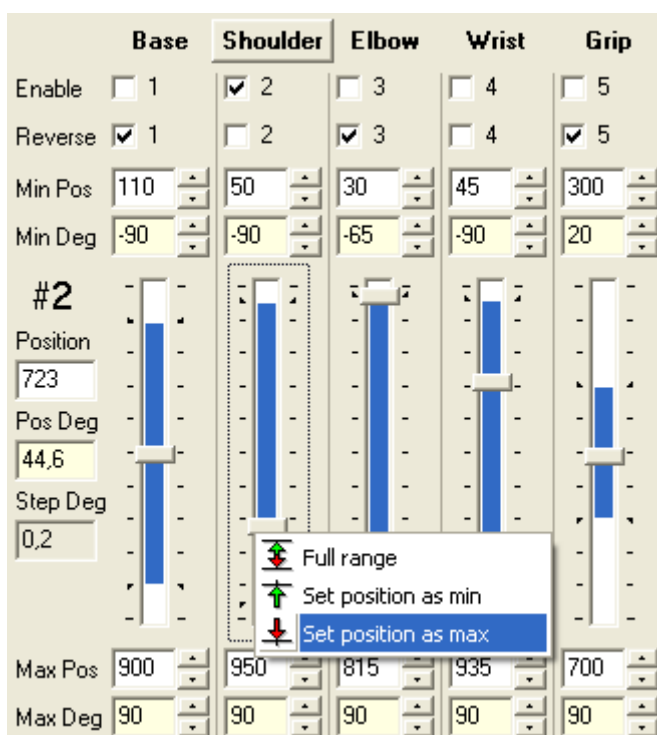
Obrázek 28 Nástroj pro kalibraci, všechna serva jsou vypnutá (položka Enable není zaškrtnutá)

**SHOULDER (RAMENO)**

S kalibrací by se mělo začít na servu č. 2, které výrobce nazývá *Shoulder*. Začíná se s ním proto, že celou ruku zvedne do výšky, tudíž nehrozí kolize s podstavou.

Posuvník nastavíme do poloviny, zapneme napájení a zaškrtneme pole **Enable** u serva č. 2. Ruka se přesune na pozici, kterou jsme nastavili. Nyní pomalu hýbáme posuvníkem a kontrolujeme minima i maxima. Podle výrobce by měla být poloha tohoto kloubu v obou rovinách horizontální. V našem případě jsme zvolili maxima mírně nad touto hranicí, abychom zamezili kolizím s jinými objekty. Myslíme si, že hlavně pro nezkušené uživatele to bude tak výhodnější.

Samotné **maximum** a **minimum** nastavíme tak, že posuvníkem najedeme na požadované místo a klikneme na něj pravým tlačítkem myši. Maximum nastavujeme dole (volba **Set position as max**) a minimum nastavujeme nahoře (volbou **Set position as min**). Více prozradí obrázek.

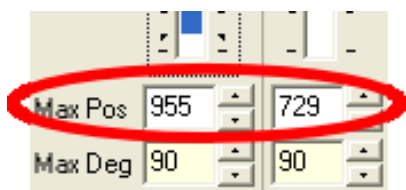


Obrázek 29 Nastavení maxima u ramenního kloubu

**BASE (ZÁKLADNA)**

Nyní můžeme přistoupit k nastavení hlavní otočné základny robota. Zaškrtneme si tedy políčko **Enable** a pokračujeme v dalších krocích.

Rozsah otáčení by měl být přesně 180°. Každé servo je trochu jiné a je možné, že ke správnému nastavení budeme muset posunout rozsah otáčení směrem nahoru či dolů. V našem případě jsme museli snižovat rozsah maxima a zvyšovat rozsah minima. To provedeme změnou hodnoty **Max Pos** a **Min Pos** směrem nahoru či dolů.



Obrázek 30 Pomocí šipek měníme hodnotu maxima

**WRIST (ZÁPĚSTÍ)**

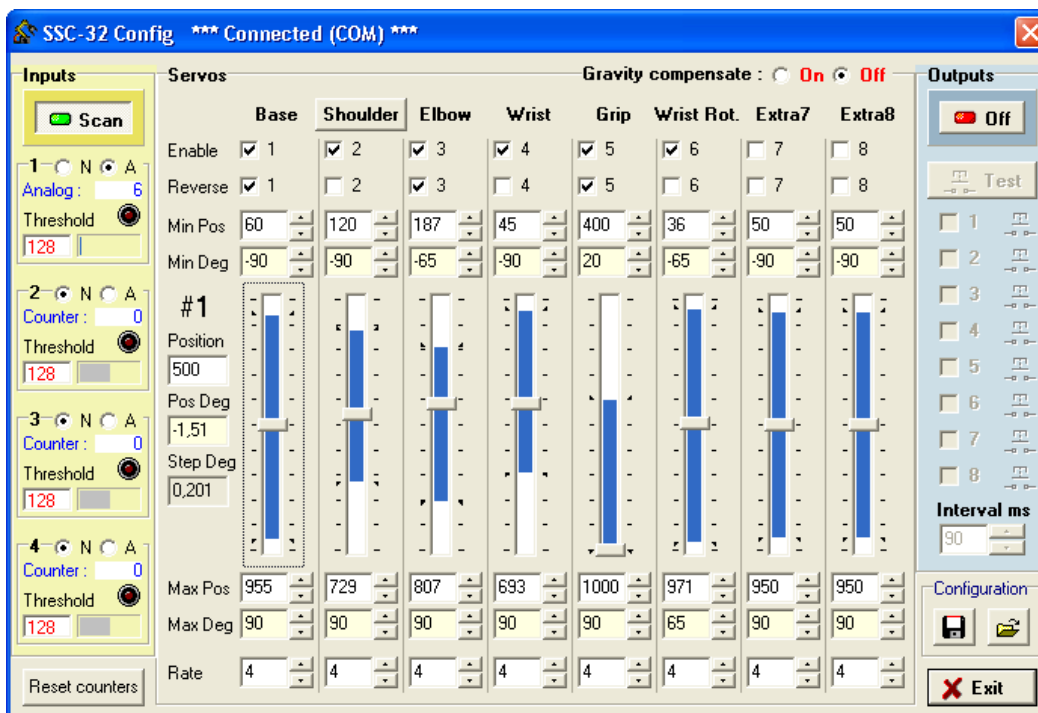
Po zaškrtnutí tlačítka **Enable** u políčka *Wrist* provedeme kalibraci pohyblivého zápěstí v horizontální rovině. Jelikož jsme použili nástavec pro připevnění bezdrátové kamery a IR senzoru, museli jsme upravit maximální možnou výchylku zápěstí (tedy hodnotu **Set position as max**), neboť by jinak hrozila kolize. Postupujeme naprosto stejně jako u předchozích bodů.

**GRIP (RUKOJEŤ)**

Postupujeme opět stejně jakou u předchozích bodů. Nastavujeme hodnoty servomotoru pro svírací rukojeť. V našem případě jsme museli kompletně předělat hodnoty pro kalibraci. V základním nastavení byly hodnoty uprostřed, to ale neumožňovalo úplné sevření rukojeti. Museli jsme posunout maximální hodnotu na úplný konec spodního maxima – v této poloze je ruka sevřena.

**WRIST ROTATE (ROTACE ZÁPĚSTÍ)**

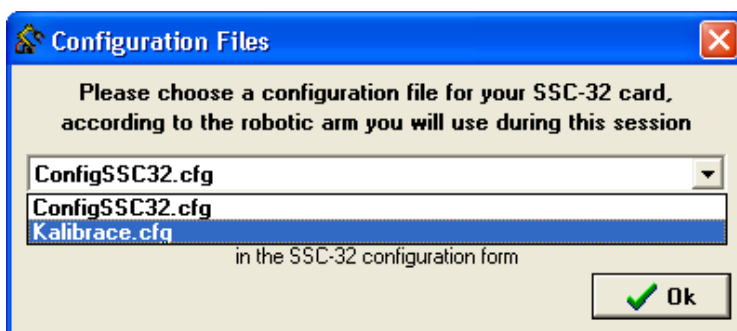
Tento modul jsme zakoupili zvlášť a slouží k rotaci celé části zápěstí. Zapojuje se jako další samostatné servo. Jelikož je modul nabízen stejným výrobcem, rovnou s ním počítá i v softwaru RIOS. Tudíž nenastal žádný problém ani v tomto případě. Pouze jsme museli rozšířit maximální a minimální hodnoty, neboť se využívá celý rozsah serva, který zde díky převodům činí 180°.



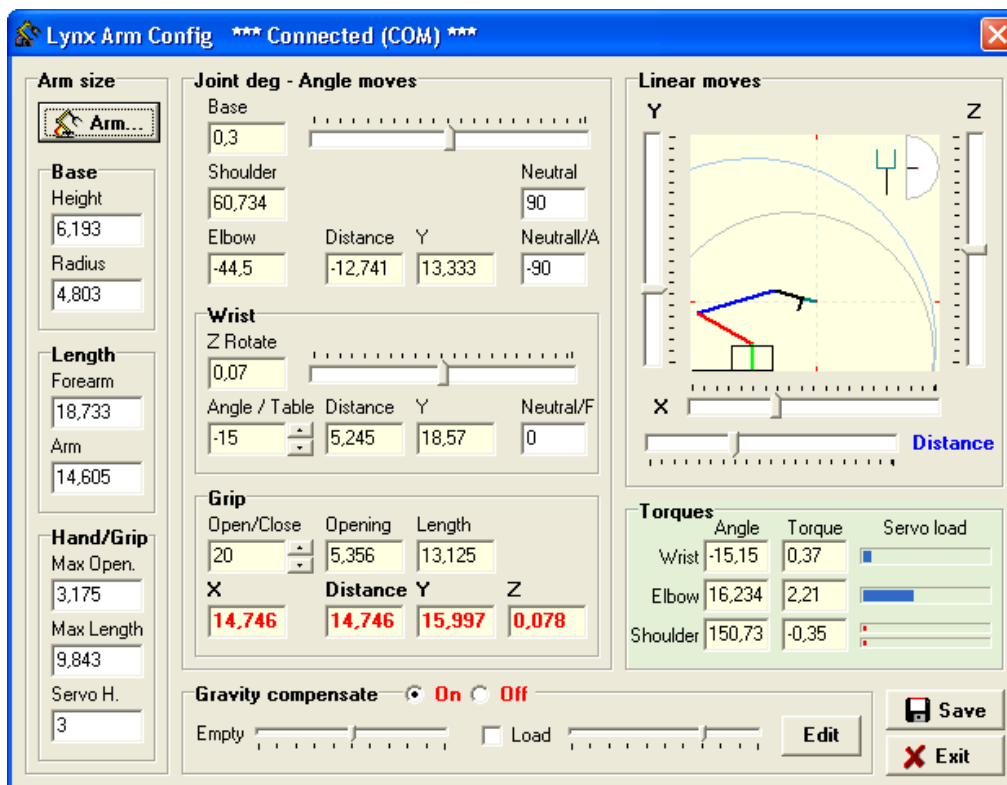
Obrázek 31 Zde vidíme kompletně provedenou kalibraci pro náš typ robotické ruky. Všimněte si rozdílů oproti obrázku Obrázek 28, kde je základní nastavení.

Program a řídicí jednotka nabízí ještě ovládání dalších 2 servomotorů, které my už nemáme. Další servo by se dalo využít např. pro vakuovou přísavku pro chytání předmětů, kterou výrobce nabízí. My ale necháme kanály 7 a 8 vypnuté, abychom zamezili nežádoucím stavům.

Když máme proces kalibrace hotov, musíme si konfigurační soubor uložit klepnutím na ikonu diskety v pravém dolním rohu. Doporučujeme si soubor nějak příhodně nazvat. Tento soubor je uložen v klasickém textovém formátu, tudíž je mezi programy přenosný. V našem případě to je soubor **kalibrace.cfg**. Ten budeme používat pro celou práci s robotem.



Obrázek 32 Při dalším spuštění programu se otevře okno, v kterém musíme vybrat vytvořený konfigurační soubor

5.3.3 TLAČÍTKO ARM 

Obrázek 33 Program ARM slouží k nastavení důležitých parametrů

Jak vidíme na obrázku výše, v této aplikaci se nastavují všechny parametry ruky. Aplikace je univerzální pro všechny robotické ruce, takže si musíme nastavit parametry přesně pro náš typ. K tomu slouží tlačítko **Arm...** vlevo nahoře. Po kliknutí na něj se spustí formulář, kde jsme zvolili:

- Typ ruky (Arm): AL5D
- Základna (Base): New Design
- Rukojeť (Gripper): druhý typ vpravo
- Rotace zápěstí (Wrist rotate): Yes – první typ

Všechny hodnoty se nám po správném výběru nastaví samy. Výběr ruky musíme provádět s vypnutým napájením servomotorů, jinak hrozí kolize!

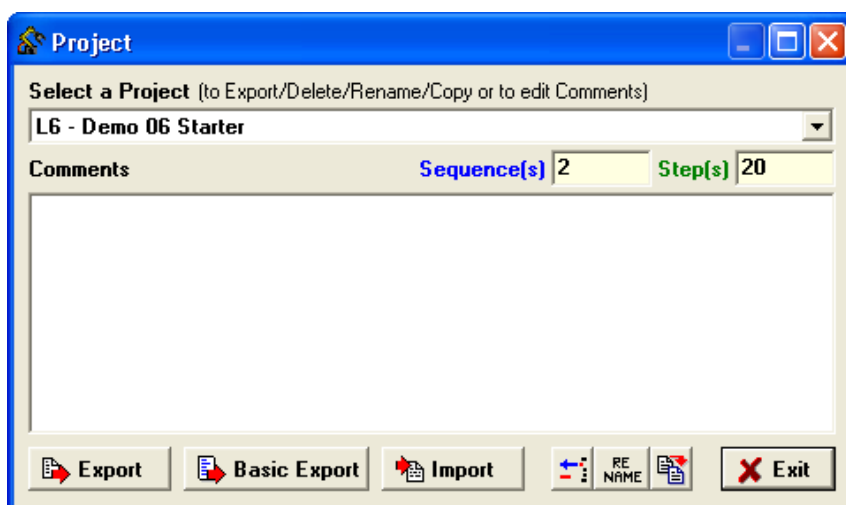
Zajímavá možnost je kompenzace gravitace (**Gravity compensate**), kdy software upravuje naklonění ruky a proudové impulsy do serv na základě zatížení (hmotnosti zvedaného předmětu). Obojí se dá nastavit pomocí těžítek *Empty* a *Load*. Kdo by

potřeboval podrobnější nastavení, klepne na tlačítko *Edit* a nastaví si vlastní hodnoty, např. pokud chceme zvedat těžší předměty.

V pravé části programu vidíme schematické znázornění ruky a tažítka pro osy X, Y a Z. Zde si můžeme poprvé vyzkoušet chování ruky v reálném čase. Program využívá tzv. inverzní kinematiky - s rukou pohybujeme v jedné ose a všechny ostatní se dopočítávají dle potřeby – ruka se po nich posouvá. Pokud měníme hodnotu posuvníku osy X, ruka se pohybuje v ose X, ale ostatní klouby se udržují ve stejné pozici – když si to představíme na příkladu, tak bychom takto nakreslili na papír rovnou čáru směrem od základny ven. A stejně funguje inverzní kinematika ve všech osách v případě, že s nimi nechceme hýbat samostatně.

Jestliže máme ruku jakkoliv upravenou, můžeme si nastavit vlastní hodnoty velikosti. Vše potřebné najdeme v levém sloupci **Arm size**.

#### 5.3.4 TLAČÍTKO PROJECT



Obrázek 34 Okno pro práci s hotovými projekty

Tato aplikace slouží k editaci a exportu již vytvořených projektů z části pro programování ruky. Obsluha je jednoduchá. Vybereme si projekt a exportujeme ho do požadovaného formátu databáze \*.CSV.

Důležitá je volba *Basic Export*. Ta provede export příkazů do jazyka Basic. Program v tomto jazyce používá mikrokontrolér Basic Atom 28 Pro, který využívá výrobce u dražších robotů nebo při kombinaci několika stavebnic. Například pásový podvozek a

robotická ruka. Díky mikrokontrolér je schopen robot provádět program bez připojení k počítači, čili chová se jistým způsobem autonomně.

Máme také možnost projekt importovat ze souboru formátu \*.CSV. Tuto možnost použijeme, pokud chceme přenést uložené projekty z programu nainstalovaného na jiné stanici. Všechny projekty jsou zakódovány a uloženy v souboru **Moves.dbf** a **Moves.MDX**, které jsou uloženy v kořenovém adresáři programu. Takže pokud nechceme provádět import, stačí do nového programu zkopírovat tyto 2 soubory.

### 5.3.5 TLAČÍTKO XYZ SYS

Slouží, pokud chceme používat jiný souřadnicový systém, než je nastaven. Ten odpovídá normě ISO:

- Osa X – horizontální rovina
- Osa Y – vertikální rovina
- Osa Z – prostorová hloubka – záporný směr

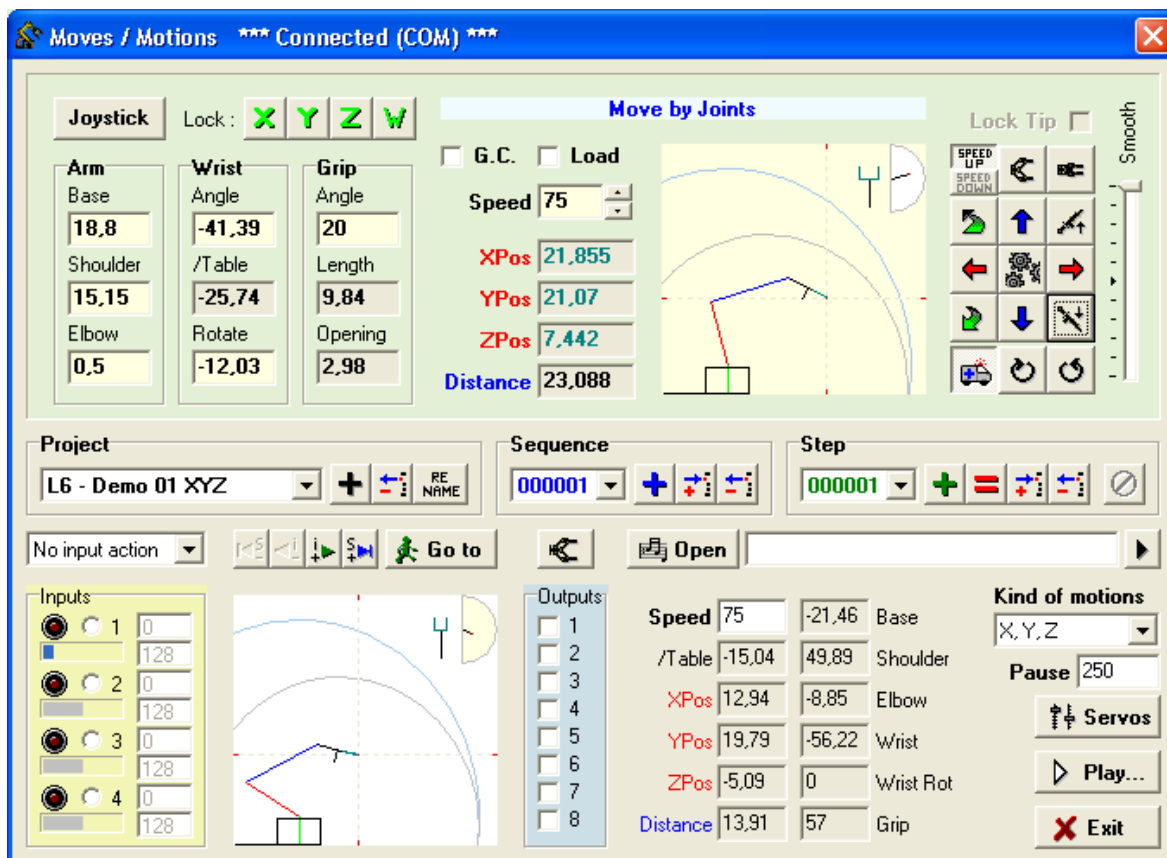
Druhou možností kombinace je podle strojů CNC, kde jsou osy prohozeny. Změna se provede pouze v prostředí programu RIOS, nikoliv hardwarově v robotické ruce. Zobrazení bude tedy vypadat odlišně podle nastavení. Doporučujeme používat RIOS zobrazení.

## 5.4 ZELENÝ CELEK – OVLÁDACÍ ČÁST

Jakmile se jednou robotická ruka správně nastaví a zkalibruje, na sekci nastavení se už nemusí vůbec sahat a používají se pouze 2 nástroje z ovládací části. Nacházejí se zde 2 aplikace – *Moves* (Pohyby) a *Play* (Přehrát). Každá má svůj účel, proto si je obě popíšeme podrobněji.

### 5.4.1 MOVES (POHYBY)

Můžeme říci, že toto je základní aplikace pro práci s robotickou rukou. Odtud ji můžeme ovládat pomocí myši, pomocí joysticku či jiné periferie. Navíc zde probíhá samotné programování pohybů ruky, které se zde dělí na programování sekvencí a programování kroků. Také si zde volíme jednotlivé režimy pohybu ruky. Více prozradí obrázek.



Obrázek 35 Hlavní okno aplikace Moves

Aplikace je rozdělena na 2 hlavní části.

**Horní** je vybarvena zeleně a slouží pro řízení robotické ruky. V její levé části vidíme přesně údaje o polohách jednotlivých kloubů. V prostřední části vidíme polohy jednotlivých os (X, Y, Z). V pravé je v reálném čase vykreslována poloha přesná poloha ruky v souřadnicovém systému. Každý kloub je znázorněn jinou barvou pro lepší rozpoznání. Zároveň se přizpůsobují obrazce elipsoidů, které při aktuálním natočení všech kloubů ruka opisuje. Toto velice usnadní práci s rukou v prostoru, protože to velice dobře demonstruje, jak se v reálných podmínkách roboti pohybují. Myslíme si, že je to velice dobrý výukový prvek.

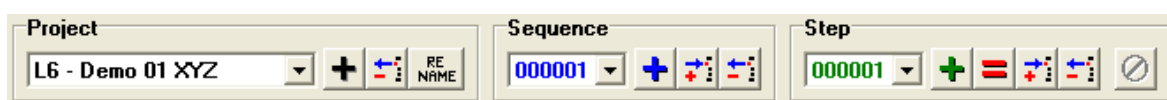
**Dolní** část slouží k vytváření projektů z jednotlivých pohybů ruky. Vidíme zde vedle sebe rozbalovací nabídky s projekty, sekvencemi a kroky, včetně ovládacích prvků pro jejich vytváření, mazání či přidávání. Pod těmito nabídkami je opět náhled se schematickou pozicí v prostoru. Tentokrát je ale zobrazena pozice vybraného kroku z jednotlivých sekvencí. Můžeme si tedy procházet jednotlivé kroky a kontrolovat pozici



celé ruky. V levé části vidíme monitorování vstupů. V našem případě je zapojen pouze vstup A. Vpravo vidíme souřadnice všech kloubů a napravo dole vidíme 3 tlačítka. **Servos** ovládá dodatečná serva, která ale my nemáme zapojená. Tlačítko **Play** spouští druhou aplikaci, která slouží k podrobnějšímu přehrávání projektů, včetně vkládání podmínek a cyklů mezi sekvence. **Exit** ukončí tuto aplikaci.

Představili jsme si obecně dostupné prvky a nyní se podíváme podrobněji na vytváření projektu.

### VYTVÁŘENÍ PROJEKTŮ



Obrázek 36 Volby pro postupné vytváření projektu

Samotné programy, které si budeme poté moci přehrát, výrobce řeší pomocí projektů. **Projekt** je tedy celistvý program, pomocí kterého řešíme nějakou kompletní událost či problém (např. přenesení předmětu).

Celý projekt se skládá ze samostatných **sekvencí** (Sequence) a sekvence se skládají z jednotlivých **kroků** (Steps).

#### Sekvence

Chápejme je jako obecnější souhrn několika kroků, které vedou k dílčímu výsledku. To může být např. uchopení předmětu, uvedení ruky do parkovací polohy či čekání na povel z libovolného senzoru. Obvykle se tedy projekt skládá z několika sekvencí. Záleží ale na jeho rozsahu. Čím složitější vytváříme projekt, tím důkladněji bychom měli volit správnou hierarchii sekvencí a kroků. Přináší to lepší přehlednost a do budoucna také lepší možnosti případných oprav či vylepšení.


#### Kroky



Jsou to jednotlivé fáze pohybu ruky. Záleží na nás, jak budou velké. Jeden krok bývá obvykle např. otočení základny, sevření rukojeti či nastavení určité polohy. Opět záleží na složitosti projektu. Ale obecně je lepší členit pohyby ruky do více kroků. Pohyb je potom plynulejší a máme opět více možností pro opravu či editaci.

Je důležité naučit se používat toto členění projektů hned od začátku, neboť některé rozšířené funkce programu jsou dostupné pouze pro sekvence, nikoliv pro kroky. To platí hlavně pro podmínky a cykly, které můžeme do projektu vkládat v aplikaci **Play**. Chápejme to tedy tak, že když projekt postavíme pouze z kroků v jedné sekvenci, nebudeme poté moci používat podmínku IF, která bude vykonávat různé sekvence v závislosti na hodnotě vstupního IR čidla.



### Ovládací prvky pole projektů


U každého pole jsou umístěna tlačítka, pomocí kterých ovládáme celý proces vytváření projektu.

Tlačítko  vytváří nový projekt. Otevře se dialogové okno, kde zadáme jeho název. Zároveň se vytvoří nová sekvence 000001 a v ní se vytvoří krok 000001. Tyto hodnoty značí počáteční polohu robota. Proto bychom ho měli uvést do tohoto stavu ještě předtím, než projekt vytvoříme.



Tlačítko  celý projekt smaže a tlačítko  projekt přejmenuje.


### Ovládací prvky pole sekvencí



Zde editujeme sekvence jednotlivého projektu. Kliknutí na tlačítko  vytvoří novou sekvenci v aktuálním projektu. V této sekvenci je automaticky vytvořen nový krok 000001, který navazuje na poslední krok z předchozí sekvence. Tlačítko  celou sekvenci včetně naprogramovaných kroků smaže.


Tlačítko  vloží novou sekvenci mezi již vytvořené. Toho použijeme, pokud potřebujeme přidat do hotového projektu další mezikrok.

### Ovládací prvky pole kroků

Tlačítko  vkládá ve zvolené sekvenci nový krok, který navazuje na ten předchozí. Jak již bylo řečeno dříve, krk by měl být nějaký dílčí pohyb určité sekvence, která určuje nějaký složitější celek.  námi vybraný krok maže.

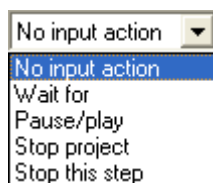
Tlačítko  přepíše hodnotu námi vybraného kroku novou polohou, kterou jsme nastavili v ovládací části.

Tlačítko  vkládá nová krok mezi 2 námi dříve vytvořené. Někdy se také může hodit tlačítko , které zde funguje jako tlačítko Zpět, vrací vše o krok zpět.

Pod projektem se nachází lišta, kde jsou ovládací prvky pro posun po sekvencích a krocích. S tím, jak se v projektu posouváme, mění se i náhledy schematického zobrazení ruky a pokud je zapojená, po stisku tlačítka  se ruka přesune na námi vybranou pozici.

#### Možnosti spouštění projektu v závislosti na vstupu


Zde se poprvé setkáváme s IR vstupním senzorem pro vzdálenost. Na obrázku dole vidíme rozsah akcí, které můžeme nastavit pro spouštění projektu.



Obrázek 37 Reakce programu na vstupní senzor

- **No input** – neprovede se nic a po stisku tlačítka Play se projekt spustí
- **Wait for** – Program bude čekat, dokud neobdrží hodnotu z čidla, která odpovídá námi nastavené hodnotě *Threshold* (prahu). Uvedeme si příklad. Nastavili jsme hodnotu 50 – po odečtení této hodnoty z grafu, který jsme vytvořili, zjistíme, že to odpovídá vzdálenosti 12 cm od senzoru. Jakmile se přiblížíme rukou či nějakým předmětem blíže než je 12 cm, projekt se spustí a jednou provede
- **Pause/Play** – na základě prahové hodnoty ze senzoru se projekt spustí nebo pozastaví
- **Stop project** – zastaví probíhající projekt, pokud je splněna zadaná podmínka (hodnota prahu)
- **Stop this step** – program hlídá hodnoty ze senzoru a jakmile je splněna podmínka, vynechá se při vykonávání programu krok, který jsme označili

## OVLÁDÁNÍ ROBOTICKÉ RUKY POMOCÍ TLAČÍTEK

V horním zeleném poli máme všechny potřebné nástroje ovládání robota. Důležité je hlavně pole tlačítek napravo od znázornění polohy. Zde jsou všechny ovládací prvky, na které se podrobněji podíváme. Uprostřed se nachází důležité tlačítko , kterým přepínáme 3 různé módy ovládání.

### Move by joints (ovládání kloubů)



Tuto možnost využijeme, pokud chceme kontrolovat samostatně každý kloub robota. Náš typ má 6 stupňů volnosti, tudíž ovládáme samostatně 6 pohonů. Symbol ruky nahoře svírá a rozvírá rukojeť. Symbol kruhu dole ovládá natáčení rukojeti. Symboly ruky vpravo pohybují zápěstím nahoru a dolů. Zelené šipky otáčejí základnou. Modré šipky pohybují loktem ruky a červené pohybují hlavním ramenem. Tlačítko záchrany okamžitě vypne napájení všech serv, kdybychom náhodou udělali nějaký špatný pohyb.

Tento druh ovládání je nejjednodušší na pochopení, a pokud potřebujeme s rukou přesně manipulovat, je také nejvhodnější. Na druhou stranu ale vyžaduje od uživatele jistou představivost, jak s jednotlivými klouby v prostoru pohybovat tak, abychom dosáhli maximálního rozsahu pohybu ruky. Doporučujeme tento mód používat pro vytváření nových projektů.

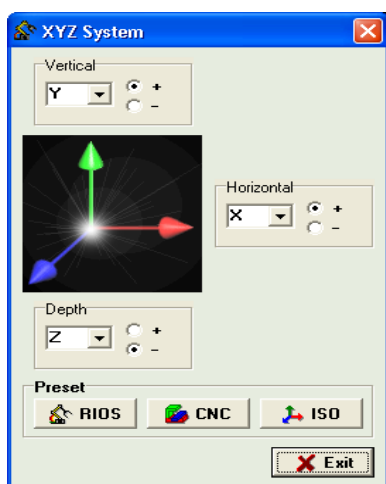
### Move by X, Y and Z (pohyb v osách X, Y a Z)



Jak už je patrné z názvu, pohybujeme s rukou přesně v souřadnicovém systému. Jelikož se ruka pohybuje v 3D prostoru, tak k jeho popsání potřebujeme 3 osy – X, Y a Z. Pomocí **šipek doleva a doprava** ovládáme osu X a ruka se pohybuje dopředu či dozadu. **Šipky nahoru a dolů** ovládají osu Y a ruka se pohybuje nahoru či dolů. Tučné **šipky po stranách vlevo** ovládají osu Z, čili rukou pohybuje do stran. Dále jsou zde samostatné ovladače pro ovládání natáčení zápěstí a svírání rukojeti. Ovšem při pohybu s jednou osou se virtuální body ostatních os udržují na jednom bodě. Takže kdy pohybují osou Z, ruka se pohybuje pouze po přímce osy Z, ale body os X a Z zůstávají stejné. Přesvědčit se o tom můžeme na reálných hodnotách, které jsou zobrazeny vedle obrázku ruky.

Zde se již využívá pokročilých výpočtů inverzní kinematiky. Tudiž tím, jak pohybujeme rukou ve všech třech osách, řešíme pouze pomyslný bod, kam chceme, aby se ruka dostala, ale už neřešíme to, jak musíme jednotlivé klouby správně natočit. To za nás řeší software RIOS.

A právě zde vidíme napojení na výuku geometrie na ZŠ. Děti zde uvidí v praxi, jak se aplikují zákony geometrie na reálné věci. Abstraktní učební látka se tak u žáků přetvoří do praxe a zároveň upevní. Když se podíváme na Bloomovu taxonomii cílů, tak aplikace a analýza poznatků je velice vysoko. Žáci budou muset přemýšlet, v jaké ose musí robotem pohybovat, aby dosáhli určeného bodu.



Obrázek 38 Zobrazení os pro robotickou ruku

Move by Distance, Y and Base Angle (pohyb vzdáleností, v ose y a natočení základny)



Zde už se nepohybujeme pouze po jednotlivých osách, ale i v prostoru, tudíž využíváme celý souřadnicový systém. Opět se zde pro výpočet polohy ruky využívá inverzní kinematiky. Samotná základna se otáčí volně **tučnými šipkami vlevo**. A z každého bodu podle natočení základny pohybujeme rukou dopředu a dozadu (ovládají tenké šipky se zeleným bodem) či nahoru a dolů (ovládají bílé šipky). Do toho ještě ovládáme natočení či zvednutí zápěstí. Reálné hodnoty souřadnic se vypisují vedle schématu ruky.

Tento mód se pro děti nejsložitější na pochopení, neboť už musí mít z geometrie teoretické znalosti o tom, jak se předměty pohybují v trojrozměrných souřadnicových systémech. Uvedeme na příkladu. Když základnu natočíme o  $45^\circ$  a pohybujeme rukou dopředu, už se nepohybujeme pouze na ose X, ale současně se mění i hodnoty osy Z atd.

Ruka může sloužit jakou praktický příklad toho, jak se teoretické znalosti matematiky a geometrie uplatňují v praxi.

Posuvník **Smooth** vedle ovládacích prvků reguluje jemnost a plynulost pohybu ruky, doporučujeme ho nechat co nejvyšší.

Tlačítko **Load** značí, že ruka nese nějaký předmět. Upraví se tedy podle toho i gravitační kompenzace. Zvláště u těžších předmětů doporučujeme používat. Tato volba jde přiřazovat jednotlivým krokům.

### **OVLÁDÁNÍ ROBOTICKÉ RUKY POMOCÍ JOYSTICKU**

Doteď jsme se bavili pouze o ovládání ruky pomocí myši. Toto ovládání je sice přesné a doporučujeme ho pro vytváření nových projektů, ale také je nepohodlné a nepružné v čase. Jsme schopni ovládat současně pouze jeden prvek. Proto výrobce umožnil používat pro kontrolu ruky i joystick či joypad.

Z našich zkušeností víme, že lépe z těchto dvou zařízení vychází joystick. Je ale potřeba vybrat ten správný. Abychom obsloužili správně všechny klouby na ruce, potřebujeme joystick, který má více os než 2. My jsme vybrali **Saitek Cyborg Evo**.

Je to zařízení vyšší třídy, které mimo klasických os obsahuje i natáčecí rukojeť a páku pro regulaci plynu – tu jsme také využili jako další osu. Navíc disponuje velkým počtem tlačítek, která slouží pro obsluhu programu či svírání rukojeti.

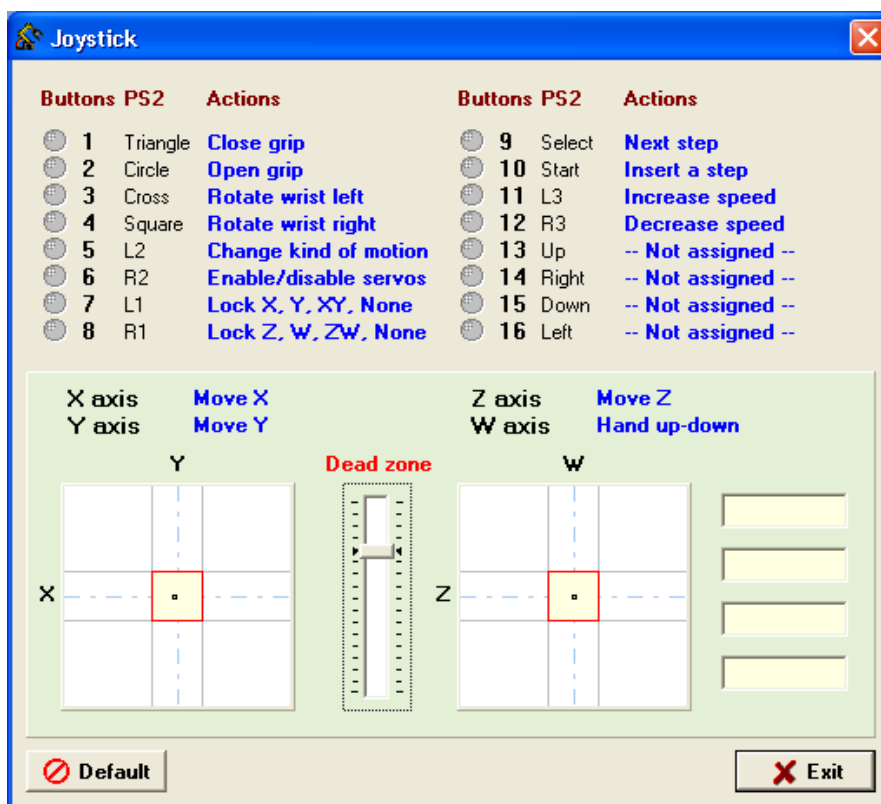
#### Nastavení joysticku

Program RIOS detekuje připojený joystick sám, stačí, když je správně nainstalován v operačním systému. V aplikaci **Moves** klepneme na tlačítko **Joystick**, které je aktivní pouze v případě připojení kompatibilních periférií. Dostaneme se do nastavení.

Nahoře vidíme tlačítka, která jsou dostupná. Dole vidíme osy. V našem případě jsme nastavili osy **X**, **Y** a **Z** shodně na joysticku i robotické ruce. Osa **W** je původem pedál plynu, který na ruce ovládá vertikální pohyb zápěstí. Doporučujeme nastavit relativně velkou **Dead zone** (nepohyblivá oblast), vystřídáme se tím zbytečných pohybů ruky.

Na tlačítkách si nastavíme ještě svírání rukojeti a také její otáčení. To by měli být nejdostupnější tlačítka – většinou na vrchu joysticku. Doporučujeme si také přiřadit na

dostupné tlačítko volbu **vypnutí všech serv.** Tato funkce se bude hodit hlavně při práci s menšími dětmi. U nich hrozí možnost, že s rukou o něco zavadí a mohly by ji zničit.



Obrázek 39 Příklad vhodně nastaveného joysticku v programu RIOS

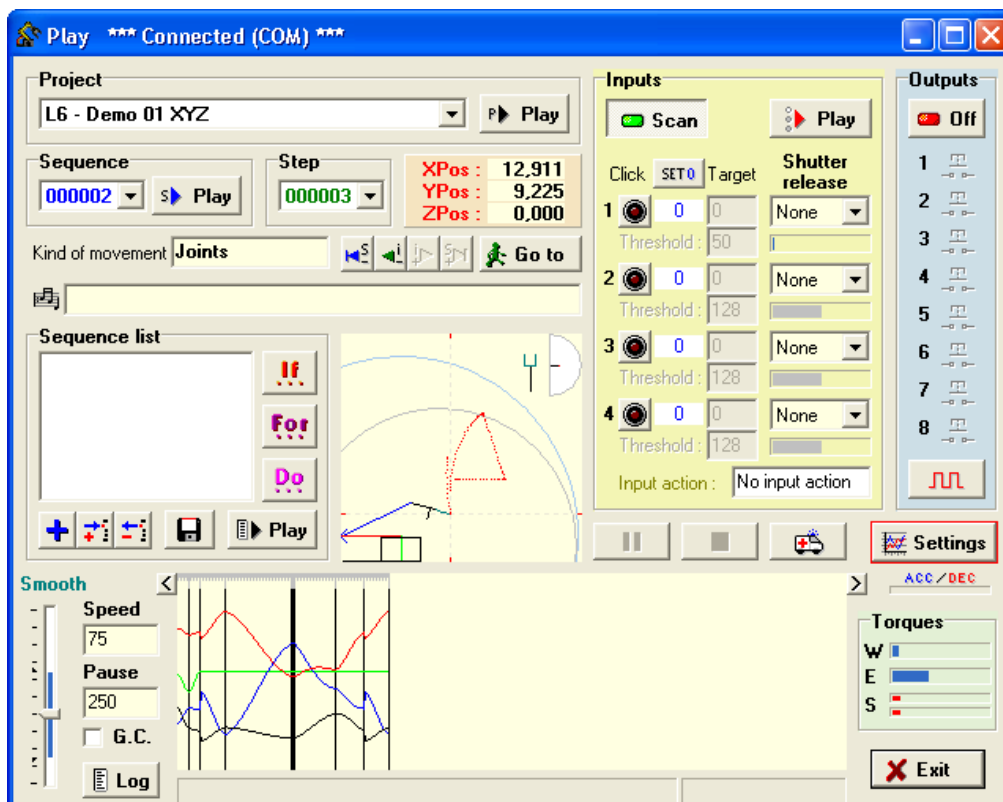
Na obrázku výše uvádíme jedno z možných nastavení joysticku pro pohodlnou práci s robotickou rukou. Funkce tlačítkům i osám se přiřazují kliknutím pravým tlačítkem myši z dostupného seznamu.

#### Blokace os


Jestliže nechceme používat některou osu joysticku, můžeme si ji pomoci těchto tlačítek **Lock : X Y Z W** zablokovat. Takto můžeme vypnout všechny osy joysticku a nemusíme ho pak vypojoovat z počítače. Když pracujeme s rukou pouze pomocí myši, důrazně doporučujeme osy vypnout. Vyhneme se tak problémům s ovládním.

#### 5.4.2 PLAY (PŘEHRÁVÁNÍ PROJEKTŮ)

Poslední aplikací v programu RIOS je Play. Zde se nachází všechny možnosti pro pokročilé spouštění a editaci vytvořených projektů.



Obrázek 40 Okno Play

Aplikace je rozdělena na 3 části. Horní část obsahuje ovládací prvky pro projekty, sekvence a kroky. Pomocí ovládacích prvků se můžeme mezi nimi libovolně přesouvat a tlačítkem  **Go to** přesunout robotickou ruku přímo na pozici z vybraného kroku.

Ve spodní části vidíme schematický náhled pozice ruky. Důležité je pole **Sequence list**. U něj vidíme 3 tlačítka – **If**, **For** a **Do**. To jsou rozšířené možnosti spouštění projektu tak, že do něj zahrneme cykly či podmínky na základě určitých hodnot. Tím jsme schopni reagovat například na hodnoty ze vstupních senzorů, případně jsme schopni opakovat určitou sekvenci několikrát za sebou. Budeme se jim věnovat podrobněji.

Dole se nachází grafické znázornění trajektorie kloubů ruky závislé na čase. Vedle se nachází zelené pole **Torgues**, které zobrazuje zatížení jednotlivých os.

Červeně označené je tlačítko **Settings**. Tím nastavujeme některé důležité parametry pohybu a regulace pohonů ruky. Od výrobce jsou nastaveny dobře, není důvod s nimi hýbat. Když se vyskytnou některé problémy, např. cukavý pohyb ruky, můžeme zkusit změnit hodnotu linearity pohybu (tím upravujeme časové konstanty PID






regulátoru). Doporučujeme zde zapnout volbu **Show trajectories**. Jak můžeme vidět na obrázku 40, aplikace vykresluje trajektorii ruky.

Ve žluté části vpravo vidíme hodnoty 4 vstupů, které se používají pro senzory. My máme zapojen jeden IR senzor do vstupu 1. V náhledu vidíme aktuální hodnotu a nastavenou prahovou hodnotu.


V modrém poli vpravo vidíme výstupy. Ale jelikož je nepoužíváme, ponecháme je vypnuté. Výstupy se používají, pokud kombinujeme několik stavebnic Lynxmotion dohromady a potřebujeme, aby si mezi sebou předávaly hodnoty.

### VYTVÁŘENÍ CYKLŮ A PODMÍNEK IF

Podmínky a cykly pracují pouze se sekvencemi, proto jsme dříve zdůrazňovali, že je důležité si projekt dobře rozvrhnout a nějaké dílčí celky či části projektu, se kterými chceme dále pracovat, zahrnovat do sekvencí.

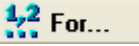



Na samotné vkládání a mazání slouží tlačítka dole pod seznamem sekvencí. Tlačítko  vybranou sekvenci. Tlačítko  vloží novou sekvenci mezi již vložené a  vloženou sekvenci smaže. Když máme potřebné sekvence vloženy, můžeme začít se vkládáním cyklů.

### Vložení cyklu FOR

Pro účely ukázky jsme si vytvořili jednoduchý program, který se skládá ze dvou sekvencí. Jedná, natočí ruku doleva a zamává, druhá natočí ruku doprava a zamává. Každá sekvence je tvořena 3 kroky. Začneme s tím, že si vložíme obě sekvence do seznamu tlačítkem +. Klikneme na tlačítko . Otevře se nám následující okno.



Obrázek 41 nabídka cyklu For

Klikneme na sekvenci 000001, do pole **Loop** napíšeme číslo 3. To znamená, že se celý cyklus provede 3. Nyní klikneme na tlačítko . Tím jsme si vybrali vložení tohoto cyklu. Kliknutím na tlačítko  se nám cyklus vloží před sekvenci 000001, což jsme chtěli. Tímto jsme si stanovili, kde cyklus začne. Cyklus se ale také musí ukončit. Klikneme tedy na sekvenci 000002, vybereme tlačítko  a klikneme na tlačítko .

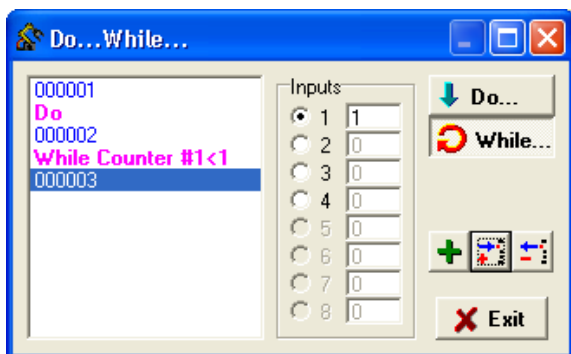
Na obrázku Obrázek 41 vidíme správně provedený cyklus For, který obě sekvence zopakuje třikrát.

Na tomto příkladu vidíme, že žáci se takto nenásilnou formou učí základy programování. Cykly a podmínky jsou stěžejní část každého programovacího jazyka a zde se setkávají s velice podobnou syntaxí, jako je např. v jazyce Pascal. Ale vše je zde grafickou formou, navíc je možnost si vlastní program hned ozkoušet v praxi.

#### Vložení cyklu Do...While





U cyklu For provádíme vybrané sekvence po určitý počet předem daných opakování a samotný obsah cyklu se nemusí provést ani jednou. Takto se nazývá *cyklus s podmínkou na začátku*. U cyklu Do...While je tomu jinak. Je *cyklus s podmínkou na konci* a rozdíly jsou 2. Obsah cyklu se provede vždy alespoň jednou a bude se provádět do doby, než bude zadaná podmínka splněna. Proto pozor na konstrukci tohoto cyklu. Může se nám stát, že se program bude provádět pořád dokola, dokud ho neukončíme manuálně.

Je důležité, abychom žáky předtím, než budou s cykly pracovat, seznámili s nezbytnou teorií. Musí dobře chápat rozdíly mezi nimi. Navíc pokud to pochopí pro účely programování robotické ruky, nebudou mít problémy ani v budoucnu v objektovém programování, kde je konstrukce a syntaxe velice podobná.



Obrázek 42 Správná konstrukce cyklu Do

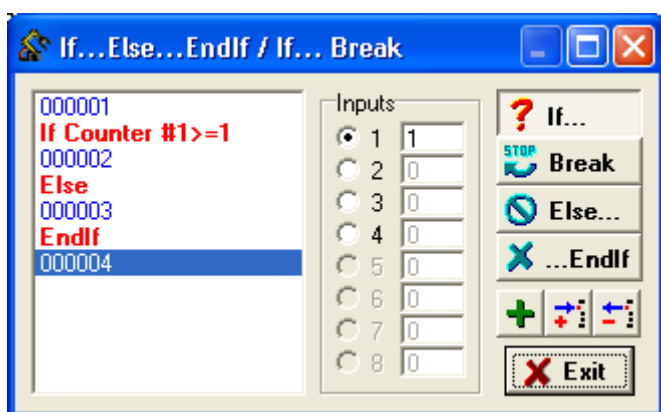
Pro lepší názornost jsme si předchozí příklad upravili. Vložili jsme do něj třetí sekvenci, které na levé straně mává uživateli. A právě na tuto sekvenci aplikujeme cyklus Do. Naším požadavkem je, aby ruka neustále mávala, dokud nebude mít zpětnou odezvu od uživatele – přiblíží se k ní a taky jí zamává. Poté ruka bude pokračovat a bude mávat na druhou stranu.

Klikneme tedy na tlačítko . Vybereme sekvenci **000002**, do pole **Inputs** zapíšeme hodnotu **1** – to znamená, že se cyklus překloupí po 1 splnění podmínky v počítadle podle hodnoty prahu, který je nastaven na 40 (15 cm). Vybereme tlačítko **Do** a klikneme na . Poté vybereme sekvenci **000003**, klikneme na  **While...** a opět klikneme na . Hotový cyklus by měl vypadat jako na obrázku Obrázek 42.

### Podmínka IF

Zde se setkávají žáci nejspíše poprvé s podmínkou If – Else. Opět zde vidíme důležitý přesah do více předmětů, hlavně matematiky a informatiky. V programování jsou konstrukce IF a jejich varianty jedny z nejdůležitějších. Ostatně jak jinak bychom ošetřili v systémech různé stavy, chybové chování apod. Proto je důležité, aby se žáci naučili správně používat podmínky IF.

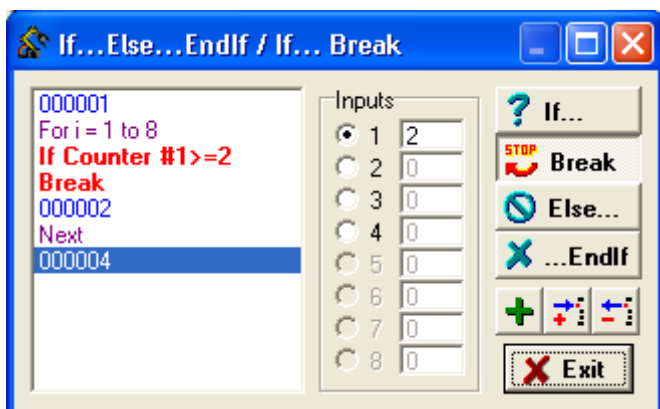
Máme několik možností konstrukce podmínky. Více prozradí obrázek.



Obrázek 43 Jedna z možností konstrukce IF – Else

Začínáme tak, že vždy vložíme přes známé tlačítko **Insert** podmínku **If** před námi vybranou sekvencí. Tou se ptáme, co chceme, aby se stalo při splnění této podmínky. Vybereme vstup 1 a zadáme do něj číslo 1. Když se splní podmínka a začloníme hned na začátku sekvence 000001 senzor, counter se změní na 1 a provede se sekvence 000002.

Když tuto podmínku nesplníme, provede se rovnou sekvence 000003 a poté sekvence 000004 (ta se provede vždy). Když chceme programu říci, že chceme, aby se místo sekvence 2 provedla sekvence 3, musíme před ni vložit **Else (jinak)**. Právě to zařídí to, že se provede sekvence 000003 namísto 000002. Podmínku musíme náležitě ukončit příkazem **EndIf**. To byla jedna konstrukce. Ale jak vidíme na obrázku, je jich mnohem více.



Obrázek 44 Jiná konstrukce If kombinovaná s cyklem For

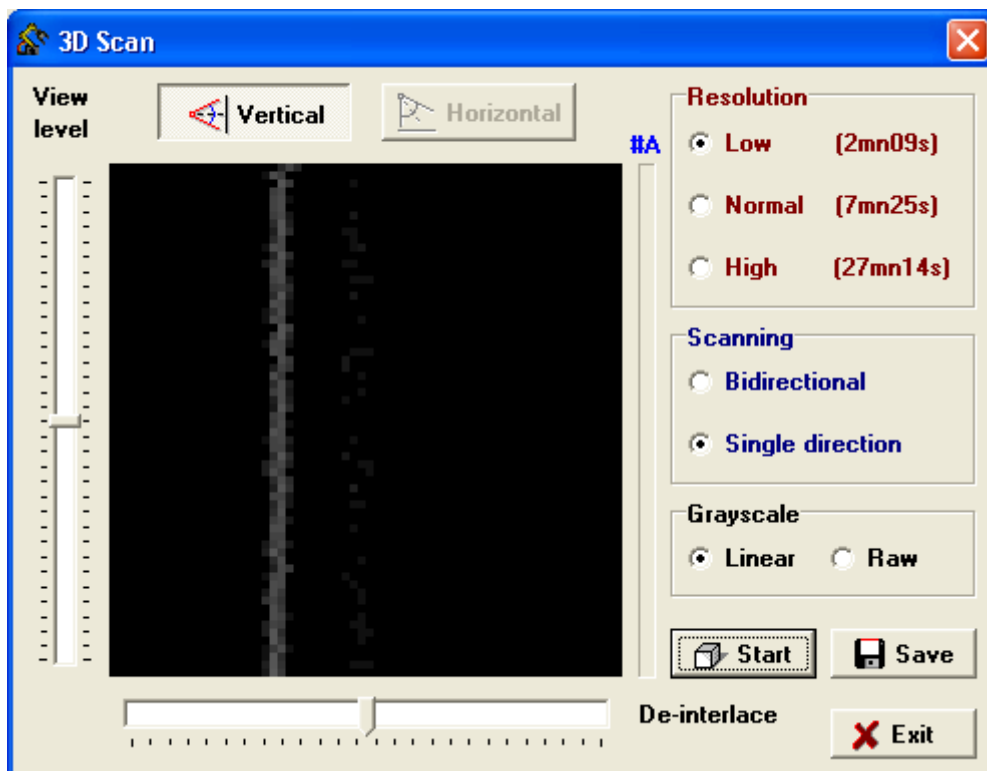
Použili jsme upravený příklad s cyklem **For**. Ten se opakuje osmkrát (mává na levé straně). My ale chceme, aby přestal mávat, pokud mu odpoví 2 lidé (zamávají IR čidlu ve vzdálenosti cca 15 cm). Cyklus se poté přeruší a skočí rovnou na sekvenci **000004**.

Vložili jsme tedy do cyklu podmínku **IF** s počítadlem splněných vstupů (nastavený na **2**). Hned za ní jsme vložili konec podmínky **Break**. Ten způsobí to, že při splnění podmínky se přeruší cyklus (sekvence 000002) a program rovnou přejde na poslední sekvenci. Více prozradí Obrázek 44.

Konstrukcí je ještě mnohem více, další možnosti jsou např. *If...Else...Break a If...EndIf*. Myslíme si ale, že pro pochopení konstrukce podmínek a cyklů v programu RIOS výše uvedené příklady stačí.

## 5.5 3D SCAN 3D Scan

Velice zajímavý nástroj, který využívá IR senzoru ke skenování prostoru. Se starším senzorem dosahoval program lepší výsledky, neboť měl větší rozsah. S tím, který máme k dispozici my, se nám naskytne pouze přibližný obraz toho, jak vypadá prostor před robotickou rukou.



Obrázek 45 Získaný obraz prostoru před rukou - bílý pruh je tužka vložená před senzor

Senzor funguje tak, že obsahuje IR vysílač (diodu) a IR přijímač. Infračervené paprsky se tedy odrážejí od předmětů před senzorem. Čím je předmět blíže, tím se vrátí do senzoru více paprsků a tím větší hodnotu poté senzor posílá do vstupu SSC-32.

Samotná funkce 3D Scan funguje tak, že ruka se pohybuje postupně po řádcích a snímá předem určený rozsah obrazu, který se z nich skládá. Bližší bod je znázorněn světlou barvou a vzdálený je černě.


Můžeme nastavit rovinu snímání, kvalitu a druhy skenování (jedním směrem či oběma). Čím kvalitnější nastavení, tím bude snímací čas delší (max. 28 minut).

Na tomto příkladu se žáci zábavnou formou seznámí s fyzikálními principy nejen IR světla. Jelikož noktovizory a jiná podobná zařízení fungují na podobném principu, bude pro ně vytváření 3Dscanů jistě velice lákavé. Navíc tím, jak budou postupně vytvářet lepší a lepší obrazy skutečnosti, nebudou vůbec vnímat to, že se vlastně učí fyzice.

## 5.6 SOCKET SERVER A SOCKET CLIENT

Toto je velice zajímavá funkce, kterou výrobce přidal v několika posledních verzích programu. Jak už z názvu vyplývá, v programu RIOS si spustíme vlastní server a poté se k němu můžeme pomocí internetu připojit.

### SPUŠTĚNÍ SERVERU

Server spustím kliknutím na tlačítko  **Server**. Otevře se nám okno, které vidíme na obrázku Obrázek 46. Ovládání je opravdu jednoduché. V aplikaci vidíme naši lokální adresu (127.0.0.1), adresu **IP v síti LAN** (192.168.1.2). Na adresu LAN IP se budeme připojovat z cizího počítače, proto si ji musíme zapamatovat. Důležité je také **číslo portu** (55000). To musíme také znát, neboť ho spolu s IP adresou zadáváme do přihlašovacího formuláře. Poté stačí pouze stisknout tlačítko **Activate** a server se spustí.

Musíme pamatovat na to, že jakmile se pohybujeme v lokální síti, je možné, že jsou omezena práva. Proto když nebude spojení fungovat, měli bychom se poradit se správcem sítě, který musí přidat výjimku do **Firewallu** Windows, případně musí přidat přímo na **Routeru** lokální síti port pro přeposílání. Tím si zajistíme maximální funkčnost.

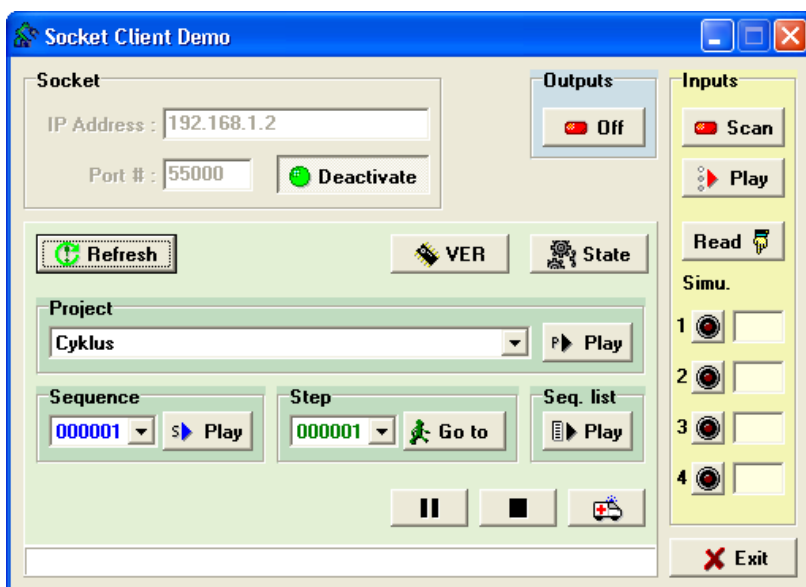


Obrázek 46 Nastavení a spuštění Socket serveru

### PŘIPOJENÍ A OBSLUHA SOCKET KLIENTA


Na druhé straně internetu musíme mít aplikaci, která se k serveru připojí. Výrobce ji dodává jako samostatný program s názvem **Socket Client Demo**. Jak vidíme na obrázku Obrázek 47, obsahuje pouze omezené ovládací prvky oproti Programu RIOS.

Nahoře jsou volby pro připojení. V zelené části vidíme volby pro spuštění již hotových projektů, včetně samotných sekvencí a kroků. Dole jsou ovládací tlačítka (Pauza, Spuštění a vypnutí všech pohonů). Ve žluté části napravo vidíme 4 vstupy, které můžeme ovládat. Tlačítkem **Scan** se vstupy zapnou, tlačítkem **Read** se načtou hodnoty ze senzorů.



Obrázek 47 Socket klient

### Připojení k serveru

Do formuláře nahoře zadáme známou **IP adresu** serveru. Do pole **Port** zadáme číslo portu, které jsme si zvolili (55000). Nakonec klikneme na **Activate**. Pokud jsme vše nastavili správně, program by se měl připojit. Spojení si můžeme ověřit tlačítkem **VER**, které nám vrátí verzi softwaru RIOS. Důležité je tlačítko  **Refresh**, které načte ze serveru všechny uložené projekty z databáze.

### Didaktické hledisko

Použití internetového serveru vytváří z didaktického hlediska příležitost pro mezipředmětové vazby. Nejenom, že budou muset žáci řešit komplexní konstrukci projektu s robotickou rukou, ale navíc se k tomu ještě přidá problematika síťového připojení počítačů. Nabízí se vazba s informačními technologiemi, konkrétně se sítěmi. Záleží na učiteli, do jaké hloubky se bude chtít problematice sítí věnovat. Žáci by si při nějaké komplexnější úloze či projektu mohli zkusit dokonce nakonfigurovat router a firewall pro správnou komunikaci. Do toho by šla vysvětlit problematika adresování v internetu a lokálních sítích atd.

## 6 PROBLEMATIKA PŘÍMÉ A INVERZNÍ KINEMATIKY

V předchozí kapitole byly několikrát zmíněny pojmy jako inverzní či přímá kinematika. Zatím ale nebylo vysvětleno, co tyto pojmy znamenají. Proto si je pojdme vysvětlit. Kinematika pro robotiku se zabývá pohybem vzájemně spojených nepružných struktur. Pod tímto krkolomným pojmem si představme obyčejnou lidskou ruku. Ta je tvořena nepohyblivými částmi (kosti) a ty jsou mezi sebou propojeny klouby. A právě klouby umožňují pohyb celých rukou v prostoru. Prostor ale musí být nějak definován, většinou matematicky pomocí souřadnic. Potom říkáme, že se pohybujeme v určitém souřadnicovém systému (prostor).

A podobně jako u lidské ruky je realizován pohyb u naší robotické ruky. Obsahuje pevné části, klouby (realizovány pomocí ložisek) a pohon (servomotor). Náš typ ruky se pohybuje v souřadnicovém systému X, Y a Z. Každá osa řeší pohyb do jiné strany. A pozice v prostoru se určuje kombinací všech těchto tří číselných hodnot (tomu říkáme vektorový součet. Matematický popis pohybu řeší různé druhy kinematiky.

### 6.1 PŘÍMÁ KINEMATIKA

Vysvětleme si obecný princip. V tomto druhu kinematiky postupně určujeme pozici či natočení jednotlivých kloubů celé soustavy (v našem případě robotické ruky se 6 klouby). Postupným vektorovým součtem všech pozic kloubů získáme koncovou polohu konečného stupně ruky.

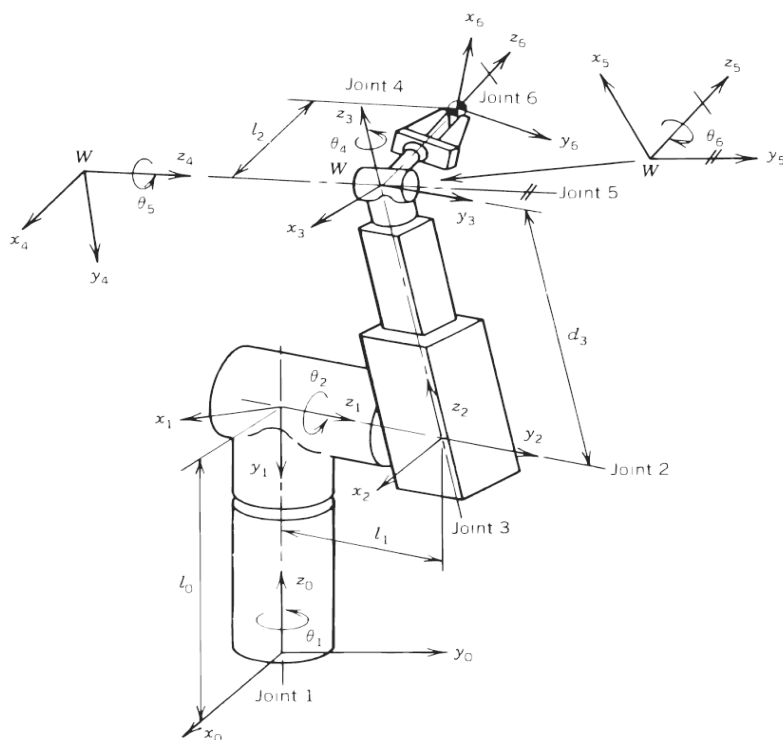
Samotný princip je velice jednoduchý. Postupně natáčíme jednotlivé osy tak, abychom došli k cíli (např. uchopení předmětu). Když toto aplikujeme na robotickou ruku, tak v aplikaci **Moves** programu RIOS tento druh kinematiky implementuje mód **Move by joints** (pohyb samostatných os). Když si ale zkusíme realizovat pohyb do určitého bodu, zjistíme, že musíme dopředu předvídat, do jaké pozice každý kloub natočit tak, abychom skutečně došli k cíli. Když se spleteme v jednom kloubu, musíme poté změnit (přepočítat) pozice všech ostatních. Tato metoda je tedy jednoduchá na pochopení, ale už ne ta na realizaci. Proto se dnes spíše používá tzv. Inverzní kinematika.



## 6.2 INVERZNÍ KINEMATIKA

Můžeme rovnou říci, že se v podstatě jedná o opačný postup než v přímé kinematice. Máme tedy v prostoru definován určitý bod X (tomu se říká efektor). A právě do tohoto bodu se snažíme dostat. Musíme ale početně vyřešit, jak natočit jednotlivé klouby celé soustavy, abychom se tam dostali. A to je právě to nejsložitější. Naštěstí za nás toto řeší samotný software RIOS či Flowbotics. Znamená to, že už nemusíme řešit, jak natočit jednotlivé klouby, ale pouze programu řekneme, že chceme, aby se ruka posunula dopředu o 10 cm, a on sám spočte všechny potřebné natočení kloubů a poté je sám provede. Příkladem inverzní kinematiky budiž mód **Move by X, Y and Z** v aplikaci **Moves**. Ještě názornější příklad je v programu **Flowbotics studio**, který zmíníme dále v textu.

Čím větší je počet kloubů, tím je výpočet složitější, neboť do něj musíme zahrnout více proměnných. Navíc program musí zároveň i počítat s omezeními aktuální ovládané soustavy, kalibračními údaji apod. Program RIOS například pro názornost zobrazuje uživateli na schematickém zobrazení ruky elipsy či kruhy, ke kterým se v aktuální poloze může ruka maximálně dostat. Ty se mění v reálném čase podle aktuálních dat pozice ruky.



Obrázek 48 Inverzní kinematika manipulátoru se 6 stupni volnosti<sup>7</sup>

<sup>7</sup> Převzato z: <http://cmp.felk.cvut.cz/cmp/courses/ROB/roblec/inverse6DOF-notecz.pdf>

## 7 ALTERNATIVNÍ SOFTWARE PRO OVLÁDÁNÍ ROBOTICKÉ RUKY

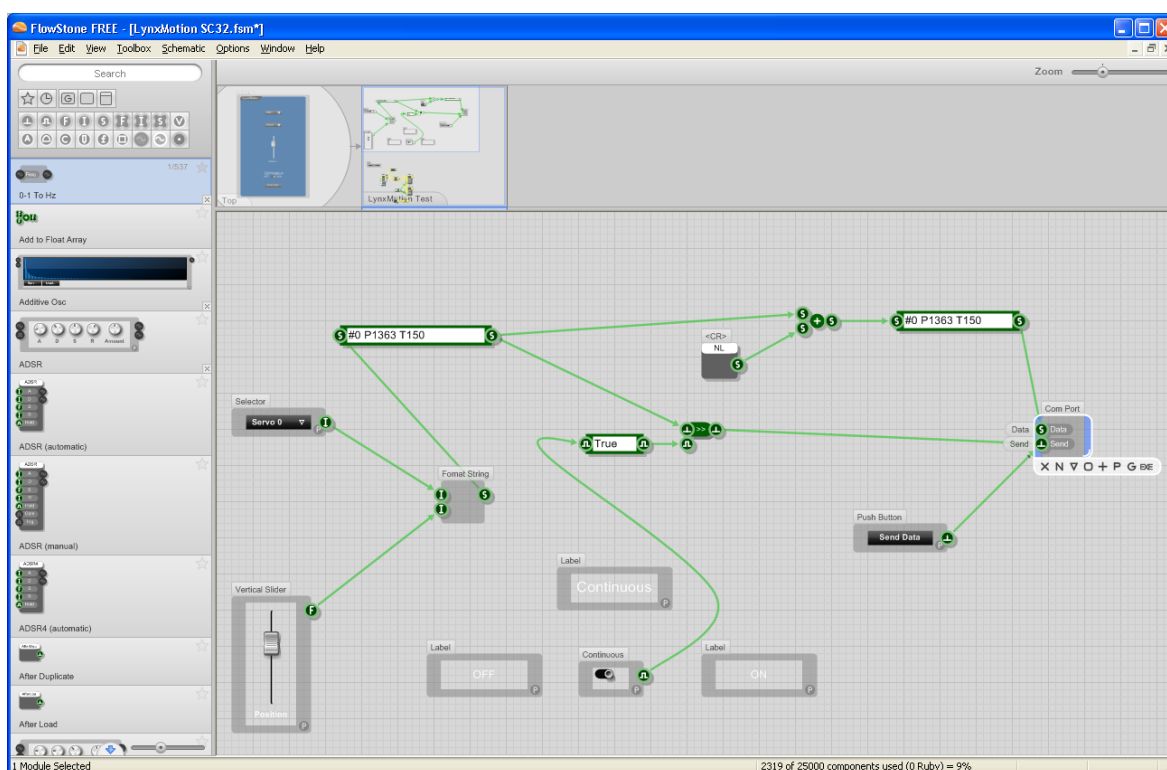
O variantách softwaru, který dodává LynxMotion ke svým stavebnicím, jsme se podrobněji zmínili v kapitole 3. Většinou se jedná o placený software vyvíjen pro účely robotických stavebnic této firmy.

Ale na trhu se vyskytuje i komplexnější software pro ovládání robotů. Hlavní zástupce je Microsoft Robotics Developer Studio 4. Je dostupný ke stažení zdarma na této adrese: <http://www.microsoft.com/robotics/>. Jedná se o komplexní nástroj pro simulaci prostředí robota. Ovšem pro účely základního vzdělávání se jedná o příliš složitý program a vyžaduje pokročilou znalost programovacích jazyků. Tento nástroj se hodí spíše pro univerzitní sféru či profesionální nasazení než pro vzdělávání.

Můžeme ale najít jiný a mnohem zajímavější program – Flowstone.

### 7.1 FLOWSTONE

Nejedná se o jednoúčelový program pro ovládání robotů, ale o komplexní grafické vývojové prostředí. Můžeme ho použít pro vývoj celé řady aplikací s různým nasazením. Např. pro ovládání robotů, automatizaci, DSP signálová centra a další.



Obrázek 49 Grafické programovací rozhraní programu Flowstone

Veškeré programování probíhá pomocí grafických symbolů – to je mnohem logičtější. Tyto symboly představují jednotlivé funkční celky (např. spínač, posuvník, zesilovač, logický člen, PID regulátor a další). Moduly vidíme na ploše vlevo. Ten, který zrovna potřebujeme, přetáhneme na plochu uprostřed. Musíme dobře rozvážit, jaké budeme potřebovat a jak chceme celý program zkonstruovat.

Moduly mezi sebou propojujeme pomocí pomyslných vodičů. Tím určujeme, jak se mezi nimi vyměňují data. Můžeme definovat různé datové typy (Boolean, Text, číslo). Když máme program hotový, můžeme ho zkompilovat do jednoho EXE souboru, který je přenositelný. K běhu vytvořené aplikace tedy nepotřebujeme samotné vývojové prostředí.

V současné době se tento program stává velice oblíbený, hlavně v USA, neboť s trochou cviku může i nezkušený uživatel vytvářet vlastní aplikace.

Nemyslíme si, že žáci na ZŠ by zvládli nějaké složité programování, hodí se spíše na SŠ. Zde se dá využít na výuku číslicové elektroniky, elektroniky apod. Zmiňujeme ho proto, že prakticky všechny volně dostupné aplikace pro ovládání robotů od LynxMotion jsou vyvíjeny v tomto prostředí. Měli bychom proto mít alespoň obecnou představu, o co se jedná.

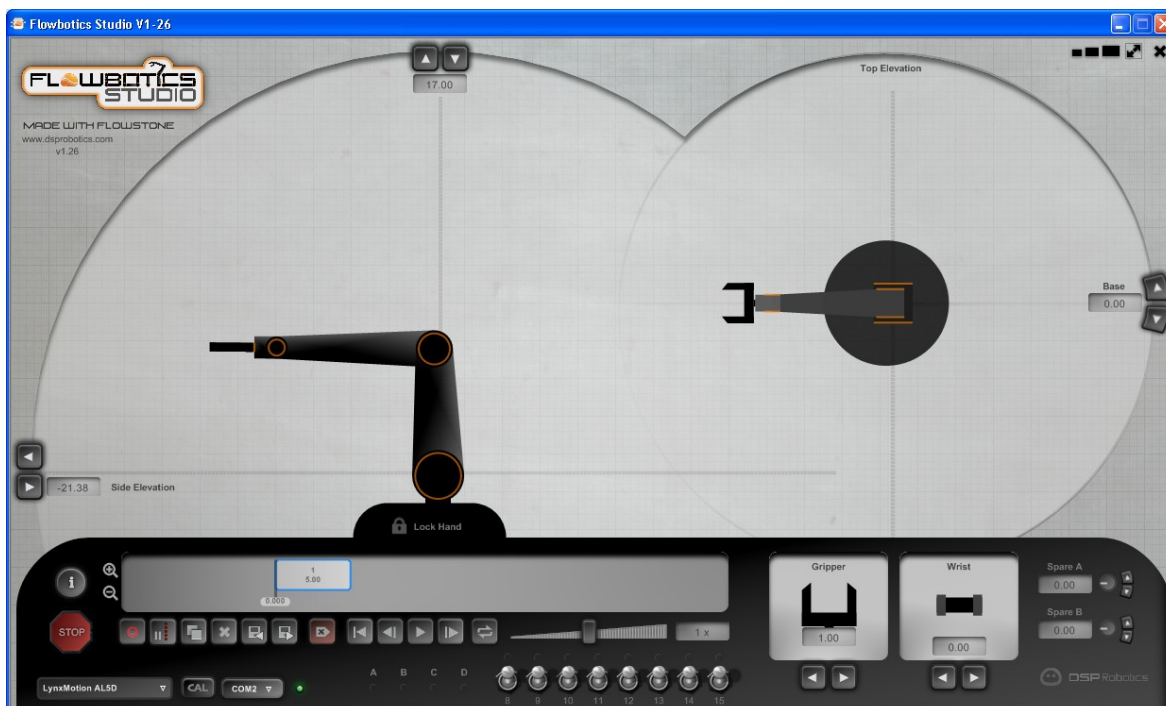
Na obrázku Obrázek 49 vidíme jednoduchý program pro ovládání jednoho kloubu ruky AL5D. I tak je program poměrně složitý a obsahuje mnoho modulů, přičemž některé z nich si musíme sami upravit. Ale myslíme si, že je zde jasně vidět logika programování.

Na formulář rozmístíme ovládací prvky, které poté propojujeme s dalšími moduly (např. sériový port, převodník text na číslo, pulzní ovládání serva atd.) a nastavujeme jejich vlastnosti a výstupy.

Aplikace je v základní verzi volně ke stažení na portálu firmy DSP Robotics: <http://www.dsrobotics.com/download.php>. Obsahuje omezení na počet vložených modulů v aplikaci a nelze kompilovat EXE soubory. Technická podpora a manuály přímo od výrobce jsou zde: <http://www.dsrobotics.com/support/>.

## 7.2 FLOWBOTICS STUDIO

Tato aplikace byla vytvořena ve vývojovém prostředí Flowstone a je dostupná zdarma na adrese <http://www.dsrobotics.com/support/viewtopic.php?f=71&t=416>, a to včetně zdrojových kódů, takže si ji můžeme dále upravovat podle našich potřeb. Umožňuje ovládání robota pomocí myši v reálném čase a je to výborný didaktický nástroj na demonstraci principu funkce inverzní kinematiky a pohybu v prostoru vůbec.




Obrázek 50 Náhled na prostředí aplikace Flowbotics Studio

Je opravdu jednoduchá na ovládání. Napřed si musíme vybrat ve výběrovém poli náš typ robotické ruky (AL5D), program se sám přizpůsobí. Poté zvolíme správný komunikační port a můžeme ruku ovládat -



Samotné ovládání ruky probíhá tak, že klikáme do kružnic, které určují maximální možnou polohu ruky. Konec ruky se přesune na místo, kam jsme klikli. To znamená, že se musela aplikovat metoda inverzní kinematiky pro správné natočení všech kloubů ruky. Ostatní prvky ruky (otočné zápěstí a rukojeť) ovládáme myší pomocí grafických prvků vpravo dole. Otáčení základny ovládáme myší na pravé části obrazovky. Pravým tlačítkem měníme polohu zápěstí, které také můžeme uzamknout v jedné poloze (tlačítko **Lock**).

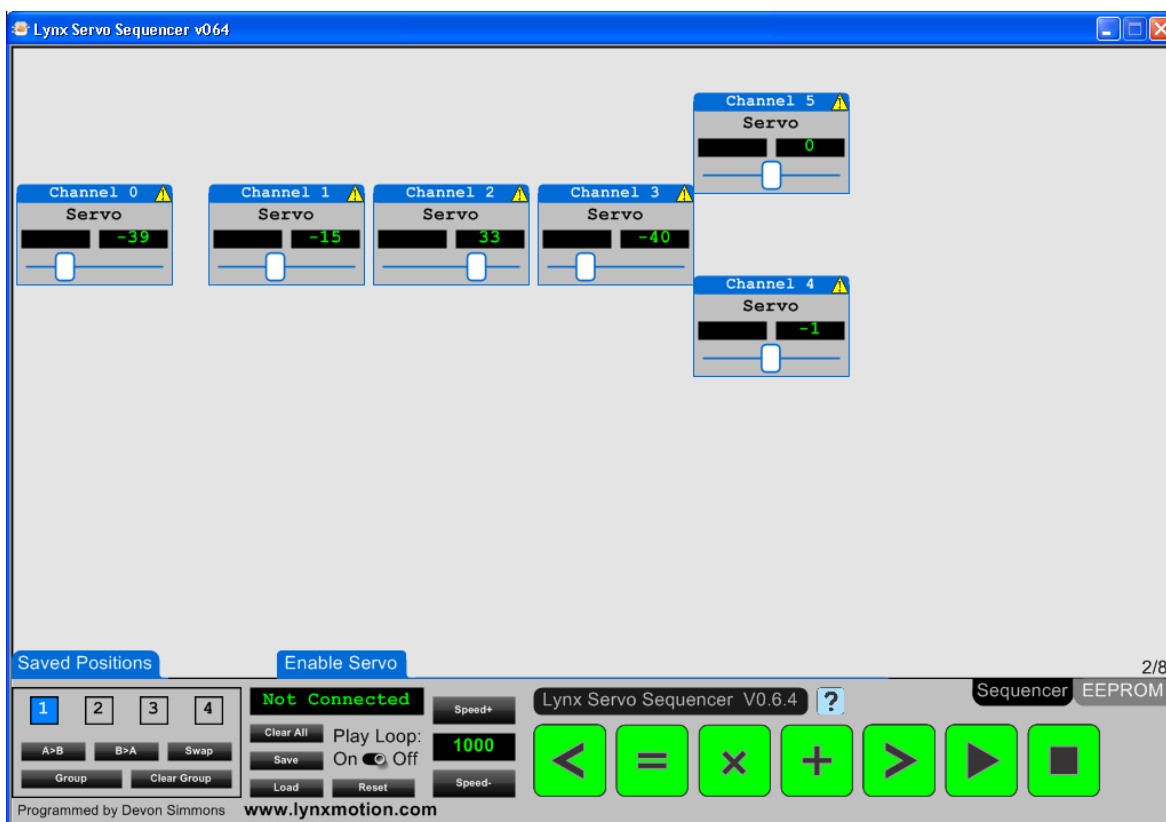
Z jednotlivých poloh ruky můžeme sestavovat sekvenci. K tomu slouží ovládací prvky pod rukou. Každý pohyb má jeden blok, který je závislý v čase. Tím také upravujeme plynulost pohybu.

Program obsahuje i jednoduchou nápovědu a popis ovládacích prvků. Tu si zapneme kliknutím na tlačítko . Tu doporučujeme hned na začátku použít. Rychle objasní všechny funkce programu.

Zároveň tento program doporučujeme hlavně pro začátečníky či mladší děti. Sice nenabízí tolik možností programování jako RIOS, ale má mnohem přehlednější a jednodušší ovládání myši a i nezkušení uživatelé dosahují rychle poměrně dobrých výsledků. Navíc má poměrně líbivé uživatelské rozhraní, které více upoutá pozornost dětí.

### 7.3 LYNX SERVO SEQUENCER

Další aplikace, které je nabízena zdarma a je vytvořena v prostředí FlowStone. Dostupná je na adrese <http://www.lynxmotion.com/p-832-free-download-lynx-servo-sequencer.aspx>. Jedná se o zcela odlišný způsob programování robota.



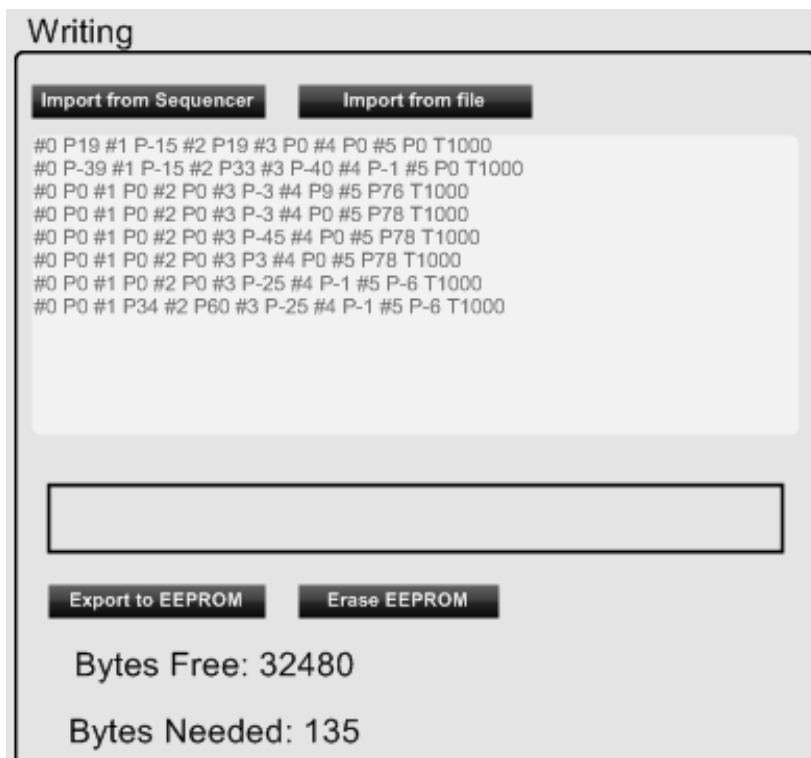
Obrázek 51 Prostředí programu Lynx Servo Sequencer

Na obrázku výše vidíme hlavní pracovní plochu programu. Programování probíhá tak, že každé servo je představováno **jedním blokem**, u kterého nastavujeme jeho polohu (hodnotu). To znamená, že pokud je ruka tvořena 6 servy, musíme zde mít 6 bloků. Ty jsou značeny jako **Channel 0 až 5**. Každá pozice ruky se zaznamená do jedné sekvence. Polohu serva nastavíme pomocí posuvníku v bloku. Doporučujeme provádět změny poloh po menších krocích. Ruka by se jinak mohla dostat do nežádoucího stavu.

Na ovládání tvorby programu slouží zelená tlačítka vpravo dole. Novou sekvenci vložíme tlačítkem **+**. Posouváme se tlačítky **<>** a celý projekt spustíme symbolem **Play**. **X** maže jednu sekvenci. Vidíme, že ovládání je opravdu jednoduché a srozumitelné. Rychlost každé sekvence se nastavuje tlačítky **Speed +** nebo **-**. Tím určíme, kolik dáme každé sekvenci času na to, aby změnila polohu ruky.

Takto sestavíme celý program, který poté můžeme spustit či uložit tlačítkem **Save**. V levé dolní části programu vidíme ještě ovládací prvky pro ukládání sekvencí. Můžeme si uložit sekvenci a tu poté pro usnadnění práce používat i v jiných projektech.

Zajímavá je možnost EEPROM vpravo dole. Tím se otevře okno, které překládá sekvence do formátu kódu, v kterém se posílají data do řídicí jednotky.



Obrázek 52 Program ve formátu pro řídicí jednotku SSC-32

Vidíme, že program se posílá po sekvencích. Každá sekvence obsahuje údaje o polohách každého serva. A jelikož má řídicí jednotka vlastní paměť, můžeme do ní tento program přímo poslat. V našem případě toho nemůžeme nijak využít. Pokud ale používáme jednotky Basic Board Atom Pro či jiné, je to velice užitečná vlastnost.

V této aplikaci si můžeme vygenerovat zdrojový kód projektu a ten poslat do řídicí jednotky. A jelikož má Basic Atom vlastní výpočetní jednotku a překladač jazyka Basic, nemusíme už poté mít robota spojeného s počítačem. Velké využití toto najde např. u stavebnic hexapodů či podvozků pro autonomní pohyb.

Z didaktického hlediska může být tento program užitečný pro žáky v tom, že si uvědomí, jak se jednotlivé pohyby ruky skládají v jeden celek. Pro ty pokročilejší může být zajímavé vidět, jak vypadá zdrojový kód sekvencí, čili jak spolu počítač a robot komunikují.

### **SHRNUTÍ KAPITOLY**

Na předchozích 2 programech jsme si ukázali, že existuje mnoho alternativ pro programování robota. Také jsme viděli, že oba programy mají zcela jiný přístup k samotnému vytváření sekvencí a dokonce využívají rozdílné druhy kinematiky.

Výhodou obou aplikací je to, že jsou zdarma. Od výrobce máme možnost zakoupit robotickou ruku i bez softwaru RIOS. Díky tomu bude o cca 800 Kč levnější. Pro základní práci si s těmito 2 programy vystačíme. Ale pro pokročilejší práci určitě doporučujeme zakoupit i program RIOS. Z hlediska nabídky funkcí nebudeme litovat. Ale rozhodně bude dobré, když budeme všechny tyto programy podle aktuálních požadavků kombinovat. Jen tak dosáhneme pružnosti práce u našich studentů.

## 8 PRAKTICKÉ PŘÍKLADY PRO VYUŽITÍ VE VÝUCE NA ZŠ

V této kapitole jsou uvedeny vybrané příklady, které demonstrují možnosti využití robotické ruky pro výuku. Snažili jsme se o maximální propojení teorie s praxí. Je důležité, aby žáci pochopili, že teoretické znalosti technických oborů jsou důležité pro to, abychom je poté byli schopni aplikovat i v praxi. S některými příklady se žáci mohli setkat na Dnech vědy a techniky v Plzni 2011. Každý bude mít teoretický popis, zadání a metodickou pomoc s cíli pro učitele. Soubory s řešením programů a videa s náhledy správného provedení budou uloženy na přiloženém CD ve složce videa.

### 8.1 SEZNÁMENÍ S ROBOTICKOU RUKOU – ZAMÁVÁNÍ

Jedná se o příklad, které je jednodušší a měl by Vám posloužit pro seznámení s robotickou rukou. Zatím se nebudou vytvářet nějaké složité konstrukce, ale budete pouze pohybovat s rukou v reálném čase. Ti šikovnější si vyzkouší sestavení první sekvence. Pro tyto účely nám dobře poslouží program Flowbotics Studio.

#### ZADÁNÍ

Milí studenti, poprvé se setkáváte s robotickou rukou. Začněte tím, že si ji pořádně prohlédnete. Zkuste pohledem zjistit, kolik má robotická ruka kloubů. Poté se podívejte na svoji ruku a udělejte si srovnání. Kdo má víc – člověk nebo ruka?

Výsledek předejte učiteli. Ten poté ruku zapojí, zapne a spustí Vám ovládací software Flowbotics Studio. Podívejte se na pracovní plochu programu a řádně se s ní seznamte. Vidíte stejný obrázek ruky i na ploše programu. Klikněte tedy myší do plochy.

1. Co se s rukou stalo? Nyní si ověřte, zda jste určili správný počet pohyblivých kloubů.
2. Nyní zkuste docílit toho, aby bylo rameno ruky v pravém úhlu vůči lokti.
3. Naučte se ovládat otočné zápěstí a svírací rukojeť
4. Zkuste rukou zamávat spolužákovi. Výsledek snažení ukažte učiteli.

Pokud jste vše zvládli, zkuste si vytvořit první naprogramovanou sekvenci toho, jak ruka 2x zamává. Napovíme, že sekvence se vytváří pomocí ovládacích prvků jako na vašem DVD přehrávači.



### **METODICKÁ POMOC**

Předpokládá se to, že žáci nemají žádné zkušenosti s programováním ani s robotikou. Proto je dobré, když učitel provede teoretický úvod do robotiky, o souřadnicových systémech apod. Dá se využít mezipředmětové vazby s dalšími předměty jako matematika, informatika, fyzika atd. To záleží na učiteli.

Žáci budou pracovat ve skupinkách a dostanou pracovní listy se zadáním. Cílem příkladu je to, aby se seznámili s robotickou rukou. Zjistí, jakým způsobem se realizuje pohyb, a to porovnájí s lidským tělem. Dalším cílem je, že se naučí pracovat s programem Flowbotics studio, který představuje základní ovládací prvky pro robota. Postupně se naučí s robotem pohybovat v prostoru a nastavit jeho přesnou polohu dle zadání. Ti šikovnější se naučí vytvářet sekvence složené z několika dílčích kroků.

Pro zkušenější či větší studenty můžeme poté zadat, aby podobný příklad realizovali i v softwaru RIOS a poté oba programy porovnali. Navíc u vyšších ročníků můžeme do tohoto příkladu zahrnout i výuku (inverzní) kinematiky.

### **8.2 PŘENESENÍ PŘEDMĚTU**

Na tomto příkladu se důkladně seznámíte s prostředím programu RIOS. Zatím se nebudeme věnovat pokročilejším metodám programování či studování jednotlivým druhů kinematiky. Pouze se naučíme, jak může probíhat tvorba jednoduchého programu pro přenesení předmětu.

#### **ZADÁNÍ**

Představte si, že se stanete hlavním technikem přes roboty v továrně a postaví Vás před úkol, kdy musíte automaticky provést pomocí robotické ruky přenesení předmětu z jednoho místa na druhé. A jelikož je velice objemný, nemůže to provést člověk, ale pouze robot.

V praxi to vypadá tak, že těžký balík přijede na lince na určené místo. Operátor spustí program, který tento balík přenesse na korbu nákladního auta, které stojí na předem určené pozici.

1. Zapněte robotickou ruku a spusťte program RIOS. V něm spusťte aplikaci Moves. Zatím použijte pouze pohybový mód pro jednotlivé klouby – **Move by joints**
2. Na pracovní desce máte namalované body 1 a 2. Z bodu 1 přeneste předmět do bodu 2. Sestavte podle toho správně projekt. Používejte správné členění na projektu na sekvence a kroky.
3. Změřte si vzdálenost páky ruky (pouze loketní část, dolní neuvažujte). Zvažte si předmět, který přenášíte a z hmotnosti vypočtete sílu, která na robotickou ruku působí při zvednutém předmětu.
4. Výsledek ukažte učiteli

#### **METODICKÁ POMOC**

Cílem toho příkladu je seznámit žáky se pokročilejší prací v softwaru RIOS. Naučí se správně vytvářet projekt. Důležité je, aby učitel ohlídal správnou posloupnost vytváření sekvencí a kroků v nich. Měly by být 4. Žáci uvidí, jak se může využívat robotických manipulátorů v praxi a jaké jsou možnosti jejich programování.

V příkladu je zahrnut i výpočet síly na páce, který závisí na délce ramene a hmotnosti zvedaného předmětu. Obsahuje mezipředmětovou vazbu s Fyzikou, předpokládáme tedy návaznost na probírané učivo v tomto předmětu. Díky tomu budou mít žáci teoretické znalosti propojení s praxí a přispěje to upevnění látky.

### **8.3 PŘENESENÍ PŘEDMĚTU S VYUŽITÍM JOYSTICKU**

Zde navazujeme na příklad předchozí. Budeme chtít přenést předmět z jednoho místa na druhé. Avšak nebudeme pro to sestavovat projekt, ale využijeme k tomu periferní zařízení k počítači. Tím může být joystick nebo joypad. V našem případě joystick, který možná někteří z Vás používají k hraní počítačových simulátor. Uvidíte, že může najít i jiné využití, a to i v praxi.

## ZADÁNÍ

V průmyslových provozech se často používá ovládání robotických částí pomocí joysticků. Výhodou tohoto zařízení je celkem jednoduchá manipulace a přirozený pohyb v osách pro uživatele. A vy si právě takovéto ovládání vyzkoušíte sami.

1. Připojte k počítači joystick a otevřete program RIOS. V něm spusťte aplikaci **Moves**. Vidíte, že se spojil s aplikací. Po kliknutí na tlačítko Joystick se podívejte na jeho nastavení. Kolik můžete ovládat os robota na vašem typu zařízení?
2. Nyní si změňte ovládací prvky podle vašich preferencí. Důležité jsou hlavně osy. Ty si přiřaďte tak, aby bylo ovládání pro vás přirozené. Na políčkách v nastavení vidíte přímo náhledy, kam se osy pohybují.
3. Seznamte se důkladně s ovládáním a přeneste zvolený předmět z jednoho místa na druhé. Pro plynulejší a rychlejší pohyb využijte některý z módů inverzní kinematiky – doporučujeme hlavně **Move by distance, Y and Base**.
4. Porovnejte možnosti využití ovládání programem či periferií. U obou analyzujte výhody i nevýhody. Výsledek předvedte učiteli.

## METODICKÁ POMOC

Příklad navazuje na předchozí. Využíváme stejné přípravy a zadání je stejné. Chceme přenést předmět z jednoho místa na druhé. Žáci ale využívají k ovládání joystick. Cílem je, aby se seznámili s ovládáním pomocí periferií. Simulujeme tím reálné podmínky průmyslových provozů (např. Pilsen steel).

Důležité je, aby byl joystick co možná nejkvalitnější a měl 4 samostatné osy. To znamená, že páka musí být otočná – to realizuje osu otáčení základny. Čtvrtou osu pro posun zápěstí většinou řešíme ovládáním plynu na joysticku. Záleží na učiteli, jestli nechá přiřazení kláves na žácích nebo provede nastavení za ně.

V programu RIOS je vhodné používat módy pohybu, které vychází z inverzní kinematiky. Proto by zde měla být návaznost na probrané učivo v matematice, které se týká druhů zobrazování a pohybu v prostoru.

Vyučující musí dávat pozor na to, aby se ruka neponičila, neboť špatným pohybem joysticku se může např. zápěstí zarazit o předmět a nastane přetížení serva či ulomení některé části robota. Je vhodné provést před samotným úkolem kontrolu skupin, zda umí ruku správně ovládat a zda mají správně zkalibrovaný joystick.

### 8.4 SIMULACE PYROTECHNICKÉHO ROBOTA

V dnešní době se můžeme setkat s roboty v různých lidských činnostech. Často se nasazují na činnosti, které jsou pro člověka nebezpečné. Příkladem toho je robot na zneškodňování trhavin. Využívají ho policejní a armádní sbory po celém světě. My si jednoho takového zkusíme nasimulovat. Jelikož nemáme podvozek, který robota dopraví na místo, musíme si ho pouze představit.

#### ZADÁNÍ

Nyní se vžijete do role policejního pyrotechnika, který ovládá dálkově robota pro zneškodnění bomby. Váš robot je už na místě a musíte s pomocí ruky na něm umístěné vytáhnout detonátor z plastické trhavin. Jako v reálném případě budete ruku ovládat pomocí joysticku. Obraz bude přenášén pomocí bezdrátové kamery, která je na ruce umístěna.

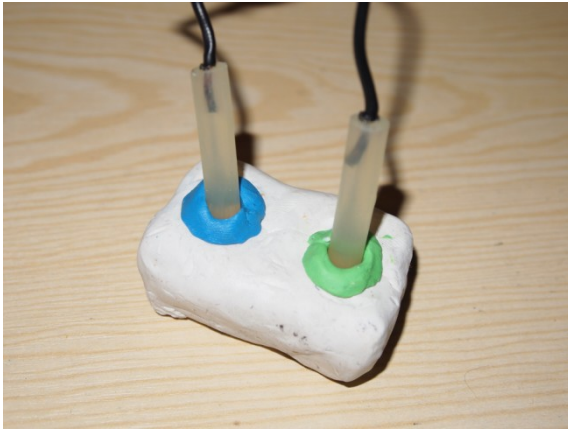
1. Zapojte ruku a spusťte ovládací software RIOS.
2. Zkontrolujte, zda vše v pořádku funguje, zde není čas na chyby.
3. Zapojte bezdrátovou kameru k **LCD televizi** a zkontrolujte, zda vidíte obraz.
4. Pokuste se vyndat detonátor z trhavin. Máte pouze jeden pokus! Musíte se orientovat pouze z obrazu kamery.
5. Detonátory jsou 2. Odstranit ale musíte pouze jeden. Většinou to bývá ten zelený.

#### METODICKÁ POMOC

Jedná se o poměrně komplexní příklad, který slouží k procvičení ovládnání ruky pomocí joysticku. Cílem úlohy je vytáhnout pomyslný detonátor z trhavin a deaktivovat tím bombu. Žáci tím získají příklad reálného využití robota. Navíc úloha obsahuje velký prvek zábavy, což by mohlo mít pozitivní motivační efekt. Doporučujeme, aby se vytvořily týmy. Můžeme uspořádat soutěž o to, který tým dokáže vyndat detonátor nejrychleji.

Dalším cílem je to, že se žáci seznámí s bezdrátovým přenosem obrazového signálu. Zde záleží na učiteli, do jaké hloubky je chce s uvedeným tématem seznámit. Čím vyšší ročník, tím více můžeme najít mezipředmětových vazeb s fyzikou, hlavně s výukou elektromagnetických polí či optiky.

Jelikož se zde používá poměrně složité pomůcky, učitel by se měl důkladně na hodinu připravit a všechny dílčí prvky vyzkoušet.



Obrázek 53 Imitace plastické trhaviny s rozbuškou



Obrázek 54 Sestava pro ovládání ruky pomocí bezdrátové kamery a joysticku

## 8.5 VYKONÁNÍ POSLOUPNOSTI PŘÍKAZŮ S VYUŽITÍM CYKLŮ

Navážeme na první příklad, kde jsme se seznámili s ovládáním robotické ruky. Úkolem bylo zamávat uživateli. Jelikož byl určen pro začátečníky, využívalo se jiného a odlišného softwaru a navíc příklad nebyl nijak programově ošetřen. Proto se seznámíme se dvěma druhy cyklů – **for** pro určitý počet opakování a **do...while** pro opakování na základě splnění podmínky, a původní příklad tak rozšíříme.

### ZADÁNÍ

Představte si, že potřebujete zajistit to, aby robotická ruka prováděla určité sekvence vícekrát. Šlo by i sekvenci vytvořit vícekrát ručně, ale to je složité. Pro tento případ se používají tzv. **cykly**.

Cyklus **for** slouží k opakování jedné sekvence vícekrát podle našeho zadání. Cyklus **do...while** opakuje sekvenci podle zadaného kritéria. Kritérium tohoto cyklu bude infračervení čidlo. Znamená to tedy, že se bude opakovat, dokud se k čidlu nepřiblížíme a program to nevyhodnotí jeho splnění. IR čidlo se používá i v reálných podmínkách k měření vzdálenosti.

1. Zapojte ruku, ověřte si programu RIOS, zda IR čidlo funguje. Mělo by být na 1. vstupu. Pokud je skenování vypnuto – zapněte ho.
2. Vytvořte projekt pro mávání na uživatele. Ten bude mávat na obě strany. Napřed zamává napravo. Tam provede mávací proces **3x**. Poté se otočí nalevo. Zde bude mávat tak dlouho, dokud mu min **2** uživatelé neodpoví, tzn., dokud se k ruce nepřiblíží na vzdálenost menší než **15 cm** – použijeme poprvé IR senzor.
3. Nejprve sestavte program teoreticky – dobře si rozmyslete, kolik budete potřebovat sekvencí pro správné splnění projektu. Ukažte návrh učiteli, který ho schválí.
4. Z příloženého grafu odečtete hodnotu IR čidla pro vzdálenost 15 cm, kterou musíte zadat do podmínky **do...while**.
5. Kolik tedy budete potřebovat podmínek a kde je budete vkládat? Konzultujte s učitelem.

**METODICKÁ POMOC**

Jde o komplexní příklad na složitější konstrukce projektů s oběma druhy využitelných cyklů. Žáci se zde poprvé setkávají s IR čidlem pro měření vzdálenosti, které interaktivitu práce s robotickou rukou dále zvětšuje. Jelikož se jedná o nové a poměrně složité téma na pochopení, doporučujeme tento příklad zařazovat do vyšších a zkušenějších ročníků. Zvláště pak cyklus **do...while** dělá žákům obvykle problémy.



Obrázek 55 Správné řešení úlohy

Cílem je to, aby žáci pochopili rozdíl mezi těmito dvěma druhy cyklů a zároveň aby věděli, proč se používají. Jelikož je konstrukce cyklů podobná jako u oficiálních programovacích jazyků, je dobré, abychom žákům ukázali i jiné příklady. Vhodná je kombinace s výukou programování např. v prostředí Baltazar či želví grafika (Logo).

V příkladu se vyskytuje i práce s grafem, neboť musí odečíst pro čidlo správnou hodnotu pro vzdálenost **15cm**. Mezipředmětové vazby se dají najít ve fyzice, kde by mohl učitel vysvětlit princip funkce IR čidla.

Učitel by měl být dobře připraven a doporučujeme před samotnou realizací provést teoretický úvod s praktickými ukázkami správné konstrukce cyklů...

## 8.6 PŘENESENÍ PŘEDMĚTU S VYUŽITÍM PODMÍNKY IF

V tomto příkladu využijeme kombinaci cyklů, podmínek If i hodnot z IR senzoru. Pokusíme se totiž nasimulovat ošetření chybového stavu robotické ruky na výrobní lince. Ruka bude přesouvat předměty z pásu podle vytvořených sekvencí a senzor bude hlídat, jestli nenastala chyba. V tom případě se činnost ruky zastaví.

### ZADÁNÍ

Představte si, že jste řešiteli projektu u automatizované výrobní linky. Pomocí robotické ruky musíme přenést předměty z pásu do nákladního auta. Pomocí IR senzoru budeme hlídat, zda nenastal chybový stav – čili jestli se nestala na pracovišti nehoda. Do automobilu se vejde 5 2tunových předmětů. Chybový stav znamená, že se na lince vzpříčil či převrhl nějaký předmět, který senzor detekuje a okamžitě se musí zastavit činnost linky.

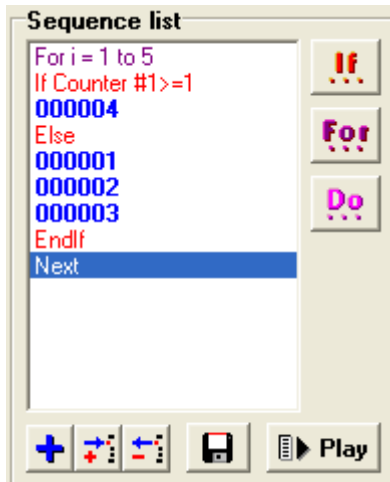
1. Zapněte ruku a zkontrolujte funkčnost IR senzoru.
2. Pokuste se teoreticky sestavit rozvržení programu. Jedná se o komplexní úlohu, kde budeme využívat jak cyklů, tak podmínky `if`. Z teorie byste měli znát jejich druhy. Zkuste navrhnout jejich správné konstrukce. Jaký druhy cyklu použijete? Jak bude vypadat podmínka `if`?
3. Z hodnoty v grafu odečtete vzdálenost senzoru od předmětu 30cm.
4. Po schválení učitelem vytvořte projekt a nechte ho zkontrolovat vyučujícím.

### METODICKÁ POMOC

Komplexní příklad, kde se pro správné řešení využívá téměř všech možností modelu robotické ruky AL5D. V projektu je základní činnost realizována podmínkou `for`, která opakuje vybrané sekvence `5x` (do naložení automobilu). Pomocí podmínky `if` kontrolujeme chybový stav. Musíme nastavit nízkou hodnotu vzdálenosti, záleží na konstrukci čidla a jeho umístění – lepší vyzkoušet na konkrétním modelu.

Podmínkou `if-else-endif` se bude řešit funkce samotného čidla a počet opakování přenesení předmětu (`5x`). V cyklu `for` bude vložena podmínka `if`, která kontroluje hodnotu čidla. Když překročí mez, podmínka `if` dalším cyklu skočí sekvenci 000004 a provede se další cyklus snížený o 1. V opačném případě se provedou sekvence za příkazem `else`.





Obrázek 56 Správné řešení úlohy

Vhodné by bylo propojit výuku s programováním v informatice či jinými předměty, kde se nabízí tato problematika. Jelikož se snažíme simulovat reálné podmínky výroby, mohli bychom uspořádat exkurzi do továren s využitím mechatroniky či přehrát dokumentární film, aby žáci viděli, jak vypadá mechatronika v reálu.

Doporučujeme před realizací příkladu provést důkladnou teoretickou výuku o tvorbě a smyslu cyklů a podmínek.

## 9 ZÁVĚR

Cílem této práce bylo ukázat možnosti, které nabízí stavebnice robotické ruky od firmy Lynxmotion. V dnešní době je potřeba v dětech podnítit zájem o technickou složku vzdělávání a v moderním školství se neustále hledají nové didaktické nástroje, který by žáky tímto směrem aktivizovaly. Robotika jistě patří mezi ně.

V první části jsme se pokusili najít v současných českých školských dokumentech prostor, zda vůbec můžeme robotiku na ZŠ vyučovat. Zjistili jsme, že je to možné a to hned v několika předmětech (Fyzika, Informatika, Technická výchova, Matematika). Vytvořili jsme tedy doporučení, kdy a jak toto realizovat v českém základním školství.

V další části se zabýváme nabídkou robotických stavebnic vhodných pro výuku se zaměřením na modely firmy Lynxmotion. Ty mají výhodu, že je můžeme kombinovat a tudíž i rozšiřovat jejich možnosti, což jsme si i ukázali.

Dále se zabýváme samotnou konstrukcí konkrétního typu ruky, včetně několika vlastních vylepšení a přidaných senzorů. Na ruku jsme připevnili bezdrátovou kameru a infračervené čidlo pro měření vzdálenosti, u kterého jsme provedli měření pro odečtení správných hodnot z grafu.

Další část se zabývá softwarem RIOS, který je spolu s rukou dodáván. Je provedena analýza jeho možností s důrazem na funkce, které by mohl učitel často pro výuku využívat. Tato část je koncipována jako průvodce pro učitele, který se s tímto programem ještě nesetkal. Zahrnut je také kvalitní a volně dostupný alternativní software, který nabízí alternativu k výše zmíněnému.

V poslední části jsme se pokusili vytvořit několik vzorových příkladů, které využívají všechny možnosti robotické ruky. Zmínit můžeme např. přesouvání předmětů, ovládání na dálku pomocí kamery apod. Každý příklad obsahuje úvod, zadání pro studenty a metodické pokyny pro učitele. Pro větší názornost je ke každému příkladu natočeno video se správným řešením.

Po přečtení této práce by se měl čtenář orientovat v možnostech nasazení robotiky pro výuku na ZŠ s důrazem na obsluhu robotické ruky Lynxmotion AL5D spolu se softwarem RIOS, včetně tvorby didakticky správných příkladů.

**10 SEZNAM OBRÁZKŮ**

Obrázek 1 Náš model již sestavené ruky .....	13
Obrázek 2 Vlajková loď hexapodů .....	13
Obrázek 3 Ukázka vynikající modularity stavebnic Lynxmotion .....	14
Obrázek 4 Řídící jednotka s procesory Basic Atom .....	16
Obrázek 5 Všechny součásti robotické ruky po vybalení z krabice .....	17
Obrázek 6 Správné vedení kabelů hlavního svazku .....	20
Obrázek 7 Problematické místo na zápěstí .....	20
Obrázek 8 Schéma zapojení IR senzoru .....	22
Obrázek 9 IR senzor a vyrobený kabel pro připojení .....	22
Obrázek 10 Naměřené hodnoty z IR senzoru .....	23
Obrázek 11 Celá sestava senzoru a kamery upevněná na ruce .....	24
Obrázek 12 Nefunkční redukce .....	25
Obrázek 13 Tato redukce fungovala bez chyb .....	25
Obrázek 14 Nastavení přenosu pro FTDI .....	26
Obrázek 15 Nastavení pokročilých možností ovladače .....	27
Obrázek 16 Redukce od Omron electronics .....	27
Obrázek 17 Nastavení rychlosti přenosu .....	28
Obrázek 18 Prostředí programu LynxTerm .....	29
Obrázek 21 Schéma připojení na řídicí desce SSC-32 .....	31
Obrázek 19 USB Bluetooth adaptér .....	31
Obrázek 20 modul connectBlue OBS410i .....	31
Obrázek 22 Schéma zapojení kabelů od modulu connectBlue .....	32
Obrázek 23 Kompletní modul připravený k zapojení .....	33
Obrázek 24 Zobrazení komunikačního portu pro adaptér .....	34
Obrázek 25 Nastavení Bluetooth COM portu .....	34
Obrázek 26 Úvodní obrazovka programu RIOS .....	35
Obrázek 27 Dotaz na inicializaci do základní polohy serv .....	37
Obrázek 28 Nástroj pro kalibraci, všechna serva jsou vypnutá (položka Enable není zaškrtnutá) .....	37
Obrázek 29 Nastavení maxima u ramenního kloubu .....	38
Obrázek 30 Pomocí šipek měníme hodnotu maxima .....	39
Obrázek 31 Zde vidíme kompletně provedenou kalibraci pro náš typ robotické ruky. Všimněte si rozdílů oproti obrázku 28, kde je základní nastavení. ....	40
Obrázek 32 Při dalším spuštění programu se otevře okno, v kterém musíme vybrat vytvořený konfigurační soubor .....	40
Obrázek 33 Program ARM slouží k nastavení důležitých parametrů .....	41
Obrázek 34 Okno pro práci s hotovými projekty .....	42
Obrázek 35 Hlavní okno aplikace Moves .....	44
Obrázek 36 Volby pro postupné vytváření projektu .....	45
Obrázek 37 Reakce programu na vstupní senzor .....	47
Obrázek 38 Zobrazení os pro robotickou ruku .....	49
Obrázek 39 Příklad vhodně nastaveného joysticku v programu RIOS .....	51
Obrázek 40 Okno Play .....	52
Obrázek 41 nabídka cyklu For .....	53

Obrázek 42 Správná konstrukce cyklu Do .....	54
Obrázek 43 Jedna z možností konstrukce IF – Else .....	55
Obrázek 44 Jiná konstrukce If kombinovaná s cyklem For .....	56
Obrázek 45 Získaný obraz prostoru před rukou - bílý pruh je tužka vložená před senzor..	57
Obrázek 46 Nastavení a spuštění Socket serveru .....	58
Obrázek 47 Socket klient.....	59
Obrázek 48 Inverzní kinematika manipulátoru se 6 stupni volnosti .....	61
Obrázek 49 Grafické programovací rozhraní programu Flowstone .....	62
Obrázek 50 Náhled na prostředí aplikace Flowbotics Studio .....	64
Obrázek 51 Prostředí programu Lynx Servo Sequencer .....	65
Obrázek 52 Program ve formátu pro řídicí jednotku SSC-32 .....	66
Obrázek 53 Imitace plastické trhaviny s rozbuškou .....	73
Obrázek 54 Sestava pro ovládání ruky pomocí bezdrátové kamery a joysticku .....	73
Obrázek 55 Správné řešení úlohy.....	75
Obrázek 56 Správné řešení úlohy.....	77

**11 SEZNAM LITERATURY**

1. **VÚP.** *Rámcový vzdělávací program pro základní vzdělávání.* Praha : VÚP, 2007.
2. Botball Educational Robotics. *EUROPEAN CONFERENCE ON EDUCATIONAL ROBOTICS.* [Online] IEEE, 2012. [Citace: 29. květen 2012.] <http://ecer12.pria.at/index.php?page=home>.
3. **Jurča, Jaroslav.** V Písku u Jablunkova se děti učí fyziku s roboty z Lega. *Skupina ČEZ - Média.* [Online] 2011. [Citace: 29. květen 2012.] <http://www.cez.cz/cs/pro-media/tiskove-zpravy/3560.html>.
4. **Písek, SPŠ a VOŠ.** Aplikovaná robotika. *SPŠ a VOŠ Písek.* [Online] SPŠ a VOŠ Písek, 2009. [Citace: 29. květen 2012.] <http://www.sps-pi.cz/projekty/aplikovana-robotika/>.
5. Dětské robotické stavebnice. *ICT Školám.* [Online] 2011. [Citace: 29. květen 2012.] <http://ict-skolam.eu/?link=7>.
6. TERECoP. *TERECoP.* [Online] SOCRATES PROGRAMME, COMENIUS, 2009. [Citace: 29. KVĚTEN 2012.] <http://www.terecop.eu/>.
7. **Lynxmotion, Inc.** Lynxmotion guides. *Lynxmotion.* [Online] 2010. [Citace: 30. květen 2012.] <http://www.lynxmotion.com/driver.aspx?Topic=assem01>.
8. —. Lynxmotion Tech Support. *Lynxmotion.* [Online] 2010. [Citace: 31. květen 2012.] <http://www.lynxmotion.net>.
9. **RCR.** *Robot revue : magazín ze sveta robotiky.* Praha : RCR, 2009.
10. **Novák, Petr.** *Mobilní roboty : pohony, senzory, řízení.* Praha : BEN, 2005. ISBN 80-7300-141-1.
11. **Geršl, Vladimír.** *Inverzní kinematika se zaměřením na herní aplikace.* Plzeň : FAV ZČU, 2008.
12. **Smutný, Vladimír.** Inverzní kinematika manipulátoru se 6-ti stupni volnosti a řemi protínajícími se osami. *Výukový portál robotiky.* [Online] 2008. [Citace: 31. květen 2012.] <https://cw.felk.cvut.cz/doku.php/courses/a3b33rob/prednasky>.

## 12 RESUMÉ

This thesis deals with the problematic of using the robotic arm LynxMotion Company for teaching on the secondary schools. It is also framed to be used as a guide for teachers of technical subjects, who have never met with the robotic. In today's times students need to get the positive approach to the technical subjects and the using of the robotics and similar interactive didactical tools could it achieve.

The first part of the thesis makes an analysis of the RVP for secondary schools in order to find the subjects, in which could be robotics use. The positive finding is that the teaching of the robotic could be included in several educational areas of RVP. They are "The human and the world of work", "Informative and communicative technologies" and "The Human and the nature". Within these areas are educated more subjects, so we deal with the inter-subject relations with the application of the robotic to the usual teaching. This describes the first part of the thesis. The consequential part tries to make an analysis of possibilities of the robotic construction set of the LynxMotion Company, including the construction itself of the particular arm AL5D. The chapter is dedicated to the most common problems, which may occur during the construction or communication with the PC.

In the next part is described the programming environment of the programme RIOS, which is supplied with the robotic arm. The emphasis is placed on the important elements in terms of programming didactic. Since this is the complex tool, we try to describe only parts, which could the teacher use in the teaching. We also try to make an analysis of the alternative freeware software that could be used for teaching with the robotic arm.

The last part contains several sample exercises of using the robotic arm, which have the potential to attract the children's attention to these problems. For example, the moving of object, the different reactions of the arm depending on the infra-red sensor or remote control that simulates the police pyrotechnical robot. Some of the examples have been tested in practice on the Days of technique in Pilsen in 2011.

## 13 PŘÍLOHY

Na přiloženém CD jsou uložena videa, která jsme natočili pro lepší názornost toho, jak by měly být příklady realizovány.

Ve složce *RIOS* je uložen konfigurační soubor kalibrace pro konkrétní typ robotické ruky, kterou jsme měli k dispozici a soubor databáze, kde jsou uloženy projekty, které jsme vytvořili. Přiložen je také soubor ve formátu MS Excel 2007, kde jsou výsledky měření IR senzoru včetně grafu.

Ve složce *Programy* je umístěn volně šiřitelný software, který jsme také využívali pro vybrané příklady.