



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

## VÝVOJ A NÁVRH NÍZKONÁKLADOVÉHO MANIPULÁTORU PRO INTERAKCI S OKOLÍM

DEVELOPMENT AND DESIGN OF LOW COST AND ENVIRONMENT INTERACTION MANIPULATOR

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Vojtěch Štěpánek

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Radek Knoflíček, Dr.

BRNO 2018

## Zadání diplomové práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Student:	<b>Bc. Vojtěch Štěpánek</b>
Studijní program:	Strojní inženýrství
Studijní obor:	Výrobní stroje, systémy a roboty
Vedoucí práce:	<b>doc. Ing. Radek Knoflíček, Dr.</b>
Akademický rok:	2017/18

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

### Vývoj a návrh nízkonákladového manipulátoru pro interakci s okolím

#### Stručná charakteristika problematiky úkolu:

Cílem diplomové práce je navrhnout takový manipulátor, který nepatří do typů běžně nabízených ke komerčním účelům, což jsou tzv. průmyslové. Návrh netypizovaného a nestandardního robotu, jež je podřízen účelu a použití a také případným nákladům na výrobu prototypu ve vývojové a prototypové dílně zadavatele DP (tj. za cenu minimalizace nákladů na pořízení komponent a výrobu).

#### Cíle diplomové práce:

1. Zhodnocení současného stavu v oblasti vývoje a výroby manipulátorů nebo robotů, které jsou určeny pro interakci s člověkem (např. pro univerzitní výzkum nebo v oblasti tzv. zábavné robotiky)
2. Návrh koncepce přiměřeně složitého manipulátoru, zohledňující tyto parametry:
  - a. Počet stupňů volnosti a morfologie
  - b. Dosažitelná přesnost polohy
  - c. Velikost pracovního prostoru
  - d. Vlastní rozměry a hmotnost
  - e. Druhy pohonů a převodů
  - f. Autonomnost chování z hlediska výbavy senzorickým a řídicím systémem,to vše za předpokladu dobré dostupnosti komponent pořízených nákupem i popřípadě vlastní výrobou
3. Tvorba konstrukční dokumentace návrhu robotu
4. S přihlédnutím k účelu a použití robotu návrh jedné nebo více úloh, včetně pracovního cyklu manipulace
5. Koncový efektor manipulátoru pro předmět manipulace volte nákupem, popřípadě vlastním ideovým návrhem
6. Odhad nákladů na případnou výrobu jednoho kusu manipulátoru
7. Řešení bezpečnosti modelového pracoviště (včetně a analýzy rizik celku i komponent)

**Seznam doporučené literatury:**

KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. První vydání. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016. ISBN 978-80-214-4828-5.

SHIGLEY, Joseph Edward, Charles R. MISCHKE a Richard G. BUDYNAS, VLK, Miloš (ed.). Konstruování strojních součástí. 1. vyd. Přeložil Martin HARTL. V Brně: VUTIUM, 2010. Překlady vysokoškolských učebnic. ISBN 978-80-214-2629-0.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2017/18.

V Brně, dne 27. 10. 2017



doc. Ing. Petr Blecha, Ph.D.  
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **ABSTRAKT**

Tato práce se věnuje návrhu nízkonákladového robotického manipulátoru typu SCARA. Práce je sestavena chronologicky podle dílčích vývojových stádií manipulátoru. Představí metody z oblasti inverzní kinematiky, které určují vztah mezi natočením kloubů a souřadnicemi koncového efektoru. Dále práce představí vývojové diagramy pro řízení celého zařízení a přehledně vysvětlí řídicí software pro mikro kontrolér Arduino MEGA.

## **ABSTRACT**

This thesis is focused in design low cost robotic manipulator known as SCARA. Chapters are sorted chronological by degrees of manipulator development. Thesis will present methods of inverse kinematics, that determines intaraction between joint rotation and cartesian coordinates of gripper. Next it introduces flowcharts for controlling machine and explains the control software determinater especially for microcontroller called Arduino MEGA.

## **KLÍČOVÁ SLOVA**

Robotický manipulátor, SCARA robot, Arduino, servisní robotika, inverzní kinematika

## **KEYWORDS**

Robotic manipulator, SCARA robot, Arduino, service robot, inverse kinematics

## **BIBLIOGRAFICKÁ CITACE**

ŠTĚPÁNEK, V. *Vývoj nízkonákladového manipulátoru pro interakci s okolím*, Brno, Vysoké učení technické v Brně, Fakulta strojního inženýrství. 2018, 85 s., Vedoucí diplomové práce doc. Ing. Radek Knoflíček, Dr.

## **PODĚKOVÁNÍ**

Velké poděkování patří doktorovi Radkovi Knoflíčkovi za cenné rady a postřehy při sestavování osnovy celé práce, pomoc během výběru optimálního kinematického uspořádání robotu a konzultace při provádění výpočtů a programování kompletního zařízení. Dále bych chtěl poděkovat svým rodičům za podporu během celého studia a za jejich rady a myšlenky během dříve absolvovaných projektů z oblasti konstruování hobby obráběcích strojů, které přispěli dosažení co nejlepších výsledků.

## **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že tato práce je mým původním dílem a zpracoval jsem ji samostatně pod vedením doc. Ing. Radka Knoflíčka, Dr.. Všechny použité zdroje jsou uvedeny v seznamu na konci této práce. Hypotézy, jež se neodkazují na znalosti získané během studia nebo na základě literatury, jsou potvrzeny nebo vyvráceny sestavením funkčního prototypu zařízení, jehož vývoj je v práci detailně popsán.

V Brně dne 22.5.2018

.....  
Vojtěch Štěpánek

# OBSAH

<b>ÚVOD</b> .....	<b>15</b>
<b>MOTIVACE</b> .....	<b>17</b>
<b>1 DEFINICE MANIPULÁTORU</b> .....	<b>19</b>
<b>2 ROZDĚLENÍ MANIPULÁTORŮ</b> .....	<b>19</b>
2.1 Mechanické manipulátory bez aktivních pohonů .....	19
2.2 Elektro-mechanické balancery.....	20
2.3 Teleoperátory .....	21
2.4 Automatické manipulátory s tvrdým řídicím programem .....	22
2.5 Automatické manipulátory s pružným řídicím programem.....	22
<b>3 POUŽÍVANÉ KINEMATICKÉ DVOJICE MANIPULÁTORŮ</b> .....	<b>23</b>
3.1 Translační kinematické dvojice .....	23
3.1.1 Realizace translační kinematické dvojice.....	24
3.2 Rotační kinematické dvojice.....	24
<b>4 POUŽÍVANÉ KOMBINACE KINEMATICKÝCH DVOJIC</b> .....	<b>25</b>
<b>5 ROBOTIKA OKOLO NÁS</b> .....	<b>29</b>
5.1 Servisní roboty .....	29
5.2 Servisní roboty ve zdravotnictví .....	29
5.3 Servisní roboty v bezpečnostních složkách .....	31
5.4 Humanoidní roboty .....	35
5.4.1 Robot ASIMO .....	35
5.5 Servisní roboty v domácím prostředí.....	37
5.5.1 Funkce inteligentního domu .....	37
5.5.2 Spotřebiče inteligentního domu.....	38
5.6 Robotický mazlíček AIBO .....	39
5.7 Robotické vysavače .....	40
<b>6 NÁVRH MANIPULÁTORU URČENÉHO PRO INTERAKCI S OKOLÍM</b> .....	<b>41</b>
<b>7 VÝBĚR VHODNÉ KINEMATIKY PRO DANOU APLIKACI</b> .....	<b>41</b>
<b>8 ROZBOR KONSTRUKČNÍCH ČÁSTÍ</b> .....	<b>43</b>
8.1 Podložka .....	43
8.2 Rozmístění krokových motorů .....	43
8.3 Velké rameno .....	43
8.4 Malé rameno .....	44
8.5 Výsuvný píst s přísavkou .....	44
8.6 Gravitační zásobník .....	44
<b>9 POUŽITÉ TECHNOLOGIE PRO VÝROBU</b> .....	<b>45</b>
<b>10 VÝPOČET NEPŘÍMÉ KINEMATIKY MANIPULÁTORU</b> .....	<b>47</b>
10.1 Výpočet pomocí explicitní metody.....	47
10.2 Výpočet pomocí metody transformační matice .....	49
<b>11 VÝVOJOVÉ DIAGRAMY FÁZÍ PROGRAMU</b> .....	<b>51</b>
11.1 Podoba digitálních číslic a indexy jednotlivých bodů .....	52
11.2 Čtení hodnoty času z modulu DC3231 .....	52
11.3 Sestavení vektorů požadovaných přesunů .....	53
11.4 Plánování přesunů kuliček mezi pozicemi .....	54

11.5	Krokový motor .....	54
11.6	Výpočet kroků potřebných pro přesun mezi souřadnicemi .....	56
11.7	„Paralelní“ řízení krokových motorů .....	56
11.8	Řízení rychlosti krokových motorů .....	58
11.8.1	Konstantní frekvence otáčení .....	58
11.8.2	Zrychlování a zpomalování po rampě .....	58
11.8.3	Zrychlování a zpomalování po S-křivce .....	61
11.9	Referování zařízení po zapnutí .....	61
11.10	Nulování zařízení .....	62
11.11	Schémata zapojení vstupů a výstupů .....	62
11.12	Blokové schéma programu HODINY .....	64
<b>12</b>	<b>RIZIKA BĚHEM POUŽÍVÁNÍ MANIPULÁTORU.....</b>	<b>65</b>
12.1	Úraz elektrickým proudem .....	65
12.2	Spolknutí či vdechnutí malých částí zařízení .....	65
12.3	Úraz skřípnutím mezi pohyblivými rameny .....	65
12.3.1	Řízení rychlosti manipulátoru během krizových situací .....	65
12.3.2	Optická závora .....	66
<b>13</b>	<b>TECHNICKÉ PARAMETRY ZAŘÍZENÍ .....</b>	<b>67</b>
<b>14</b>	<b>FINANČNÍ BILANCE ZAŘÍZENÍ .....</b>	<b>67</b>
	<b>ZÁVĚR .....</b>	<b>69</b>
	<b>SEZNAM POUŽITÉHO SOFTWARE .....</b>	<b>71</b>
	<b>SEZNAM POUŽITÝCH ZKRATEK A PROMĚNNÝCH .....</b>	<b>73</b>
	<b>SEZNAM POUŽITÝCH ZDROJŮ .....</b>	<b>77</b>
	<b>ZDROJE POUŽITÝCH OBRÁZKŮ .....</b>	<b>81</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>83</b>

## ÚVOD

V dnešní době se výrobci obráběcích strojů a celkově automatizovaných výrobních a montážních systémů předhánjí v maximálních dosahovaných rychlostech, v mikronových přesnostech, teplotních kompenzacích, korekcích atd. Všechny výrobní systémy se stávají složitějšími, vyžadují sofistikovaný systém řízení, kvalitní a drahé senzory a moderní konstrukční postupy. Tato práce je však směřována opačným směrem. Popisuje návrh robotického manipulátoru typu RRT se dvěma rotačními klouby a jedním výsuvným pístem s přísavkou. Manipulátor je navržen a sestaven s ohledem na požadavek minimalizace nákladů. Při stavbě jsou použity levné a pro tuto aplikaci dostačující krokové motory a navíc je celé zařízení řízeno pomocí mikro kontroléru Arduino, který sice nedosahuje takového výkonu jako průmyslové řídicí jednotky, ale jeho pořizovací cena je v řádu stokorun.

Celý manipulátor je navržen pro interakci se svým okolím a byla pro něj vytvořena ukázková aplikace z oblasti zábavné robotiky. Manipulátor obsluhuje pracovní prostor 550x350mm a sestavuje kuličky do předem připraveného pole. Za pomoci časové značky řídicí modul rozhodne o podobě jednotlivých číslic a podle toho přeskládá černé kuličky, jež tvoří podobu výsledných číslic. Představený program je dělený na dílčí podprogramy, takže je možné relativně jednoduchým způsobem měnit podprogram s plánováním trajektorie a tedy „naučit“ robota libovolné manipulační aplikaci. Práce by měla sloužit jako učební text i úplným začátečníkům z oblasti konstrukce a programování a měla by poskytovat dostatek klíčových slov pro snazší proniknutí do jádra problému. Dále vysvětluje podstatu robotické transformace v podobě inverzní kinematiky a je doplněna množstvím vývojových diagramů, které představují tok informací během celého řídicího programu.

## MOTIVACE

Motivací pro zvolení daného tématu byla možnost sestavení vlastního funkčního robotu. Zkonstruovat celé zařízení „od nuly“ pro mě bylo velkým přínosem, neboť jsem dostal příležitost vyzkoušet si sestavit celý kinematický řetězec manipulátoru tak, aby bylo možné jednotlivé díly smontovat, obsluhovat požadovaný pracovní prostor a přitom vytvořit relativně malé zařízení, které nebude působit masivním dojmem. Dalším velkým přínosem pro mě byla možnost proniknout do podstaty robotické transformace a vztahů mezi natočením ramen a kartézskou souřadnicí koncového efektoru. Veliké plus také shledávám v praktickém sestavení celého zařízení, jež některé předpoklady potvrzuje, ale některé vyvrací v podobě snížené funkčnosti zařízení v podobě vibrací a jiných negativních projevů. Věřím, že si práce najde své čtenáře, kteří si podle ní taktéž sestaví vlastní nízkonákladový robot.

# 1 DEFINICE MANIPULÁTORU

Podle normy ISO 8373 je průmyslový manipulátor definován takto:

*Manipulátor je stroj, ve kterém mechanismus sestává ze sady členů (segmentů) navzájem spojených otočnou nebo posuvnou vazbou za účelem uchopení a/nebo přemísťování předmětů (obrobků nebo nástrojů), obvykle v několika stupních volnosti.*

*pozn. 1 : Manipulátor může být řízen operátorem, programovatelným elektronickým regulátorem nebo jakýmkoliv logickým systémem (například vačkovým zařízením či zařízením s pevně zapojenou logikou).*

*pozn. 2 : Manipulátor neobsahuje koncový efektor. [5]*

## 2 ROZDĚLENÍ MANIPULÁTORŮ

Manipulátor je strojní zařízení, které slouží zpravidla k přemísťování obrobků, sestavování montážních celků či manipulaci s nástrojem. Manipulátor může být jednoúčelový stroj bez automatického řízení, který je plně závislý na ovládní lidskou obsluhou. Může být vybavený primitivním řídicím systémem, který může fungovat diskrétním způsobem (zapnuto/vypnuto, zasunuto/vysunuto, upnuto/odepnuto) nebo způsobem spojitým (polohování na přesnou pozici, upnutí stanovenou silou, řízení otáček a rychlostí atd.).

### 2.1 Mechanické manipulátory bez aktivních pohonů

Do této oblasti spadají nejjednodušší typy manipulátorů a lze je přirovnat spíše k vyvažovacím zařízením, které v průmyslu ulehčují lidskou práci. Činným prvkem těchto manipulátorů jsou mechanické pružiny, hydraulické písty s ručním pohonem či prostá rovnováha na páce. Z nejjednodušších manipulátorů lze zmínit například paletový vozík, kde hydraulický převod umožňuje zvedat velice těžká břemena, která by lidská síla nebyla schopna uzvednout. Dalším zástupcem může být mechanický pružinový balancer, na kterém bývají zavěšeny různé pomůcky ve výrobních buňkách. Mechanické balancery se vyrábí buď s pevně nastavenou silou pružiny či mohou být konstruovány jako stavitelné pro širší oblast použití.



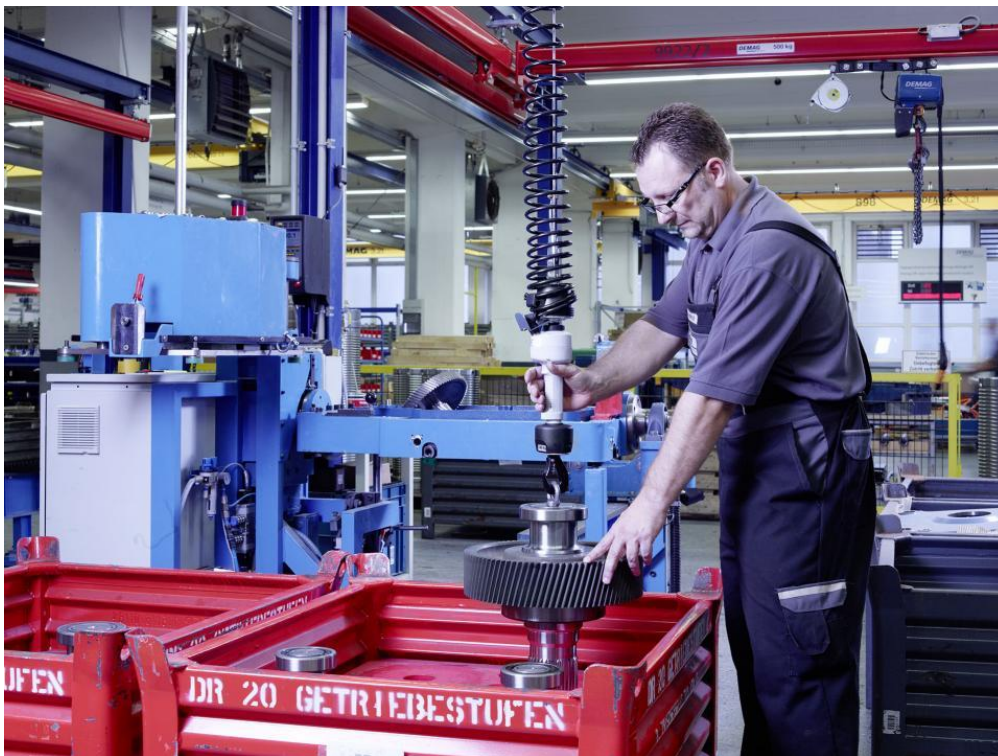
Obr. 1 Balancery pro upevnění ručního nářadí do 10.9 kg. [OBR1]

Montážní pracovník není nucen zvedat tyto pomůcky, ale stahuje je dolů na lanku balanceru. Vhodnou volbou předpětí pružiny se navíc téměř vyruší tíhová síla náradí s napínací silou lanka. Na manipulaci s náradím tedy není potřeba použít téměř žádnou sílu a manipulace je pro obsluhu velice příjemná. Dalšími zástupci jsou různé manuální jeřáby, které najdou velké zastoupení v oblastech manipulace se součástmi, které nejsou umístěny na paletě. Zde lze uvést například jeřáb pro montáž s motorem osobního automobilu.

## 2.2 Elektro-mechanické balancery

Pouze malá úroveň elektronické výbavy dokáže zlepšit vlastnosti mechanického balanceru. Na trhu se ve velké míře objevují balancery s elektrickým pohonem, které dokáží velice pružně reagovat na zatěžující sílu. Balancer se zpravidla skládá ze sloupu a otočné konzole, která vymezuje jeho pracovní prostor zpravidla kruhové výseče různého poloměru a úhlového rozsahu.

Princip činnosti je velice jednoduchý. Tyto balancery mají charakteristické madlo, které není zvenčí ničím zajímavé, nicméně vnitřek ukrývá důmyslný senzorický systém na principu tenzometru. Systém vyhodnocuje rozdíl sil od spodního háku, na kterém je umístěno břemeno, a silou zvedacího lana. Naměřený rozdíl mezi silami je právě síla, kterou působí operátor na madlo. Pokud tento rozdíl míří vzhůru, systém zareaguje navíjením nosného lana a celý systém se začne pohybovat směrem vzhůru a naopak. Pokud operátor přestane působit silou, systém se zastaví. Nevýhoda těchto balancerů spočívá v tom, že součást je držena ve vzduchu pouze silou od motoru. Výpadek elektrické energie by mohl způsobit pád břemene, a proto musí být tyto balancery vybaveny mechanickou brzdou nebo šnekovou samosvornou převodovkou.



Obr. 2 Elektro-mechanický balancer určený pro manipulaci s těžkými obrobky [OBR2]

## 2.3 Teleoperátory

Nezastupitelnou oblast manipulace s obrobkem zaujímají teleoperátory. Jsou definovány jako paralelní manipulátory, které vytváří přesné pohyby na základě pokynů lidské obsluhy. Umožňují provádět práce, které jsou pro člověka nevhodné a to zejména z důvodu velkých sil potřebných pro manipulaci, přesnosti manipulace a nebezpečnosti pracovního prostředí.

Jako kompenzátor nedostatku pracovní síly by mohl být zmíněn například bagr. Pro člověka fyzicky velice náročnou práci zprostředkovávají hydraulické válce, kterým dodává energii hydrogenerátor poháněný zpravidla spalovacím motorem. Člověk zde v pozici operátora řídí elektronicky jednotlivé rozváděcí ventily hydraulického obvodu. Trendem v řízení těchto strojů je užití proporcionálních ventilů, které ovládají rychlost výsuvu na základě míry řídicího signálu. V některých aplikacích se zavádí systém tzv. elektrohydraulického PID. Tento systém funguje dle majitele firmy Echidna na principu zpětnovazební tlakové smyčky. Firma jakožto výrobce řezacích pil na kámen pro bagry používá tento systém pro zajištění stálé řezné síly nástroje. V praxi tento systém funguje tak, že v měkčím materiálu umožňuje řezat vyšším posuvem. Po najetí do tvrdšího podloží se zpomalí řezný posuv. Výsledkem tohoto typu řízení se zvyšuje rychlost řezání za současného zvýšení životnosti řezných břitů.[6]



Obr. 3 Kotoučová pila vybavena elektrohydraulickým PID [OBR3]

Další oblastí teleoperátorů jsou stroje, které zvyšují přesnost lidského pohybu. Tato zařízení získávají uplatnění zejména v oblasti medicíny jako operační manipulátory, které ve zmenšeném měřítku provádí operátérův pohyb. Zároveň jsou nejmodernější manipulátory tohoto typu osazeny řídicím systémem, který je schopen filtrovat nechtěný třes rukou a díky tomu jsou operační pohyby velice přesné a čisté. Pokud opustíme oblast nemocnice, jsou tyto operátory nasazovány například v oblasti velice přesného pájení a osazování desek plošných spojů. Zde je ale nutno podotknout, že se takový postup

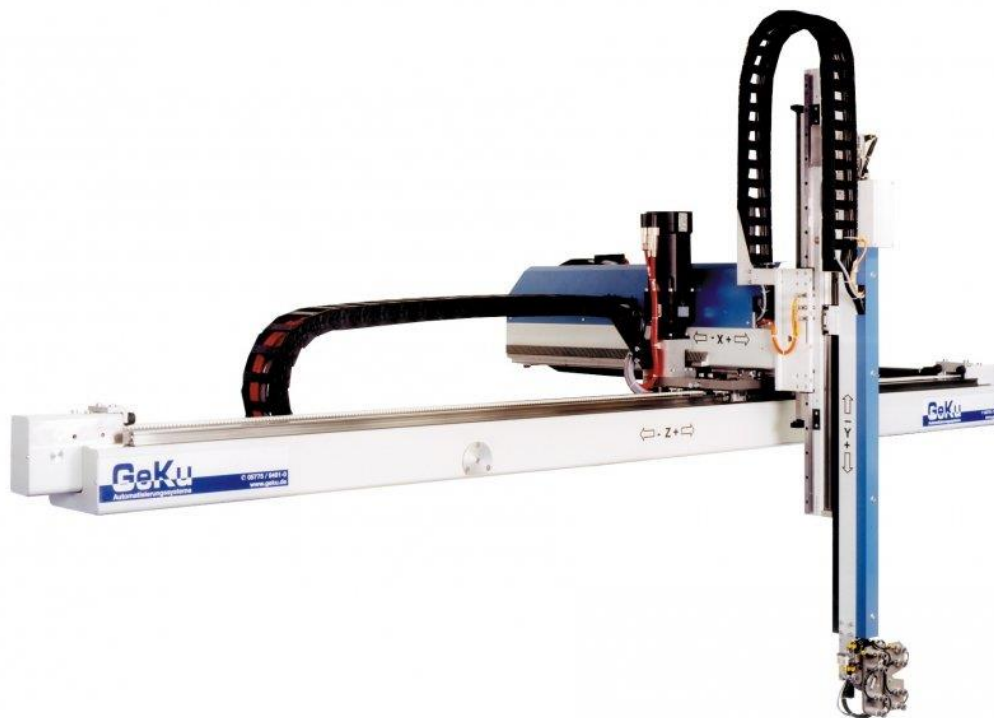
uplatňuje pouze v oblastech výjimečných oprav nebo výroby prototypů. V hromadné výrobě se nasazují tzv. osazovací automaty.

## 2.4 Automatické manipulátory s tvrdým řídicím programem

Tyto manipulátory sice pracují v automatickém pracovním cyklu, ale tento cyklus jde jen velice obtížně měnit. Nasazují se v hromadných výroбах, kde se předpokládá obrovský počet opakovaných cyklů, či v oblastech, kde je jistota, že bude vyžadován stále stejný pohyb manipulátoru. Zde může být zmíněno rameno automatické výměny nástroje.

Tvrdá automatizace manipulátorů byla dříve zajištěna vačkovým mechanismem. Velké počáteční náklady na návrh a výrobu vačkových převodů byla vyvážena stálou přesností pohybu a dokonalé synchronizací díky mechanické vazbě mezi jednotlivými komponentami soustrojí.

V dnešní době se i na jednoduché manipulátory, které vyžadují automatickou činnost, používají elektronické systémy. Důvod této volby je zejména v klesajícím trendu nákladů na pořízení takového manipulátoru a rostoucí tlak na pružnost výroby a adaptivitu manipulátoru na aktuální vnější podmínky.



Obr. 4 Kartézský manipulátor určený pro vyjímání obrobku ze vstřikolisu [OBR4]

## 2.5 Automatické manipulátory s pružným řídicím programem

Tato oblast manipulace vyžaduje rychlou reakci manipulátoru na aktuální okolní situaci. Proto jsou manipulátory vybaveny svým řídicím systémem nebo řídicím systémem stroje, který obsluhují. Manipulátor je schopný se rozhodovat na základě vstupní informace, jakou je například čárový kód na součásti, jaké komponenty vyndat ze zásobníku a kam je umístit. Paletizační automaty se poté podle půdorysu palety umí rozhodnout, jak na paletu co nejhospodárněji umístit krabice atd.

Z této oblasti mě na veletrhu zaujal zejména logistický systém – „AUTOMATICKÝ SKLADNÍK“. Systém byl složen z několika komponent. Na základě požadavku vyjmutí součásti systém zvolil příslušnou polici rotomatu. Dále systém obsahoval kolový bateriový manipulátor s uchopovacím mechanismem. Manipulátor se mezi jednotlivými rotomaty pohyboval za pomoci lokální sítě na principu GPS. Přesné navádění k zásobníkům bylo poté zajištěno přesnějšími senzory. Manipulátor vyjmul příslušnou krabici z police a dovezl ji k výdejnímu pultu. Zde se tedy nejednalo o samostatný manipulátor, ale o pružnou skladovací manipulační buňku.



Obr. 5 Robotický zakladač od firmy Viastore [OBR5]

Pro aplikace, které vyžadují rychlé rozhodování a složité rozhodovací algoritmy, se dostáváme do oblasti složitějších řídicích systémů a celkově se začínáme pohybovat v oblastech průmyslových robotů, které v dnešní době otevírají svět dosud nevídaných možností. Nasazení nejmodernějších řídicích systémů startuje v oblastech výběru neuspořádaných součástek z kontejneru a končí v oblasti kooperativních robotů, kde si s montážním pracovníkem doslova podává šroubovák.

## 3 POUŽÍVANÉ KINEMATICKÉ DVOJICE MANIPULÁTORŮ

### 3.1 Translační kinematické dvojice

Kinematická dvojice translační neboli posuvná umožňuje lineární posuv dvou těles po sobě. Podle relativnosti pohybu jednotlivých součástí vůči sobě se translační kinematické dvojice dělí na další tři podskupiny

**Suportové/saňové** – při tomto konstrukčním provedení se pohybuje kratší součást po delším vedení, pro ilustraci je možno si představit suport kartézského manipulátoru typu MANTA či pohyb celého robotu po stacionární pojezdovém ústrojí.

**Smykadlové** – tento typ uspořádání má obrácené role součástí. Kratší díl je nyní nosnou částí, po které se pohybuje delší část s vodicími drážkami nebo profilovým kuličkovým vedením.

**Výsuvné/teleskopické** – velice hospodárný typ uspořádání. Tělesa se v tomto případně zasouvají navzájem do sebe a je možno dosáhnout velikého rozpětí. Teleskopická kinematická dvojice může být jednoduchá nebo může být složena z několika stupňů a umožnit tím ještě větší rozsah. [3]

### 3.1.1 Realizace translační kinematické dvojice

Posuvnou kinematickou dvojici lze realizovat dvojnásobným způsobem. Prvním způsobem je použití kluzného vedení a naproti němu stojí vedení profilové s valivými elementy, které si zejména pro absenci kluzného tření a s ním souvisejícím Stick-Slip efektu získává velkou oblibu. Řádově nižší hodnoty valivého tření umožňuje zvyšovat dynamičnost celého systému, jež si moderní průmysl vyžaduje. V průmyslu se nejčastěji setkáváme s profilovým valivým vedením, kde je vozík vůči kolejnici schopen odebrat 5 stupňů volnosti a zachovat pouze požadovaný pohyb ve směru osy. Jako oběhové elementy jsou zde použity kuličky nebo válečky podle velikosti působících sil. Vozíky lze vyrábět s různým předpětím elementů.

Předpětí má veliký vliv na tuhost translační kinematické dvojice. Stejně tak má ale velký vliv na sílu potřebnou na rozpohybování dvojice, takže pro dynamičtější aplikace se volí nižší předpětí a pro přesnější aplikace naopak. I pro své velké výhody valivého vedení existují oblasti, kde zůstává obliba na straně vedení kluzného. Kluzné vedení má daleko menší zástavbové rozměry.[1]

## 3.2 Rotační kinematické dvojice

Rotační kinematická dvojice umožňuje rotaci kolem jedné osy souřadného systému. Rotační kinematické dvojice si získaly velkou oblibu v oblasti průmyslových robotů, neboť kloubové uspořádání umožňuje skvělé využití tzv. multiúhlového pracovního prostoru. Rotace může být rozdělena do dvou skupin.

**Rotační** – v tomto případě je pro pohyb charakteristické, že otočný pohyb probíhá kolem podélné osy a může probíhat do nekonečna. To znamená, že rozsah otočného pohybu není úhlově omezen. Z oblastí průmyslových strojů je typickým zástupcem otočný stůl při kinematickém uspořádání 3 translační osy v nástroji a dvě osy rotační v tzv. kolébce. U průmyslových robotů lze najít kinematické dvojice rotační ke konci otevřeného kinematického řetězce, například otáčení koncového efektoru. Fyzicky však není využívána vlastnost nekonečného otáčení, neboť ke koncovému efektoru vedou pracovní média (stříkácí pistole, podtlaková přísavka, svařovací hořák) a při rotaci o více než 360° by došlo k poškození technologie, kterou je robot vybaven.

**Kyvné** – kyvné kinematické uspořádání lze reprezentovat jako rotaci ramene s dané délky o definovaný úhel kolem osy, jež je na toto rameno kolmá. Kinematickou dvojici kyvnou je možné si představit jako klasický dveřní pant či loket ruky. Umožňuje rotaci ramene manipulátoru kolem příčné osy a ve většině případů konstrukční upořádání neumožňuje otáčení o celých 360°.

## 4 POUŽÍVANÉ KOMBINACE KINEMATICKÝCH DVOJIC

Počet kinematických dvojic určuje, jaký typ oblasti bude manipulátor schopen obsluhovat. Jedna kinematická dvojice umožňuje obsluhovat 1D prostor v podobě přímky pro translační typ KD nebo prostou kružnici pro rotační typ KD. Pro dvě KD je možné obsluhovat rovinu pro uspořádání translace-translace, část válcové plochy pro uspořádání rotace-translace. V poslední řadě zbývá uspořádání rotace-rotace, které umožňuje obsluhovat opět rovinu, která je však ohraničena oblouky.

Pracovní prostor ve tvaru plochy je velkým omezením, a proto se přidává ještě jedna kinematická dvojice, která zajistí rozšíření do prostoru. Pokud sečteme všechny kinematické uspořádání, dostaneme celkem 165 jejich kombinací. Dovolím si proto zmínit pár nejznámějších vč. jejich pracovního prostoru.

### **TTT – translace-translace-translace**

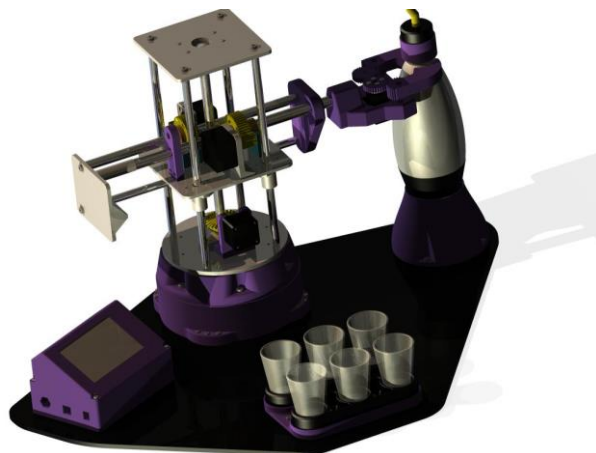
Kinematické uspořádání určené pro obsluhování kvádrového prostoru. Jelikož směry translací v jednotlivých osách kopírují směry os souřadného systému, je řízení takového uspořádání velice jednoduché. Kvádrový pracovní prostor je předurčen pro paletizační operace. Obrovskou výhodou takového uspořádání je i stálá orientace součástí v prostoru, takže takový typ manipulátoru nepotřebuje další složité orientační ústrojí. Navíc je možné dosahovat rozsahů jednotlivých os více než 2000 mm. Tento typ uspořádání reprezentuje např. zařízení MANTA.[1]



Obr. 6 Kartézský manipulátor firmy Wittmann [OBR6]

### **RTT – rotace-translace-translace**

Pokud je potřeba obsadit cylindrický prostor, je toto uspořádání ideální. Manipulátory typu RTT, reprezentovány strojem VERSATRAN s nosným sloupem a výsuvným ramenem kolmým na vertikální osu. Nevýhodou tohoto uspořádání může být změna orientace součástí kolem vertikální osy. Naopak se toho dá skvěle využít u výměníků nástrojů s externím zakladačem. Místa pro nástroje jsou umístěna hvězdicově v policích a manipulátor může jednotlivé nástroje pohodlně vkládat i bez dalšího orientačního zařízení. [7]

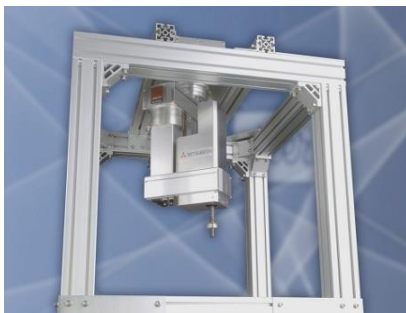


Obr. 7 Robotické pracoviště s cylindrickým manipulátorem vyvinutým na VUT – Alcbot [OBR7]

### **RRT – rotace-rotace-translace**

Takové upořádání zajišťuje pokrytí části sférického prostoru. Tento typ manipulátoru je výhodné programovat ve sférických souřadnicích, neboť dvě rotace reprezentují natočení o úhel ve dvou navzájem kolmých rovinách a translace reprezentuje poloměr kulového prostoru. Nevýhodou uspořádání je změna orientace součástí ve dvou směrech a je nutno použít adekvátní koncový efektor, který je schopen tuto odchylku kompenzovat. Manipulační zařízení tohoto typu můžeme najít pod značkou UNIMATE

Dalším zástupcem této kinematiky může být zařízení známé pod značkou SCARA. Zde je kinematika odlišná od předchozího zástupce. První dvě rotační vazby vytváří dvouramenný mechanismus a polohují manipulátor v rovině. Další posuvná kinematická dvojice je reprezentována výsuvným mechanismem, jenž je kolmý na výše zmíněnou rovinu. Toto uspořádání má velice příznivý poměr mezi velikostí manipulátoru a obsluhovaným prostorem. Další výhodou je, že manipulátor může být zavěšen od stropu a díky tomu je možné manipulovat se součástmi na průběžném pásu. [1]



Obr. 8 Robot SCARA od firmy Mitsubishi zavěšený od stropu [OBR8]

## **RRR – rotace-rotace-rotace**

Nejsložitější systém na řízení je poslední kinematické schéma. Zařízení, které je velice oblíbené a jehož kinematika se čím dál víc objevuje v oblasti průmyslových robotů a manipulátorů, je nasazováno v průmyslu zejména pro svoji anatomičnost. Manipulátory RRR umí jako jediní pracovat v bezprostřední blízkosti osy Z na rozdíl od ostatních výše zmíněných kinematických schémat. Navíc jsou schopny se vyhýbat překážkám a dokáží dosáhnout „za zed“.



**Obr. 9 Průmyslový 3-osý robot ABB doplněný 3-osým koncovým orientačním ústrojím [OBR9]**

Nevýhodou těchto manipulátorů je jejich složité řízení a výpočet inverzní kinematiky pro správné stavění souřadnic koncového efektoru. Dříve byla složitost kompenzována přímým učením operátorem. Manipulátor poté opakoval naučené souřadnice bod po bodu (anglicky PTP – point to point) avšak průběh pohybu mezi definovanými body do jisté míry nebylo možné předem definovat. S růstem výpočetní techniky roste i výkonnost výpočtu souřadnic. Moderní řídicí systémy dokáží vypočítávat trajektorii pomocí prvků numerické matematiky. Nejen že dokáží přesně stavět definované souřadnice, ale dokáží také interpolovat mezi body po přímce či dokáží otáčet globální souřadný systém podle požadavků operátora.[2]

## 5 ROBOTIKA OKOLO NÁS

Opusťme nyní oblast průmyslové manipulace, kde najdou manipulátory obrovskou pozici při manipulaci s obrobky mezi stroji, při operacemi s nástroji a v neposlední řadě i montáž či paletizaci hotových výrobků. V dnešním světě se čím dál více objevují manipulační systémy i v oblastech denního života. U těchto automatických systémů je největší důraz kladen na užitnou hodnotu pro člověka. Z těchto systémů lze zmínit tzv. servisní roboty, robotizované hračky, roboty v domácnosti či velká robotická centra typu automatických parkovišť.

### 5.1 Servisní roboty

Pojem servisní robot vznikl přijmutím z anglického „service robot“, což znamená robot pro nevýrobní činnost, která nepřispívá k výrobě součástí, ale vytváří služby užitečné pro člověka. Je možno si pod tímto pojmem vybavit zařízení pro dálkovou deaktivaci výbušniny, robotického číšníka či servisní vozítko, které slouží pro výzkumnou činnost ve vesmíru. Zejména na roboty, které přichází do přímého kontaktu s člověkem, je kladen obrovský důraz na jejich design a adaptivitu k lidskému prostředí. Adaptivitou je myšlena zejména otázka bezpečnosti, které je při tomto kontaktu prvořadá. Je důležité si uvědomit, že i přes svoji podobu a většinou vysoký stupeň inteligence zůstávají servisní roboty stále stroji. Na rozdíl od člověka mají poměrně vyšší hmotnost, větší sílu a také uvnitř každého robotu mohou protékat člověku vysoce nebezpečné proudy.[15]

Výrobci však předcházejí nebezpečným situacím díky instalaci bezpečnostních senzorů, které v kombinaci s chytře zpracovanými bezpečnostními rutinami v programu zajistí, že servisní roboty jako mobilní číšníci a další zařízení určené pro přímý kontakt s člověkem předchází dotyku s člověkem a zároveň se vyhýbají nebezpečným zónám. Mezi nebezpečné zóny mohou být považovány schody, nakloněné části podlah či případně prahy, které by mohli způsobit pád zařízení.

Na opačné straně stojí servisní roboty, které provádějí výzkum ve vesmíru. Zde je kladen veliký důraz na spolehlivost celého zařízení, neboť vyzvednutí jakéhokoliv zařízení do vesmíru je velmi nákladné a chyba zařízení není možné tolerovat. Proto bývají nejdůležitější systémy řízení zdvojené a započítí výroby předchází rozsáhlé studie krizových situací, aby servisní vozítko ve vesmíru nezaskočily žádné nečekané situace.

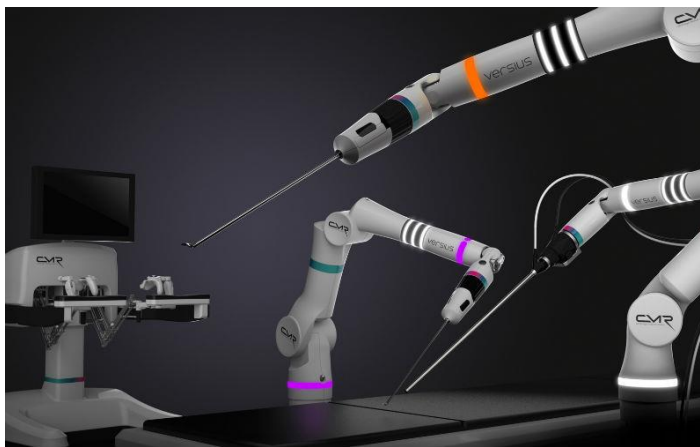
### 5.2 Servisní roboty ve zdravotnictví

Roboty v oblasti ochrany lidského zdraví mají jistě otevřenou cestu. Rozšiřují se ve velkém množství medicínských oblastí již drobných chirurgických zákroků prováděných laparoskopicky přes velké operace robotem s několika samostatně pracujícími rameny až po následnou asistenci při rekonvalescenci. Roboty si ve zdravotnictví našly své postavení zejména pro svoji přesnost. Žádný robot neoperuje sám, ale používají se jako takzvané teleoperátory. Rozšiřují tedy operační možnosti lidské obsluhy. Pomocí průmyslových kamer dokáží přiblížit místo operace či přesně vést nástroj a zvýšit přesnost díky snížení měřítka operačních pohybů obsluhy. V neposlední řadě dokáže ope-

rační manipulátor filtrovat třes rukou a tím dosahovat skvělých výsledků při operaci očí. Jelikož je minimalizován styk lidské obsluhy s pacientem během operace, eliminuje se možnost infekce.

### **Robot Versius**

Robot byl sestaven ve Velké Británii a byl představen v roce 2017. Byl zkonstruován ve spolupráci s univerzitou v Cambridge pro aplikace typu „keyhole surgery“. V doslovném překladu to znamená operace klíčovou dírkou a představuje zejména laparoskopické operace. Díky svým dlouhým a tenkým operačním nástrojům je taktéž předurčen pro operace nosních částí, operacím stěh a ke gynekologickým zákrokům. Malé operační zásahy prováděny laparoskopicky jsou velice oblíbenou metodou operace, neboť malými vpichy se zavádí dlouhé tenké nástroje a chirurgickému zákroku následuje kratší doba rekonvalescence a samozřejmě po něm zůstávají i menší jizvy. [8]



Obr. 10 Robot Versius vyvinutý ve spolupráci s univerzitou Cambridge [OBR10]

Zmíněný způsob operování však vyžaduje velice přesnou práci operátora. Jelikož se nástroje k místu operací přibližují relativně naslepo, je vyžadována skvělá znalost lidského těla. Společnost CMR (Cambridge medical robotics) se snaží zlepšovat podmínky operování, zlepšit orientaci pomocí 3D zobrazení operační scény a tím dopřát tuto operační metodu více pacientům.[9]

### **Robot DaVinci**

Operační robot vytvořený firmou Intuitive surgical z USA. Historie myšlenky robotického operačního systému sahá až do roku 1940. V tomto roce je myšlenka operačního robotu přirovnána spíše k vědecko fantastické předpovědi. Nicméně již v roce 1950 vznikla první operační nástavba, která funguje jako teleoperátor. A od té doby se robot DaVinci vyvíjí až do dnešní podoby a kolaudován pro první operace byl v roce 1999. Vývojem prošla kompletní mechanika operačních ramen, ale zejména nástroje, které pomáhají odstraňovat nechtěnou tkáň, šít, uchopoval, ale také nástroje pro odsávání krve, která se nachází v místě operace a dalších nástrojů pro práci chirurga. V České republice se robot DaVinci objevuje v roce 2005. [10]

Firma Intuitive surgical, jejíž název by mohl být přeložen jako intuitivní chirurgie, se ovšem nezabývá pouze konstrukcí svých zařízení. Dává také veliký důraz na zaškolení obsluhy pro své robotické systémy. Před zahájením skutečné operace obsluha prochází

sadou školení, která jsou připravena ve formě virtuálních 3D video operací. V této fázi si obsluha může vyzkoušet jak samotné operování, tak i ovládání dílčích podpůrných funkcí jako je ovládání 3D kamery, výměna nástrojů na konci pracovního ramene. Samozřejmostí je i přiřazení uživatelských funkcí nožním a ručním tlačítkům. Samozřejmostí je i ergonomie v podobě přizpůsobení ovládacího stanoviště, aby měl operátor připraveny nejlepší podmínky i při dlouhých a náročných zákrocích. [10]



Obr. 11 Operační robot daVinci a řídicí pracoviště robotu [OBR11]

### 5.3 Servisní roboty v bezpečnostních složkách

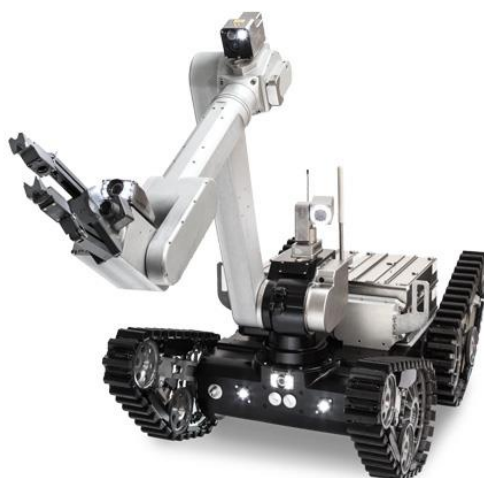
Záchrana lidského majetku a především života je jedním z prvořadých úkolů, které musí řešit bezpečnostní servisní roboty. Jejich úkoly mohou být velice různorodé. Vyskytují se zde jednoduché kolové servisní teleoperátory, pomocí kterých lze díky kameře monitorovat jejich okolí. Nasazují se v oblastech, kde je průzkum pro člověka nebezpečný, například budovy se zhoršenou statikou, okolí výbušnin, radiací zamořená místa či území nepřítel ve vojenských sporech. Dále se tyto roboty používají na místech, kde je přístup pro člověka fyzicky nemožný, tedy oblasti malých rozměrů, zatopené oblasti aj.

#### Roboty rodiny Telex

Modulární robotické systémy od německé skupiny Telerob jsou velice univerzální stroje. Lze sestavit robot přesně dle požadavků zákazníka. Podvozek lze volit mezi čtyřkolovou variantou nebo pásový. Základna je vybavena zpravidla sedmi osým manipulátorem pro zajištění dostatečné manévrovatelnosti v prostoru. Manipulátor může být dimenzován na zvedání břemen až do hmotnosti 75 kilogramů. Koncový efektor může být osazen velice širokou škálou pracovních nástrojů. Základní klešťový upínač může být vyměněn za vrtací zařízení či chemickou sondu. Může však být vybaven plamenometem nebo naopak proudnicí pro hašení požáru. Může být taktéž osazen palnou zbraní různých kalibrů či například sítí pro zpacifikování nepřítel. Nástroje mohou být montovány manuálně nebo systém podporuje automatickou výměnu koncových efektorů přímo v terénu.[11]

Je samozřejmostí, že lze systém vybavit průmyslovou kamerou pro řízení na dálku a stejně tak ho lze osadit průmyslovými kamerami. Ve volitelném příslušenství nechybí ani audio systém pro vysílání a příjem akustických signálů. Dále lze robot vybavit senzorickými čidly v podobě ultrazvuku, laseru či GPS pro lepší orientaci mezi překážkami. Při sestavování robotu nechybí bateriové zdroje energie, s nimiž robot vy-

stačí až na 8 hodin provozu. Robot je dodáván s dálkovým řídicím systémem a firma Telerob nabízí i speciální transportní boxy pro bezpečný převoz robotu na místo práce. [11]



Obr. 12 Pásový manipulátor telemax PRO stavebnicové konstrukce [OBR12]

### Robot Safety Guard

Dalším robotem firmy Telerob je také robot určený zejména pro aplikace spojené s hašením požáru. Robot se jménem Safety Guard je určen pro prvotní ohledání místa pomocí průmyslové kamery, je vybaven efektor, které dokáží manipulovat s výbušninami, transportovat tlakové láhve z ohniska požáru. Za zmínku stojí také technologické řešení efektoru, který umožňuje uzavírat kohouty a tím zabránit únikům nebezpečných látek, zejména plynům a hořlavinám. Samozřejmostí je i proudnice, která umožňuje cíleně hasit ohnisko požáru. Robot je připojen k čerpadlu pružnou hadicí a je řízen dálkovým ovládním. Velikou předností je jeho rychlé nasazení do akce, které netrvá více než 10 minut od příjezdu na místo zásahu, a úprava lokomočního ústrojí, které umožňuje jízdu po schodech. Celý robot je uzpůsoben tak, aby pracoval spolehlivě i za působení zvýšených teplot, prachu a vodě. V České republice je robot ve službách zejména pyrotechnických složek policie ČR. Na jeho vývoji se podíleli také hasiči ČR.



Obr. 13 Robot SafetyGuard [OBR13]

## Robot TEODOR

Posledním zástupcem robotů konstruovaných firmou Telerob nese název teodor z anglické zkratky „Telerob Explosive Ordnance Disposal and observation robot“. Překlad jeho názvu, který zní robot firmy Telerob určený pro pozorování a likvidaci výbušnin, přesně vystihuje oblast jeho použití. Robot TEODOR je primárně určen pro likvidaci a manipulaci s výbušninami. Řadí se do oblasti takzvaných těžkých servisních robotů a dokáže uvést náklad až 350kg. Je vybaven šestiosým manipulačním ramenem, se kterým dokáže zvedat břemena s hmotností do 100kg. Jeho předností je množství senzorů od chemických po rentgenové, které ve spojení s vnitřním programovým vybavením dokáží předpovídat nebezpečné situace, na které je obsluha během zásahové akce upozorněna. Robot TEODOR následuje svá sesterská zařízení z hlediska modularity, a tak může být zkonstruován přesně podle požadavků zákazníka.



Obr. 14 Robot TEODOR při zneškodňování výbuštiny [OBR14]

## Big Dog

Přesuňme nyní pozornost do oblasti vojenského průmyslu. Big Dog je projekt Americké firmy Boston Dynamics, který byl financován organizací DARPA ( Defence Advanced Research Projects Agency ). Tuto zkratku by bylo možné přeložit jako: „Agentura pro pokročilé projekty v oblastech výzkumu a obrany“. DARPA je zaštitěna ministerstvem obrany spojených států a rozhoduje o financování výzkumu nových technologií ve vojenském průmyslu. [12]

Big Dog byl vyvinut pro transport nákladů do oblastí, kde se standartní vozidla již nemohou pohybovat. Big Dog připomíná velkého čtyřnohého psa. Díky svému lokomočnímu ústrojí, u kterého byly kola a pásy nahrazeny čtyřmi nohama, dokáže zdolávat náklony terénu až 35°. Dále je schopen se pohybovat v sutinách, mělké vodě, ve sněhu a na ledu. Není překvapením, že zvládá chodit po schodech, a tak je schopen doprovázet vojáky v libovolném terénu. [12]

Energii pro jeho pohyb dodává 2-taktní spalovací motor. Energie je přeměňována hydrogenerátorem na energii hydraulickou. Hydraulický olej je poté pomocí sestavy servoventilů rozváděn do čtyř nohou. Každá noha má dva hydraulické válce v kyčelním kloubu, jeden válec pro kolenní kloub a jeden válec pro stavění chodidla vůči podložce. Zpětnou vazbu pro řídicí jednotku tvoří polohové senzory z kloubů, silové a momentové snímače kloubů, snímač síly mezi podložkou a chodidlem. Dalšími prvky jsou akcelerometry, které určují orientaci celého těla. S pomocí těchto snímačů již řídicí jednotka dokáže udržet stabilitu robotu. Pro orientaci v prostoru je Big Dog vybaven ultrazvukovými senzory a GPS navigací. Je schopen následovat lidský doprovod a nebo se pohybovat autonomně po naplánované trase. [13]



Obr. 15 Robot BigDog při zkouškách v náročném terénu [OBR15]

Ačkoliv se projekt robotické „tašky“ zdá na první pohled jako velice přínosný pro oblast transportu zásob a zařízení, byl projekt Big Dog v roce 2015 zastaven, neboť spalovací motor způsobuje relativně velký hluk, a tak není pro strategické operace vhodný. Současně s projektem Big Dog vyvíjela Boston Dynamics také menšího bateriového bratra pod názvem Spot. Spot však nedosahoval takového výkonu a kapacita jeho baterie výrazně omezila jeho akční rádius. Proto byl i Spot vyloučen z projektu dopravy vojenského zařízení. Boston Dynamics však nezavrhl kinematické zpracování robotu Spot a představil robot SpotMini, který je určen pro manipulační aplikace, které prezentoval na ukázce otevírání dveří za pomoci robotu SpotMini. [14]



Obr. 16 Nástupce projektu BigDog nese název SpotMini. Na obrázku včetně manipulačního ramene.[OBR16]

## 5.4 Humanoidní roboty

Oblast humanoidní robotiky se rozvíjí v podstatě stejně dlouho, jako robotika samotná. Již všem známý Golem někteří přirovnávají k humanoidnímu robotu. Jeho tělo přirovnávají k lokomočnímu ústrojí a Šém (kuličku, která jej oživuje) vnímají jako jeho řídicí systém. Pokud se však vrátíme k definici: „Humanoidní robot je robot, který svou konstrukcí a zevnějškem připomíná člověka. Jeho končetiny mají zpravidla mnoho stupňů volnosti, aby dokázal reprezentovat klouby lidského těla. Pokud se podobnost blíží reálnému člověku, označují se jako Androidi.“

### 5.4.1 Robot ASIMO

Název ASIMO vznikl z anglického Advanced Step in Innovative MObility (Pokrokové kráčející mobilní zařízení nové éry). Je robotem, který je nejvíce rozšířen v povědomí veřejnosti. K této skutečnosti přispívá zejména jeho velká podobnost k fyziologii lidského těla, neboť roboty ASIMO vypadají jako člověk v bílé kombinéze s batohem na zádech. Se svými celkem 26 stupni volnosti se stává skvělým společníkem a dosahuje plynulých a elegantních operačních pohybů.[16]



Obr. 17 Robot ASIMO provádějící servisní činnost všedního dne [OBR17]

Historie vývoje robotu ASIMO se datuje od roku 1986. Firma Honda tímto projektem započala návrat k dřívějšímu vývoji, ze kterého přebral technologii dvounohého kráčejícího pohybu. Veliký důraz byl kladen na podmínku, že robot má být připraven na přímou interakci s člověkem. Jeho vzhledová stránka tedy musí odpovídat požadavkům člověka, musí být bezpečný a zároveň musí být příjemný v oblasti komunikace v každodenním životě. Tím vznikla první generace robotického zařízení, které připomínalo dvě nohy a trup.



Obr. 18 Historický vývoj robotů firmy HONDA [OBR18]

Velkým problémem při vývoji robotu byl způsob jeho pohybu. Každý si může vzpomenout na své první krůčky, které byly velice náročné z hlediska stability. Jak vytvořit robot, který bude dokázat chodit bez jediné chybičky, neboť případný pád na člověka si nemůže žádných okolností dovolit, ale zároveň bude chůze definována podle algoritmu, který má své hranice? Vývojem vzniklo zařízení, které má dvě nohy a každá noha má 6 stupňů volnosti. Kyčelní kloub lze ovládat rotací kolem všech tří souřadnicových os, koleno umožňuje rotaci v jednom směru a chodidlo pak dokáže vytvářet poslední dva náklony. Všechny klouby jsou vybaveny momentovými a silovými senzory a dále je měřena poloha natočení jednotlivých kloubů. Dále je systém vybaven akcelerometry a senzory normálových sil od podložky.[16]

Pro dosažení stabilní chůze používá robot ASIMO tři dílčí oblasti interakce. První je vyhodnocení reakčních sil od podložky. Kombinace informací ze sady senzorů na chodidle dokáže robotu předat informace o tvaru a tvrdosti podložky a dále dokáže robot informovat o stabilitě kontaktu chodidla s podložkou.

Druhá oblast řízení zabezpečuje předcházení pádu. Gravitační síla a síly pro požadované zrychlení či zpomalení spolu tvoří tzv. inerciální sílu. Naproti ní stojí síly reakční od chodidel. Tyto síly a momenty tvoří uzavřenou silovou soustavu, která musí být ve statické rovnováze. Síla, která by byla v rozporu se statickou rovnováhou, by byla takzvaná pádová síla a došlo by k pádu celého zařízení ve směru pádového momentu od této síly. O korekci pádové síly se stará vnitřní algoritmus robotu upravováním natočení jednotlivých kloubů. Tato oblast se jmenuje řízení ZMP z anglického Zero Moment Point. [17]

Třetí oblast se zabývá predikcí následného pohybu. Robot na základě současné konfigurace kloubů a pohybu v kombinaci s dalším požadovaným průběhem trajektorie, interakcí s překážkami a dalšími faktory plánuje provedení dalšího kroku. Je bezpodmínečně nutné, aby se robot během pohybu nedostal do konfigurace, při které nedokáže dodržet statickou rovnováhu buď z důvodů přetížení některého z kloubů, nebo omezení jejich pohyblivosti. Druhým požadavkem je poté plynulost provedeného kroku bez nutnosti použití trhavých pohybů a bez kymáčení celé konstrukce. Tento typ řízení plánování kroku byl označen jako i-WALK (inteligentní krok). [18]



Obr. 19 Robot ASIMO v porovnání s výškou dospělého člověka [OBR19]

Robot ASIMO je svou ergonomií plně přizpůsoben lidské společnosti. Se svojí výškou 1200mm nepůsobí mezi lidmi dominantně. Taková vlastnost by mohla v některých lidech evokovat pocit nadřazenosti přítomného robotu. Výška 1200mm byla stanovena jako optimální, neboť robot dokáže uchopovat pohodlně objekty z konferenčního stolu, ale je i dostatečně vysoký na to, aby si s člověkem mohl podávat objekty z ruky do ruky. Navíc při své výšce 1200mm hledí přímo do očí sedícímu člověku v křesle, a tak opět vytváří příjemné prostředí. ASIMO je vytvořen pouze pro obsluhu člověka. Pracuje jako mobilní číšník, jako vítací robot a průvodce. Není určen pro výzkumné a obranné aplikace a stejně tak není určen pro jakékoliv práce ve stavu beztlíže či v podmořském prostředí. [19]

## 5.5 Servisní roboty v domácím prostředí

Pojem inteligentní dům je dnes skloňován ve všech pádech. Roboty, jakožto automatická zařízení, které se dokáží přizpůsobovat požadavkům obsluhy a svému okolí, najdeme v podstatě kdekoli v moderní domácnosti. Tyto systémy dokáží zabezpečovat naše domy, šetřit elektrickou energii, zlepšovat náš komfort, dokáží nás bavit dlouhé hodiny svou přítomností a o tom všem vedou přehledné statistiky. Jaké systémy dnes mohou obsahovat lidské domácnosti??

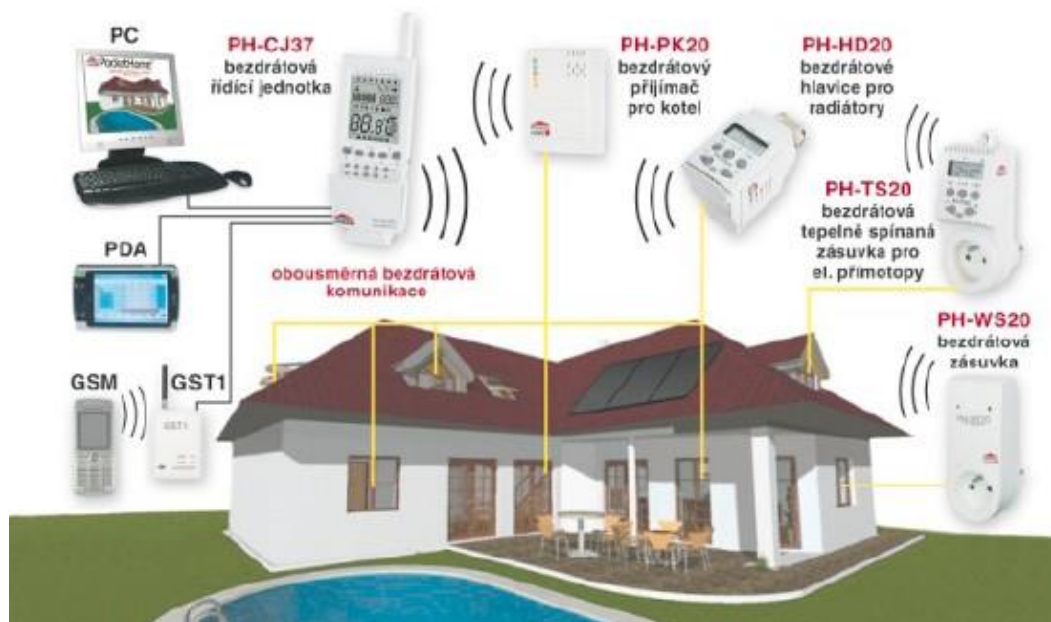
### 5.5.1 Funkce inteligentního domu

**Vytápění** - Mezi nejčastější systémy patří řízení toku energie. Chytrý systém rozvodu teplé vody do radiátorů dokáže díky množství řízených ventilů řídit vytápění a klimatizaci v celé domácnosti. Systém dokáže udržovat v každé místnosti komfortní teplotu a dokáže teplo vhodným způsobem rekuperovat. Inteligentní vytápěcí systémy jsou dálkově spojeny s chytrým telefonem uživatele, který si může prostřednictvím internetu dálkově měnit nastavení svého domu.

**Řízení osvětlení** – Klasickým žárovkám v moderní domácnosti téměř odzvonilo. Na trh přicházejí úspornější zářivkové trubice a velký nárůst poptávky získaly také LED zdroje. Právě moderní typ LED osvětlení představuje úsporný zdroj světla, který zároveň disponuje svoji variabilitou. Z barevných LED pásek dokáže řídicí jednotka sestavit libovolnou barvu vyzařovaného světla. Moderní dům tedy dokáže ovlivnit naše „buňky bdělosti“. Modré světlo stimuluje naše pracovní nasazení a pozornost. Proto denní osvětlení obsahuje velký podíl této barvy a světla vyzařují odstíny barev, které jsou známé jako studená bílá. Na rozdíl tomu ve večerních hodinách, které jsou určeny spíše na odpočinek a přípravu ke spánku se mění i teplota barvy osvětlení. Modrá barva se začne ze spektra vytrácet a nastoupí převaha žlutých a oranžových odstínů, které více uklidňují. Samozřejmostí je i řízení osvětlení pomocí telefonu. Zde však systém řízení sahá ještě dále. Telefon dokáže díky aplikacím rozpoznat fáze našeho spánku a dokáže na ně reagovat. Dnes je již možné sestavit systém, který po usnutí zhasne všechna osvětlení a naopak dokáže znovu simulovat úsvit při probuzení. Zároveň dokáží systémy v kooperaci s inteligentním domem například ovládat venkovní žaluzie a tím přístup denního světla.

**Zabezpečovací systém** – Kamerové systémy či senzory nežádoucího vloupání jsou již tradiční záležitostí v oblasti zabezpečení. Systém při nečekané situaci okamžitě upozorňuje majitele například pomocí SMS zprávy, ale je možné kontaktovat pult zabezpečovací agentury, která na základě hlášení o narušení vyšle bezpečnostní hlídku. Samo-

zřejmostí je vzdálený přístup ke všem kamerám, z nichž některé mohou být otočné, a čidlům v domácnosti, takže si majitel či operátor může vytvořit představu o vzniklé situaci. Nebezpečí však nepředstavuje pouze vniknutí zloděje. Veliká hrozba se skrývá za vznikem požáru či únikem plynu. Proto mezi senzory inteligentního domu nechybí kouřové a plynové senzory, které jsou velmi citlivé a jsou schopny reagovat již na velice malou přítomnost těchto látek. V návaznosti na přesnou identifikaci jsou zavedeny systémy ochrany. V případě identifikace kouře jsou to automatické hasící systémy a únik plynu je zastaven automatickým uzavřením hlavního plynového uzávěru. [20]



Obr. 20 Prvky řízení inteligentního domu [OBR20]

### 5.5.2 Spotřebiče inteligentního domu

Spotřebiče moderní domácnosti jsou vybaveny speciálními senzory pro zlepšení jejich vlastností a mohou být opět propojeny s chytrým telefonem. Inteligentními spotřebiči lze postupovat od inteligentní pračky a sušičky, přes elektronické zubní kartáčky s připojením na Wi-Fi, řízení akvária až po chytrou televizi. Autor vybral pro ukázkou inteligence lednici.

Obyčejná lednice se poté mění na chytrou lednici, která zvládá hlídat lhůtu spotřeby jednotlivých potravin uvnitř a dokáže měnit teplotu jednotlivých polic podle potravin, jež se v daném sektoru nachází. Stejně tak dokáže efektivně využívat technologii ionizování vzduchu kolem potravin a likvidovat bakterie, které čerstvé potraviny znehodnocují. Navíc její řídicí systém dokáže spolupracovat s databází receptů a lednice uživateli napovídá, jaké pokrmy lze z potravin uvnitř uvařit. [21] Přílišná modernizace však má i své odpůrce, kteří rozsah pokroku vtípně glosují. Cituji komentáře z nezávislé diskuse: „Už aby dělali i s vnitřní kamerou. Pak bych do ni nemusel koukat tak dlouho, když je otevřená.“ „s tou vnitřní webkou se nedá než souhlasit, je nutné zkontrolovat z práce, či manželka nezapomněla na pivo :) a situaci potom případně alternativně vyřešit...“ [22]

V následujícím odstavci si autor dovolí uvést vlastní poznatky z vývoje chytrého kávovaru, jehož prezentaci viděl na výstavě „JA studentská firma roku 2017“ v Praze 5.4.2017.

*„Na výstavě JA studentská firma roku jsem viděl výsledek vývoje chytrého kávovaru. Projekt se zabíral interakcí kávovaru s aplikací v mobilním telefonu. Tento systém umožnil plánování času vaření kávy, ale zároveň umožňoval zadat množství vody a hrubost mletí. Projekt byl prezentován se slovy: „Nic Vás neprobudí lépe, jako vůně čerstvě připravené kávy.“ Představený systém samozřejmě umožňoval ovládání pomocí internetového rozhraní a poskytoval možnost tvorby uživatelských profilů pro uložení oblíbené konfigurace připravované kávy.“ [konec citace]*

## 5.6 Robotický mazlíček AIBO

Ani v oblasti robotických mazlíčků v domácnosti nezůstává servisní robotika pozadu. Robotický pejsek AIBO je vyroben s velkým důrazem na svůj design. Společnost Sony při vývoji tohoto pejska zpracovávala návrh s ohledem na jeho roztomilost, aby si rychle získal velkou oblibu mezi dětmi, ale i ve světě dospělých. Jeho dvacet dva stupňů volnosti představuje kinematiku zařízení, které dokáže věrně napodobovat gesta skutečného pejska a se svojí velikostí kolem 30 cm představuje velikost průměrného štěněte. Jeho oči jsou tvořeny dvěma kamerami, díky kterým dokáže rozeznávat okolní svět. Dále je vybaven stereo mikrofonem, který mu poskytuje možnost slyšet povely z okolního prostředí, ale díky duálnímu mikrofonu získá přehled i o směru, odkud povely vychází a následuje je. Mezi sensorickým vybavením nechybí ani dotykové a pohybové senzory, tudíž je pejsek citlivý například na hlazení a dokáže registrovat přicházejícího páníčka.[23,24]

Firma sony se mazlíčkovi AIBO věnovala již dříve, avšak v době ekonomické krize musela vývoj robotu pozastavit. V roce 2018 se však japonská firma chystá představit nejmodernějšího mazlíčka, jenž bude schopen „dovádět“ na jedno nabití celé dvě hodiny, dokáže si vybudovat virtuální citové vazby mezi jednotlivými členy rodiny, zvládá rozeznávat své přátele pomocí hlasu a je schopen aportovat elektronickou kostičku. Dokáže jednoduché povely jako například „dej pac“ a díky speciálnímu algoritmu se během svého života učí novým povelům a reakcím na své okolí. Žádný robotický pes AIBO tedy není stejný. Robotický mazlíček bude k dostání i v České republice a bude nosit cenovku mezi 45-50 tisíci korun. [25]



Obr. 21 Robotický mazlíček AIBO [OBR21]

## 5.7 Robotické vysavače

Robotické vysavače představují oblast servisních robotů, které jsou v lidských domácnostech nejrozšířenější. Velkou oblibu si získaly zejména pro svoji autonomnost. Fungují zcela nezávisle na lidské obsluze. Jediná nutná interakce robotického vysavače s majitelem je první „ukázka“ prostoru, ale existují i systémy, které si díky svým senzorům dokáží vytvořit plnohodnotnou mapu svého okolí. Na trhu se objevují různé varianty těchto robotů. Robotický vysavač může být vybaven různým stupněm inteligence a od toho se odvíjí také jeho cena. Ty nejlevnější roboty mají jenom pár základních senzorů, zpravidla infračervených. Za pomoci infračerveného světla a jeho odrazů dokáže detekovat překážku. Těsně před překážkou, jež může být tvořena stěnou místnosti nebo kusem nábytku, se otočí o úhel v rozmezí 90ti a 180ti stupni a pokračuje ve své jízdě. Je nutno podotknout, že taková metodika plánování pohybu je zcela náhodná a že některá stísněná místa nemusí robot vůbec vysát nebo se zde naopak zaseknout a vysávání tedy není dostatečně efektivní. [26]



Obr. 22 Robotický vysavač CleanMate [OBR22]

Problém s orientací řeší nástup moderních sensorických systémů. Z nich lze zmínit například ultrazvukové senzory, které vysílají zvuk do více stran najednou a dokáží lépe mapovat okolí robotu a tím zároveň lépe plánovat trajektorii. Dalším velice významným sensorickým systémem je vybavení robotického vysavače průmyslovou kamerou. Robotický systém snímá obrázek svého okolí až třiceti snímky za sekundu. Zpracování obrazu poté probíhá na základě hledání výrazných pixelů tj. pixelů, jež se od pixelů sousedních velmi liší svojí barvou či jasnem, neboť zde lze očekávat hranu například stolu, stěny či jiné překážky. Ve spolupráci s ultrazvukovým dálkovým senzorem si robot vytváří přesnou prostorovou mapu svého okolí a pomocí vnitřních algoritmů dokáže velice efektivně plánovat trajektorii vysílání. Je zřejmé, že pro zpracování obrazu je nutný vyšší výpočetní výkon a tyto roboty bývají dražší.

Poslední variantou navigace robotického vysavače v prostoru je vnitřní GPS. V udržovaném objektu se umístí orientační kostky, které vysílají na strop místnosti světelný infračervený signál o dané datové struktuře (vlnová délka, atd.) jež slouží pro identifikaci kostky. Robot sleduje tyto body na stropu a na základě triangulace (výpočtu pozice za pomoci sestrojování trojúhelníků s vrcholy ve známých bodech) je určována pozice robotu. Za pomoci dalších senzorů dochází opět k vytváření prostorové mapy objektu. Dalšími senzory jsou senzory, které zabraňují pádu ze schodů a sensorika, která slouží k vyhledávání dokovací stanice. Dokovací stanice slouží robotu jako zdroj elektrické energie pro napájení interních baterií a také jako stabilní místo pro parkování robotu po skončení úklidu.[26]

## 6 NÁVRH MANIPULÁTORU URČENÉHO PRO INTERAKCI S OKOLÍM

Projekt vznikl za motivace návrhu nízkonákladového manipulátoru, který má sloužit pro pochopení sériové kinematiky. Princip jeho funkce je jednoduchá. Uvnitř řídicího systému je umístěn časovač, který dokáže sledovat digitální hodiny. Na základě časové stopy řídicí systém určí hodnotu digitálních číslic. Nyní začne pomocí vnitřního algoritmu skládat černé kuličky na bílé pozadí. Tím dojde k zobrazení digitálního času v podobě HH:MM. K přeskládání číslic bude docházet v intervalu 1 minuty. Dále bude možné manipulátor nastavit tak, aby bylo možné přemísťování kuliček dočasně vypnout, aby stojem nebyl nikdo rušen například v době spánku.

Manipulátor je zajímavý zejména pro svoji kinematiku. Práce se detailně věnuje výpočtu inverzní kinematiky, která slouží k určení natočení jednotlivých kloubů podle požadované kartézské souřadnice koncového efektoru. Přímá kinematika zařízení je u sériových manipulátorů poměrně jednoduchá, neboť se souřadnice koncového efektoru dá vyjádřit jako složená goniometrická funkce. Naopak inverzní kinematika je o poznání složitější, neboť ve většině případů neexistuje analytické řešení. Musí se proto použít vhodně zvolená numerická metoda a řešení pseudopřesných kloubových souřadnic probíhá iteračně. Pojem pseudopřesných je zmíněn proto, neboť hledáme řešení, které se od absolutně přesného liší o větší či menší zvolenou odchylku.

Protože manipulátor není určen pro průmyslové aplikace, ale pro aplikace zábavné robotiky, posouvá se vizuální hledisko na přednější příčky. Věřím, že v rámci práce budou zachyceny všechny podstatné body, které umožní náhled do kinematiky manipulátoru stejně tak, jako bude umožněn náhled do principu řízení. Dále věřím, že bude pochopena oblast paralelního světa (řízení dvou pohybů zároveň za pomoci sériového mikrokontroleru Arduino).

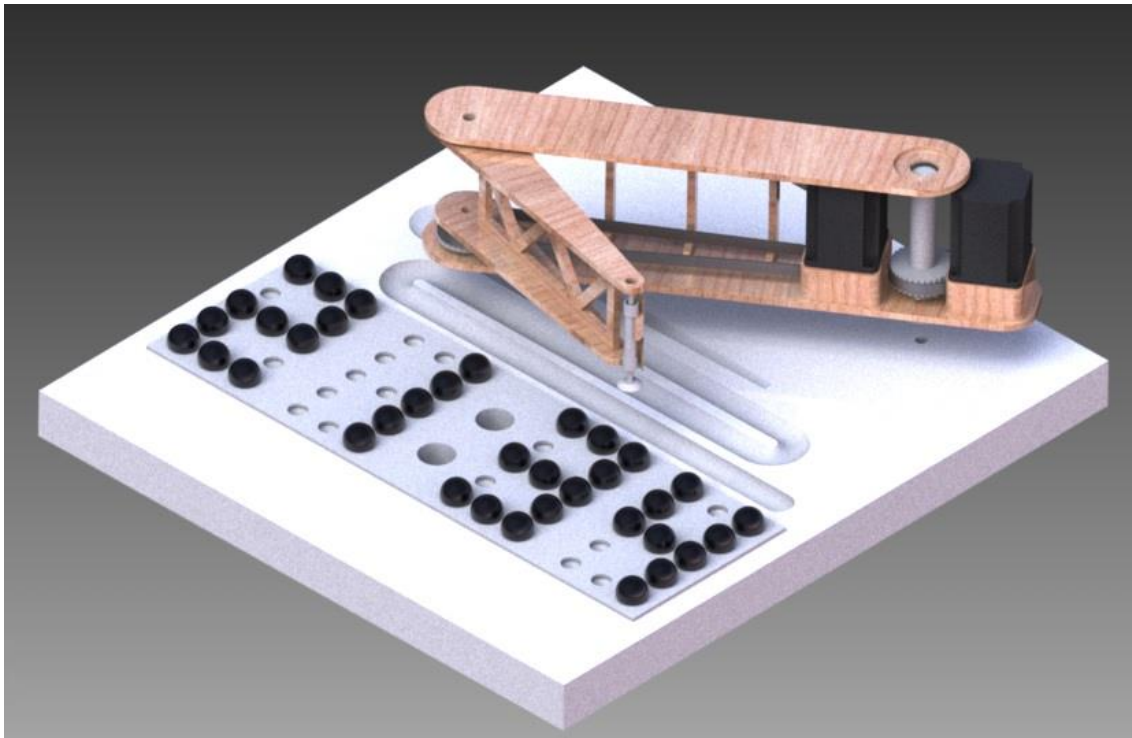
## 7 VÝBĚR VHODNÉ KINEMATIKY PRO DANOU APLIKACI

Pro jednoduchost celého mechanismu se nabízí využití kartézského uspořádání. Tohle uspořádání však s sebou nese translační kinematické dvojice, které jsou realizované jako suportové, smýkadlové či teleskopické. V každém případě však kolejnice dosahují značné délky, díky které stále musí některým směrem vystupovat. S ohledem na vizuální stránku manipulátoru bylo rozhodnuto tento typ manipulátoru zavrhnout.

Bylo nutné nalézt variantu kinematiky, která by přes své malé zástavbové rozměry dokázala obhospodařit poměrně velký operační prostor. Tak vznikla nosná myšlenka celého projektu použít kinematiku typu SCARA, která se skládá ze dvou rotačních kinematických dvojic pro vytvoření obecného rovinného pohybu v rovině XY a z následného výsuvu v ose Z, který bude reprezentovat nabírání kuliček ze zásobníku a jejich pokládání na správné místo. Při pohybu v rovině XY dochází ke změně orientace kuličky, ale již sám tvar použitého elementu napovídá, že na orientaci nebude záležet.

Na nosné podložce bude umístěn první rotační kloub, který bude umožňovat rotaci prvního ramene kolem vertikální osy  $Z$ , na konci prvního ramene bude umístěn druhý rotační kloub opět v ose  $Z$ , který bude sloužit k připojení druhého ramene. Na konci druhého ramene bude umístěn pneumatický píst s vakuovou přísavkou, který ukončí sériový kinematický řetězec.

Motory budou umístěny co nejbližší k prvnímu rotačnímu kloubu, aby se svojí hmotností projevovali co nejméně při určení momentu setrvačnosti celé konstrukce, neboť moment setrvačnosti má velký vliv na vztah mezi velikostí krouticího momentu od motorů a zrychlení pohyblivých ramen. Na základě rozvahy vznikl jednoduchý návrh celé kinematiky manipulátoru.



Obr. 23 Konceptuální návrh manipulátoru

## 8 ROZBOR KONSTRUKČNÍCH ČÁSTÍ

### 8.1 Podložka

Podložka má pro zařízení 3 základní funkce. První funkcí je vetknutí nosného čepu, na kterém je umístěna celá konstrukce manipulátoru. Nosná deska má půdorysné rozměry 600x600 mm. Dále je deska osazena zakládacím polem pro umístování kuliček. Pole je tvořeno čtyřmi číslicemi o celkovém počtu 52 (4·13) pozic. Nad rastrem číslic je umístěn gravitační zásobník. Do zásobníku zařízení odkládá kuličky v jeho horní části. Za pomoci gravitace poté kuličky probíhají skrz zásobník a řadí se postupně za sebou do pravého dolního rohu zásobníkové části. Z této pozice zařízení kuličky odebírá v případě jejich nedostatku v rámci číslicového rastru.

### 8.2 Rozmístění krokových motorů

Krokové motory jsou nejhmotnějšími prvky celého zařízení. Z pohledu momentu setrvačnosti, který závisí v první mocnině na hmotnosti a mocnině druhé na její vzdálenosti od osy otáčení, jsou to velice zásadní komponenty. Proto je s ohledem na nízký moment setrvačnosti zařízení vhodné umístit motory co nejbližší ose otáčení celého zařízení. Takový konstrukční přístup umožňuje větší uhlové zrychlení jednotlivých komponent i za použití krokových motorů s menším krouticím momentem a tedy i s menším protékajícím elektrickým proudem, což snižuje příkon celého zařízení. Krokové motory jsou v zařízení umístěny mimo osu otáčení, kterou obsluhují. S osou jsou spojeny za pomoci ozubených řemenic a řemenu. U obou převodů mezi motorem a kloubem je proveden převod do pomala, což zvyšuje krouticí moment působící na ovládání kloubu a zvyšuje teoretickou přesnost polohování zařízení.

### 8.3 Velké rameno

Při návrhu manipulátoru byl kladen důraz na ergonomii celého zařízení. Konstrukční myšlenky v začátcích návrhu byly celkem dvě. První variantou byla myšlenka, že bude konstrukce rozdělena po výšce a jednotlivá ramena budou umístěna nad sebou. Druhým a vítězným konceptem bylo vytvoření většího ramene jako obal, ve kterém bude možno skrýt motory a převody. Konstrukce většího ramene je tedy tvořena jako nosník tvaru C. Jelikož je rameno namáháno jak ohybem, tak i krouticím momentem od ramene menšího, je jeho zadní část vyplněna příhradovou konstrukcí, aby byla minimalizována jeho deformace.



Obr. 24 Detail velkého ramene s příhradovou konstrukcí

## 8.4 Malé rameno

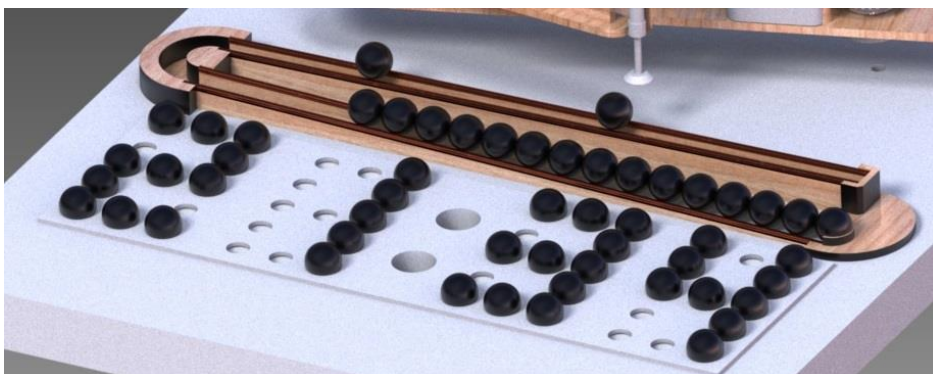
Konstrukce malého ramene byla zpracována s ohledem na požadavek jeho nízké hmotnosti, neboť je nejdálší částí od osy otáčení a má nejvýznamnější podíl na momentu setrvačnosti celé konstrukce. Malé rameno připomíná nosník tvaru I. Vrchní a spodní podstava se směrem ke konci ramene zužují a jsou spojeny svislou stěnou, jež je z důvodu snížení hmotnosti opět tvořena pouze trojúhelníkovými pruhy. Pod spodní podstavou je pevně přilepena ozubená řemenice pro přenos krouticího momentu z krokového motoru  $q_2$ .

## 8.5 Výsuvný píst s přísavkou

O výslednou manipulaci s kuličkami, kolmo směrem vzhůru a dolů, se stará pneumatický válec, který je na konci vybaven podtlakovou přísavkou, jež bude uchopovat jednotlivé kuličky. Pneumatický válec má výsuv 50 mm. Řízení uchopování bude řešeno za pomoci Arduina. První výstup bude ovládat stav pneumatického válce. Ovládání bude probíhat pouze na logické úrovni 1/0 neboli vysunuto/zasunuto. Pro aplikaci bylo užito jednočinného pneumatického válce s vratnou pružinou. Tlak ve válci lze nastavit tak, aby tlaková síla od vzduchu byla pouze o pár jednotek Newtonů větší než síla pružiny. Dochází tedy k vysunutí pístu, avšak síla pístnice se ruší se silou od pružiny a nedochází k nechtěnému namáhání konstrukce či deformaci při opření kuličky o zakládací pole či o zásobník. I řízení tvorby podtlaku probíhá na binární úrovni. Pro vytvoření vakua bude užito pneumatické součástky zvané ejektor. Do ejektoru vstupuje stlačený vzduch, prochází před zúžením, kde se zvýší jeho rychlost a sníží tlak. Za výstupem z trysky dochází k vytvoření podtlaku, jež je využíván v připojené přísavce. Vstup stlačeného vzduchu je ovládán pomocí pneumatického ventilu 3/2, který má tři vstupy a dvě polohy, ve kterých se může nacházet.

## 8.6 Gravitační zásobník

Při návrhu gravitačního zásobníku bylo nutné řešit otázku potřebné zásobní kapacity manipulátoru. Bylo učeno, že nejmenší počet kuliček je v poli umístěno v čase 1:11. Zde je v poli pouhých 15 kuliček z celkových 52 pozic. Naopak ve druhém extrému, jež má více variant, ku příkladu 22:58, 23:28 aj., je v poli zaráz naskládáno kuliček celkem 46. Rozdíl mezi mezními hodnotami počtů kuliček v poli je právě hledaná kapacita zásobníku. Zásobník musí pojmout tedy minimálně 31 kuliček. Na základě průměru 1 kuličky  $D=23$  mm je získána celková délka zásobní dráhy 713 mm.



Obr. 25 Detail gravitačního zásobníku

## 9 POUŽITÉ TECHNOLOGIE PRO VÝROBU

Pro minimalizaci nákladů při výrobě manipulátoru bylo rozhodnuto, že všechny použité konstrukční prvky a součásti budou obrobena na strojích, jimiž autor disponuje ve své dílně. Jediné nakupované součásti jsou kuličková ložiska a ozubené řemeny, jež není možné vyrábět v primitivních podmínkách.

### **Obrábění na 3-osé CNC frézce**

Na CNC frézce byly obrobena konzoly pro upevnění krokových motorů. Konzoly jsou obrobena z bukového dřeva a na konstrukci jsou upevněny za pomoci napínacího mechanismu, aby bylo možné vytvořit předpětí v ozubeném řemenu. Za pomoci CNC frézky byl vyroben také původní gravitační zásobník kuliček. Bohužel se valivé tření mezi kuličkou a dřevěným zásobníkem ukázalo tak velké, že bylo nutné zásobník přepracovat na variantu s bočními dráhami.

### **Laserové řezání na CO2**

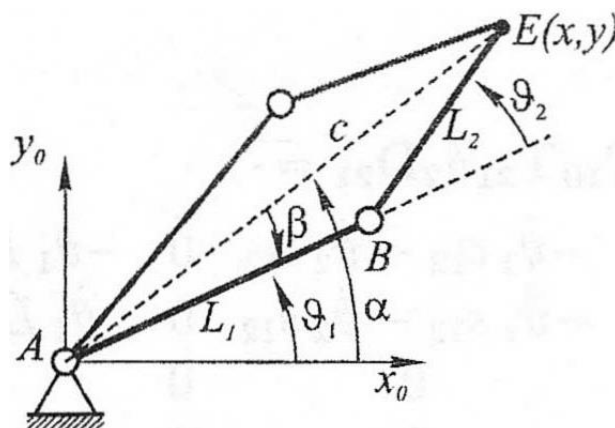
Zbylé části manipulátoru jsou vyrobeny z 3 mm tlusté překližkové desky, ze které jsou všechny komponenty vyřezány za pomoci CO2 laseru. Následně jsou jednotlivé díly spojovány do větších celků lepením. Navíc je většina nosných částí ještě vhodně přeplátována z ohledem na orientaci let dřeva. Překližkové desky 3 mm jsou vyrobené ze tří vrstev. Dvě vrstvy jsou orientovány léty podélně a prostřední vrstva kolmo. Tato struktura způsobuje, že deska má různé ohybové vlastnosti v podélném a příčném směru. Nosné části jsou tedy vyřezávány jedenkrát podélně, jedenkrát příčně a jsou lepeny do 6 mm silných vrstev, u kterých je eliminována tato negativní vlastnost tenké překližky.

## 10 VÝPOČET NEPŘÍMÉ KINEMATIKY MANIPULÁTORU

Pro výše zmíněnou aplikaci manipulátoru bude nutné řešit situace, kdy je známá požadovaná souřadnice koncového efektoru  $[X, Y]$ , avšak je neznámá konfigurace manipulátoru, která takové situaci odpovídá. Bude nutné nalézt kloubové souřadnice natočení  $[q_1, q_2]$ . Obecně se pro řešení tohoto typu úlohy používá inverzní kinematiky pomocí transformačních matic. Tato metoda je univerzální a je s ní možné řešit libovolný typ manipulátoru. Nevýhodou této metody je, že v jádru jejího řešení se skrývá numerická Gaussova metoda, takže řešení je iterační a na výpočet jedné souřadnice je vyžadováno mezi třemi až sedmi iteracemi.

Pro speciální případy manipulátorů, mezi něž patří i autorem navrhované zařízení, existuje řešení nepřímé kinematiky v podobě analytického řešení. Znamená to, že mezi jednotlivými rameny lze nalézt soustavu goniometrických funkcí, která má k sobě i inverzní funkci a lze pro ni vyjádřit explicitně hledanou proměnnou.[32]

### 10.1 Výpočet pomocí explicitní metody



Obr. 26 Kinematické schéma rovinného RR manipulátoru [OBR23]

Výpočet RR inverzní kinematiky je případ, kdy lze nalézt analytické řešení pomocí soustavy nelineárních algebraických rovnic. Za pomoci kosinové věty lze získat natočení  $\vartheta_2$ . Pro výpočet je k dispozici poloha koncového efektoru  $[X, Y]$  a délky ramen  $L_1$  a  $L_2$ . Pro výpočet je nutná znalost délky přepony  $c$ , která lze vypočítat podle Pythagorovy věty.

$$c^2 = x^2 + y^2 \quad (1)$$

Stejně, jako je délka  $c$  délkou pravoúhlého trojúhelníku v hlavním souřadném systému, je také přeponou trojúhelníku, jehož odvěsny tvoří ramena  $L_1$  a  $L_2$ . Tento trojúhelník však není pravoúhlý a proto je nutné použít kosinovou větu, jež zní: „Pro každý trojúhelník  $ABC$ , jehož strany mají délky  $a, b, c$  a jehož vnitřní úhel proti straně  $a$  má velikost  $\alpha$ , platí:  $a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos \alpha$ “ [27].

Využijme tedy kosinové věty pro náš manipulátor. Přeponu a reprezentuje délka  $c$ , strany  $b$  a  $c$  jsou poté ztvárněny délkami  $L_1$  a  $L_2$  a úhel mezi nimi lze vyjádřit jako  $\pi - \vartheta_2$ . Kosinova věta a její úpravu představují následující dva vzorce:

$$c^2 = L_1^2 + L_2^2 - 2 \cdot L_1 \cdot L_2 \cdot \cos(\pi - \vartheta_2) \quad (2)$$

$$\vartheta_2 = \pm \cos^{-1} \left( \frac{c^2 - (L_1^2 + L_2^2)}{2 \cdot L_1 \cdot L_2} \right)$$

Při úpravě se vyskytnou dva nečekané fenomény. První je, že při úpravě se mění výraz  $\pi - \vartheta_2$  na výraz  $\vartheta_2$ . Je to způsobeno vlastností funkce kosinus, jež má stejnou hodnotu pro oba zmíněné úhly. Druhým fenoménem je výskyt  $\pm$  před funkcí arkuskosinus, to je způsobeno tím, že se vyskytnou dvě možné varianty řešení. Tyto řešení jsou souměrná kolem přepony  $c$  a vznikají tedy i dvě řešení úhlu mezi rameny  $L_1$  a  $L_2$ . Kinematika navrhovaného manipulátoru je však uzpůsobena tak, že úhel  $\vartheta_2$  nabývá kladných hodnot. Algoritmus tedy vybírá správnou orientaci úhlu  $\vartheta_2$ . [32]

Vypočítat velikost úhlu  $\vartheta_1$  lze pouze nepřímo a to kombinací výpočtu úhlu  $\alpha$ , který zobrazuje úhel mezi základnou a přeponou  $c$ , od které je následně odečten úhel  $\beta$ . Úhel  $\beta$  se nachází taktéž mezi přeponou  $c$  a ramenem  $L_1$ . Úhel  $\alpha$  lze vypočítat z pravoúhlého trojúhelníku, který tvoří odvěsny  $x$ ,  $y$  a přepona  $c$ . Platí:

$$\tan \alpha = \left( \frac{y}{x} \right) \quad (3)$$

$$\alpha = \tan^{-1} \left( \frac{y}{x} \right)$$

Úhel  $\beta$  se nachází opět v trojúhelníku, který je tvořen rameny  $L_1$ ,  $L_2$  a přeponou  $c$ . Jak bylo zmíněno výše, tento trojúhelník není pravoúhlý, známe však všechny délky stran, a proto lze použít kosinovu větu pro výpočet úhlu  $\beta$  proti odvěsň  $L_2$ .

$$L_2^2 = L_1^2 + c^2 - 2 \cdot L_1 \cdot c \cdot \cos(\beta) \quad (4)$$

$$\beta = \cos^{-1} \left( \frac{L_1^2 + c^2 - L_2^2}{2 \cdot L_1 \cdot c} \right)$$

Pokud je známy oba úhly  $\alpha$  a  $\beta$ , jejich rozdílem je poté žádaná velikost kloubové souřadnice  $\vartheta_2$ .

$$\vartheta_2 = \alpha - \beta \quad (5)$$

$$\vartheta_2 = \tan^{-1} \left( \frac{y}{x} \right) - \cos^{-1} \left( \frac{L_1^2 + c^2 - L_2^2}{2 \cdot L_1 \cdot c} \right)$$

## 10.2 Výpočet pomocí metody transformační matice

Výše uvedená metoda umožňuje velice rychlý výpočet kloubových souřadnic. Funguje však pouze pro malou část kombinací kinematických dvojic a od většího počtu kloubů není vůbec použitelná. Proto bude v následující kapitole představena univerzální metoda pomocí transformačních matic. Dílčí transformační matice vznikají separací na jednoduché pohybové transformace. Je možné tedy celou transformaci souřadného systému rozdělit na rotace kolem jednotlivých souřadnicových os a translace ve směru x, y a z. Transformační matice má formát 4x4 buňky. Levá horní submatice představuje orientaci v prostoru, první tři řádky pravého sloupce odpovídá souřadnici [X, Y, Z] v prostoru. Spodní řádek matice má formát [0, 0, 0, 1], který dotváří čtvercový tvar matice a umožňuje maticové násobení. [32] Základní transformační matice lze představit 4:

### ROTACE KOLEM OSY X O $Q_x$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q_x) & -\sin(q_x) & 0 \\ 0 & \sin(q_x) & \cos(q_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### ROTACE KOLEM OSY Z O $Q_z$

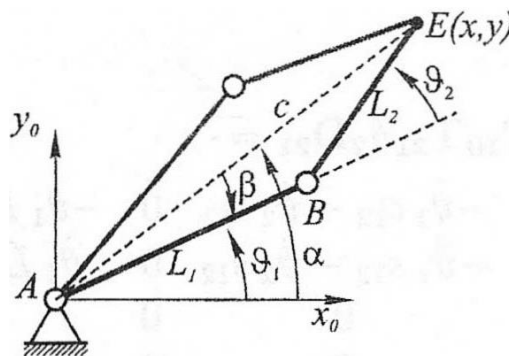
$$R_z = \begin{bmatrix} \cos(q_z) & -\sin(q_z) & 0 & 0 \\ \sin(q_z) & \cos(q_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### ROTACE KOLEM OSY Y O $Q_y$

$$R_y = \begin{bmatrix} \cos(q_y) & 0 & \sin(q_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(q_y) & 0 & \cos(q_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### TRANSLACE V SOUŘADNICOVÝCH OSÁCH O $T_x, T_y, T_z$

$$T_{xyz} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Obr. 27 Kinematické schéma rovinného RR manipulátoru [OBR23]

Převědme tedy kinematiku zvoleného RR manipulátoru na elementární transformace, které začínají v počátku souřadného systému a končí v koncovém efektoru E[x,y].

1. rotace kolem osy Z o úhel  $\vartheta_1$  v kladném smyslu (platí pravidlo pravé ruky – palec směřuje v kladném směru osy Z a prsty ukazují kladný směr otáčení)
2. posun ve směru nově vzniklé osy X o  $L_1$  v kladném smyslu
3. rotace kolem osy Z o úhel  $\vartheta_2$  v kladném smyslu
4. posun ve směru osy X o  $L_2$ .

Vyjádřeme elementární transformace pomocí vzájemnému násobení transformačních matic. Pro získání výsledné transformační matice autor využije SW Matlab.

$$T_{40} = T_{10} \cdot T_{21} \cdot T_{32} \cdot T_{43} \quad (6)$$

$$T_{40} = \begin{bmatrix} \cos(\vartheta_1) & -\sin(\vartheta_1) & 0 & 0 \\ \sin(\vartheta_1) & \cos(\vartheta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & L1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\vartheta_2) & -\sin(\vartheta_2) & 0 & 0 \\ \sin(\vartheta_2) & \cos(\vartheta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & L2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
T40 =
[ cos(v1)*cos(v2) - sin(v1)*sin(v2), - cos(v1)*sin(v2) - cos(v2)*sin(v1), 0, L2*(cos(v1)*cos(v2) - sin(v1)*sin(v2)) + L1*cos(v1) ]
[ cos(v1)*sin(v2) + cos(v2)*sin(v1),  cos(v1)*cos(v2) - sin(v1)*sin(v2), 0, L2*(cos(v1)*sin(v2) + cos(v2)*sin(v1)) + L1*sin(v1) ]
[ 0, 0, 1, 0 ]
[ 0, 0, 0, 1 ]
```

Jelikož bude manipulováno s kuličkami a nezáleží na jejich úhlové souřadnici, dále bude vynechána oblast orientace koncového efektoru, ale z transformační matice bude využito pouze pravého sloupce, který reprezentuje souřadnice [X, Y, Z]. Pro orientaci manipulovaného předmětu v prostoru by bylo nutné zařízení osadit ještě jednou rotační kinematickou dvojicí kolem osy Z. Pro úspěšnou aplikaci Gaussovi metody pro numerické řešení (princip metody bude vysvětlen v další kapitole) je nutné sestavit ještě Jakobiho matici dané kinematické soustavy. Jakobiho matice se vytvoří jako matice parciálních derivací jednotlivých souřadnic, opět k výpočtu bude použit SW Matlab. Pro tento konkrétní rovinný manipulátor lze při tvorbě Jakobiho matice lze vynechat souřadnici Z, neboť během výpočtu je stále Z rovno 0, a tedy i všechny parciální derivace by byly rovny 0. [32]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} L2 \cdot (\cos(\vartheta_1) \cdot \cos(\vartheta_2) - \sin(\vartheta_1) \cdot \sin(\vartheta_2)) + L1 \cdot \cos(\vartheta_1) \\ L2 \cdot (\cos(\vartheta_1) \cdot \sin(\vartheta_2) + \cos(\vartheta_2) \cdot \sin(\vartheta_1)) + L1 \cdot \sin(\vartheta_1) \\ 0 \end{bmatrix} \quad (7)$$

$$J = \begin{bmatrix} \frac{dx}{d\vartheta_1} & \frac{dx}{d\vartheta_2} \\ \frac{dy}{d\vartheta_1} & \frac{dy}{d\vartheta_2} \end{bmatrix} \quad (8)$$

$$J = \begin{bmatrix} -L1 \cdot \sin(\vartheta_1) - L2 \cdot (\cos(\vartheta_2) \cdot \sin(\vartheta_1) + \sin(\vartheta_2) \cdot \cos(\vartheta_1)) & -L2 \cdot (\cos(\vartheta_1) \cdot \sin(\vartheta_2) + \sin(\vartheta_1) \cdot \cos(\vartheta_2)) \\ L1 \cdot \cos(\vartheta_1) + L2 \cdot (-\sin(\vartheta_1) \cdot \sin(\vartheta_2) + \cos(\vartheta_2) \cdot \cos(\vartheta_1)) & L2 \cdot (\cos(\vartheta_1) \cdot \cos(\vartheta_2) - \sin(\vartheta_1) \cdot \sin(\vartheta_2)) \end{bmatrix}$$

Po sestavení Jakobiánu je k dispozici již dostatek vstupních údajů pro započítání výpočtu kloubových souřadnic  $\vartheta_1$  a  $\vartheta_2$ . Před započítáním ukázky postupu iteračního výpočtu si shrňme všechny známé veličiny. Kinematické schéma manipulátoru udává velikosti L1 a L2, dále víme požadované souřadnice [X, Y] koncového efektoru a je nám znám jakobián kinematické soustavy. Nyní bude představeno schéma iteračního numerického výpočtu.

```

while err>0.01 %% OPAKOVÁNÍ CYKLU, DOKUD ROZDÍL ŽÁDANÉ A VYPOČTENÉ SOUŘAD-
                NICE NEKLESNE POD DANOU MEZ
%%JAKOBIAN MANIPULÁTORU
J=
[-L2*(cos(v1)*sin(v2)+cos(v2)*sin(v1))-L1*sin(v1),-
L2*(cos(v1)*sin(v2)+cos(v2)*sin(v1));

L2*(cos(v1)*cos(v2)-sin(v1)*sin(v2))+L1*cos(v1),L2*(cos(v1)*cos(v2)-
sin(v1)*sin(v2))];

dX=X1'-X        %% ROZDÍL MEZI ŽÁDANOU A AKTUÁLNÍ POLOHOU MANIPULÁTORU
dq=inv(J)*dX    %% PŘEPOČET ROZDÍLU KARTÉZSKÉ SOUŘADNICE NA SOUŘADNICE KLOU-
                BOVÉ
q=q+dq          %% VÝPOČET NOVÉ KLOUBOVÉ SOUŘADNICE
T40=[           %% TRANSFORMAČNÍ MATICE MANIPULÁTORU
cos(v1)*cos(v2)-sin(v1)*sin(v2),-cos(v1)*sin(v2)-cos(v2)*sin(v1),
    0, L2*(cos(v1)*cos(v2) - sin(v1)*sin(v2)) + L1*cos(v1);

cos(v1)*sin(v2) + cos(v2)*sin(v1),cos(v1)*cos(v2)-sin(v1)*sin(v2),
    0, L2*(cos(v1)*sin(v2)+cos(v2)*sin(v1))+L1*sin(v1);

0, 0, 1, 0;

0, 0, 0, 1];

rE=[0;0;0;1]; %% PŘÍDAVNÁ SOUŘADNICE PRO VÝPOČET (REPREZENTUJE POZICI EFEK-
                TORU V SOUŘ. SYSTÉMU 4)

rE0= T40*rE    %% VÝPOČET NOVÉ POZICE V SOUŘADNÉM SYSTÉMU 0

err=norm(X-X1') %% VÝPOČET ODCHYLKY MEZI ŽÁDANOU A AKTUÁLNĚ VYPOČTENOU SOU-
                ŘADNICÍ

end

```

Na základě této smyčky lze získat iteračním výpočtem kloubové souřadnice  $\vartheta_1$  a  $\vartheta_2$  jako funkci požadované kartézské souřadnice  $[X, Y]$ . Iterační postup řešení však vyžaduje relativně velký výpočetní výkon, a proto bude v řízení manipulátoru použito výše rozebrané analytické řešení.

## 11 VÝVOJOVÉ DIAGRAMY FÁZÍ PROGRAMU

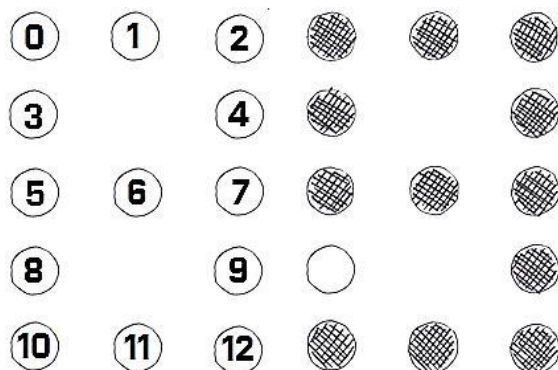
Pro řízení bude použito řídicího hardware Arduino, které bude zpracovávat stav matice kuliček na základě aktuálního času. Program separuje hodnoty na jednotlivé číslice a na základě podoby číslic vygeneruje maticovou mapu stavu, zda se na dané pozici nachází kulička nebo nikoliv. Na základě stavové mapy naplánuje přeskupení jednotlivých kuliček v rámci matice popřípadě ve spolupráci se zásobníkem. Dále bude program schopen vypočítat kloubové souřadnice na základě požadovaných souřadnic kartézských. Po vykonání přeskupení obrazce se zařízení vrací do své původní pozice.

Jelikož bude polohovací ústrojí manipulátoru řízeno za pomoci 2 krokových motorů, bude konstrukce obsahovat i dvě světelné závory, které umožní provádět referenci při každém návratu pracovního ramene a tím lze dosáhnout požadované přesnosti chodu i bez zpětné vazby v podobě senzorů pozice. Protože by tato práce měla sloužit jako učebnice pro ty, kteří si chtějí sestavit podobný manipulátor, bude použito prvků jednoduché syntaxe, aby bylo umož-

něno jednoduché pochopení kódu. Program nebude obsahovat zbytečné knihovny, neboť ty v kódu do určité míry tvoří tzv. černé skříňky, to znamená části kódu, které převádí vstupní data na údaje výstupní, ale o jejich zpracování uživatel nemá dostatečné povědomí.

### 11.1 Podoba digitálních číslic a indexy jednotlivých bodů

Digitální číslice jsou obecně známým tvarem, jenž je zpravidla tvořen třemi vodorovnými čarami a čtyřmi svislými. Jejich kombinací lze vyjádřit jakýkoliv tvar číslice. Digitální číslice pro tento účel budou tvořeny podobným systémem čar, avšak každá čára je tvořena pomocí tří kuliček viz obr. Na obrázku jsou taktéž znázorněny indexy jednotlivých pozic, tudíž lze adresovat kuličky pomocí pořadí číslice (číslíce 1. a 2. určuje hodiny, zbylé dvě minuty) a indexu v rámci číslice. Složením těchto dvou adres vzniká vektor, který má 52 hodnot, nazvěme ho STAV. Hodnoty jsou binární a 13 po sobě jdoucích hodnot určuje tvar jedné číslice. Dekódování je velice jednoduché. Pokud má vektor hodnotu dané pozice 0, kulička se zde nachází nebo má nacházet a opačně. Pro níže zobrazenou číslici 9, není kulička pouze na pozici s indexem 8, takže vektor vypadá takto:  $v=[1111111101111]$ .



Obr. 28 Rozvržení indexů pozic v rámci jedné číslice

### 11.2 Čtení hodnoty času z modulu DC3231

Protože držet stále přesnou hodnotu času pouze pomocí Arduina by nebylo možné a docházelo by ke ztrátě časové značky při vypnutí zařízení, bude konstrukce vybavena časovým modulem DC3231. Modul v sobě obsahuje krystalový oscilátor s označením TCXO. Počítáním vln, které vznikají oscilací, v kombinaci s délkou periody jedné vlny vzniká velice přesný systém pro měření času. Modul má svůj záložní zdroj energie v podobě 3V baterie, tudíž je schopen fungovat, i když je celý systém odpojen od elektrické sítě[28]. S Arduinem komunikuje pomocí sériové sběrnice I2C. Sériová sběrnice funguje díky posílání jednotlivých bitů zprávy za sebou. Zařízení se naladí na stejnou komunikační frekvenci, u sběrnice I2C na 100 kHz nebo 400 kHz. Vysílací zařízení vyšle tzv. start bit (nebo jejich sérii), druhé zařízení se aktivuje tímto bitem a začne přijímat zprávu s nalazenou frekvencí. Zpráva je uzavřena kontrolním bitem, pro ověření správnosti přijaté zprávy a bitem koncovým. [29,30]

Jelikož je problematika sériové komunikace velmi rozsáhlá a tato práce se ubírá jiným směrem, využijeme modul jako definovaný subsystém, který vrací hodnoty sekund, minut, hodin, dnu v týdnu, dne v měsíci a zároveň i rok. Konstruovaný manipulátor je připraven na zobrazení hodin a minut, a proto se využijí pouze dvě návratové hodnoty, zbylé mohou být použity například pro display při dalším rozšiřování konstrukce. Na základě přečtených hodnot budou získány základní proměnné pro řídicí algoritmus HODINA a MINUTA.

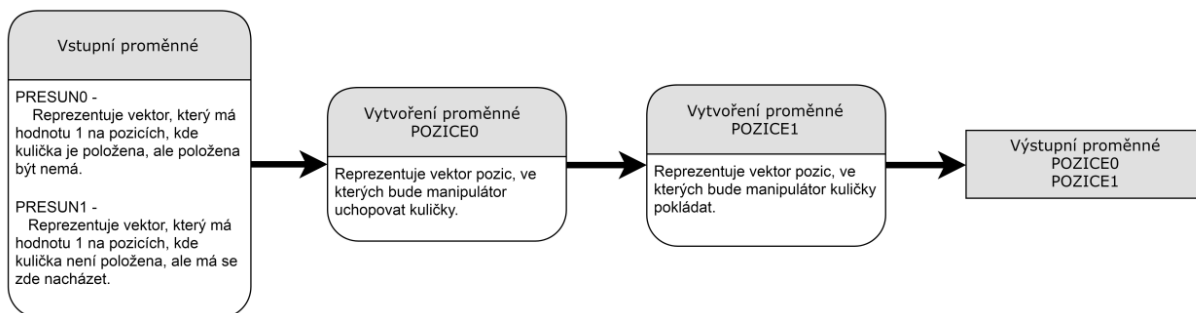


## 11.4 Plánování přesunů kuliček mezi pozicemi

Nyní přichází sestavení rozhodovacího algoritmu. V této části se projeví výhody na první pohled rozsáhlých vektorů STAV0/1 či PRESUN0/1. Tato struktura umožní řešit pole čtyř číslic jako celek, takže je možné výhodně přemísťovat jednotlivé kuličky, zkracovat trajektorie transportu a tím i čas potřebný na přestavení zobrazeného času. Metoda optimalizace přesunů je založena na principu maximálního přesunu kuliček v rámci pole a využití zásobníku až jako poslední možnost.

Pro vytvoření této metody přibudou další dvě proměnné POZICE0 a POZICE1. Tyto dva vektory budou mít délku opět 52 buněk. Jednotlivé pozice budou naplněny čísly, která označují index pozice v poli kuliček. Zbývá místa budou zaplněna indexem N z anglického NULL (v překladu prázdná buňka). Na konci této části programu mohou mít proměnné POZICE0 a POZICE1 následující podobu:

$POZICE0 = \{1, 5, 6, 8, 16, 32, 44, N, N, \dots, N\}$ ;  $POZICE1 = \{2, 7, 6, 9, 10, 22, 34, N, N, \dots, N\}$ ;  
Fyzicky tento zápis znamená, že z pozic s indexy 1, 5, 6, 8, 16, 32, 44 bude manipulátorem kulička odebrána a na pozice s indexy 2, 7, 6, 9, 10, 22, 34 bude naopak vložena. Obrázek vyjadřuje opět zjednodušené schéma, jehož rozšířenou verzi lze nalézt v příloze 6.



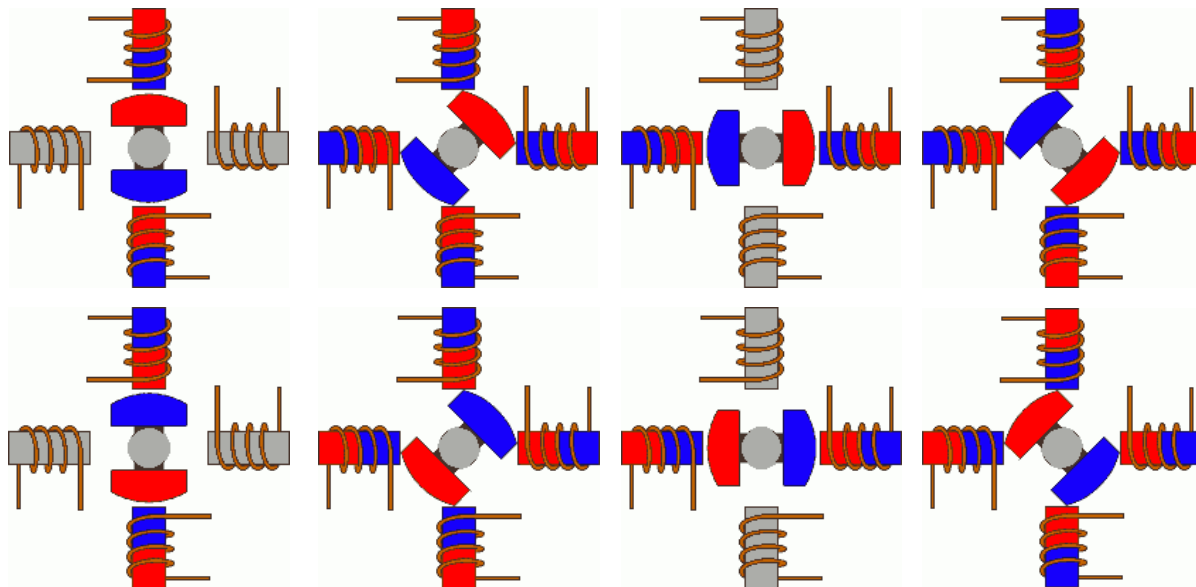
Obr. 31 Blokové schéma tvorby vektoru POZICE

Po vytvoření takto přehledných vektorů je již velice jednoduché sestavit metodu plánování trajektorie. Program bude brát postupně hodnoty pozic z vektoru POZICE1, kuličku uchopí a následně vezme souřadnici z vektoru POZICE2 a zde kuličku odloží. Tento postup se opakuje, dokud jsou v obou vektorech na pozicích indexy pozic. Ve chvíli, kdy již nebudou další indexy k dispozici, což se projeví výskytem proměnných NULL, začne manipulátor spolupracovat se zásobníkem. Zde je znatelný přínos řešení celého pole jako celku, jelikož lze přemísťovat kuličky v rámci více číslic a neprobíhá zbytečná interakce se zásobníkem, která by prodlužovala manipulační čas.

## 11.5 Krokový motor

Krokové motory patří mezi nejjednodušší servopohony, které se dají v praxi používat. Pracují krok po kroku, tedy každý pulz, respektive jeho náběžná hrana, odpovídá jednomu kroku. Krok poté odpovídá přesně danému úhlovému natočení rotoru. Rotor krokového motoru je tvořen soustavou permanentních magnetů a stator tvoří do hvězdice umístěné cívky. Cívky vytváří střídavé magnetické pole za pomoci střídavého přívádění stejnosměrného napětí a přitahují střídavě severní a jižní póly permanentních magnetů rotoru. Nejjednodušší krokové motory jsou vytvářeny jako dvoufázové, takže stator má dokola dvojici skupin cívek, označovaných jako A a B fáze. Každá ze skupin cívek může nabývat dvou stavů emitujících severní

a jižní magnetický pól. Takže celkem 4 základní kombinace magnetické. Rotor je poté dělen 50 ti zuby. Dohromady tedy 200 kroků, kterých je možno konstrukčně docílit. Moderní řídicí systémy však dokáží pracovat i s řízením napětí cívek za pomoci pulzní šířkové modulace (PWM), takže dokáží vytvářet tzv. mezikroky. Mezikroky vznikají rozdělením napájecího napětí ze stavů vypnuto/zapnuto na daný počet mikrokroků. Standartní řídicí ovladače mohou rozdělit krok na 2,4,8,16 mikrokroků. Novější systémy poté dokáží dělit hardwarový krok až na 256 mikrokroků. Pohyb motoru se stává plynulejší a pozitivní vliv má mikrokrokování i na hluk krokového motoru.



Obr. 32 Princip řízení krokového motoru s použitím mikrokrokování  $\frac{1}{4}$  [OBR24]

Mezi největší přednosti pohonu s krokovým motorem je přesně daná pozice rotoru, která je reprezentována počtem vykonaných kroků. S krokováním souvisí i velice jednoduché řízení pomocí digitálního řídicího systému. Ovladač krokového motoru vyžaduje pouze digitální informaci pro určení směru rotace motoru (tzv. DIR instrukce z anglického direction - směr) a generování pulzů pro informaci STEP (z anglického krok). Další velkou výhodou je krouticí moment, který působí i při nulových otáčkách motoru. S krouticím momentem také souvisí skutečnost, že tento motor nelze mechanicky přetížít. Po překročení maximálního krouticího momentu se motor přestane otáčet. Jedním z důležitých předností krokového motoru je také skutečnost, že funguje bez nutnosti zpětné vazby, přesto jsou v poslední době i krokové motory vybaveny snímači polohy, aby bylo při ztrátě kroku zabráněno možné kolizi v technologii.



Obr. 33 Krokový motor s přírubou NEMA23

Přes spoustu kladů má krokový motor i své zápory. Jelikož při provozování krokového motoru dochází ke změně polarit cívek, jejich indukčnost ovlivňuje proud protékající cívkou a tím i krouticí moment. Se zvyšujícími se otáčkami krouticí moment rapidně klesá. Standardně lze motor s ohledem na tuto skutečnost provozovat do 600ot./min. Někteří prodejci uvádí i 1500ot./min, ale je nutné podotknout, že při takových otáčkách už je hnací krouticí moment téměř nulový. Zmíněný problém s indukčností cívek krokového motoru řeší opět moderní typy ovladačů. Jsou napájeny vyšším napětím, než standardní ovladače a jsou vybaveny omezovačem výstupního proudu. Při vyšších otáčkách tedy dodávají obvodu větší napětí, které se projeví rychlejší změnou magnetického pole a zvýšením krouticího momentu krokového motoru. Naopak v klidu lze procházející proud omezit a tím snižovat oteplení motoru. Poslední z nevýhod je také výše zmínění hluk a vibrace spojení s krokovým trhavým pohybem. [31]

## 11.6 Výpočet kroků potřebných pro přesun mezi souřadnicemi

Pokud je nám známa již kompletní struktura prováděných přesunů mezi pozicemi, je nutné naplánovat ještě algoritmus, který bude řídit krokové motory. V programu se budou vyskytovat proměnné ACT\_POS\_Q1 a ACT\_POS\_Q2. Ty nám dávají zpětnou vazbu pro řízení natočení ve dvou ramen řízeného manipulátoru. Dále budou zavedeny proměnné POS\_Q1 a POS\_Q2, které budou získány za pomoci inverzní kinematiky z požadovaných cílových souřadnic [X,Y] koncového efektoru. Program vypočítá proměnné dQ1 a dQ2 prostým odečtením souřadnice požadované od souřadnice aktuální. Na základě dQ1 a dQ2 lze vypočítat proměnné dKROK1 a dKROK2, které reprezentují počet kroků, které musí vykonat krokové motory. Vztah mezi proměnnými dQx a dKROKx je následovný.

$$d_{KROKx}[-] = dQx [RAD] \cdot \text{počet kroků na jednotku kloubové souřadnice} \left[ \frac{1}{RAD} \right] \quad (9)$$

Pokud jsou známy potřebné počty kroků pro dosažení žádaných kloubových souřadnic, přechází program na vykonání řízení krokových motorů. Je žádané, aby byly oba krokové motory řízeny současně a bylo možno dosáhnout plynulého pohybu obou ramen. Zde se objevuje zásadní problém, neboť je nutné řídit dva krokové motory záraz, avšak procesor desky Arduino není schopen vysílat zcela paralelně řídicí signály pro krokové motory. Z tohoto důvodu autor v následující kapitole představí algoritmus, který zpracovává typ částečně paralelního řízení a zároveň řídí i rozběhový a zpomalovací průběh pohybu ramen.

## 11.7 „Paralelní“ řízení krokových motorů

Do části programu, který obsluhuje krokové motory, vstupují pouze 2 proměnné veličiny, které budou nazvány dq1 a dq2. Tyto veličiny představují počet kroků, které je nutné provést pro nastavení manipulátoru do aktuální žádané pozice. Veličiny dq1 a dq2 obsahují taktéž znaménko, které dává informaci o tom, zda je nutné kroky vykonávat v kladném směru otáčení nebo naopak. Proměnné dq1 a dq2 vznikají odečtením požadované pozice, jež se skrývá pod proměnnými next\_pos1 a next\_pos2 od aktuální pozice, která je v programu reprezentována proměnnými pos1 a pos2.

$$dq_{1(2)} = NEXT\_POS\_Q_{1(2)} - POS\_Q_{1(2)} \quad (10)$$

První část programu zpracuje údaje směru otáčení jednoduchým rozhodovacím algoritmem s podmínkou „když“. Pokud je proměnná  $dq_1$  kladná, nastaví smysl otáčení záporně a naopak. Pro druhý motor je postup naprosto shodný. Následně program vyhodnotí, která z proměnných  $dq_x$  má větší absolutní hodnotu. Motor obsluhující tuto proměnnou bude pro následný vykonaný pohyb označený jako master a bude se pohybovat maximální možnou rychlostí, aby bylo požadované pozice dosaženo v co nejkratším časovém intervalu. Druhý motor, označen jako slave, bude sledovat průběh rychlosti motoru prvního, avšak vykoná poměrně menší počet kroků vůči motoru master.

Poměr mezi počtem kroků motorů master a slave je reprezentován další proměnnou  $r$  (z anglického ratio). Proměnná  $r$  se vypočítá jako poměr kroků, které musí vykonat master, ku počtu kroků, které jsou požadovány po motoru druhém. Z předchozího odstavce je zřejmá skutečnost, že proměnná  $r$  bude vždy větší než 1.

$$r = \frac{dKROK_{MASTER}}{dKROK_{SLAVE}} > 1 \quad (11)$$

Nyní jsou již známy všechny potřebné proměnné, které jsou nutné pro paralelní řízení krokovým motorem. Celou proceduru řízení bude představovat FOR cyklus, který vykoná stejný počet kroků, jako je hodnota  $dq_{MASTER}$ . Po vykonání kroku 1. motoru následuje rozhodovací algoritmus, jež rozhodne o vykonání kroku druhého motoru. Zde bude vytvořena další proměnná  $j$ , která bude získána součinem pořadí kroku for cyklu  $i$  a proměnné  $r$ .

$$j[integer] = i[integer] \cdot r[float] \quad (12)$$

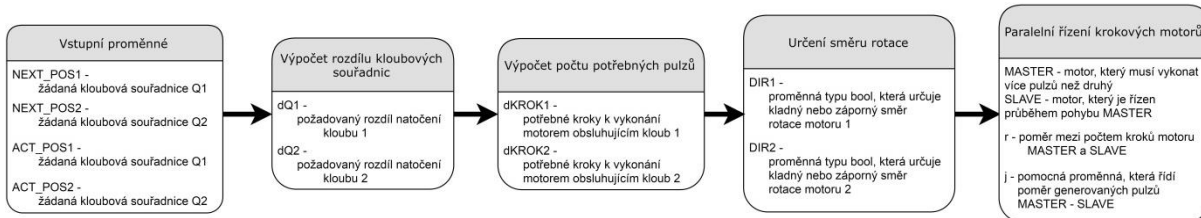
Jak je zmíněno již v symbolické rovnici. Program využije přechodu mezi datovým typem proměnných. Proměnná  $i$  je celé číslo, tedy datový typ integer, a  $r$  je číslo desetinné, datový typ float. Výsledkem násobení těchto dvou čísel vznikne obecně desetinné číslo, ale celočíselný datový typ proměnné  $j$  vše za desetinnou čárkou zahodí. Program této vlastnosti využije a krok pro motor slave bude generovat pouze v případě, když proměnná  $j$  změní svojí hodnotu oproti předchozímu průchodu for cyklem. Jako předchozí hodnota bude zavedena proměnná  $j_0$ . Pro praktické vysvětlení celé problematiky zmíním příklad.

### Příklad:

Požadují, aby motor MASTER vykonal 6 kroků a motor SLAVE kroky pouze 2. Vypočtu proměnnou  $r$ , která nabývá hodnoty 3. Vstupní hodnoty do FOR cyklu :  $i=\{1:6\}$ ,  $r=3$ ,  $j_0=0$

1.  $i=1, j=0.33 \rightarrow 0, j_0=0$  --- GENERUJI KROK PRO 1. MOTOR
2.  $i=2, j=0.67 \rightarrow 0, j_0=0$  --- GENERUJI KROK PRO 1. MOTOR
3.  $i=3, j=1.00 \rightarrow 1, j_0=0$  --- GENERUJI KROK PRO 1. a 2. MOTOR
4.  $i=4, j=1.33 \rightarrow 1, j_0=1$  --- GENERUJI KROK PRO 1. MOTOR
5.  $i=5, j=1.67 \rightarrow 1, j_0=1$  --- GENERUJI KROK PRO 1. MOTOR
6.  $i=6, j=2.00 \rightarrow 2, j_0=2$  --- GENERUJI KROK PRO 1. a 2. MOTOR

Výsledkem procedury je generování 6ti kroků pro první motor a 2 pro motor druhý. Je nutné podotknout, že během tohoto algoritmu může dojít ke ztrátě jednoho kroku v závěrečné fázi polohování. Avšak tuto skutečnost je možné zanedbat, neboť generované počty kroků jsou ve stovkách až tisících, takže jeden krok představuje zanedbatelnou chybu. Ve prospěch mluví také skutečnost, že se zařízení po provedení jednoho přesunu kuliček automaticky kalibruje na referenčních optických senzorech. Příloha 7 obsahuje rozsáhlou verzi níže uvedeného blokového schématu.



Obr. 34 Blokové schéma paralelního řízení krokových motorů

## 11.8 Řízení rychlosti krokových motorů

V předchozí kapitola se zaměřuje na distribuci kroků mezi jednotlivými krokovými motory. Nyní je však nutné specifikovat, s jakou frekvencí bude Arduino vysílat pulzy ke krokovým motorům a tím přímo ovládat průběh jejich rozjezdu a rychlosti. Průběh této frekvence je nepřímým ovládan za pomoci čekací funkce delay (z anglického zpoždění, prodloužení) resp. delay-Microseconds, jejíž funkce je stejná, pouze se do funkce dosazuje v mikrosekundách. Tato funkce pozastaví chod programu mezi jednotlivými pulzy a ovládá výstupní frekvenci. Výstupní frekvenci pulzů lze ovládat základními třemi způsoby.

### 11.8.1 Konstantní frekvence otáčení

Prvním způsobem, kterým se dá výstupní frekvence pulzů řídit, je frekvence konstantní. Motor se při tomto způsobu řízení točí konstantními otáčkami po celou dobu pohybu. Výhodou tohoto způsobu řízení je fakt, že proměnná  $d$ , která představuje dobu čekání, je po celou dobu konstantní. Doba čekání vychází nepřímým způsobem z požadované frekvence.

$$d = \frac{1\,000\,000}{f} [\mu s] \quad (13)$$

Nevýhodou tohoto přístupu k řízení jsou veliké rázy při rozjezdu a zpomalování zařízení. Změna rychlosti z hodnoty nulové na provozní probíhá skokově. Tento skok vyžaduje zrychlení zařízení směřující do nekonečna. Do nekonečna stoupá i požadovaný krouticí moment na hřídeli motoru. Tohoto jevu se nedá v praxi docílit. Fyzickým projevem tohoto řízení jsou veliké rázy a kmitání při pohybu a po přetížení krokového motoru dochází také ke ztrátě kroku a může dojít až ke kolizi zařízení. Proto je možné tento způsob řízení použít pouze v pomalu se pohybujících mechanismech.

### 11.8.2 Zrychlování a zpomalování po rampě

Dalším způsobem, kterým lze řídit frekvenci pulzů, je rozjezd a brždění pomocí rampy, která je tvořena přímkou. Pokud je počet požadovaných kroků menší než počet kroků nutných pro rozjezd na maximální rychlost a zastavení, má průběh rychlosti tvar rovnooramenného trojú-

helníku. V opačném případě má tvar souměrného lichoběžníku. Tento pohyb je lehčí na řízení než dále zmíněný pohyb po S-křivce, proto bude dopodrobna rozebrán princip výpočtu všech proměnných potřebných pro definici rychlosti. Pohyb po rampě je definován dvěma základními veličinami. První veličinou je směrnice, která reprezentuje velikost úhlového zrychlení  $\alpha$ . Druhou veličinou je maximální rychlost otáčení.

Maximální rychlost je zpravidla určená požadavky technologie, v níž je motor použit. Pro aplikaci v konstruovaném manipulátoru je ji možné v rozumných mezích zvolit. Pro zvolenou aplikaci přesunu kuliček byla zvolena maximální rychlost manipulátoru 90°/s. Maximální frekvence pulzů se poté vypočítá za pomoci trojčlenky z kombinace rychlosti [°/s] a počtu pulzů na vykonání 1°. Vztah představuje následující rovnice.

$$f = \omega_{max} * STEP_{ROT_{MAX}} [Hz] \quad (14)$$

$$f = 90 * 151.7037 = 13653 \text{ Hz}$$

Další neznámou je směrnice, se kterou může frekvence růst. Na tuto veličinu má vliv zejména moment setrvačnosti a kroučící moment dodávaný pomocí krokových motorů. Moment setrvačnosti byl experimentálně zjištěn za pomoci 3D modelu v programu Autodesk Inventor a jeho velikost kolem vertikální osy Z je:  $I_z = 0,047662 \text{ kg} \cdot \text{m}^2$ . Moment setrvačnosti byl odečten pro provozní konfiguraci s rozložením hmoty nejdále od osy otáčení, tedy pro ramena natočena v jedné přímce. V modelu jsou použity krokové motory s kroučícím momentem 0.9 Nm. Kroučící moment krokového motoru je nutné ještě vynásobit převodovým poměrem mezi řemenicí motoru a řemenicí spojenou se základnou. Pro konstruovaný manipulátor je počet zubů motorové řemenice  $z_1=15$  a počet zubů druhé řemenice  $z_2=32$ . Převodový poměr je tedy 32/15. Maximální zrychlení  $\alpha$  vypočítáme následujícím vzorcem.

$$M \cdot \frac{z_1}{z_2} = I \cdot \ddot{\varphi} \quad (15)$$

$$\ddot{\varphi} = \frac{M \cdot \frac{z_1}{z_2}}{I} = \frac{0,9 \cdot \frac{32}{15}}{0,0477} = 40,25 \text{ Rad} \cdot \text{s}^{-1}$$

$$[DEG \cdot \text{s}^{-1}] = [RAD \cdot \text{s}^{-1}] \cdot \frac{360}{2 \cdot \pi}$$

$$\ddot{\varphi} = \frac{40,25 \cdot 360}{2 \cdot \pi} = 2306,25 \text{ DEG} \cdot \text{s}^{-1}$$

Pokud je známa maximální požadovaná úhlová rychlost a úhlové zrychlení, jehož je manipulátor schopen dosáhnout, lze spočítat dobu rozběhu. S použitím rozběhového času lze vyjádřit úhel, o který se kloub během rozběhu natočí. Tento úhel bude potřebný pro vypočtení počtu kroků, přes které bude probíhat rozjezd a zastavení.

$$T = \frac{\omega}{\dot{\varphi}} = \frac{90}{2306,25} = 0,039 \text{ s} \quad (16)$$

$$\varphi = \omega_0 \cdot T + \frac{1}{2} \alpha \cdot T^2 = \frac{1}{2} \cdot 2306,25 \cdot 0,039^2 = 1,75^\circ \quad (17)$$

$$d_{KROK} = \varphi \cdot STEP_{ROT_{MAX}} = 1,75 \cdot 151,703694 = 265,48 \cong 266 \text{ kroků} \quad (18)$$

V této fázi výpočtu jsou známy všechny potřebné veličiny pro výpočet žádané směrnice  $K$ , která určuje strmost změny frekvence. Pro výpočet bude zaveden prakticky ověřený předpoklad. Rozběhová křivka nebude začínat čistě od nuly, neboť by na začátku docházelo k velikým čekacím prodlevám, ale začne se skokově na frekvenci 750 kroků za sekundu (odpovídá přibližně  $5^\circ$  za sekundu). Tento skok rychlosti můžeme považovat za přijatelný a předejde se tak velice pomalému rozběhu od 0. Pro počáteční frekvenci blíží se nule by se čas čekání blížil k nekonečnu podle následujícího vzorce.

$$\lim_{x \rightarrow 0} \frac{1}{k \cdot x} \gg \infty \quad (19)$$

Po zavedení počáteční nenulové frekvence  $f_0 = 750 \text{ Hz}$  se rozběh výrazně urychlí a zlepší se rychlost polohování celého manipulátoru. Pro zjištění směrnice  $K$  uijeme soustavu 2 lineárních rovnic, které odpovídají dvěma okrajovým podmínkám  $f_0 = 750 \text{ Hz}$  a  $f_{750} = 13563 \text{ Hz}$ .

$$f = K \cdot i + b \text{ [Hz]} \quad (20)$$

$$750 = K \cdot 0 + b \rightarrow b = 750$$

$$13653 = K \cdot 266 + 750$$

$$K = \frac{13653 - 750}{266} = 48,5 \text{ [-]}$$

$$f = 48,5 \cdot i + 750 \text{ [Hz]}$$

$$T = \frac{1\,000\,000}{f} = \frac{1\,000\,000}{48,5 \cdot i + 750} \text{ [\mu s]} \quad (21)$$

Díky tomuto výpočtu je již znám průběh proměnné  $d=T$ , která se bude předepisovat do funkce delayMicroseconds. Tímto postupem bude docíleno lineárního nárůstu úhlové rychlosti a pozvolnému zastavení celého zařízení.

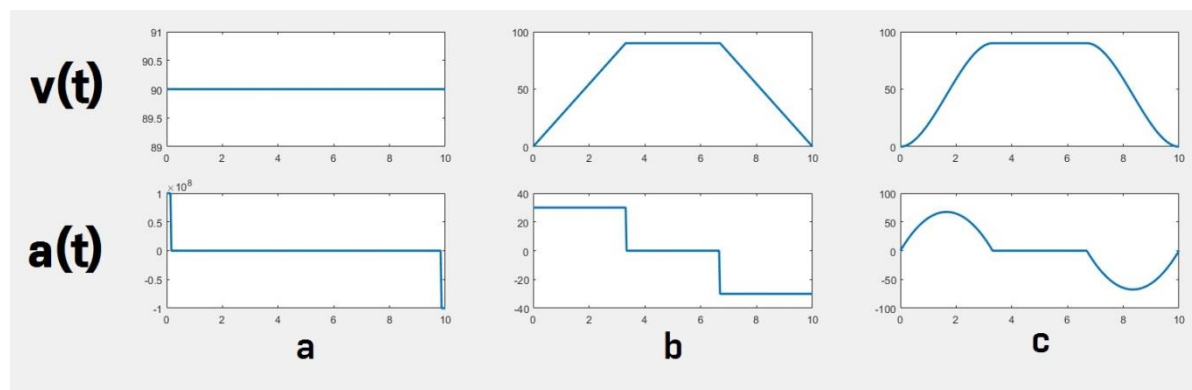
### 11.8.3 Zrychlování a zpomalování po S-křivce

Při tomto způsobu řízení rychlosti se mění předpis funkce, která reprezentuje průběh frekvence pulzů pro krokové motory. Kromě počáteční a koncové rychlosti se předepisuje ještě velikost zrychlení v počátečním a koncovém bodě. Aby nedocházelo ke skokovým změnám, jsou velikosti zrychlení v krajních bodech rovny nule. Tím je dosaženo hladkého průběhu rychlosti a dráhy během zrychlování a zpomalování. Výpočet koeficientů je však poměrně složitější. Nyní jsou předepsány celkem 4 okrajové podmínky, počáteční rychlost, zrychlení a koncová rychlost, zrychlení. Aby bylo možné dodržet všechny 4 podmínky, je nutné tedy reprezentovat průběh rychlosti polynomem se 4 nezávislými koeficienty. Tento polynom má průběh pomocí následujícího vzorce

$$y = A_3 \cdot x^3 + A_2 \cdot x^2 + A_1 \cdot x + A_0 \quad (22)$$

$$\frac{dy}{dt} = 3 \cdot A_3 \cdot x^2 + 2 \cdot A_2 \cdot x + A_1 \quad (23)$$

Do původní rovnice se dosazují okrajové podmínky, které předepisují rychlosti na počátku a konci zrychlování. Do rovnice derivované se poté předepisují okrajové podmínky pro zrychlení. Řešením 4 nelineárních rovnic jsou získány 4 parametry, které definují průběh rozběhové křivky. Výpočet terciální rovnice je však náročnější než výpočet lineární funkce. Tato skutečnost by zbytečně zpomalovala průběh programu, neboť tento výpočet probíhá při každém pulzu krokového motoru. S ohledem na fakt, že těchto cyklů mohou být jednotky až desítky tisíc během jednoho pohybu ramene, autor zůstane u řízení za pomoci přímkové rampy.



Obr. 35 Vztahy mezi rychlostí  $v(t)$  a zrychlením  $a(t)$  pro různé rozjezdové a brzdné rampy.  
 a) konstantní rychlost b) zrychlení po rampě c) zrychlení po S-křivce (polynom 3. stupně)

### 11.9 Referování zařízení po zapnutí

Zařízení je osazeno krokovými motory bez zpětné vazby. Velice lehké řízení takového servopohonu je však vyváženo skutečností, že po přivedení napájení do zařízení nemá řídicí systém žádnou informaci o aktuální konfiguraci ramen. Proto je před započítím manipulace nutné provést takzvané referování zařízení. Referování zařízení spočívá v najetí do pevné polohy. Pevná poloha je u zde konstruovaného manipulátoru reprezentována dvěma optickými snímači. První senzor snímá referenční polohu velkého ramene vůči základně a slouží pro referování polohy kloubové souřadnice  $q_1$ . Druhý senzor je umístěn mezi prvním a druhým ramenem a zajišťuje načtení výchozí kloubové pozice  $q_2$ . Referenční snímače se ve většině případů dá-

vají do téměř krajní pozice pohybové osy. Pokud by byl referenční snímač umístěn například uprostřed rozsahu pohybu, nebylo by možné přesně odhadnout, jakým směrem se má zařízení začít pohybovat.

Referování sestavovaného manipulátoru probíhá stejně jako u většiny obráběcích strojů s relativním odměřováním polohy. Ihned po zapnutí dochází k vykonání podprogramu REFEROVANI(), který zajistí najetí do výchozí pozice. Manipulátor se běžnou rychlostí rozjíždí v záporném směru kloubových souřadnic (opticky se skládá do sebe a zarovnává s horní hranou podstavy). Před vykonáním každého kroku se dotazuje optických snímačů na aktuální stav. V zařízení jsou použity optické snímače test1103, které v jednom pouzdře obsahují jak vysílač infračerveného záření, tak i přijímač v podobě fotodiody. Do mezery mezi vysílačem a diodou se během referování zasouvá clonka, jež může být tvořena z jakéhokoliv neprůsvitného materiálu.

Ve chvíli, kdy Arduino indikuje clonku v optické závoře, dočasně pozastaví pohyb jednoho z ramen a dále pohybuje ramenem druhým pro až do zaclonění druhého snímače. Po hrubém nalezení obou referenčních snímačů následuje odskok v kladném směru otáčení a operace je zakončena finálním najetím na snímače. Konečné najetí na referenční polohu bude probíhat za snížené rychlosti, aby byla zajištěna přesnost a především opakovatelnost najetí při každém spuštění. Výchozí konfigurace manipulátoru má zásadní vliv na pozicování během provozu. Chyba reference o pouhý jeden úhlový stupeň se může projevit chybou polohování i 10 mm, což může vést například k chybnému uchopení kuličky a následné ztrátě během přesunu.

### 11.10 Nulování zařízení

Již dříve zmíněna absence jakéhokoliv odměřování nebyla pro sestavení plně funkčního a spolehlivého systému přijatelná. Proto bylo rozhodnuto zavést programová opatření, jež tento problém částečně řeší. Pro tuto aplikaci vznikl podprogram nazvaný nulovani(). Náplní tohoto podprogramu není nic jiného než najetí na referenční spínače. Přiblížení však neprobíhá jako v případě referování. Manipulátor napozicuje klasicke rychlosti  $5^\circ$  před referenční pozici u každého ramene a poté probíhá dojezd sníženou rychlostí na optické závory. Program zároveň počítá kroky, jež jsou potřebné k dosažení referenční polohy. Podle počtu kroků následně vypočte odchylku od ideálních kloubových souřadnic. Odchylka mezi hodnotami je reprezentována proměnnou ERR\_POS<sub>1(2)</sub>. Pokud je alespoň jedna z těchto hodnot větší než dovolená odchylka, předpokládá se chyba polohování během procesu. Manipulátor se zastaví a vyzývá operátora k odstranění kuliček z pracovního pole do zásobníku a k restartování zařízení.

### 11.11 Schémata zapojení vstupů a výstupů

Pro kompletní zprovoznění celého zařízení je nutné HW Arduino fyzicky propojit se všemi komponentami za použití vstupně výstupních pinů, kterými Arduino disponuje. Na řídicí desce lze naléznout piny, které umožňují digitální vstup a výstup. Digitální vstup rozlišuje pouze 2 hodnoty 1/0 neboli proud přes pin teče/neteče. U digitálního výstupu se setkáváme opět s binární hodnotou, tentokrát napětí 0/5V DC. Těchto pinů je většina a jsou označeny čísly. Arduino MEGA 2560 jich má celkem 54. Celkem 15 pinů z celkového počtu umožňuje ještě vysílat analogový signál 0-5V DC, jež je realizován za pomoci pulzní šířkové modulace (z angličtiny PWM).

Dalším druhem pinů jsou piny Analogového vstupu. Tyto piny jsou osazeny digitálně analogovým převodníkem, takže dokáží měřit napětí v rozsahu 0-5V DC. U Arduina MEGA je konkrétně 10ti bitový převodník, takže lze 0-5V převádět na vstupní hodnotu 0-1023. Poslední oblastí pinů jsou piny obsluhující sériovou komunikaci. Díky ní mohou komunikovat jednotlivé desky mezi sebou, ale zároveň umožňuje i komunikaci s výstupním modulem. Nakonec je zde pin +5V DC, +3.3V DC, pin GND (ground – místní nulová hladina napětí), pin VIN, přes který se zapojí externí napájení 7-12V DC a posledním je pin RESET, který může být ovládán z externího resetovacího tlačítka. Celému zařízení dominuje USB port, který slouží pro připojení s počítačem při programování nebo diagnostice. USB port je u některých zařízení nahrazen Ethernetem. U těchto desek nalezneme port RJ-45



Obr. 36 Arduino MEGA [OBR25]

Pro vytvoření funkční aplikace mechanicko pneumatických hodin bylo nutné zajistit výstupy do krokových motorů a ovládat pneumatické ventily pro přísavku a pro jednočinný pracovní píst. Zpětná vazba je poté zajištěna dvojicí referenčních snímačů, v tomto případě optických a optické závory, která uzavírá prostor kolem zařízení. Vstupně výstupní piny Arduina jsou využity následujícím způsobem:

**digitální výstup:**

- pin 2** ovládání směru rotace motoru q1
- pin 3** vysílání pulzů pro motor q1
- pin 4** ovládání směru rotace motoru q2
- pin 5** vysílání pulzů pro motor q2
- pin 52** ovládání přísavky
- pin 53** ovládání výsuvu pístu osa Z

**analogový vstup:**

- pin A13** čtení stavu optické závory
- pin A14** čtení referenčního snímače q1
- pin A15** čtení referenčního snímače q2

**sériová komunikace s modulem reálného času**

- pin SCL** tzv. clock line – udává takt sériové komunikace
- pin SDA** vodič přenášející data

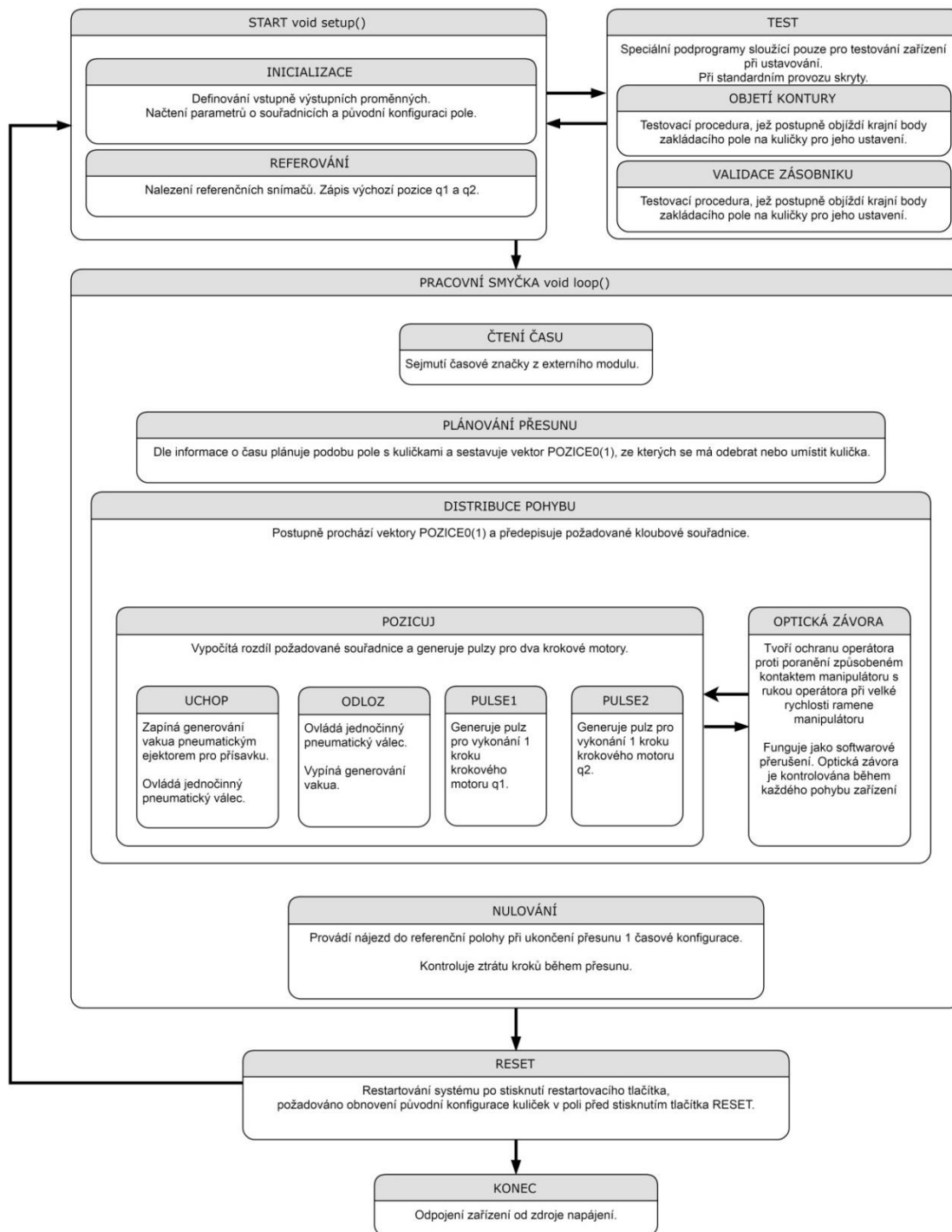
**napájení**

- pin 5V** pin s výstupním napětím 5V DC
- pin 3.3V** pin se sníženým výstupním napětím 3.3V DC
- 12V** externě dodávané napětí 12V DC
- GND** lokální nulová hladina potenciálu

Celkové graficky zpracované schéma vstupů nalezne čtenář na konci tohoto dokumentu v příloze číslo 2. Schéma výstupů je uvedeno vzápětí v příloze s pořadovým číslem 3.

## 11.12 Blokové schéma programu HODINY

Pro lepší porozumění funkce programu bylo vytvořeno blokové schéma, které zobrazuje závislosti mezi jednotlivými podprogramy. Kaskáda volání má převážně hierarchickou strukturu. Inicializační fáze programu, jež se provede pouze 1 po zapnutí zařízení nebo po provedení resetu je v kódu ohraničená funkcí void setup(). Ve schématu je reprezentována prvními dvěma bloky START a REFEROVÁNÍ, kdy se manipulátor z jakékoliv pozice ve vypnutém stavu navrácí do polohy referenční, která je určena optickými snímači. Dále už následuje pracovní smyčka, označována jako void loop(), která se cyklicky opakuje až do restartování či vypnutí přístroje. Kompletní program včetně komentářů je přiložen v příloze číslo 1.



Obr. 37 Blokové schéma programu HODINY

## 12 RIZIKA BĚHEM POUŽÍVÁNÍ MANIPULÁTORU

### 12.1 Úraz elektrickým proudem

protože se jedná o manipulátor, který je řízen za pomoci elektrické energie, je nutné zvážit, zda elektrické prvky nemohou způsobit nebezpečí uživateli. Pro eliminaci styku uživatele s elektrickým proudem je elektrická energie přiváděna pomocí standardizované koncovky, která je běžně užívána ve stolních počítačích, zásuvka je zároveň osazena i vypínačem a také pojistkou, která zařízení ochrání v případě vzniku zkratu. Délka rozvodu síťového napětí 230V AC je omezena na minimum a je vedena do spínaného zdroje, který je náležitě uzemněn ochranným vodičem. Za zdrojem vystupuje stejnosměrné napětí 12V, které lze v normálních podmínkách již považovat za bezpečné. Samozřejmostí však je, že všechny části pod napětím jsou přesto chráněny izolací a jsou použity standardní kryté svorkovnice, a je tak zabráněno nechtěnému dotyku s živými částmi. Velký důraz je kladen také na zabránění výskytu síťového napětí 230V na vnějším krytování zařízení. Prototyp je vyroben za pomoci desek na bázi dřeva či plastů. Oba materiály patří mezi izolanty, a proto není nutné vnější kryt chránit uzemněním.

### 12.2 Spolknutí či vdechnutí malých částí zařízení

Všechny části zařízení jsou spojeny se základní konstrukcí za pomoci nerozebíratelných vazeb. Jediné oddělitelné součásti jsou barevné kuličky, se kterými manipulátor pracuje. Průměr kuliček je konstrukčně stanoven na 25mm. I přestože jsou tyto elementy relativně velké, může dojít k jejich spolknutí malými dětmi. Je tedy výrobcem doporučeno používat manipulátor pouze za přítomnosti dospělé osoby.

### 12.3 Úraz skřípnutím mezi pohyblivými rameny

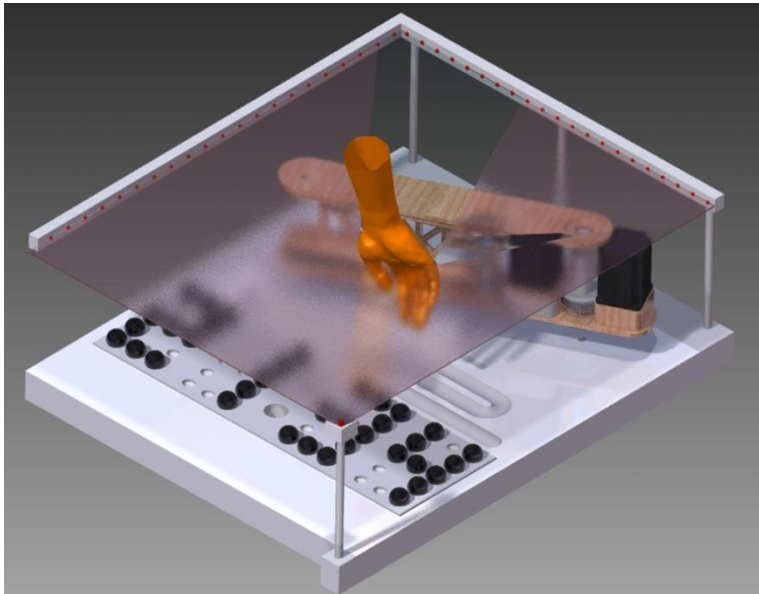
Dalším krizovým aspektem je skřípnutí částí ruky, zejména prstů, mezi pohyblivá ramena či mezi první rameno a základnu. V konstrukci jsou použity krokové motory s omezeným kroučícím momentem. Střížná síla, která vzniká mezi rameny je dostatečně malá na to, aby nedošlo k poranění ustřížením. Krokové motory při přetížení začnou tzv. ztrácet krok, přestanou se otáčet a celá konstrukce se zastaví. Jak však bylo zmíněno výše, systém při výskytu cizího tělesa zpomalí svůj pohyb, a proto by ke kontaktu nemělo vůbec docházet. Dalším rizikem použití manipulátoru je kontakt s částmi konstrukce během vykonávání přesunů.

#### 12.3.1 Řízení rychlosti manipulátoru během krizových situací

Při paralelním studiu programového kódu pro Arduino čtenář narazí na výskyt proměnné w (z anglického Waiting – čekání). Tato proměnná se vyskytuje v pozici násobku každého předpisu ve funkci delayMicroseconds a přímo tedy ovlivňuje rychlost manipulátoru za současného zachování tvaru rozběhových a zpomalovacích křivek. Pro standardní podmínky provozu manipulátoru je proměnná w rovna jedné a pohybové vlastnosti odpovídají standardním hodnotám, jež byly vypočteny v předchozí části práce. Pokud však nastane nečekaná situace, proměnná w začne nabývat hodnot větších než jedna. Tato skutečnost bude mít vliv na snížení frekvencí pulzů pro krokové motory a zpomalení mechanismu.

### 12.3.2 Optická závora

Pro zvýšení celkové bezpečnosti zařízení je nad manipulátorem sestavena optická závora. Závora je tvořena jedním vysílačem laserového paprsku a soustavou fotorezistorů. Laserový vysílač je konstruován speciálním způsobem, takže vycházející paprsek nemá podobu jednoho bodu jako například u laserového ukazovátka, ale vychází s vysílače v podobě laserové linie. Do dvou protilehlých stran od vysílače budou umístěny fotorezistory. Fotorezistor je polovodičová součástka, která mění odpor svého P-N přechodu právě na základě míry osvětlení. Celkem 38 fotorezistorů je zapojeno do série. Sériové zapojení rezistorů zajistí, že při přerušení paprsku cizím předmětem či závada některé ze součástí zapříčiní veliký nárůst celkového odporu světelné závory, který lze pohodlně detekovat jako poruchový stav.



Obr. 38 Ovlivnění roviny světelných paprsků vniknutím cizího předmětu

O stavu optické závory se bude starat samo Arduino. Před každou pohybovou instrukcí dojde ke zjištění aktuálního stavu závory a případnému bezpečnostnímu zásahu. Při narušení bezpečnostního prostoru dochází ke zpomalení celého manipulátoru. Prvotním návrhem byla skoková změna parametru  $W$  popsaného v předchozím odstavci. Parametr  $W$  se měl například 5x zvětšit a tím rychlost pohybu manipulátoru omezit na pětinu. Po pominutí krizové situace se měl parametr  $W$  opět skokově vracet ke své původní hodnotě. Je však zřejmé, že při skokové změně parametru  $W$  a s tím i skokové změně rychlosti manipulátoru by docházelo k velkým rázům, což není žádoucí. Rázy by způsobovaly vibrace ramene, ale především by mohly způsobit přetížení krokových motorů a následně ztrátu kroku, jež má velice negativní vliv na zařízení.

Z výše zmíněného důvodu byl zvolen postup, který umožňuje iterační nárůst nebo naopak pokles velikosti proměnné  $W$ . Pokud bude narušena optická závora, proměnná  $W$  se při každém generovaném pulzu bude postupně zvyšovat, dokud nedosáhne maximální hodnoty, která je nastavena z výroby. Po pominutí krizové situace se proměnná  $W$  opět postupně vrací do své původní hodnoty. Díky pružné změně parametru  $W$ , bude mít manipulátor při případné kolizi s končetinou operátora velice malou rychlost. Snížená rychlost umožňuje operátorovi zareagovat ucuknutím. V každém případě se minimalizuje možnost způsobení zranění střetem s pohybujícími se částmi.

## 13 TECHNICKÉ PARAMETRY ZAŘÍZENÍ

Vstupní napětí: **230V AC, 50Hz**

Typ napájecí sítě: **TN-S**

Pracovní napájení manipulátoru: **12V DC**

Počet os: **2+1 (2 elektrické, 1 pneumatická)**

Princip řízení elektrických os: **krokový motor + převod ozubeným hřebenem**

Princip řízení pneumatické osy: **pneumatický lineární motor**

Typ kinematiky: **SCARA** (rotace-rotace-translace)

Délky kinematických ramen: **350 mm; 235 mm**

Rozsah vertikální osy: **50 mm**

Velikost pracovního prostoru: **500 x 300 x 50 mm**

Maximální úhlová rychlost ramen: **90 °/s**

Maximální úhlové zrychlení ramen: **2306,25 °/s<sup>2</sup>**

Typ laseru optické závory: **červený 650 nm, 20 mW**

Řídící jednotka: **Arduino MEGA 2560**

Řízení krokových motorů: **DRIVER M542H (až 25600 kroků/otáčku)**

## 14 FINANČNÍ BILANCE ZAŘÍZENÍ

Jedním z klíčových bodů návrhu byla minimalizace nákladů sestavovaného manipulátoru. Autor níže uvádí reálné náklady na pořízení jednotlivých komponent. Do cenové kalkulace není zahrnut konstrukční a programátorský návrh. Taktéž není do cenové kalkulace zanesen strojní čas potřebný na výrobu součástí.

Součást	množství	cena/mj	celkem
Krokový motor 23HS8610	2	470 Kč	940 Kč
Driver krokového motoru	2	457 Kč	914 Kč
Arduino MEGA 2560	1	237 Kč	237 Kč
Modul reálného času	1	27 Kč	27 Kč
Nepájivé pole	1	70 Kč	70 Kč
Externí zdroj 12V	1	161 Kč	161 Kč
Pneumatický rozváděč 3/2	2	374 Kč	748 Kč
Regulátor tlaku	1	184 Kč	184 Kč
Přísavka s ejektorem	1	482 Kč	482 Kč
Jednočinný pneumatický válec	1	430 Kč	430 Kč
Ložiska	2+2	27; 50 Kč	154 Kč
Kuličky	45	6 Kč	270 Kč
Překližka 3mm	1.8 m <sup>2</sup>	80 Kč	144 Kč
Závitová tyč M20	0.2	70 Kč	14 Kč
Ozubené řemeny T5	2	270; 353 Kč	623 Kč
Spotřební materiál	-	-	300 Kč
<i>Návrh a programování manipulátoru</i>	<i>280 hod</i>	<i>320 Kč</i>	<i>89 600 Kč</i>
<b>celkem</b>			<b>5698 Kč</b>

## ZÁVĚR

V době pokročilých technologií a různých počítačových simulací je možné zkonstruovat prakticky jakoukoliv součást. Veliký přínos této práce však shledávám ve skutečnosti, že práce přistupuje k zařízení jako k celku. Celý vývoj začíná s čistým papírem a myšlenkou vytvořit interaktivní zařízení, které nebude mít velké zástavbové rozměry, pro uživatele bude jednoduché na obsluhu, estetické a jak již zadání práce předesílá, nebude mít vysoké pořizovací náklady.

S ohledem na tyto skutečnosti vzniká krok za krokem zařízení, na které se nabalují jednotlivé prvky konstrukce a řízení, se kterými se autor musí potýkat. Nejprve přichází otázka volby kinematiky zařízení, která bude dobře plnit svoji funkci a zároveň nebude těžké zpracovat úlohy kinematiky inverzní. Každá kinematická dvojice má svoji nepřesnost. U translace to může být vůle v pohybovém šroubu, u rotace zase úhlová vůle, jež s rostoucí délkou ramen lineárně zvyšuje odchylku od žádané polohy. A s rostoucím počtem kinematických dvojic se nepřesnosti sčítají a je čím dál těžší udržet přesnost polohování koncového efektoru. Taktéž roste moment setrvačnosti celého zařízení, čím vyšší je vzdálenost od základní osy otáčení. Je nutné správně volit rozložení hmoty a sílu motorů. Dále je nutné zahrnout i výpočet zrychlení celého zařízení a o to víc oblast brždění a maření kinetické energie.

V neposlední řadě je nutné analyzovat všechna úskalí spojená s operacemi, které má manipulátor vykonávat. Po prvním nahrání programu, zadání parametrů a spuštění následuje ještě relativně obsáhlá oblast lazení celého zařízení, aby bylo schopno plnit funkci, pro kterou je určeno. V laických podmínkách, ve kterých manipulátor vznikal, není možné dodržet takové tolerance, jaké jsou požadovány na klasický průmyslový manipulátor. Jsou tedy nutné dodatečné korekce polohy kuliček, zrychlení a úpravy původního programu.

Jsem rád, že mi díky diplomové práci bylo umožněno absolvovat další projekt, který v sobě zahrnuje teoretické znalosti nabyté během školní docházky, ale vyžaduje i řadu praktických zkušeností ať již z vlastní předchozí činnosti v oblasti návrhu obráběcích strojů pro hobby použití, či celosvětové databáze v podobě různých technických portálů. Výsledkem několikaměsíční práce je manipulátor, jež na první pohled neohromí velkou skupinu lidí. Naopak je nutné podotknout, že to není manipulátor zcela jednoúčelový. Program je napsán strukturovaně to řady podprogramů, takže za pomoci úpravy distribučního kódu lze manipulátor v relativně krátké době přeprogramovat na jinou manipulační činnost, ale lze ho vybavit joystickem či jinou vstupní periferií a tím diametrálně změnit jeho účel.

Do budoucna plánuji využít manipulátor v oblastech propagačních či reklamních. Zde bude vyžadováno vylepšení v oblasti uchopování kuliček. Přísavka bude osazena podtlakovým snímačem, jež bude umět detekovat přísátí kuličky a zároveň její ztráty. Dále bude zlepšena procedura sběru kuliček. Nepřesnost zařízení občas způsobí chybné pozicování, díky kterému nedojde k přísátí kuličky. Tato skutečnost bude moci být registrována za pomoci podtlakového senzoru. Následně dojde ke korekci polohy efektoru a opětovnému pokusu o zdvihnutí. Dále dojde ke změně gravitačního zásobníku, jež je nyní označen za nejslabší článek zařízení. Program bude také využit v následném projektu živého obrazu. Celé zařízení bude umístěno tentokrát vertikálně a pneumatickou manipulaci nahradí elektromagnetické systémy.

Práce poskytuje řadu poznatků z oblastí teoretických, ale i praktických zkušeností. Poskytuje řadu klíčových slov, podle kterých si čtenář může vyhledávat další informace z oblasti svého zájmu. Poskytuje námět na tvorbu a sestavení funkčního zařízení, zejména část automatizace, ale zároveň čtenáře vybízí k přemýšlení nad problémem v souvislostech za dosažením chtěného cíle. Práce je zakončena finanční bilancí, která čtenáři přináší náhled na podíl jednotlivých oblastí stroje na konečnou finanční částku.

## SEZNAM POUŽITÉHO SOFTWARE

### **Autodesk Inventor Professional 2013**

CAD systém pro modelování prostorových součástí, jejich složení do sestavy a analýzu kolizí jednotlivých prvků. Dále umožňuje základní napětíové analýzy a samozřejmě je generování 2D výkresů součásti či sestavy

### **EdgeCAM 11.0**

CAM software pro plánování technologie třískového obrábění. Při vývoji byl použit pro díly vyráběné frézováním na tříosé frézce. EdgeCAM je schopen sdílet 3D modely s CAD systémem Inventor přes soubory typu s příponou .sat. a úpravy modelu se se plynule přenáší mezi programy a adaptivně upravují technologický postup a generované dráhy.

### **ArtSoft MACH3**

Řídicí systém pro ovládání CNC frézky. Instaluje se standardně do personálního počítače a s hardwarem frézky komunikuje přes paralelní port počítače. Výhodou tohoto softwaru je možnost užití tzv. G-kódu, jež je v průmyslu naprostým standardem pro řízení obráběcích strojů.

### **Autodesk ArtCAM**

CAD systém určený pro práci s vektory. Je zaměřen na hobby použití, avšak je to skvělý nástroj pro převádění bitmapové grafiky na grafiku vektorovou a následné kombinování a ořezávání vektorů, vkládání textu a tvorba různých mase a polí. Je ideálním nástrojem pro vstupu do následujícího programu LaserCAD.

### **LaserCAD**

Program je primárně určen pro řízení CO2 laserového řezacího stroje. Program umožňuje plánování trajektorií na základě dodané vektorové grafiky, umožňuje rozlišovat operace na laserové řezání, kde dokáže řídit výkon laseru a posuvovou rychlost a dále na laserové gravírování čili popisování. Zde opět dokáže řídit výkon laseru a tím i hloubku textu.

### **MathWorks Matlab**

Program Matlab od firmy MathWorks umožňuje za pomoci interaktivního programového prostředí vytvářet výpočtové modely zařízení. Při návrhu konstrukce byl použit pro výpočet kloubových souřadnic pomocí inverzní kinematiky. Dále byl tento program použit generování a výpočtu transformačních matic a Jakobiho matice složené z parciálních derivací kartézských souřadnic dle souřadnic kloubových. V neposlední řadě byl Matlab využit pro výpočet průměrného doby přesunu kuliček během změny času. Velikou výhodou programu je možnost generování grafů na základě vypočtených hodnot, čehož bylo využito při vizualizaci rozjezdových a brzdných průběhů rychlostí. V neposlední řadě disponuje Matlab programovou částí s názvem Simulink. Simulink dokáže na základě vstupu mechanických veličin vytvořit fyzikální model celého zařízení a umožňuje optimalizace jeho řízení.

## SEZNAM POUŽITÝCH ZKRATEK A PROMĚNNÝCH

### použití zkratky v dokumentu

CAD	Computer Aided Design -počítačová podpora konstrukce a návrhu
CAM	Computer Aided Manufacturing - počítačová podpora plánování technologie
TTT	Uspořádání kinematických dvojic - Translace, Translace, Translace
RTT	Uspořádání kinematických dvojic - Rotace, Translace, Translace
RRT	Uspořádání kinematických dvojic - Rotace, Rotace, Translace
RRR	Uspořádání kinematických dvojic - Rotace, Rotace, Rotace
PID	Proporčně Integračně Derivační regulace
tzv.	takzvaně
GPS	Global Positioning System - globální systém určování pozice
KD	Kinematická Dvojice
PTP	Point To Point - označení pohybu bod po bodu, kdy nezáleží na trajektorii mezi body
CMR	Cambridge medical robotics
USA	Unated States of Amerika - Spojené státy americké
3D	označení 3-dimenzionálního prostoru
ČR	Česká republika
DARPA	Defence Advanced Research Projects Agency - Agentura pro pokročilé projekty v oblastech výzkumu a obrany
ASIMO	Advaced Step in Innovative MOBility - pokrokové kráčeující mobilní zařízení nové éry
ZMP	Zero Moment Point - Bod nulového momentu, vyjadřuje bod statické rovnováhy při řešení kráčení robotu
LED	Light Emitting Diode - dioda vyzařující světlo
SMS	Short message service - systém krátkých zpráv. Označení pro krátké textové zprávy přenášené telefonní sítí
JA	Junior Achievement - [Úspěch mladých studentů] je nejstarší a nejrozsáhlejší vzdělávací nezisková organizace na světě.
HH	Označení pro 2 číslice, které značí počet hodin v systému 24 hod/den
MM	Označení pro 2 číslice, které značí počet hodin v systému 60 min/hod
I2C	Inter-Integrated Circuit - sběrnice pro sériovou komunikaci mezi zařízeními
DC3231	Označení pro modul přesného času s oscilátorem
TCXO	Temperature Compensated Crystal Oscillators - Teplotně kompenzované krystalové oscilátory
HW	Hardware - označuje fyzicky existující vybavení stroje a zařízení
SW	Software - představuje veškeré programové vybavení stroje a zařízení
NULL	Vychází z latinského nullus - nic, žádná hodnota
PWM	Pulse Width Modulation - Modulace napětí za pomoci změny šířky pulzu
STEP	Vykonání jednoho kroku krokového motoru
DIR	Proměnná označující směr otáčení krokového motoru
RAD	označení úhlové jednotky - Radián
DEG	označení úhlové jednotky - uhlový stupeň
MASTER	Označení nadřazeného ze dvou krokových motorů
SLAVE	Označení podřazeného ze dvou krokových motorů
AC	Alternating Current - střídavý proud

DC	Direct Current - stejnosměrný proud
USB	Universal Serial Bus - univerzální sériová sběrnice
RJ-45	Standartní koncovka síťových kabelů

### **proměnné použité pro výpočty v dokumentu**

X	Poloha bodu ve směru osy x kartézského souřadného systému
Y	Poloha bodu ve směru osy y kartézského souřadného systému
Z	Poloha bodu ve směru osy z kartézského souřadného systému
q1	Označení kloubové souřadnice mezi 1. ramenem a základnou
q2	Označení kloubové souřadnice mezi 1. a 2. ramenem
L1	Délka 1. ramene manipulátoru
L2	Délka 2. ramene manipulátoru
q1	Označení kloubové souřadnice mezi 1. ramenem a základnou
q2	Odklon 2. ramene od ramene prvního
$\beta$	úhel mezi 1. ramenem a spojnicí mezi body [0,0] a [X,Y] souřadného systému
$\alpha$	Úhel mezi kartézskou osou x a spojnicí mezi body [0,0] a [X,Y] souřadného systému
x	označení osy x
y	označení osy y
z	označení osy z
Tx	Transformační matice pro posuv v ose x
Ty	Transformační matice pro posuv v ose y
Tz	Transformační matice pro posuv v ose z
Rx	Transformační matice pro rotaci kolem osy x
Ry	Transformační matice pro rotaci kolem osy y
Rz	Transformační matice pro rotaci kolem osy z
J	Jakobián složený z parciálních derivací $dx/dq1$ , $dx/dq2$ , $dy/dq1$ , $dy/dq2$
T40	Transformační matice představující transformaci mezi souřadným systémem koncového efektoru a počátkem
rEO	Pozice koncového efektoru v globálním souřadném systému
f	frekvence generování pulzů
M	Krouticí moment krokových motorů
I	Moment setrvačnosti zařízení
T	Délka periody mezi pulzy pro krokové motory
$\phi$	úhel natočení
$d\phi/dt$	úhlová rychlost
$d^2\phi/dt^2$	úhlové zrychlení
INF, $\infty$	označení pro nekonečna, použito v limitách

### **proměnné programu Arduino**

STAV1	52 pozicové pole určující požadovaný stav digitálních číslic
STAV0	52 pozicové pole určující aktuální stav digitálních číslic
HODINA	proměnná obsahující počet hodin v systému 24hod/den
MINUTA	proměnná obsahující počet minut v systému 60min/hod
PRESUN0	52 pozicové pole určující místa, kde kulička je, ale položena být nemá
PRESUN1	52 pozicové pole určující místa, kde kulička není, avšak po dokončení přesunu má být

---

POZICE0	52 pozicové pole obsahující indexy pozic, odkud má být kulička odebrána
POZICE1	52 pozicové pole obsahující indexy pozic, kam má být kulička položena
ACT_POS_Q1	aktuální hodnota kloubové souřadnice 1. kloubu
ACT_POS_Q2	aktuální hodnota kloubové souřadnice 2. kloubu
NEXT_POS_Q1	následující hodnota kloubové souřadnice 1. kloubu
NEXT_POS_Q2	následující hodnota kloubové souřadnice 2. kloubu
dQ1	rozdíl kloubové souřadnice 1. kloubu
dQ2	rozdíl kloubové souřadnice 2. kloubu
dKROK1	požadovaný počet kroků k vykonání motorem ovládaný 1. kloub
dKROK2	požadovaný počet kroků k vykonání motorem ovládaný 2. kloub
r	poměr mezi požadovanými počty kroků ( $dKROK_{max}/(dKROK_{min})$ )
j	celočíslná proměnná, která slouží pro rozhodnutí o přidělení pulzu motoru SLAVE
i	iterační konstanta vyžívána během for cyklů v programu
DIR1	proměnná určující směr otáčení 1. ramene
DIR2	proměnná určující směr otáčení 2. ramene
d	proměnná označující čekání v rámci programu
w	časová konstanta, která je závislá na okolí pracujícího manipulátoru
photo1	stav optické závory pro referenci polohy 1. ramene
photo2	stav optické závory pro referenci polohy 2. ramene
REF_POS1	referenční poloha ramene 1
REF_POS2	referenční poloha ramene 2
ERR_POS1	Rozdíl mezi referenční polohou a fyzicky vykonanou polohou ramene 1
ERR_POS2	Rozdíl mezi referenční polohou a fyzicky vykonanou polohou ramene 2

## SEZNAM POUŽITÝCH ZDROJŮ

- [1] KOLÍBAL, Zdeněk a Radek KNOFLÍČEK. *Morfologická analýza stavby průmyslových robotů*. Košice: Viena, 2000, 178 s. : il. ISBN 80-88922-27-5.
- [2] KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. Brno: Vysoké učení technické v Brně - nakladatelství VUTIAM, 2016, 787 stran : ilustrace (převážně barevné), portréty. ISBN 978-80-214-4828-5.
- [3] MAREK, Jiří. *Konstrukce CNC obráběcích strojů III*. Praha: MM publishing, 2014, 684 s. : il. ISBN 978-80-260-6780-1.
- [4] SHIGLEY, Joseph Edward, Charles R. MISCHKE, Richard G. (Richard Gordon) BUDYNAS, Martin HARTL a Miloš VLK. *Konstruování strojních součástí*. V Brně: VUTIAM, 2010, xxv, 1159 s. : il. ; 26 cm. ISBN 978-80-214-2629-0.
- [5] ČSN ISO 8373. *Roboty a robotická zařízení – Slovník*. 2. vydání. Published in Switzerland: ISO copyright office, 2012.
- [6] KNOWLEDGE CORNER. *ECHIDNA* [online]. Austrálie, 2017 [cit. 2018-05-20]. Dostupné z: <https://www.echidna-rocksaws.com.au/KnowledgeCorner/KnowledgeCorner.php>
- [7] TEJCHMANOVÁ, M. *Vývoj a výroba nízkonákladového robotu pro interakci s okolím*, Brno, Vysoké učení technické v Brně, Fakulta strojního inženýrství. 2016, 79 s., Vedoucí diplomové/bakalářské práce doc. Ing. Radek Knoflíček, Dr.
- [8] Chirurgický robotický systém Versius. *CMR surgical* [online]. Cambridge, 2017 [cit. 2018-05-18]. Dostupné z: <https://cmrsurgical.com/versius/>
- [9] Může být nejmenší chirurgický robot na světě největším průkopníkem laparoskopických operací?. *Techworld* [online]. Cambridge, 2017 [cit. 2018-05-18]. Dostupné z: <https://www.techworld.com/startups/could-worlds-smallest-surgical-robot-make-keyhole-surgery-mainstream-3668452/>
- [10] Moving Surgery Forward. *INTUITIVE SURGICAL* [online]. 2017 [cit. 2018-05-19]. Dostupné z: <https://www.intuitivesurgical.com/>
- [11] Rodina Telex. *TELEROB* [online]. Německo, 2018 [cit. 2018-05-19]. Dostupné z: <https://www.telerob.com/de/produkte/telex-familie>
- [12] BigDog. *Boston Dynamics* [online]. Massachusetts, 2018 [cit. 2018-05-19]. Dostupné z: <https://www.bostondynamics.com/bigdog>
- [13] Robotický "nosič" vykazuje ohromující reflexy. *New scientist* [online]. 2006 [cit. 2018-05-19]. Dostupné z: <https://www.newscientist.com/article/dn8802-robotic-pack-mule-displays-stunning-reflexes/#.VXdBjUYbCUk>

- [14] Boston Dynamics překračuje další hranici se svým psem na otevírání dveří. *The Guardian* [online]. Londýn, 2018 [cit. 2018-05-19]. Dostupné z: <https://www.theguardian.com/technology/2018/feb/13/boston-dynamics-spot-mini-door-opening-dog-crosses-threshold>
- [15] KÁRNÍK, Ladislav. *VYUŽITÍ SERVISNÍCH ROBOTŮ V NESTROJÍRENSKÝCH APLIKACÍCH*. Ostrava, 2011. Studijní opora. Vysoká škola báňská – Technická univerzita Ostrava, Fakulta strojní. [online]. [cit. 2018-05-19]. Dostupné z: <http://projekty.fs.vsb.cz/147/ucebniopory/978-80-248-2728-5.pdf>
- [16] ASIMO - Frequently Asked Questions. In: *ASIMO by Honda* [online]. Japonsko, 2004 [cit. 2018-05-19]. Dostupné z: <http://asimo.honda.com/downloads/pdf/honda-asimo-robot-fact-sheet.pdf>
- [17] CARDINAUX, Raphaël. *Mobile Robot modeling, Simulating and Programming*. [online]. Lausanne, 2014 [cit. 2018-05-19]. Semestrální projekt. Polytechnická škola Louisiana. Dostupné z: <https://www.cyberbotics.com/reports/asimo.pdf> .
- [18] MCQUEEN, Julia. Honda's Humanoid Robot, ASIMO, Inducted into Robot Hall of Fame™. In: *ASIMO by Honda* [online]. PITTSBURGH, 2004 [cit. 2018-05-19]. Dostupné z: [https://asimo.honda.com/ASIMO\\_DCTM/News/renditions/pdf/ASIMO--Robot\\_Hall\\_of\\_Fame.pdf](https://asimo.honda.com/ASIMO_DCTM/News/renditions/pdf/ASIMO--Robot_Hall_of_Fame.pdf)
- [19] ASIMO - technical information. In: *ASIMO by Honda* [online]. Japonsko, 2007 [cit. 2018-05-19]. Dostupné z: <http://asimo.honda.com/downloads/pdf/asimo-technical-information.pdf>
- [20] Inteligentní dům – chytrá domácnost. *Gevara* [online]. Praha, 2018 [cit. 2018-05-19]. Dostupné z: <http://www.gevara.cz/chytra-domacnost/inteligentni-dum-chytra-domacnost>
- [21] 4-Door Flex. *Samsung* [online]. [cit. 2018-05-19]. Dostupné z: <http://www.samsung.com/ca/consumer/home-appliances/refrigerators/4-door-flex/RF32FMQDBSR/AA/>
- [22] Samsung T9000: chytrá lednička s Androidem. *Svět Androida* [online]. 2013 [cit. 2018-05-19]. Dostupné z: <https://www.svetandroida.cz/samsung-t9000-chytra-lednicka-s-androidem/#comments>
- [23] Growth | aibo. *AIBO* [online]. [cit. 2018-05-19]. Dostupné z: <https://aibo.sony.jp/en/feature/feature3.html>
- [24] Sony chce přivést zpět na scénu svého robotického mazlíčka Aiba. *OTechnice.cz* [online]. 2017 [cit. 2018-05-19]. Dostupné z: <https://www.otechnice.cz/sony-chce-privest-zpet-na-scenu-sveho-robotickeho-mazlicka-aiba/>

- [25] Sony stakes its reputation on Aibo's return. *Nikkei Asian Review* [online]. 2018 [cit. 2018-05-19]. Dostupné z:  
<https://asia.nikkei.com/magazine/20180118/Business/Sony-stakes-its-reputation-on-Aibo-s-return?page=1>
- [26] CleanMate QQ-6S. *CleanMate* [online]. [cit. 2018-05-20]. Dostupné z:  
<http://www.cleanmate.cz/cleanmate-qq-6s/>
- [27] Kosinová věta. *Využití internetu ve výuce goniometrie na střední škole* [online]. Praha [cit. 2018-05-19]. Dostupné z:  
[http://kdm.karlin.mff.cuni.cz/diplomky/motyckova/Stranky\\_s\\_aplety/Kosinova\\_veta.html](http://kdm.karlin.mff.cuni.cz/diplomky/motyckova/Stranky_s_aplety/Kosinova_veta.html)
- [28] DS3231 - Extrémně přesný RTC/TCXO Krystal. *Maxim integrated* [online]. [cit. 2018-05-19]. Dostupné z:  
<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>
- [29] I2C. *SparkFun* [online]. [cit. 2018-05-19]. Dostupné z:  
<https://learn.sparkfun.com/tutorials/i2c>
- [30] Arduino a I2C sběrnice. *ITnetwork.cz* [online]. 2016 [cit. 2018-05-19]. Dostupné z:  
<https://www.itnetwork.cz/hardware-pc/arduino/arduino-a-i2c-sbernice>
- [31] Princip - krokový motor. *Pohonnatechnika.cz* [online]. [cit. 2018-05-19]. Dostupné z:  
<http://www.pohonnatechnika.cz/skola/motory/krokovy-motor>
- [32] GREPL, Robert. *Kinematika a dynamika mechatronických systémů* [skriptum]. Olomoučany: AKADEMICKÉ NAKLADATELSTVÍ CERM, s.r.o. Brno, 2007 [cit. 2018-05-20]. ISBN 978-80-218-3530-8.

## ZDROJE POUŽITÝCH OBRÁZKŮ [CIT. 2018-05-19]

- [OBR1] <https://www.nederman.com/en/products/product?product=89171>
- [OBR2] <http://www.demagcranes.com/en-us/products/hoist-units/compact-hoists/e-balancer>
- [OBR3] <http://www.echidna-rocksaws.com.au/Gallery/Photos.php?keyword=All&thumb=5&pic=360>
- [OBR4] <http://www.plasticportal.cz/cs/podla-cinnosti/luger-spol-s-ro/f/716/>
- [OBR5] <https://www.viastore.com/en/warehouse-systems/controls/>
- [OBR6] <http://www.directindustry.fr/prod/wittmann-battenfeld-france/product-78380-949621.html>
- [OBR7] TEJCHMANOVÁ, M. *Vývoj a výroba nízkonákladového robotu pro interakci s okolím*, Brno, Vysoké učení technické v Brně, Fakulta strojního inženýrství. 2016, 79 s., Vedoucí diplomové/bakalářské práce doc. Ing. Radek Knoflíček, Dr.
- [OBR8] <http://www.istrojirenstvi.cz/automatizace/precist.php?navez=novy-kompaktni-robot-scara-je-z-hlediska-nakladu-efektivni-alternativou-k-spider-robotum&id=64>
- [OBR9] <https://digitaltmuseum.org/021027315068/robot?aq=place%3A%22Bryne%22&i=2>
- [OBR10] <https://www.medgadget.com/2017/11/cambridge-medical-robotics-minimal-access-surgery-versius.html>
- [OBR11] <http://www.davincisurgery.com/da-vinci-surgery/da-vinci-surgical-system/>
- [OBR12] <http://www.telerob.com/de/produkte/telemax-familie/telemax-plus>
- [OBR13] <https://storage.pozary.cz/article/4/8/482839147c830/4c819df3e80c6.1400.jpg>
- [OBR14] <http://www.militarysystems-tech.com/military-gallery/cobham-mission-equipment---unmanned-systems/teodor---bomb-disposal-robot-in-action>
- [OBR15] <https://inhabitat.com/google-acquires-big-dog-and-cheetah-robot-manufacturer-boston-dynamics/>
- [OBR16] <https://www.bostondynamics.com/spot-mini>
- [OBR17] [http://asimo.honda.com/ASIMO\\_DCTM/News/images/inline/HME\\_delivery\\_close\\_inline2.jpg](http://asimo.honda.com/ASIMO_DCTM/News/images/inline/HME_delivery_close_inline2.jpg)
- [OBR18] <http://world.honda.com/ASIMO/history/>
- [OBR19] [http://asimo.honda.com/ASIMO\\_DCTM/News/images/inline/group200x200.jpg](http://asimo.honda.com/ASIMO_DCTM/News/images/inline/group200x200.jpg)
- [OBR20] <http://www.reco.cz/technologie/inteligentni-domacnost/pockethome-inteligentni-dum>
- [OBR21] <https://aibo.sony.jp/en/feature/feature3.html>
- [OBR22] <http://www.cleanmate.cz/cleanmate-qq-6s/>
- [OBR23] GREPL, Robert. *Kinematika a dynamika mechatronických systémů* [skriptum]. Olomoučany: AKADEMICKÉ NAKLADATELSTVÍ CERM, s.r.o. Brno, 2007 [cit. 2018-05-20]. ISBN 978-80-218-3530-8.
- [OBR24] [http://www.pcbheaven.com/wikipages/How\\_Stepper\\_Motors\\_Work/](http://www.pcbheaven.com/wikipages/How_Stepper_Motors_Work/)
- [OBR25] [https://www.tme.eu/cz/details/a000067/vyvojove-kity-arduino/arduino/arduino-mega2560-rev3/?brutto=1&gclid=Cj0KCQjwre\\_XBRDVARIsAPf7zZjFaNuFglCSib3ZHDUjup-q4fJKRROyV7PgQyecZgqT7GT1ucwzIAcaAqKBEALw\\_wcB](https://www.tme.eu/cz/details/a000067/vyvojove-kity-arduino/arduino/arduino-mega2560-rev3/?brutto=1&gclid=Cj0KCQjwre_XBRDVARIsAPf7zZjFaNuFglCSib3ZHDUjup-q4fJKRROyV7PgQyecZgqT7GT1ucwzIAcaAqKBEALw_wcB)

## SEZNAM PŘÍLOH

Příloha 1	KOMPLETNÍ KÓD PRO HW ARDUINO	P1
Příloha 2	SCHÉMA ZAPOJENÍ VSTUPŮ	P9
Příloha 3	SCHÉMA ZAPOJENÍ VÝSTUPŮ	P11
Příloha 4	VÝVOJOVÝ DIAGRAM ČTENÍ ČASU Z EXTERNÍHO MODULU	P13
Příloha 5	SESTAVENÍ VEKTORŮ POŽADOVANÝCH PŘESUNŮ KULIČEK	P15
Příloha 6	BLOKOVÉ SCHÉMA TVORBY VEKTORU POZICE	P17
Příloha 7	BLOKOVÉ SCHÉMA PARALELNÍHO ŘÍZENÍ KROKOVÝCH MOTORŮ	P19
Příloha 8	FOTOGRAFIE REÁLNÉHO ZAŘÍZENÍ	P21
Příloha 9	TECHNICKÁ DOKUMENTACE	EXTERNĚ V ZAKLÁDACÍ LIŠTĚ
	SCHÉMA SESTAVY	
	2-2018-161070 -MANIPULÁTOR HODINY	
	VÝKRES SESTAVY	
	1-2018-1 MANIPULÁTOR HODINY	
	VÝKRES SOUČÁSTI	
	2-2018-2 – VELKÉ RAMENO	
	2-2018-3 – GRAVITAČNÍ ZÁSOBNÍK	
	3-2018-4 – ZAKLÁDACÍ POLE	
	3-2018-5 – MALÉ RAMENO	
	4-2018-6 – KONZOLA Q1	
	4-2018-7 – KONZOLA Q2	
	4-2018-8 – SPOJOVACÍ ČEP	

# PŘÍLOHA 1 - KOMPLETNÍ KÓD PRO HW ARDUINO

```
// procedury casoveho modulu
#include "Wire.h"
#define DS3231_I2C_ADDRESS 0x68
byte decToBcd(byte val){return( (val/10*16) + (val%10) );} // prevod decimalnich cisel na binarni a naopak
byte bcdToDec(byte val){return( (val/16*10) + (val%16) );}

// definice pouzivanych promennych
byte *second; byte *minute; byte *hour; byte *dayOfWeek; byte *dayOfMonth; byte *month; byte *year;
int hodina;
byte minuta;
int l; // promenna ovladajici for cyklus pri distribuci pohybu
int i; // inkrement pro for cykly obecne
int ii; // pomocna promenna pro vypocty kroku
int j=0; // promenna ovladajici generovani kroku SLAVE motoru
int j0=0; // stav promenne j v predchozim kroku
int k=0; // promenna obsluhujici distribuji hodnot v promenne pozice
// kloubove souradnice pole
float
q1[52]={7.21,6.60,6.73,12.72,11.97,18.55,17.50,17.31,24.93,22.87,32.48,29.89,28.84,8.65,10.74,13.57,13.57,1
8.15,18.70,20.39,22.71,23.83,27.34,29.19,30.36,32.15,27.61,33.42,39.70,30.71,41.40,33.99,38.58,43.55,37.46,
46.09,41.17,44.91,49.00,51.27,57.89,64.35,51.73,63.73,52.88,58.40,63.95,54.59,64.86,56.81,61.54,66.40};
float
q2[52]={107.70,99.12,91.46,114.66,97.97,122.88,113.41,105.33,132.92,113.73,146.26,133.32,123.57,79.78,73.96
,68.74,86.32,75.56,93.52,87.86,82.89,101.51,90.82,110.49,104.55,99.49,56.30,53.90,52.34,63.79,60.15,71.58,6
9.50,68.16,79.75,76.47,88.44,86.47,85.22,51.75,52.65,54.42,59.61,62.06,67.65,68.42,69.95,75.99,78.19,84.75,
85.47,86.90};
float q1_zas_in =67.37; // kloubove souradnice vkladaciho mista do zasobniku
float q2_zas_in =29.99;
float q1_zas_out=64.76; // kloubove souradnice odebiraciho mista ze zasobniku
float q2_zas_out=43.5;
float NEXT_POS1=0; // promenna urcujici pozadovane kloubove souradnice
float NEXT_POS2=0;
float ACT_POS1=3; // aktualni kloubove souradnice
float ACT_POS2=11;
float REF_POS1=3; // referencni kloubove souradnice
float REF_POS2=11;
float ERR_POS1=0; // odchylka predikujici ztratu kroku motoru
float ERR_POS2=0;
// float STEP_ROT1=151.703694; // pocet kroku na vykonani 1° kloubove souradnice pri 1/128 deleni kroku
// float STEP_ROT2=132.740741;
float STEP_ROT1=19.6; // pocet kroku na vykonani 1° kloubove souradnice pri 1/16 deleni
float STEP_ROT2=16.6;
float w=8; // koeficient cekani... 1 pro mikrokrokovani 1/128, 8 pro mikrokrokovani 1/16
float ww=0; // pomocna promenna k w, slouzi pro navraceni k puvodni hodnote po zpomaleni
float dQ1 = 0; // rozdil kloubovych souradnic
float dQ2 = 0;
int dKROK1=0; // pozadovany pocet pulzu k dosazeni dQx
int dKROK2=0;
float dKROK1_float=0; // pozadovany pocet pulzu k dosazeni dQx, který umožňuje výpočet promenne r -
ratio
float dKROK2_float=0;
float r=1; // pomer kroku r - ratio
boolean smer1=false; // urceni smeru otaceni ramen
boolean smer2=false;
int STEP1 = 38; // aktualni pocet kroku od stavu Q1 = Q2 = 0
int STEP2 = 199;

boolean cislice1[13]={0,0,1,0,1,0,1,0,1,0,0,1}; // Podoba jednotlivych cislic
boolean cislice2[13]={1,1,1,0,1,1,1,1,0,1,1,1};
boolean cislice3[13]={1,1,1,0,1,1,1,1,0,1,1,1};
boolean cislice4[13]={1,0,1,1,1,1,1,1,0,1,0,0};
boolean cislice5[13]={1,1,1,1,0,1,1,1,0,1,1,1};
boolean cislice6[13]={1,1,1,1,0,1,1,1,1,1,1,1};
boolean cislice7[13]={1,1,1,0,1,0,1,1,0,1,0,0};
boolean cislice8[13]={1,1,1,1,1,1,1,1,1,1,1,1};
boolean cislice9[13]={1,1,1,1,1,1,1,1,0,1,1,1};
boolean cislice0[13]={1,1,1,1,1,1,0,1,1,1,1,1};
boolean stav1[52]={NULL}; // deklarace pole STAV
boolean stav0[52]={NULL};
boolean presun0[52]={0}; // deklarace pole presun
boolean presun1[52]={0};
int pozice0[52]={99}; // deklarace pole pozice
int pozice1[52]={99};

int photo1=0; // stav optické závory pro referenci polohy 1. ramene
int photo2=0; // stav optické závory pro referenci polohy 2. ramene

void setup()
{
  Serial.begin(9600); // zahajeni seriové komunikace s PC
  Wire.begin(); // zahajeni seriové komunikace s modulem realneho casu
  for(i=0;i<52;i++){pozice0[i]=99;} // nulovani vseh bunek pole pozice
  for(i=0;i<52;i++){pozice1[i]=99;}
}
```

```

pinMode(2,OUTPUT); // nastaveni vystupnich pinu pro ovladani motoru
pinMode(3,OUTPUT); // 2 - smer Q1, 4 - smer Q2
pinMode(4,OUTPUT); // 3 - krok Q1, 5 - krok Q2
pinMode(5,OUTPUT);
pinMode(52,OUTPUT); // ovladani vakua
pinMode(53,OUTPUT); // ovladani vysunuti pistu
// pin A14,INPUT // cteni referencniho snimace q1
// pin A15,INPUT // cteni referencniho snimace q2
// pin A13,INPUT // cteni stavu optické závory

for (i=0;i<26;i++){stav0[i]=1;} // deklarace počáteční podoby pole kulicek (zde dve cislice zleva plne kulicek)
ww=w; // zavedení pomocné proměnné, pro návrat k původní hodnotě w při změně rychlosti v rámci světelné závory
referovani(); // provedení referování manipulátoru
}

void loop()
{ cteni_casu();
  planovani_presunu();
  distribuce_pohybu();
  if (analogRead(14)>10 || analogRead(15)>10) {nulovani();}

  /* objeti_kontury();
   validace_zasobniku */
}

void cteni_casu(){
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year; // deklarovani promennych sekunda, minuta ... rok
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year); // cteni hodnot z modulu realneho casu
  Serial.print(hour,DEC);Serial.print(":");Serial.print(minute,DEC);Serial.print(":");Serial.println(second,DEC); // vypis aktualni hodiny, minuty, sekundy
  hodina=hour;
  minuta=minute;

  // PLANOVANI 1. CISLICE
  if (hodina <10){for (i=0;i<13;i++){stav1[i]=0;}}
  if (hodina>9 && hodina<20){for (i=0;i<13;i++){stav1[i]=cislicel[i];}}
  if (hodina>19){for (i=0;i<13;i++){stav1[i]=cislice2[i];}}

  // PLANOVANI 2. CISLICE
  if (hodina==0 || (hodina-10)==0 || (hodina-20)==0 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice0[i];}}

  if (hodina==1 || (hodina-10)==1 || (hodina-20)==1 )
    {for (i=0;i<13;i++){stav1[i+13]=cislicel[i];}}

  if (hodina==2 || (hodina-10)==2 || (hodina-20)==2 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice2[i];}}

  if (hodina==3 || (hodina-10)==3 || (hodina-20)==3 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice3[i];}}

  if (hodina==4 || (hodina-10)==4 || (hodina-20)==4 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice4[i];}}

  if (hodina==5 || (hodina-10)==5 || (hodina-20)==5 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice5[i];}}

  if (hodina==6 || (hodina-10)==6 || (hodina-20)==6 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice6[i];}}

  if (hodina==7 || (hodina-10)==7 || (hodina-20)==7 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice7[i];}}

  if (hodina==8 || (hodina-10)==8 || (hodina-20)==8 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice8[i];}}

  if (hodina==9 || (hodina-10)==9 || (hodina-20)==9 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice9[i];}}

  // PLANOVANI 3. CISLICE
  if (minuta <10)
    {for (i=0;i<13;i++){stav1[i+26]=cislice0[i];}}

  if (minuta>9 && minuta<20)
    {for (i=0;i<13;i++){stav1[i+26]=cislicel[i];}}

  if (minuta>19 && minuta<30)
    {for (i=0;i<13;i++){stav1[i+26]=cislice2[i];}}

  if (minuta>29 && minuta<40)
    {for (i=0;i<13;i++){stav1[i+26]=cislice3[i];}}

```

```

if (minuta>39 && minuta<50)
    {for (i=0;i<13;i++){stav1[i+26]=cislice4[i];}}

if (minuta>49 && minuta<60)
    {for (i=0;i<13;i++){stav1[i+26]=cislice5[i];}}

// PLANOVANI 4. CISLICE
if (minuta==0 || (minuta-10)==0 || (minuta-20)==0 || (minuta-30)==0 || (minuta-40)==0 || (minuta-50)==0 ||
(minuta-60)==0)
    {for (i=0;i<13;i++){stav1[i+39]=cislice0[i];}}

if (minuta==1 || (minuta-10)==1 || (minuta-20)==1 || (minuta-30)==1 || (minuta-40)==1 || (minuta-50)==1 ||
(minuta-60)==1)
    {for (i=0;i<13;i++){stav1[i+39]=cislice1[i];}}

if (minuta==2 || (minuta-10)==2 || (minuta-20)==2 || (minuta-30)==2 || (minuta-40)==2 || (minuta-50)==2 ||
(minuta-60)==2 )
    {for (i=0;i<13;i++){stav1[i+39]=cislice2[i];}}

if (minuta==3 || (minuta-10)==3 || (minuta-20)==3 || (minuta-30)==3 || (minuta-40)==3 || (minuta-50)==3 ||
(minuta-60)==3)
    {for (i=0;i<13;i++){stav1[i+39]=cislice3[i];}}

if (minuta==4 || (minuta-10)==4 || (minuta-20)==4 || (minuta-30)==4 || (minuta-40)==4 || (minuta-50)==4 ||
(minuta-60)==4)
    {for (i=0;i<13;i++){stav1[i+39]=cislice4[i];}}

if (minuta==5 || (minuta-10)==5 || (minuta-20)==5 || (minuta-30)==5 || (minuta-40)==5 || (minuta-50)==5 ||
(minuta-60)==5)
    {for (i=0;i<13;i++){stav1[i+39]=cislice5[i];}}

if (minuta==6 || (minuta-10)==6 || (minuta-20)==6 || (minuta-30)==6 || (minuta-40)==6 || (minuta-50)==6 ||
(minuta-60)==6)
    {for (i=0;i<13;i++){stav1[i+39]=cislice6[i];}}

if (minuta==7 || (minuta-10)==7 || (minuta-20)==7 || (minuta-30)==7 || (minuta-40)==7 || (minuta-50)==7 ||
(minuta-60)==7)
    {for (i=0;i<13;i++){stav1[i+39]=cislice7[i];}}

if (minuta==8 || (minuta-10)==8 || (minuta-20)==8 || (minuta-30)==8 || (minuta-40)==8 || (minuta-50)==8 ||
(minuta-60)==8)
    {for (i=0;i<13;i++){stav1[i+39]=cislice8[i];}}

if (minuta==9 || (minuta-10)==9 || (minuta-20)==9 || (minuta-30)==9 || (minuta-40)==9 || (minuta-50)==9 ||
(minuta-60)==9)
    {for (i=0;i<13;i++){stav1[i+39]=cislice9[i];}}

Serial.print("Stav0 "); // vypis aktualniho stavu pole stav 0
for(i=0;i<13;i++){Serial.print(stav0[i]);Serial.print(" ");}
for(i=0;i<13;i++){Serial.print(stav0[13+i]);Serial.print(" ");}
for(i=0;i<13;i++){Serial.print(stav0[26+i]);Serial.print(" ");}
for(i=0;i<13;i++){Serial.print(stav0[39+i]);Serial.print(" ");}
Serial.println();

// procedury casoveho modulu
#include "Wire.h"
#define DS3231_I2C_ADDRESS 0x68
byte decToBcd(byte val){return( (val/10*16) + (val%10) );} // prevod decimalnich cisel na binarni a naopak
byte bcdToDec(byte val){return( (val/16*10) + (val%16) );}

// definice pouzivanych promennych
byte *second; byte *minute; byte *hour; byte *dayOfWeek; byte *dayOfMonth; byte *month; byte *year;
int hodina;
byte minuta;
int l; // promenna ovladajici for cyklus pri distribuci pohybu
int i; // inkrement pro for cykly obecne
int ii; // pomocna promenna pro vypocty kroku
int j=0; // promenna ovladajici generovani kroku SLAVE motoru
int j0=0; // stav promenne j v predchozim kroku
int k=0; // promenna obsluhujici distribuji hodnot v promenne pozice
// kloubove souradnice pole
float
q1[52]={7.21,6.60,6.73,12.72,11.97,18.55,17.50,17.31,24.93,22.87,32.48,29.89,28.84,8.65,10.74,13.57,13.57,1
8.15,18.70,20.39,22.71,23.83,27.34,29.19,30.36,32.15,27.61,33.42,39.70,30.71,41.40,33.99,38.58,43.55,37.46,
46.09,41.17,44.91,49.00,51.27,57.89,64.35,51.73,63.73,52.88,58.40,63.95,54.59,64.86,56.81,61.54,66.40};
float
q2[52]={107.70,99.12,91.46,114.66,97.97,122.88,113.41,105.33,132.92,113.73,146.26,133.32,123.57,79.78,73.96
,68.74,86.32,75.56,93.52,87.86,82.89,101.51,90.82,110.49,104.55,99.49,56.30,53.90,52.34,63.79,60.15,71.58,6
9.50,68.16,79.75,76.47,88.44,86.47,85.22,51.75,52.65,54.42,59.61,62.06,67.65,68.42,69.95,75.99,78.19,84.75,
85.47,86.90};
float q1_zas_in =67.37; // kloubove souradnice vkladaciho mista do zasobniku
float q2_zas_in =29.99;
float q1_zas_out=64.76; // kloubove souradnice odebiraciho mista ze zasobniku
float q2_zas_out=43.5;
float NEXT_POS1=0; // promenna urcujici pozadovane kloubove souradnice
float NEXT_POS2=0;
float ACT_POS1=3; // aktualni kloubove souradnice
float ACT_POS2=11;

```

```

float REF_POS1=3;          // referencni kloubove souradnice
float REF_POS2=11;
float ERR_POS1=0;         // odchylka predikujici ztratu kroku motoru
float ERR_POS2=0;
// float STEP_ROT1=151.703694; // pocet kroku na vykonani 1° kloubove souradnice pri 1/128 deleni kroku
// float STEP_ROT2=132.740741;
float STEP_ROT1=19.6;     // pocet kroku na vykonani 1° kloubove souradnice pri 1/16 deleni
float STEP_ROT2=16.6;
float w=8;                // koeficient cekani... 1 pro mikrokrokování 1/128, 8 pro mikrokrokování 1/16
float ww=0;               // pomocna promenna k w, slouzi pro navraceni k puvodni hodnotě po zpomaleni
float dQ1 = 0;            // rozdíl kloubových souradnic
float dQ2 = 0;
int dKROK1=0;             // pozadovany pocet pulzu k dosazeni dQx
int dKROK2=0;
float dKROK1_float=0;    // pozadovany pocet pulzu k dosazeni dQx, ktery umoznuje vypocet promenne r -
ratio
float dKROK2_float=0;
float r=1;                // pomer kroku r - ratio
boolean smer1=false;     // urceni smeru otaceni ramen
boolean smer2=false;
int STEP1 = 38;           // aktualni pocet kroku od stavu Q1 = Q2 = 0
int STEP2 = 199;

boolean cislice1[13]={0,0,1,0,1,0,0,1,0,1,0,0,1}; // Podoba jednotlivych cislic
boolean cislice2[13]={1,1,1,0,1,1,1,1,0,1,1,1,1};
boolean cislice3[13]={1,1,1,0,1,1,1,1,0,1,1,1,1};
boolean cislice4[13]={1,0,1,1,1,1,1,1,0,1,0,0,1};
boolean cislice5[13]={1,1,1,1,0,1,1,1,0,1,1,1,1};
boolean cislice6[13]={1,1,1,1,0,1,1,1,1,1,1,1,1};
boolean cislice7[13]={1,1,1,0,1,0,1,1,0,1,0,0,1};
boolean cislice8[13]={1,1,1,1,1,1,1,1,1,1,1,1,1};
boolean cislice9[13]={1,1,1,1,1,1,1,1,0,1,1,1,1};
boolean cislice0[13]={1,1,1,1,1,1,0,1,1,1,1,1,1};
boolean stav1[52]={NULL}; // deklarace pole STAV
boolean stav0[52]={NULL};
boolean presun0[52]={0}; // deklarace pole presun
boolean presun1[52]={0};
int pozice0[52]={99}; // deklarace pole pozice
int pozice1[52]={99};

int photo1=0; // stav optické závory pro referenci polohy 1. ramene
int photo2=0; // stav optické závory pro referenci polohy 2. ramene

void setup()
{
  Serial.begin(9600); // zahajeni seriove komunikace s PC
  Wire.begin(); // zahajeni seriove komunikace s modulem realneho casu
  for(i=0;i<52;i++){pozice0[i]=99;} // nulovani vseh bunek pole pozice
  for(i=0;i<52;i++){pozice1[i]=99;}
  pinMode(2,OUTPUT); // nastaveni vystupnich pinu pro ovladani motoru
  pinMode(3,OUTPUT); // 2 - smer Q1, 4 - smer Q2
  pinMode(4,OUTPUT); // 3 - krok Q1, 5 - krok Q2
  pinMode(5,OUTPUT);
  pinMode(52,OUTPUT); // ovladani vakua
  pinMode(53,OUTPUT); // ovladani vysunutí pistu
  // pin A14,INPUT // cteni referencniho snimace q1
  // pin A15,INPUT // cteni referencniho snimace q2
  // pin A13,INPUT // cteni stavu optické závory

  for (i=0;i<26;i++){stav0[i]=1;} // deklarace počáteční podoby pole kulicek (zde dve cislice zleva plne
kulicek)
  ww=w; // zavedeni pomocné proměnné, pro návrat k původní hodnotě w při změně rychlosti v rámci světelné
závory
  referovani(); // provedení referování manipulátoru
}

void loop()
{
  cteni_casu();
  planovani_presunu();
  distribuce_pohybu();
  if (analogRead(14)>10 || analogRead(15)>10) {nulovani();}

  /* objeti_kontury();
  validace_zasobniku */
}

void cteni_casu(){
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year; // deklarovani
promennych sekunda, minuta ... rok
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year); // cteni hodnot z
modulu realneho casu
  Se-
  rial.print(hour,DEC);Serial.print(":");Serial.print(minute,DEC);Serial.print(":");Serial.println(second,DEC
); // vypis aktualni hodiny, minuty, sekundy
  hodina=hour;
  minuta=minute;

  // PLANOVANI 1. CISLICE
  if (hodina <10){for (i=0;i<13;i++){stav1[i]=0;}}

```

```

if (hodina>9 && hodina<20){for (i=0;i<13;i++){stav1[i]=cislice1[i];}}
if (hodina>19){for (i=0;i<13;i++){stav1[i]=cislice2[i];}}

// PLANOVANI 2. CISLICE
if (hodina==0 || (hodina-10)==0 || (hodina-20)==0 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice0[i];}}

if (hodina==1 || (hodina-10)==1 || (hodina-20)==1 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice1[i];}}

if (hodina==2 || (hodina-10)==2 || (hodina-20)==2 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice2[i];}}

if (hodina==3 || (hodina-10)==3 || (hodina-20)==3 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice3[i];}}

if (hodina==4 || (hodina-10)==4 || (hodina-20)==4 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice4[i];}}

if (hodina==5 || (hodina-10)==5 || (hodina-20)==5 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice5[i];}}

if (hodina==6 || (hodina-10)==6 || (hodina-20)==6 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice6[i];}}

if (hodina==7 || (hodina-10)==7 || (hodina-20)==7 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice7[i];}}

if (hodina==8 || (hodina-10)==8 || (hodina-20)==8 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice8[i];}}

if (hodina==9 || (hodina-10)==9 || (hodina-20)==9 )
    {for (i=0;i<13;i++){stav1[i+13]=cislice9[i];}}

// PLANOVANI 3. CISLICE

if (minuta <10)
    {for (i=0;i<13;i++){stav1[i+26]=cislice0[i];}}

if (minuta>9 && minuta<20)
    {for (i=0;i<13;i++){stav1[i+26]=cislice1[i];}}

if (minuta>19 && minuta<30)
    {for (i=0;i<13;i++){stav1[i+26]=cislice2[i];}}

if (minuta>29 && minuta<40)
    {for (i=0;i<13;i++){stav1[i+26]=cislice3[i];}}

if (minuta>39 && minuta<50)
    {for (i=0;i<13;i++){stav1[i+26]=cislice4[i];}}

if (minuta>49 && minuta<60)
    {for (i=0;i<13;i++){stav1[i+26]=cislice5[i];}}

// PLANOVANI 4. CISLICE
if (minuta==0 || (minuta-10)==0 || (minuta-20)==0 || (minuta-30)==0 || (minuta-40)==0 || (minuta-50)==0 ||
    (minuta-60)==0)
    {for (i=0;i<13;i++){stav1[i+39]=cislice0[i];}}

if (minuta==1 || (minuta-10)==1 || (minuta-20)==1 || (minuta-30)==1 || (minuta-40)==1 || (minuta-50)==1 ||
    (minuta-60)==1)
    {for (i=0;i<13;i++){stav1[i+39]=cislice1[i];}}

if (minuta==2 || (minuta-10)==2 || (minuta-20)==2 || (minuta-30)==2 || (minuta-40)==2 || (minuta-50)==2 ||
    (minuta-60)==2 )
    {for (i=0;i<13;i++){stav1[i+39]=cislice2[i];}}

if (minuta==3 || (minuta-10)==3 || (minuta-20)==3 || (minuta-30)==3 || (minuta-40)==3 || (minuta-50)==3 ||
    (minuta-60)==3)
    {for (i=0;i<13;i++){stav1[i+39]=cislice3[i];}}

if (minuta==4 || (minuta-10)==4 || (minuta-20)==4 || (minuta-30)==4 || (minuta-40)==4 || (minuta-50)==4 ||
    (minuta-60)==4)
    {for (i=0;i<13;i++){stav1[i+39]=cislice4[i];}}

if (minuta==5 || (minuta-10)==5 || (minuta-20)==5 || (minuta-30)==5 || (minuta-40)==5 || (minuta-50)==5 ||
    (minuta-60)==5)
    {for (i=0;i<13;i++){stav1[i+39]=cislice5[i];}}

if (minuta==6 || (minuta-10)==6 || (minuta-20)==6 || (minuta-30)==6 || (minuta-40)==6 || (minuta-50)==6 ||
    (minuta-60)==6)
    {for (i=0;i<13;i++){stav1[i+39]=cislice6[i];}}

if (minuta==7 || (minuta-10)==7 || (minuta-20)==7 || (minuta-30)==7 || (minuta-40)==7 || (minuta-50)==7 ||
    (minuta-60)==7)
    {for (i=0;i<13;i++){stav1[i+39]=cislice7[i];}}

```

```

if (minuta==8 || (minuta-10)==8 || (minuta-20)==8 || (minuta-30)==8 || (minuta-40)==8 || (minuta-50)==8 ||
(minuta-60)==8)
    {for (i=0;i<13;i++){stav1[i+39]=cislice8[i];}}

if (minuta==9 || (minuta-10)==9 || (minuta-20)==9 || (minuta-30)==9 || (minuta-40)==9 || (minuta-50)==9 ||
(minuta-60)==9)
    {for (i=0;i<13;i++){stav1[i+39]=cislice9[i];}}

Serial.print("Stav0      "); // vypis aktualniho stavu pole stav 0
for(i=0;i<13;i++){Serial.print(stav0[i]);Serial.print(" ");}
for(i=0;i<13;i++){Serial.print(stav0[13+i]);Serial.print(" ");}
for(i=0;i<13;i++){Serial.print(stav0[26+i]);Serial.print(" ");}
for(i=0;i<13;i++){Serial.print(stav0[39+i]);Serial.print(" ");}
Serial.println();

{
pulse2();
j=i/r;
if (j!=j0)
    {pulsel();}
delayMicroseconds(1);
digitalWrite(3,LOW); digitalWrite(5,LOW);
// opticka_zavora();
delayMicroseconds((1000000/(i*66.43+2000))*w);
j0=j;
}

for (i=200; i<(dKROK2-200); i++ )
{
pulse2();
j=i/r;
if (j!=j0)
    {pulsel();}
delayMicroseconds(1);
digitalWrite(3,LOW); digitalWrite(5,LOW);
// opticka_zavora();
delayMicroseconds(70*w);
j0=j;
}

ii=0;
for (i=0; i<200; i++ )
{
pulse2();
j=i/r;
if (j!=j0)
    {pulsel();}
delayMicroseconds(1);
digitalWrite(3,LOW); digitalWrite(5,LOW);
// opticka_zavora();
delayMicroseconds((1000000/(15286-66.43*i))*w);
j0=j;
} }

ACT_POS1=NEXT_POS1;
ACT_POS2=NEXT_POS2;

Serial.println("napozicovano ");
Serial.print("Aktualni pozice manipulatoru q1 : ");Serial.println(ACT_POS1);
Serial.print("Aktualni pozice manipulatoru q2 : ");Serial.println(ACT_POS2);

}

void pozicuj_konstant()
{
dQ1=NEXT_POS1-ACT_POS1; // ZJISTENI POTREBNE ZMENY UHLU
dQ2=NEXT_POS2-ACT_POS2;

dKROK1=abs(dQ1)*STEP_ROT1; // VYPOCET POTREBNYCH KROKU
dKROK1_float=abs(dQ1)*STEP_ROT1;
dKROK2=abs(dQ2)*STEP_ROT2;
dKROK2_float=abs(dQ2)*STEP_ROT2;

Serial.print("dKROK1= : ");Serial.println(dKROK1);
Serial.print("dKROK2= : ");Serial.println(dKROK2);

if(dQ1>0)
    {digitalWrite(2,HIGH); smer1=true;} // PRIRAZENI SMERU POHYBU
else {digitalWrite(2,LOW); smer1=false;}

if(dQ2>0)
    {digitalWrite(4,HIGH); smer2=true;}
else {digitalWrite(4,LOW); smer2=false;}

if (dKROK1>dKROK2)
{

```

```

r=dKROK1_float/dKROK2_float;
// Serial.print("ratio= : ");Serial.println(r);

j0=0;
for (i=0; i<dKROK1;i++)
{ pulse1();
j=i/r;
if (j0!=j)
{pulse2();}
delayMicroseconds(1);
digitalWrite(3,LOW); digitalWrite(5,LOW);
j0=j;
delayMicroseconds(70*w);
}

}
else { // else cast rozhodnuti if (dKROK1>dKROK2)
r=dKROK2_float/dKROK1_float;
j0=0;

for (i=0; i<dKROK2;i++)
{ pulse2();
j=i/r;
if (j0!=j)
{pulse1();}
delayMicroseconds(1);
digitalWrite(3,LOW); digitalWrite(5,LOW);
j0=j;
delayMicroseconds(70*w);
}
}

ACT_POS1=STEP1/STEP_ROT1;
ACT_POS2=STEP2/STEP_ROT2;

Serial.println("napozicovano ");
Serial.print("Aktualni pozice manipulatoru q1 : ");Serial.println(ACT_POS1);
Serial.print("Aktualni pozice manipulatoru q2 : ");Serial.println(ACT_POS2);

}

void pulse1()
{
digitalWrite(3,HIGH); // generování pulzu 1. motoru
if (smer1==true){STEP1++;} else {STEP1--;}
}

void pulse2()
{
digitalWrite(5,HIGH); // generovani pulzu 2. motoru
if (smer2==true){STEP2++;} else {STEP2--;}
}

void uchop()
{Serial.println("UCHOPUJI KULICKU");
digitalWrite(52,HIGH); // zapni vakuum
digitalWrite(53,HIGH); // vysun pist
delay(400);
digitalWrite(53,LOW); // zasun pist
delay(200);
}

void odlož()
{Serial.println("POKLADAM KULICKU");
digitalWrite(53,HIGH); // vysun pist
delay(200);
digitalWrite(52,LOW); // vypni vakuum
delay(300);
digitalWrite(53,LOW); // zasun pist
delay(200);
}

void referovani()
{
if (analogRead(A14)<10){photo1=1;}else {photo1=0;} // je v REF spinaci 1 prepazka ??
if (analogRead(A15)<10){photo2=1;}else {photo2=0;} // je v REF spinaci 2 prepazka ??

smer1=false; digitalWrite(2,LOW); // nastavi zaporny smer otaceni pro obe ramena
smer2=false; digitalWrite(4,LOW);
while (photo1==0 || photo2==0)
{
if (analogRead(A14)<10){photo1=1;} else {photo1=0;}
if (analogRead(A15)<10){photo2=1;} else {photo2=0;}

if (photo1==0){pulse1();} // generuje pulzy, dokud neni prepazka zasunuta na obou snimacich
if (photo2==0){pulse2();}
delayMicroseconds(10);
}
}

```

```

digitalWrite(3,LOW); digitalWrite(5,LOW);
delayMicroseconds(300*w);
}

delay(300);
smer1=true; digitalWrite(2,HIGH); // zmena smeru otaceni na kladny smysl
smer2=true; digitalWrite(4,HIGH);

for (i=0; i<100;i++) // provedeni odskoku o cca 5 °
{pulse1();
pulse2();
delayMicroseconds(10);
digitalWrite(3,LOW); digitalWrite(5,LOW);
delayMicroseconds(800*w); }
delay(100);
smer1=false; digitalWrite(2,LOW);
smer2=false; digitalWrite(4,LOW);

photo1=0; photo2=0;
while (photo1==0 || photo2==0){
if (analogRead(A14)<10){photo1=1;} else {photo1=0;} // faze najezdu zpomalenu rychlosti pro presnejsi
vysledek
if (analogRead(A15)<10){photo2=1;} else {photo2=0;}
if (photo1==0){pulse1();}
if (photo2==0){pulse2();}
delayMicroseconds(10);
digitalWrite(3,LOW); digitalWrite(5,LOW);
delayMicroseconds(1000*w);}
ACT_POS1=REF_POS1; // aktualni kloubove souradnice
ACT_POS2=REF_POS2;}

void nulovani()
{NEXT_POS1=REF_POS1+5; // po skonzeni presunu kulicek navrat do REF pozice + 5°
NEXT_POS2=REF_POS2+5;
pozicuj();
STEP1=5*STEP_ROT1; // zavedeni promenne step pro vypocet odchylky ztratu kroku
STEP2=5*STEP_ROT2;
smer1=false; digitalWrite(2,LOW);
smer2=false; digitalWrite(4,LOW);

if (analogRead(A14)<10){photo1=1;} else {photo1=0;} // face priblizeni
if (analogRead(A15)<10){photo2=1;} else {photo2=0;}

while (photo1==0 || photo2==0){
if (photo1==0){pulse1();}
if (photo2==0){pulse2();}
delayMicroseconds(10);
digitalWrite(3,LOW); digitalWrite(5,LOW);
delayMicroseconds(500*w);
if (analogRead(A14)<10){photo1=1;} else {photo1=0;}
if (analogRead(A15)<10){photo2=1;} else {photo2=0;}}
ERR_POS1=abs(STEP1/STEP_ROT1); // vypocet uhlove odchylky od referencni polohy
ERR_POS2=abs(STEP2/STEP_ROT2);

if (abs(ERR_POS1)>2 || ERR_POS2>2){
while (true)
{ Serial.println("ZTRATA KROKU!! STOP !! ODSTRANTE KULICKY A ZMACKNETE TLACITKO RESET."); delay(1000);}}
ACT_POS1=REF_POS1; // aktualni kloubove souradnice
ACT_POS2=REF_POS2; }

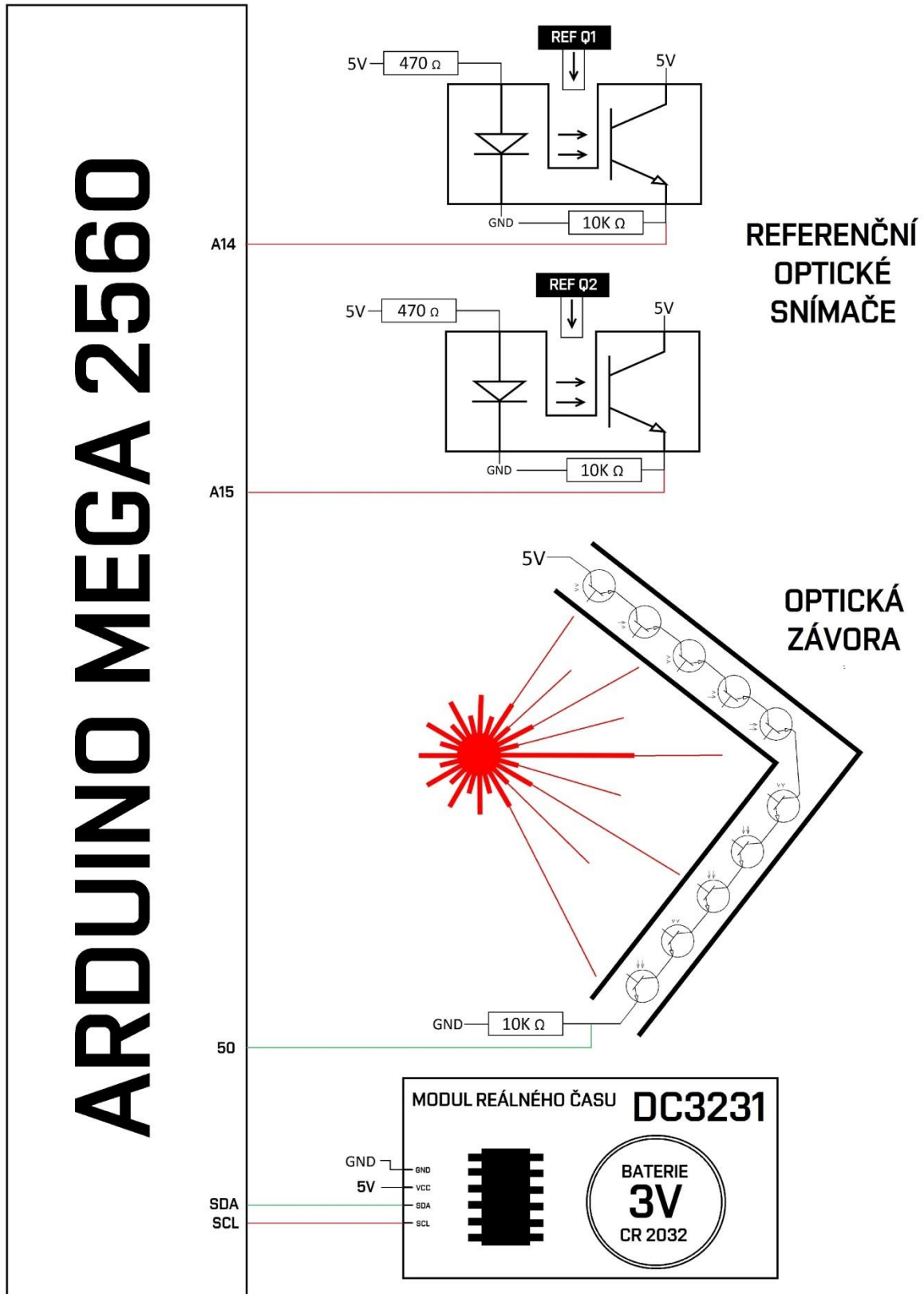
void opticka_zavora()
{
if (analogRead(A13)<10 && w<70) {w=w+0.5;} // postupne zpomalovani pri narušení pracovniho prostoru
if (analogRead(A13)>10 && w>ww) {w=w-0.5;} // postupne zrychlovani pri bezporuchovem stavu

void objeti_kontury() // postupne objeti 4. krajnich bodu pole
{ NEXT_POS1=q1[0]; NEXT_POS2=q2[0]; pozicuj(); delay(1000);
NEXT_POS1=q1[10]; NEXT_POS2=q2[10]; pozicuj(); delay(1000);
NEXT_POS1=q1[51]; NEXT_POS2=q2[51]; pozicuj(); delay(1000);
NEXT_POS1=q1[41]; NEXT_POS2=q2[41]; pozicuj(); delay(1000);}

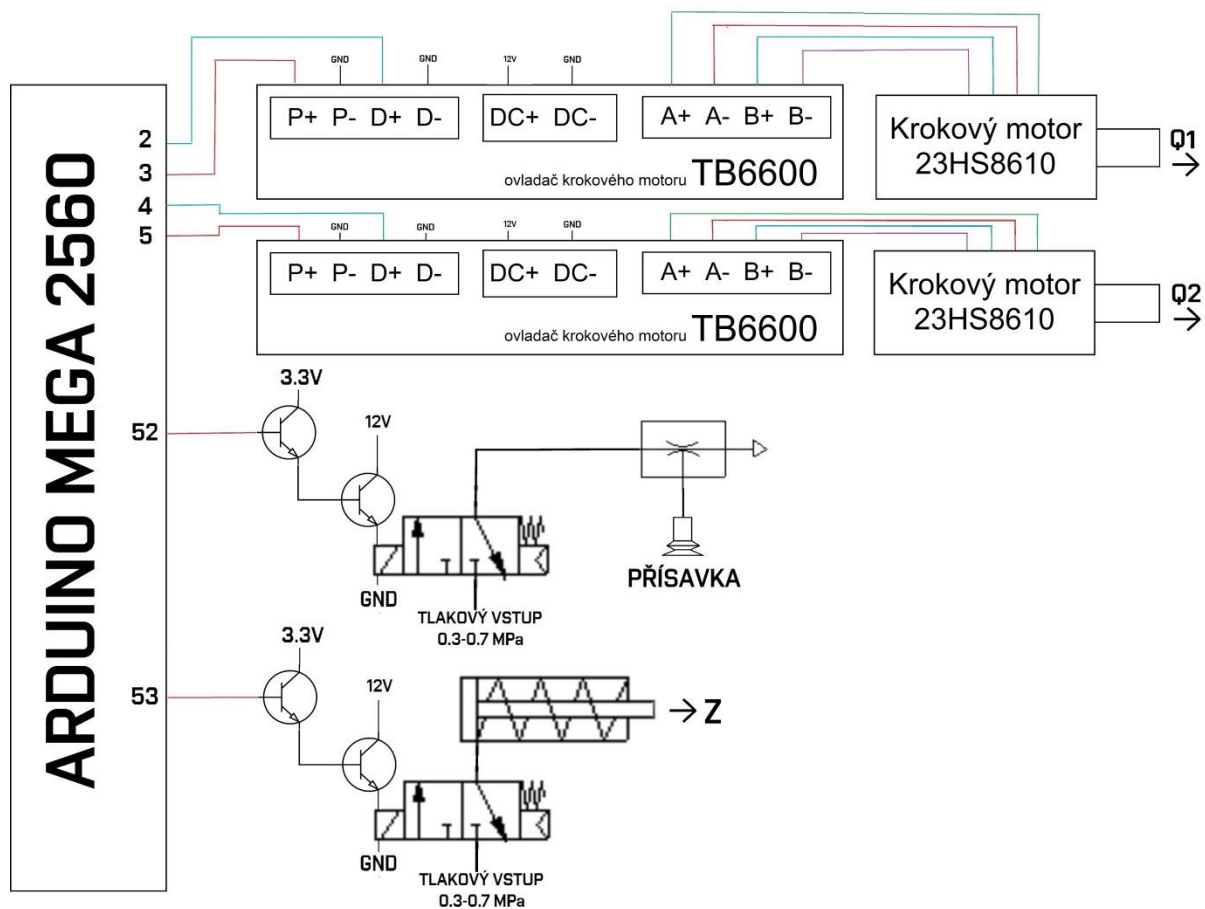
void validace_zasobniku() // postupne prejezdeni mezi vstupnim a vystupnim bodem zasobniku
{ NEXT_POS1=q1_zas_in; NEXT_POS2=q2_zas_in; pozicuj(); delay(1000);
NEXT_POS1=q1_zas_out; NEXT_POS2=q2_zas_out; pozicuj(); delay(1000);}

```

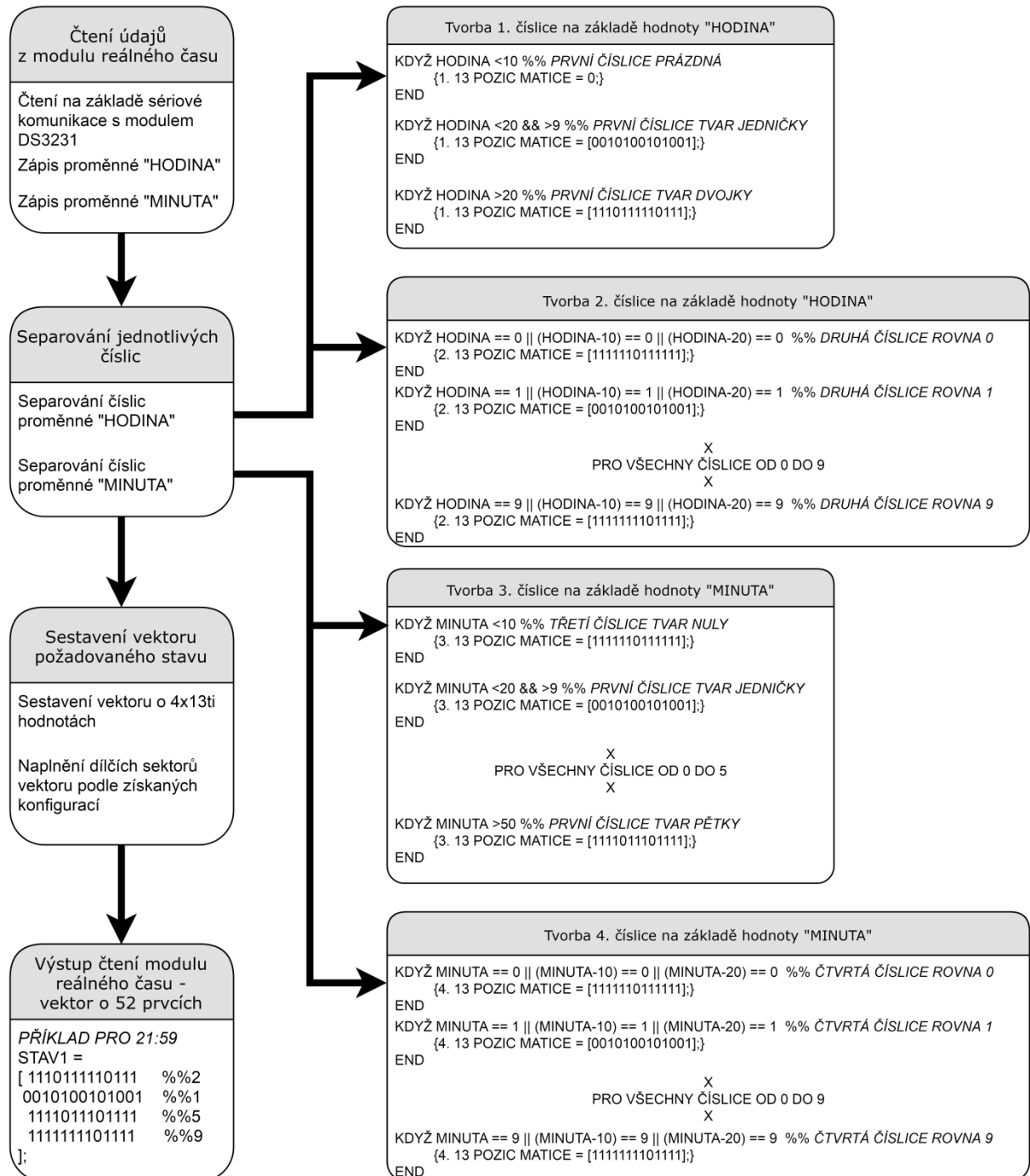
## PŘÍLOHA 2 – SCHÉMA ZAPOJENÍ VSTUPŮ



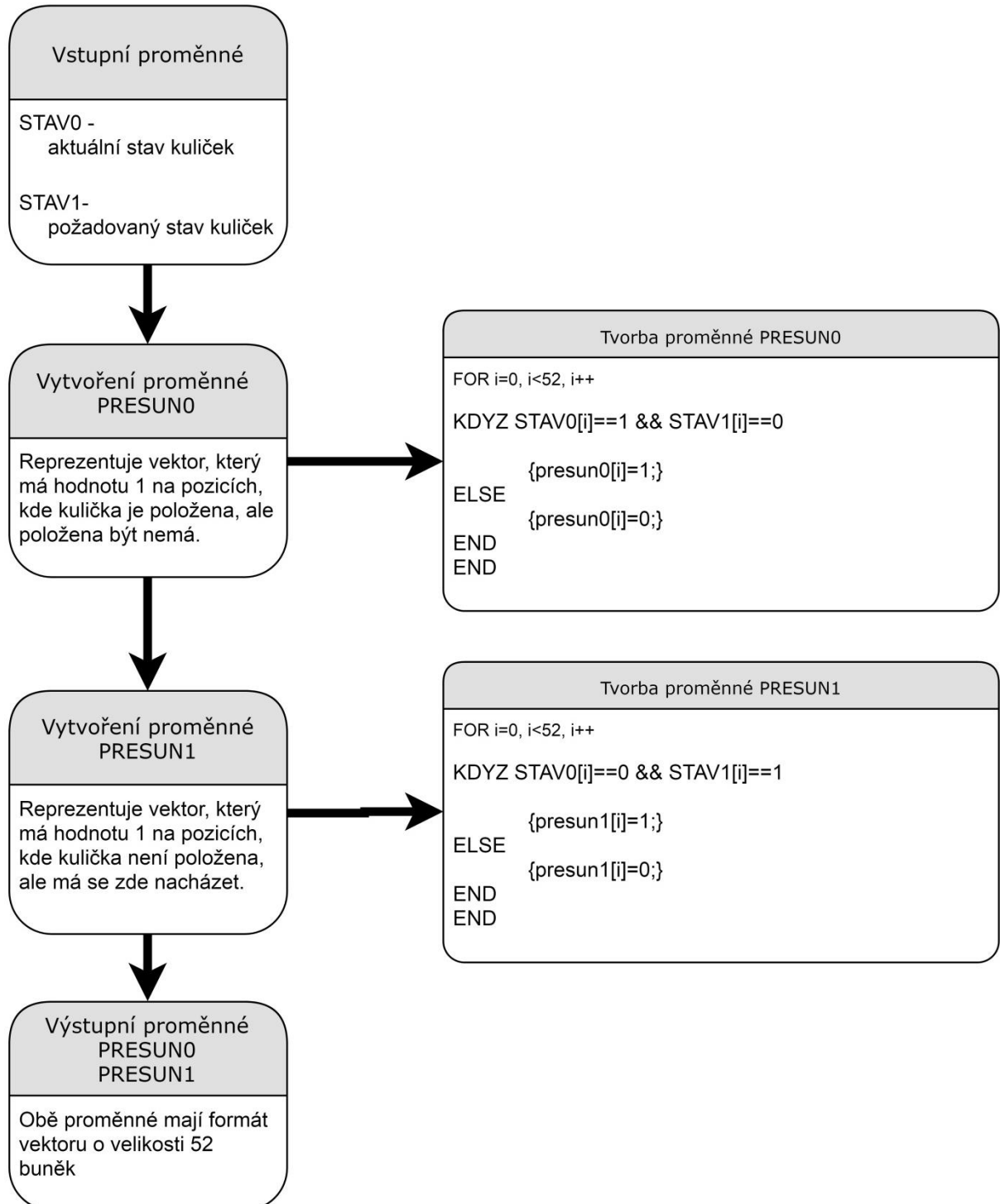
## PŘÍLOHA 3 – SCHÉMA ZAPOJENÍ VÝSTUPŮ



# PŘÍLOHA 4 - VÝVOJOVÝ DIAGRAM ČTENÍ ČASU Z EXTERNÍHO MODULU

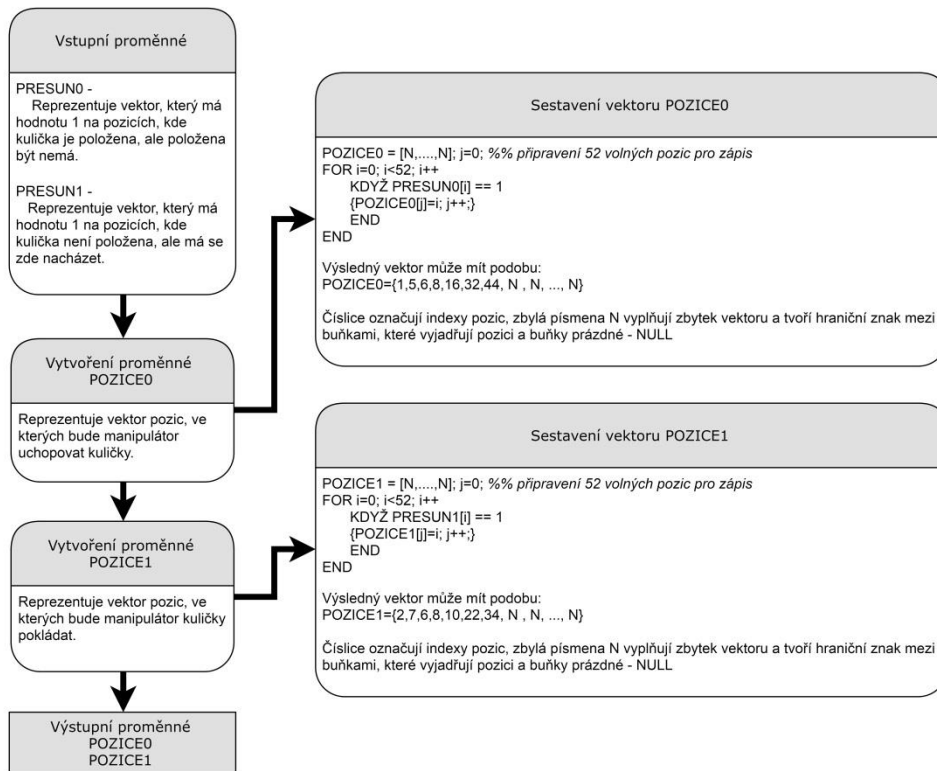


## PŘÍLOHA 5 - SESTAVENÍ VEKTORŮ POŽADOVANÝCH PŘESUNŮ KULIČEK

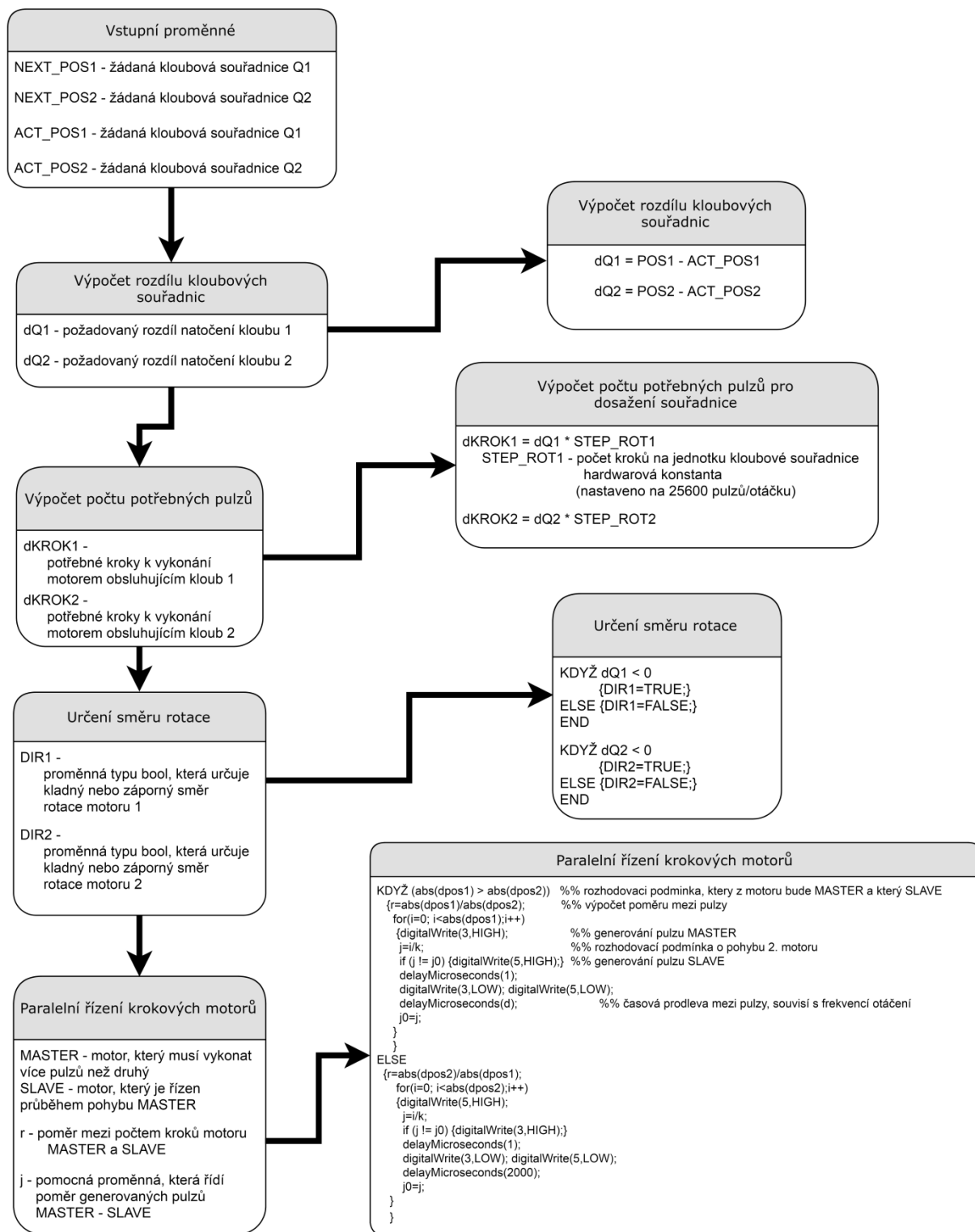


# PŘÍLOHA 6 - BLOKOVÉ SCHÉMA TVORBY VEKTORU POZICE

## PLÁNOVÁNÍ PŘESUNŮ KULIČEK MEZI POZICEMI



# PŘÍLOHA 7 - BLOKOVÉ SCHÉMA PARALELNÍHO ŘÍZENÍ KROKOVÝCH MOTORŮ



## PŘÍLOHA 8 - FOTOGRAFIE REÁLNÉHO ZAŘÍZENÍ

