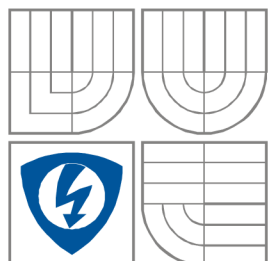


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLÓGIÍ
ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION DEPARTMENT OF
MICROELECTRONICS

AKCELEROMETREM OVLÁDANÉ ROBOTICKÉ RAMENO

ACCELEROMETER DRIVEN ROBOT

Bakalářská práce
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ RYŠAVÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL PAVLÍK, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav mikroelektroniky

Bakalářská práce

bakalářský studijní obor
Mikroelektronika a technologie

Student: Tomáš Ryšavý

ID: 155231

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Akcelerometrem ovládané robotické rameno

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte systém pro řízení robotického ramena. Řízení ramene bude využívat informace o poloze ruky operátora zjištěné akcelerometrem. Systém bude založen na jednoduchém osmibitovém mikrokontroleru.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 10.2.2015

Termín odevzdání: 4.6.2015

Vedoucí práce: Ing. Michal Pavlík, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Háze, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Cílem této práce je navrhnout nový řídicí systém robota, který simuluje manipulační činnosti ve výrobním procesu. Původní řídicí systém vykazuje určité funkční nedostatky, zvolené komponenty mají technologická resp. programová omezení. Z tohoto důvodu bude původní řídicí systém modernizován.

Zvolený úkol byl vyřešen nahrazením některých komponent modernějšími moduly. S tím souvisí i změna programovacího jazyka a implementace nového programového kódu. Důraz byl kladen i na konektivitu řídicího systému s vnějším prostředím. Nejdůležitější změnou konfigurace řídicího systému je využití gyroskopu a akcelerometru. Tyto moduly posunují technologickou část a funkcionalitu na výrazně vyšší úroveň.

Navrhované řešení poskytuje nezbytný základ pro praktické uskutečnění změn. Práce obsahuje detailní výčet a popis uvažovaných modulů. Návrh uvažuje pouze komponenty Arduino. Tím je zajištěna 100% kompatibilita a otevřenost řídicího systému pro další rozšíření. Zde se jedná zejména o možnost přímého řízení robota pomocí vnějšího zařízení jako je počítač nebo mobilní telefon.

Abstract

The goal of this work is to propose new robot controlling system. This robot simulates handling activities within manufacturing process. Original controlling system shows some defects, selected components have technological and software limitation. Due to this will be current controlling system modernized.

Chosen task has been solved replacing some components by modules using higher technology. This is related to applying new programming language and implementing new program source code. Emphasis has been put to connectivity with external environment as well. The most important change in controlling system configuration is using gyroscope and accelerometer modules. These modules provide significantly higher level for technological part and whole functionality.

Proposed solution provides requisite base to apply practical realization of changes. This work contains detail list and description of all proposed modules. Only Arduino components are taken into consideration. This way will be covered 100% compatibility and controlling system will be open for future extension. Controlling system will have option for direct robot driving by external devices like computer or cell phone.

Klíčová slova

robot, konstrukce, řídicí systém, Arduino, akcelerometr

Keywords

robot, construction, control system, Arduino, accelerometer

Bibliografická citace

RYŠAVÝ, T. *Akcelerometrem ovládané robotické rameno*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 32 s. Vedoucí bakalářské práce: Ing. Michal Pavlík, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Akcelerometrem ovládané robotické rameno jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce prof. Ing. Michalu Pavlíkovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

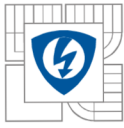
.....

(podpis autora)



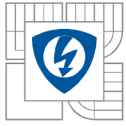
Obsah

| | |
|-----------------------------------------------------|-----|
| Seznam obrázků | vii |
| Úvod | 1 |
| 1 Mechanika | 2 |
| 1.1 Konstrukce | 2 |
| 1.2 Pohony a způsob řízení | 3 |
| 2 Řídicí systém | 5 |
| 2.1 Původní řídicí systém | 5 |
| 3 Nový řídicí systém | 6 |
| 3.1.1 Specifikace nedostatků a chyb: | 6 |
| Řešení nedostatků a chyb: | 6 |
| 3.1.2 Arduino..... | 7 |
| 3.1.3 Hlavní řídicí jednotka..... | 9 |
| 3.1.4 Zobrazovací jednotka LCD | 10 |
| 3.1.5 Snímání pohybu ovladače | 11 |
| 3.1.6 Bluetooth moduly | 13 |
| 3.1.7 Řídicí jednotka 2 | 14 |
| 4 Realizace nového řídicího systému | 15 |
| 4.1 Hlavní blok robota..... | 15 |
| 4.2 Dálkové ovládání..... | 17 |
| 5 Ovládací program | 20 |
| 5.1 Program řídicí jednotky (uvnitř robota) | 20 |
| 5.1.1 Inicializace | 20 |
| 5.1.2 Základní menu..... | 21 |
| 5.1.3 Nastavení | 21 |
| 5.1.4 Výběr módu..... | 22 |
| 5.1.5 Program ovladače | 24 |
| 5.1.6 Obslužný program bluetooth modulu..... | 26 |
| 5.1.7 Automatický režim | 27 |
| 5.2 Program řídicí jednotky (uvnitř ovladače) | 28 |
| 5.2.1 Inicializace | 28 |
| 5.2.2 Zaznamenávání akcelerometrem..... | 29 |
| 6 Závěr..... | 31 |
| Seznam použitých zdrojů | 32 |



Seznam obrázků

| | |
|--------------------------------------------------------|----|
| Obrázek 1: Pohled na konstrukci | 2 |
| Obrázek 2: Pohled na hlavní podstavec | 3 |
| Obrázek 3: Nejmenší použitý pohon (mikroservo) | 4 |
| Obrázek 4: Způsob řízení modelářského servomotoru..... | 4 |
| Obrázek 5: Původní řídicí systém | 5 |
| Obrázek 6: Arduino | 7 |
| Obrázek 7: Blokové schéma..... | 8 |
| Obrázek 8: Hlavní řídicí jednotka | 9 |
| Obrázek 9: LCD displej..... | 10 |
| Obrázek 10: Převodník pro LCD | 10 |
| Obrázek 11: Gyroskop & Akcelerometr | 12 |
| Obrázek 12: Bluetooth modul | 13 |
| Obrázek 13: Řídicí jednotka 2..... | 14 |
| Obrázek 14: USB konektor | 15 |
| Obrázek 15: Konektor pro ovladač | 16 |
| Obrázek 16: Elektrické schéma řídicího systému | 17 |
| Obrázek 17: USB propojka | 17 |
| Obrázek 18: Elektronika ovladače | 18 |
| Obrázek 19: Elektrické schéma ovladače..... | 19 |
| Obrázek 20: Dokončený ovladač | 19 |
| Obrázek 21: Struktura menu | 23 |
| Obrázek 22: Hodnoty z akcelerometru..... | 24 |

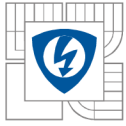


Úvod

Robot je stroj, který je navržen a zkonstruován tak, aby mohl nahradit lidskou pracovní sílu v místech, kde jsou náročnější pracovní podmínky. Konstrukce a míra umělé inteligence je přizpůsobena místu, kde bude robot instalován a provádět předem naprogramované činnosti.

Důvodem nasazení robotů namísto lidské pracovní síly bývá většinou rychlost a přesnost prováděných činností. Dalším důvodem použití robotů mohou být těžké nebo nebezpečné pracovní podmínky (nedostatek vzduchu, vysoká teplota, prašnost, atd.).

V rámci projektu byl navržen a zhotoven model robota, který simuloval manipulátory ve výrobním procesu. Vlivem nedokonalosti řídicího systému vykazoval robot provozní chybové stavy. V rámci tohoto projektu, který navazuje na závěrečnou středoškolskou práci, bude navržen nový řídicí systém, který vykazované nedostatky odstraní.

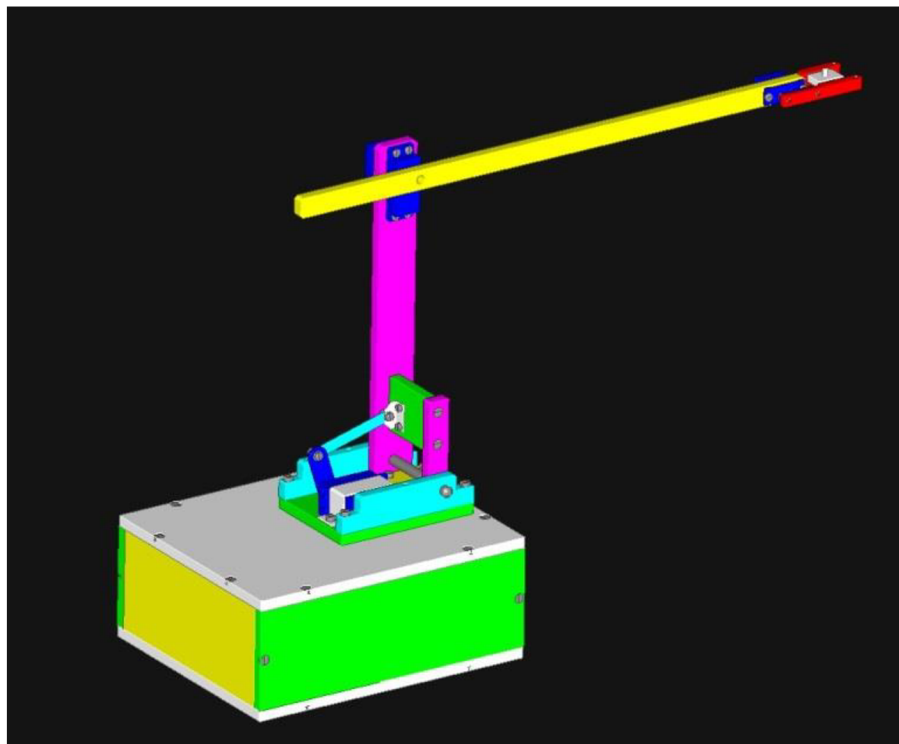


1 Mechanika

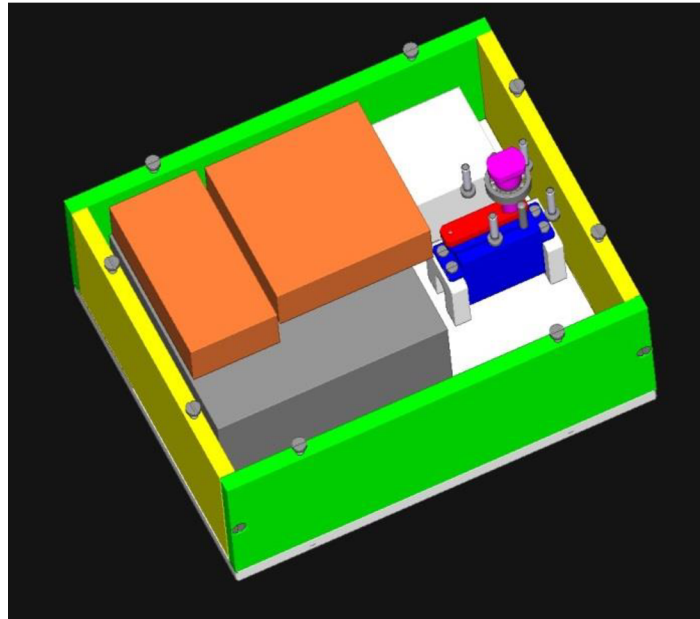
Kapitola je zaměřena na konstrukční část výroby robota. Dělí se na dvě podkapitoly. První podkapitola „KONSTRUKCE“ je zaměřena na jednoduchý popis dílů a nastínění funkce ramene. Druhá podkapitola „POHONY“ informuje o použitých servomotorech, jejich umístění a zvoleném typu. Konstrukční a mechanické řešení bylo navrženo primárně pro funkční účely, nikoli z hlediska designu.

1.1 Konstrukce

Většina dílů modelu robota byla vyrobena z materiálu Makrolon. Tento materiál je lehký, dobře zpracovatelný a vykazuje odolnost vůči nárazům. Samotná konstrukce modelu robota se skládá z několika dílů. V hlavním podstavci krychlovitého tvaru je umístěn zdroj napětí, veškerá elektronika a mechanismus na otáčení ramene se servomotorem č. 1. Malá otočná základna zajišťuje uložení hlavního ramene. Pohyb tohoto ramene ovládá přes páku servomotor č. 2. Na konci hlavního ramene je umístěn servomotor č. 3. Ten otáčí druhým ramenem, jež má malé přesazení pro umístění závaží, aby mohlo vyrovnávat hmotnost delšího konce ramene. Zde je umístěn servomotor č. 4, který ovládá poslední a nejdůležitější část robota. Touto částí jsou mechanické prsty řízené servomotorem č. 5. Síla stlačení těchto prstů je definovaná silou pružinky, která tyto dva prsty stahuje. Pružinka nemůže být příliš silná z důvodu malé rozevírací síly servomotoru č. 5.



Obrázek 1: Pohled na konstrukci



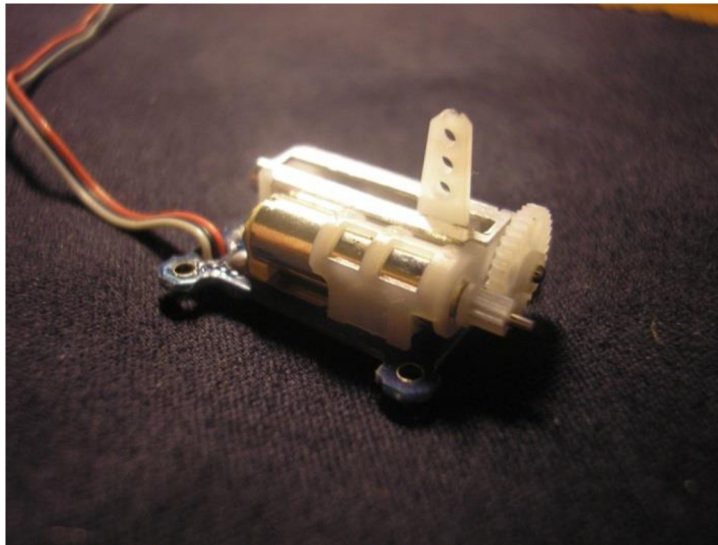
Obrázek 2: Pohled na hlavní podstavec

1.2 Pohony a způsob řízení

Pohonné jednotky na model robota byly vybrány dle typu provedení, přesnosti, síly a váhy samotného motoru. Těmto požadavkům vyhověly modelářské servomotory, které se vyrábí ve velikosti mikro, mini, standard a maxi. Velikost motoru určuje i samotnou hmotnost motoru, jež je velice podstatná vlastnost pro modely. Další důležitý parametr servomotorů je typ převodu. Ty mohou být plastové nebo kovové. Na modelu jsou použity obě zmíněné varianty, přičemž u více namáhaných částí jsou převody kovové (klouby ramen). Nevýhodou modelářských servomotorů je pohyb omezený na 180° . Tento parametr zásadním způsobem omezuje pohyblivost základny robota. S ohledem na rozměrovou dispozici robota nelze nahradit modelářské servomotory jiným typem, který toto stupňové omezení nemá.

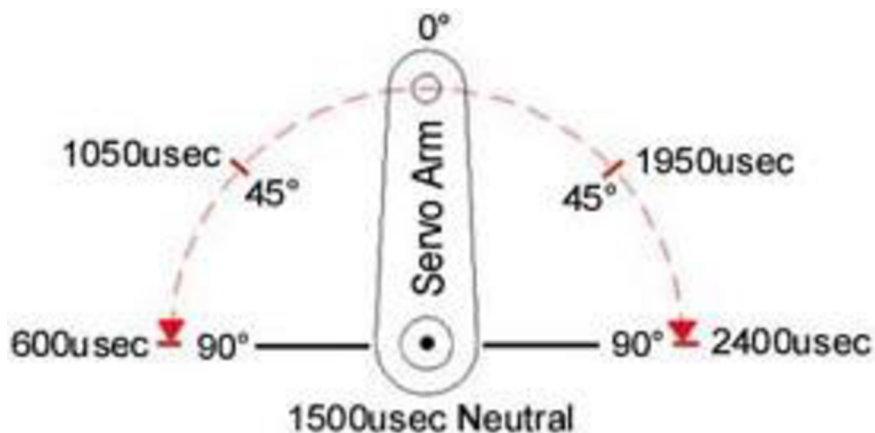
Použité servomotory:

- č. 1- síla 5 kg/cm, plastové převody, pohyb kruhový,
- č. 2- síla 10 kg/cm, kovové převody, pohyb kruhový,
- č. 3- síla 10 kg/cm, kovové převody, pohyb kruhový,
- č. 4- síla 3 kg/cm, kovové převody, pohyb kruhový,
- č. 5- síla 1 kg/cm, plastové převody, pohyb lineární.



Obrázek 3: Nejmenší použitý pohon (mikroservo)

Ovládací impuls je přiváděn obvykle na žlutý vodič, v případě použitého modelářského mikroserva byl žlutý vodič nahrazen bílým. Mikroprocesor posílá v pravidelném kmitočtu 50 Hz kladný pulz, na jehož šířce závisí natočení servomotoru. Při délce pulzu 1500 μ s je servomotor v neutrální pozici (střed). Se zkracováním pulzu na 600 μ s se servomotor postupně natočí vlevo až na úroveň -90° oproti neutrální pozici. S prodlužováním pulzu na 2400 μ s se pak postupně servomotor natočí vpravo na limitní úroveň $+90^\circ$. Viz obrázek 4.[1]



Obrázek 4: Způsob řízení modelářského servomotoru¹

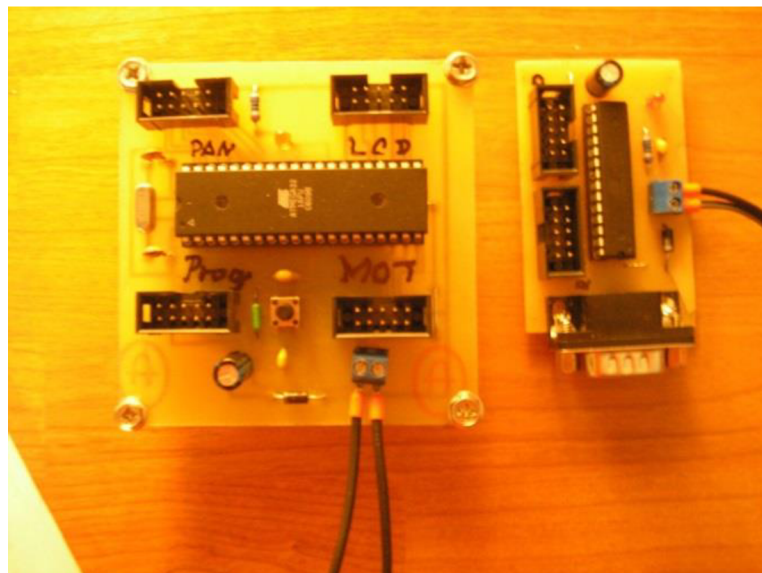
¹ [cit. 2014-12-14]. Dostupný na WWW: http://files.servo.webnode.cz/200000239-ae327af2bc/Servo_Angle_Schematic_180_3_8.jpg

2 Řídicí systém

Řídicí elektroniku v robotech je možno s určitým logickým omezením chápat jako mozek u lidí. Řídicí elektronika zajišťuje podle programové sekvence řízení všech pohybů a na základě údajů z čidel a jejich vyhodnocení udržuje kontrolu nad aktuálním stavem. Umělá inteligence má několik možných úrovní, vždy závisí na požadavku, jaké úkoly má robot plnit a v jakém prostředí se bude pohybovat. Od reálných požadavků se odvíjí náročnost na konstrukci řídicího systému.

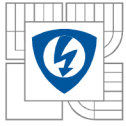
2.1 Původní řídicí systém

Elektronika tohoto robota se skládá ze dvou základních bloků s odlišným typem úkolů. Hlavní řídicí blok měl za úkol komunikovat s vnějším prostředím a řídit chod robota. Na tuto činnost byly kladeny největší nároky, a proto byl použit procesor ATmega 32 od firmy Atmel. Komunikaci s vnějším prostředím pro nastavení robota zajišťovala interaktivní konzola. Na displeji bylo zobrazováno jednoduché menu, které informovalo uživatele o nastavených hodnotách. Pomocí skupiny tlačítek klávesnice bylo možné volit jednotlivé položky a nastavovat požadované parametry. Komunikaci s druhým řídicím blokem zajišťovala sériová linka RS232. Hlavní řídicí blok odeslal hodnotu proměnné po RS232 a to vyvolalo u druhého bloku s mikroprocesorem ATmega 8 přerušení a následné přijetí hodnot, které byly odeslány na správný servomotor.



Obrázek 5: Původní řídicí systém

Tento řídicí systém měl celou řadu nedostatků a chyb, proto jej bylo potřeba vyměnit za jiný, novější a flexibilnější. Detailnější informace o novém systému jsou popsány v následující kapitole.



3 Nový řídicí systém

Nová elektronika byla navržena tak, aby minimalizovala nebo nejlépe úplně odstranila všechny vzniklé chyby a nedostatky u předešlého systému. Tyto nedostatky jsou uvedeny níže.

3.1.1 Specifikace nedostatků a chyb:

- **Občasné zasekávání řídicího systému:**

Původní programový kód byl primárně napsán za účelem zajistit požadovanou funkcionalitu, tedy svým způsobem pro ideální prostředí a při dodržení ideálních podmínek. Programový kód však neobsahoval ošetření mezních stavů. Při vzniku těchto stavů řídicí systém nebyl schopen správně zpracovat požadované hodnoty. Robot se v tom případě zaseklul nebo se dostal do nekonečné smyčky.

- **Nevyvedené programovací konektory:**

Při samotném konstrukčním návrhu nebyly zohledněny případné změny v programovacím kódu. V případě požadavku na úpravu nebo změnu v původním programu bylo nutné rozebrat boční stěnu základny. Jen tak bylo možné připojit programátor na programovací konektory jednotlivých řídicích bloků.

- **Programovací jazyk:**

Program pro řídicí systém byl napsán v jazyce BASCOM. Tento jazyk se řadí do kategorie jednodušších a je určen zejména pro ty, kteří v tomto oboru začínají a vytvářejí první aplikace.

- **Omezenost připojení dalších modulů:**

V době realizace tohoto řídicího systému nebylo zohledněno případné následující rozvíjení a zdokonalování jednotlivých řídicích bloků.

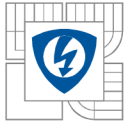
- **Komplikovaný mód manuálního řízení:**

Řídicí hodnoty pro jednotlivé modelářské servomotory bylo nutné zadávat pomocí konzole. Mód manuálního řízení nepodporoval krajní hodnoty. Uživatel mohl tedy zadat hodnotu, která se nacházela mimo definovaný rozsah a tím způsobit poškození modelářských servomotorů, které jsou na tyto stavy citlivé.

Řešení nedostatků a chyb:

- **Občasné zasekávání řídicího systému:**

Využitím programovacího jazyka Wiring se nabízejí daleko větší možnosti v oblasti programování.



- **Nevyvedené programovací konektory:**

S využitím nového řídicího systému s USB konektivitou je možné tento výstup vyvést na zadní straně základny, čímž se umožní snadný přístup k programovacímu konektoru.

- **Programovací jazyk:**

Řídicí jednotky jsou programovány v jazyce Wiring, který je podobný jako jazyk C++.

- **Omezenost připojení dalších modulů:**

Připojení dalších modulů je podporováno nově použitým multifunkčním kitem Arduino.

- **Komplikovaný mód manuálního řízení:**

Konzole s manuálním zadáváním hodnot je nahrazena ovladačem, ve kterém je umístěn gyroskop s akcelerometrem.

3.1.2 Arduino

Jedná se o jednoduchý, levný a multifunkční kit použitelný pro jednoduché i náročné operace. Arduino není jenom základní deska s procesorem, ale také celá řada modulů, jako jsou gyroskopy, akcelerometry, wifi moduly, zapisovače na SD karty, motorové moduly, LCD displeje, teplotní čidla, řadiče, ultrazvuková čidla, GPS moduly a mnoho dalších. Významnou výhodou je, že každý tento modul má dostupnou knihovnu a základní příklad programu. To ocení nejen začátečníci, ale i pokročilí.



Obrázek 6: Arduino²

Programovací jazyk Arduina se jmenuje Wiring. Podle autorů jazyka je Wiring podobný C++. V podstatě můžeme použít navykklé programovací postupy z jazyka C.

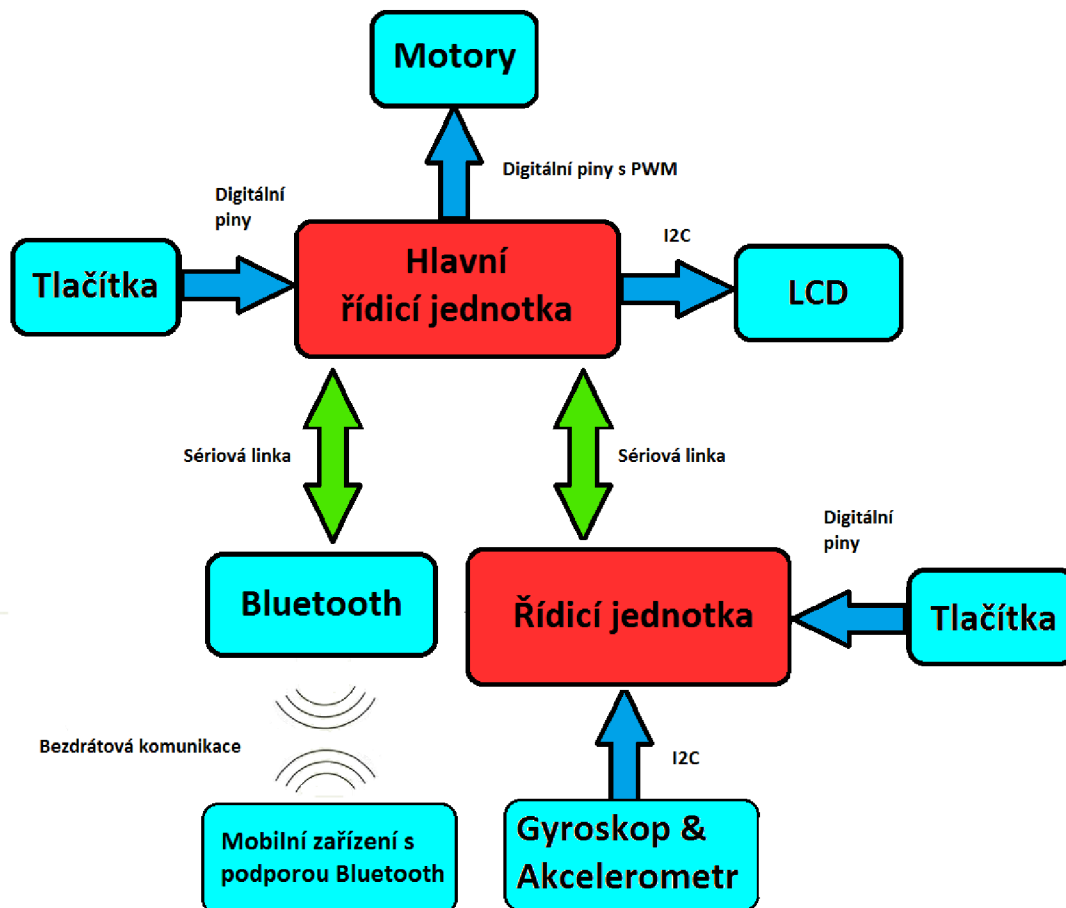
Ke kitům je jednoduše ke stažení speciální software pro programování. [2]

² [cit. 2014-12-14]. Dostupný pod licencí Creative Commons na WWW: http://arduino.cc/new_home/assets/illu-arduino-UNO.png

Výhody:

- jednoduchá propojitelnost s ostatními moduly od Arduina,
- multifunkčnost,
- jednoduché programování,
- USB konektivita,
- dobrý poměr cena/kvalita,
- nezávislá platforma (Windows, Linux, MAC),
- dobrá podpora (knihovny, návody).

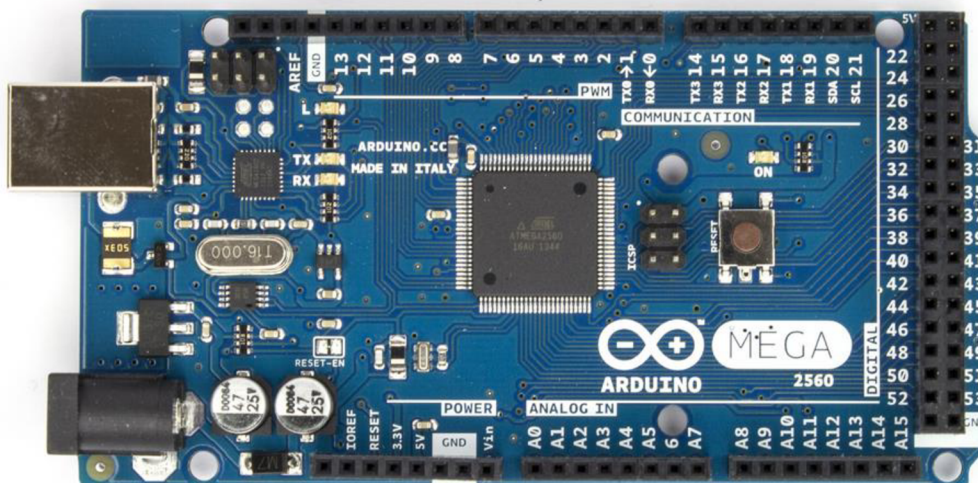
Nově navržený řídicí systém je znázorněn v následujícím blokovém schématu:



Obrázek 7: Blokové schéma

3.1.3 Hlavní řídicí jednotka

Pro hlavní řídicí jednotku je použit kit od Arduina s procesorem ATmega2560-16AU. Ten umožňuje na svých výstupech provozovat pulzní šířkovou modulaci (PWM), která je nezbytně nutná pro řízení modelářských servomotorů. Velkou výhodou tohoto kitu je integrovaný chip, který nahrazuje externí programátor. Pomocí USB kabelu je možné propojit kit s počítačem a vzájemná komunikace bude navázána. [3]



Obrázek 8: Hlavní řídicí jednotka³

Parametry použitého kitu:

- mikroprocesor: Atmel ATmega2560-16AU,
- doporučené vstupní napětí: 7-12 V,
- provozní napětí (rozhodovací): 5 V,
- DC proud na vstupních/výstupních pinech 40 mA,
- DC proud pro pin s 3.3V 50 mA,
- 16x analogových pinů,
- 54x digitálních pinů, z toho 14 může být použito pro PWM,
- paměť: 256 kB, z toho 8 kB bootloader,
- EEPROM: 4 kB,
- frekvence krystalu: 16 MHz.

³ [cit. 2014-12-14]. Dostupný pod licencí Creative Commons na WWW:
http://arduino.cc/en/uploads/Main/ArduinoMega2560_R3_Fronte.jpg

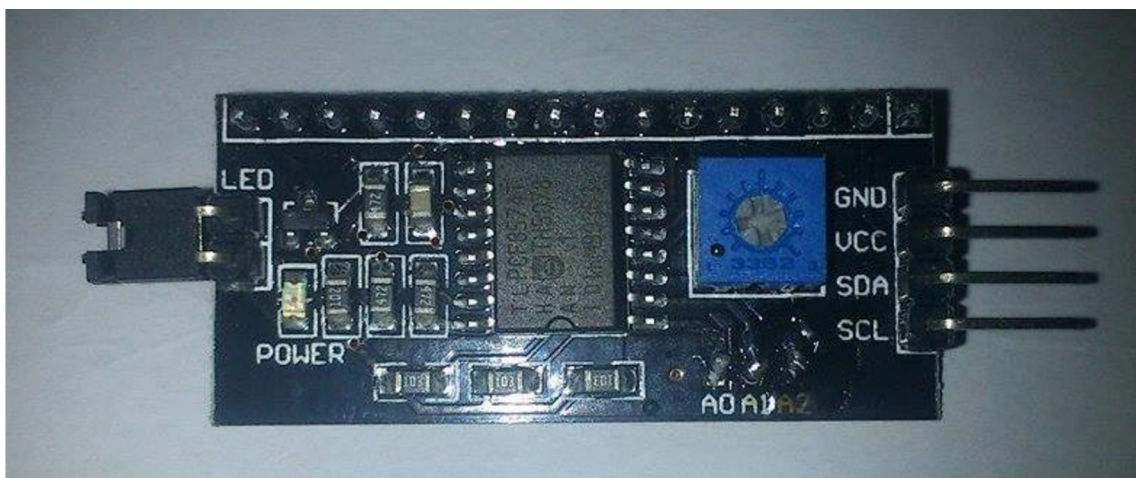
3.1.4 Zobrazovací jednotka LCD

LCD displej patří do skupiny elektronických prvků, pomocí kterých může mikroprocesor komunikovat s vnějším prostředím, v tomto případě s uživatelem.

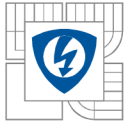


Obrázek 9: LCD displej

Jako zobrazovací jednotka je použit dvouřádkový displej s 16 modře podsvícenými segmenty a bílými znaky. Komunikace s procesorem je zajištěna přes sběrnici I2C. Samotný LCD displej I2C sběrnici nepodporuje, proto bylo potřeba zařadit mezi procesor a displej převodník od Arduina, který tuto komunikaci umožňuje. Výhodou použitého převodníku je trimr, kterým lze nastavovat kontrast LCD displeje.



Obrázek 10: Převodník pro LCD



3.1.5 Snímání pohybu ovladače

Pro snímání pohybu je nejvhodnější použít akcelerometr. Jedná se o zařízení, které měří náklony, otřesy, rotace, vibrace nebo zrychlení sil. Síly mohou být statické (zemská gravitace) nebo dynamické, které jsou způsobeny změnou pohybujícího se předmětu (snímač).

Použití akcelerometru:

- automobilový průmysl – airbagy, stabilizace podvozku ESP, chvění motoru,
- měření natočení a detekce otáčení,
- měření vibrací stroje (například vadné ložisko, které způsobí vibrace),
- měření a detekce seismické aktivity,
- měření odstředivé síly,
- notebooky volný pád => vypnutí HDD,
- náklonný mód herních ovladačů.

Dělení akcelerometrů:

Podle vnitřní struktury:

piezoelektrický jev – aktivní prvek akcelerometru je piezoelektrický materiál, který je vložen mezi dvě elektrody. Při vzniku deformace vlivem vnější síly se generuje na elektrodách napětí. Jedna strana piezoelektrického materiálu je spojena s pevným úchytem v základně snímače. Takzvaná seismická hmota je spojena s druhou stranou, která převádí zrychlení na sílu dle Newtonova zákona. Výsledná síla působí na piezoelektrický materiál. Díky piezoelektrickému efektu je výstupní náboj úměrný generované vnější síle. Když je seismická hmota konstantní, náboj výstupního signálu je úměrný zrychlení této hmoty. Pomocí tranzistoru FET se konvertuje náboj na výstupní napětí s nízkou impedancí.

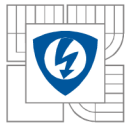
snímání kapacity – uvnitř prvku jsou mikrostruktury umístěny vedle sebe, je mezi nimi určitá kapacita. Pokud zrychlovací síla pohybuje s jednou z těchto struktur, pak se kapacita bude měnit. Tyto akcelerometry vyžadují obvod, který mění kapacitu na napětí.

horká bublina – jedná se o komoru naplněnou plynem, v jejímž středu je umístěn topný článek a na okrajích komory jsou čtyři teplotní snímače. Při klidové poloze horká bublina stoupne vzhůru ke stropu uprostřed komory akcelerometru, a tedy všechny čtyři snímače měří stejnou teplotu. Následně při působení vnějších sil dojde k náklonu akcelerometru, pak se bublina přemístí k jednomu nebo maximálně dvěma teplotním snímačům. Stejně jako u předchozího principu i tady je nutno mít obvod, který bude převádět měřené teploty na signály.[4]

Podle výstupního napětí:

analogové: výstupní napětí je obvykle závislé na zrychlení, například při 0 g je 2,5 V, pro 0,5 g je 2,6 V a pro 1 g je 2,7 V.

digitální: pro řízení výstupního napětí používají pulsní šířkovou modulaci (PWM), tedy na výstupu jsou čtvercové vlny s určitou frekvencí a vysoké napěťové úrovně budou odpovídat množství zrychlení.



Podle počtu os:

pro většinu typu projektů jsou dostačující akcelerometry dvouosé, ale pro náročnější práce lze použít akcelerometry tříosé, čímž můžeme simulovat pohyb 3D.

Maximální rozkmit:

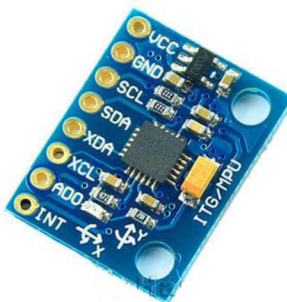
- ±1,5 g pro měření náklonu užitím zemské gravitace,
- ±2 g pro měření pohybu auta, letadla nebo robotů,
- ±5 g pro měření s neočekávanými starty nebo zastavením.

Impedance:

- vysoká - u analogových akcelerometrů je výstupní odpor od 32 kΩ,
- nízká - u piezoelektrických.

Vybraný modul s chipem MPU-6050 je integrovaný obvod s tříosým akcelerometrem, tříosým gyroskopem, teploměrem a komunikací přes sběrnici I2C. Obvod se vyznačuje vysokou přesností díky integrovanému 16 bitovému A/D převodníku.[5]

Integrovaný Digital Motion Processor™ (DMP), provádí výpočty se získanými hodnotami z gyroskopu a akcelerometru. Výsledkem této funkce je výrazné odlehčení procesů s přepočty Arduina. DMP procesor dokáže vyhodnotit získané hodnoty z akcelerometru a gyroskopu, přepočítat je na stupně a poslat Arduino. Pomocí samotného programovacího kódu lze také měnit 4 stupně citlivosti akcelerometru nebo gyroskopu.[6]



Obrázek 11: Gyroskop & Akcelerometr⁴

Parametry použitého kitu:

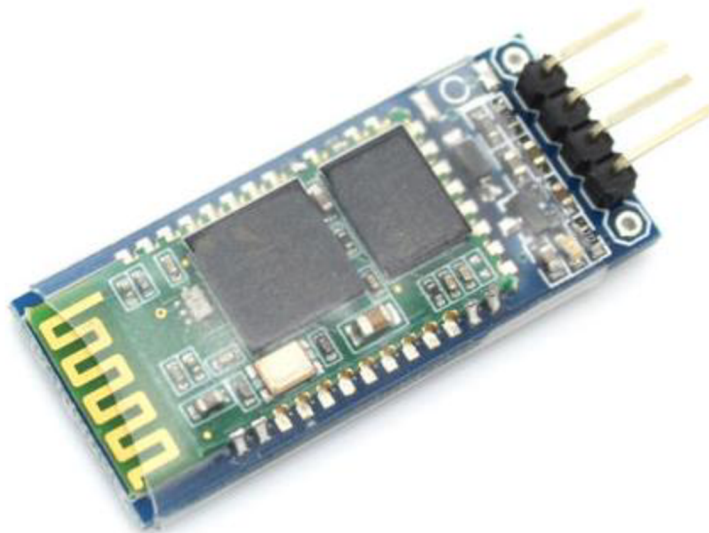
- napájecí napětí: 3-5 V,
- chip: MPU-6050,
- komunikace: I2C,
- rozsah gyroskopu +250 500 1000 2000 °/s,
- rozsah akcelerometru: ±2 ±4 ±8 ±16 g.

⁴[cit. 2014-12-14]. Dostupný pod licencí Creative Commons na WWW:
[http://i.ebayimg.com/00/s/NTAwWDUwMA==/z/99oAAOxyrjZRry9x/\\$T2eC16NHJGwFFZOo0c9YBRry9\)KBug~~60_57.JPG](http://i.ebayimg.com/00/s/NTAwWDUwMA==/z/99oAAOxyrjZRry9x/$T2eC16NHJGwFFZOo0c9YBRry9)KBug~~60_57.JPG)

3.1.6 Bluetooth moduly

Bluetooth modul Arduina bude zajišťovat bezdrátovou komunikaci mezi hlavní řídicí jednotkou a spárovaným zařízením. Bluetooth modul kromě podpory komunikace umožní připojit jakékoli zařízení s podporou bluetooth.

Použité bluetooth moduly jsou verze V2.0+EDR, tedy jejich přenosová rychlost dosahuje hodnoty až 2,1 Mbit/s ve volném prostoru. S přibývajícím překážkami vzrůstá chybovost přenesených paketů a tedy ztráta kvality spojení. Dalším velmi důležitým parametrem bluetooth modulu je jeho výkon, který má výrazný vliv na možný dosah. U použitých modulů se jedná o třídu dvě, což odpovídá orientačnímu výkonu 2,5 mW a dosah 10 m. Komunikace mezi bluetooth modulem a řídicí jednotkou je realizována po sériové lince (RX, TX data). [7]



Obrázek 12: Bluetooth modul⁵

Parametry použitých kitů:

- napájecí napětí: 3,3 V,
- pracovní frekvence: 2,4 GHz,
- specifikace: V2.0+EDR
- výstupní výkon: 2,5 mW = Class2,
- flash Memory Size: 8 Mbit,
- komunikace: sériová linka.

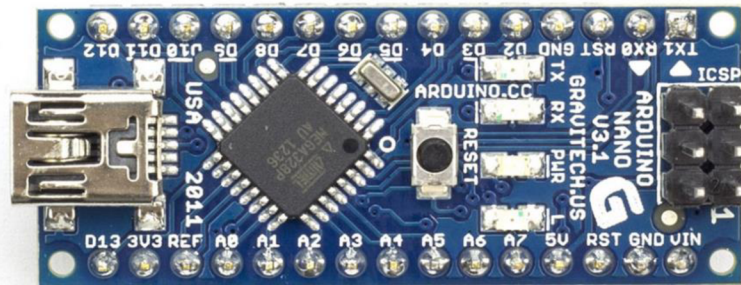
⁵[cit. 2014-12-14]. Dostupný pod licencí Creative Commons na WWW:
[http://i.ebayimg.com/00/s/MTYwMFgxNjAw/z/T9sAAOxy0NtTCtcw/\\$_57.JPG](http://i.ebayimg.com/00/s/MTYwMFgxNjAw/z/T9sAAOxy0NtTCtcw/$_57.JPG)

3.1.7 Řídicí jednotka 2

Tato řídicí jednotka je umístěna v dálkovém ovladači a jejím nejdůležitějším úkolem je předávat souřadnicové hodnoty z chipu MPU-6050 do hlavní řídicí jednotky za pomoci sériové linky. Mezi řídicí jednotkou a chipem MPU-6050, který je součástí akcelerometru, komunikace probíhá prostřednictvím sběrnice I2C.

Na zajištění této činnosti je použit opět kit od Arduina v provedení NANO, a to s mikroprocesorem ATmega328P ve zmenšené verzi. Zmenšený kit byl zvolen z důvodu rozměrových dispozic samotného ovladače.

Akcelerometr je napájen z této řídicí jednotky, v níž je zabudován stabilizátor napětí na 3,3 V. Samotná jednotka je napojena na zdroj umístěný v základně robota.[8]



Obrázek 13: Řídicí jednotka 2⁶

Parametry použitého kitu:

- mikroprocesor: Atmel ATmega328,
- doporučené vstupní napětí: 7-12 V,
- provozní napětí (rozhodovací): 5 V,
- 8x analogových pinů,
- 14x digitálních pinů, z toho 6 může být použito pro PWM,
- DC proud na vstupních/výstupních pinech 40 mA,
- paměť: 32 kB z které je použito 2 kB na bootloader,
- EEPROM: 1 kB,
- frekvence krystalu: 20 MHz,
- sběrnice I2C (SDA, SCL).

⁶ [cit. 2014-12-14]. Dostupný pod licencí Creative Commons na WWW:
http://arduino.cc/en/uploads/Main/ArduinoNanoFront_3_lg.jpg

4 Realizace nového řídicího systému

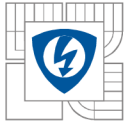
V následujících dvou podkapitolách je popsán postup realizace nového řídicího systému. Jsou zde zmíněny i problémy, které se při praktické realizaci vyskytly.

4.1 Hlavní blok robota

Nový řídicí systém nahradil původní dva bloky jedním. Tímto způsobem došlo k značné úspoře místa v základně robota. Na pravé straně základny (viz obr 14) je umístěna USB průchodka, která umožňuje vnější přístup k programovacímu konektoru na řídicí jednotce. Na protější stěně je umístěn 4 pinový konektor (viz obr 15) pro připojení dálkového ovládání. S výměnou LCD displeje souvisely konstrukční úpravy. Nový displej vyžadoval odlišné usazení, do čelní stěny bylo nutné vyfrézovat otvory na podsvětlení. Bluetooth modul byl umístěn tak, aby jeho integrovaná anténa vyzařovala signál skrze čelní stěnu. Významnou předností konstrukčního řešení je materiál Makrolon, ze kterého je robot vyroben a který výrazně signál neovlivňuje. Poslední částí úprav bylo uchycení hlavní řídicí jednotky. Konektor USB svými rozměry zabírá hodně místa, proto musela být řídicí jednotka uchycena šikmo. Po uskutečnění této úpravy konektor směřuje k hraně stěny a vznikl tak dostatečný prostor pro kabel.



Obrázek 14: USB konektor

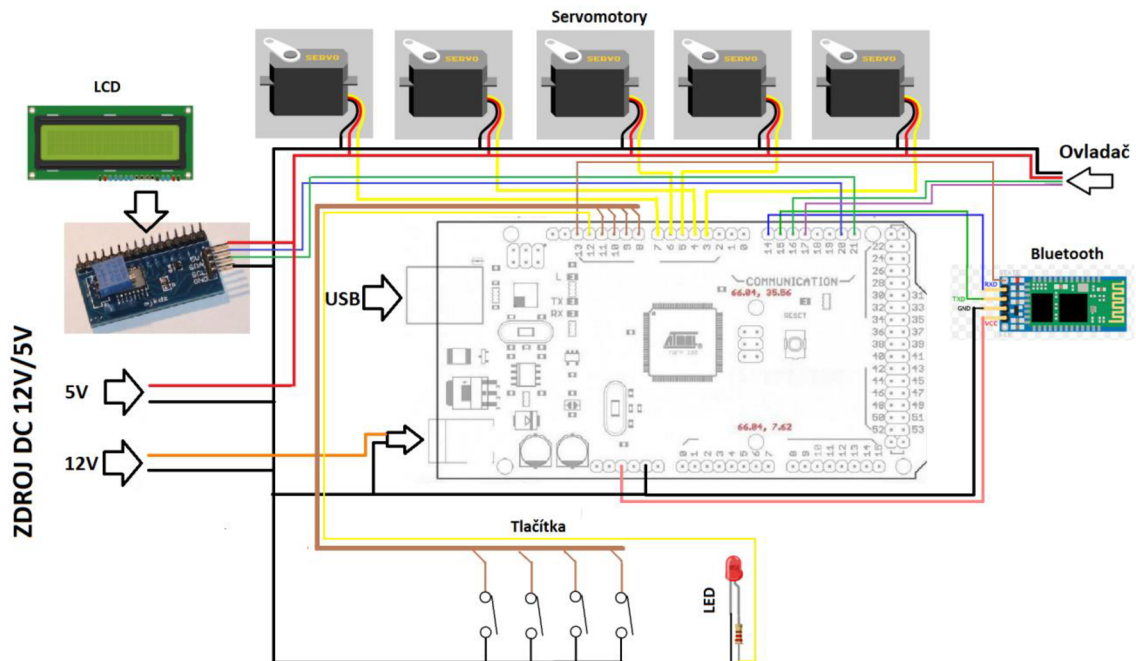


Obrázek 15: Konektor pro ovladač

Použité prvky:

- Arduino ATmega2560-16AU,
- bluetooth modul,
- LCD 16x2,
- převodník pro LCD na I2C,
- tlačítkové spínače (P-DT6SW),
- USB kabel,
- konektor,
- červená Led 3 mm,
- tavné lepidlo.

Řídicí jednotka je napájena DC 12V ze zdroje umístěného v základně robota. Propojení USB konektoru s průchodkou je realizováno za pomoci USB kabelu (viz obr 17). Bluetooth modul je napájen 5V a komunikuje s řídicí jednotkou po sériové lince. Poslední pin u bluetooth modulu slouží k vypínání nebo zapínání modulu. Všechny servomotory jsou napájeny 5V a třetí vodič je napojen na piny ŘJ, které umožňují provozovat PWM modulaci. LED displej je připojen k převodníku, který komunikuje s ŘJ po sběrnici I2C a je napájen 5V. Komunikace s ovladačem probíhá po sériové lince.



Obrázek 16: Elektrické schéma řídicího systému

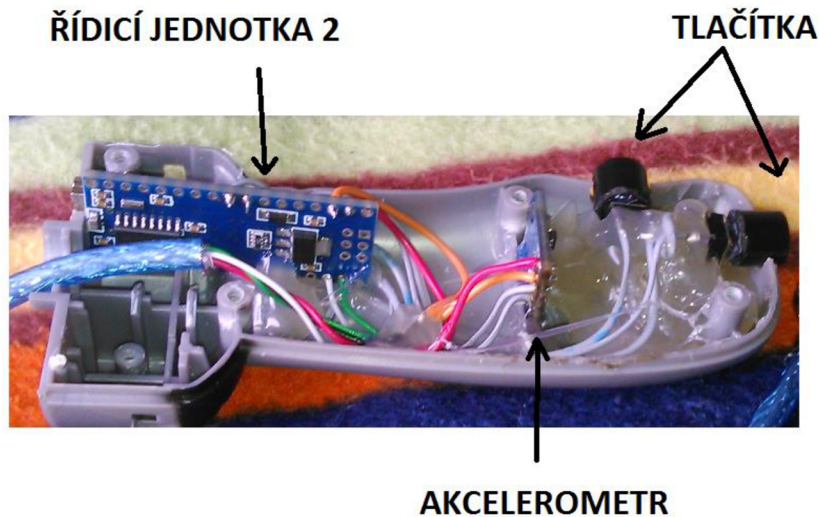


Obrázek 17: USB propojka

4.2 Dálkové ovládání

Jako pouzdro ovladače byla použita roztáček ručka z hračky FlyWheels, která svoji velikostí a tvarem nejlépe vyhovovala pro zabudování komponentů. Do horní části ovladače byly vyvrtány dva otvory pro vsazení dvou tlačítek. Na každý mikrospínač byly napájeny dva vodiče pro následné propojení s jednotkou. Tento celek je zalit tavným lepidlem ve vyvrtaných otvorech. V prostřední části je umístěn modul akcelerometru. Zde bylo nutné dodržet požadavky na přesné umístění v rovině x a y při dosažení nulového, popř. zanedbatelného odklonu od 0° . Ve spodní části je řídicí jednotka 2. Umístění řídicí jednotky umožňuje vnější přístup k programovacímu USB konektoru. Poslední částí

je propojení ovladače se základnou robota, a to za pomoci čtyřvodičového stíněného přívodního kabelu, který vstupuje do ovladače spodní částí a je napojen na řídicí jednotku.

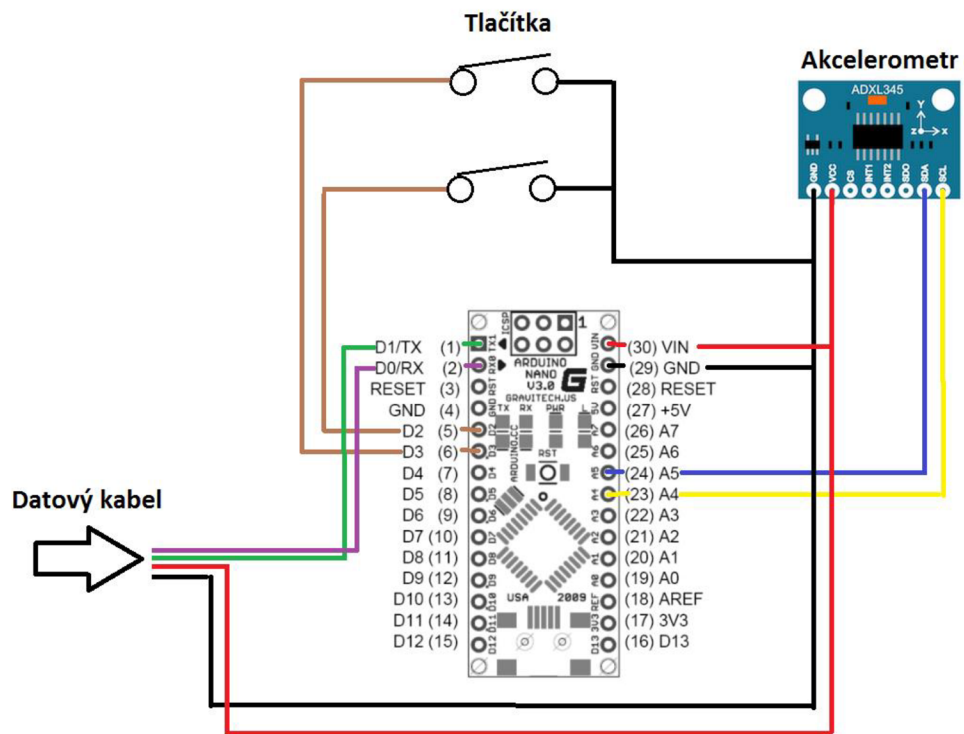


Obrázek 18: Elektronika ovladače

Použité prvky:

- Arduino NANO,
- akcelerometr,
- tlačítkové spínače (P-DT6SW),
- USB kabel,
- konektor,
- tavné lepidlo.

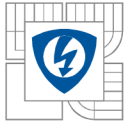
Elektronika v ovladači byla zapojena dle následujícího schématu. Tlačítka uzemňují piny řídicí jednotky na zem a přitom je využíváno interního “pull up“ odporu. Modul akcelerometru je napájen 5V přímo ze zdroje, který je umístěn v podstavci robota a datová komunikace mezi jednotkou a akcelerometrem probíhá po sběrnici I2C. Komunikace mezi jednotkou ovladače a robotem byla popsána v kapitole 4.1.



Obrázek 19: Elektrické schéma ovladače



Obrázek 20: Dokončený ovladač



5 Ovládací program

Základní programovou strukturu tvoří dva bloky. První programový blok je součástí řídicí jednotky robota, druhý programový blok je umístěn v řídicí jednotce dálkového ovladače. Každý z těchto programových bloků má jednak odlišné nároky na programovou strukturu, stejně tak existují odlišné požadavky na požadovanou funkci. Programová podpora řízení robota a dálkového ovladače musí zajišťovat optimální chod robota. Na jedné straně se tedy jedná o podporu pohybových sekvencí ramene a uchopovacích kleští, na druhé straně je nutné uplatnit určité optimalizační kroky, které řeší případné kolizní stavy. Zde se jedná zejména o uplatnění časového zpoždění mezi některými příkazy, odladění pohybových sekvencí s plynulým náběhem jednotlivých pohybů, atd. Nezbytnou součástí programové podpory je otevřenost řešení pro oba programové bloky. Jedná se o možnost uplatnit změny v algoritmu programu podle aktuálních požadavků na funkci robota, aniž by se programové bloky vytvářely znova. Stejně tak programová sekce pro dálkový ovladač může být rozšiřována.

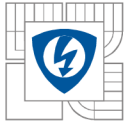
5.1 Program řídicí jednotky (uvnitř robota)

Program řídicí jednotky robota obsahuje sekce inicializace, nastavení, výběr módu a programy pro jednotlivé módy. V dalších podkapitolách jsou uvedeny detailní informace o jednotlivých sekcích. Aktuální konfiguraci bude možné v rámci dalších projektů rozšiřovat.

5.1.1 Inicializace

Inicializaci zajišťuje smyčka *main*, která proběhne pouze jednou, a to při každém zapnutí nebo restartování řídicí jednotky. První dva řádky obsahují nastavení rychlosti pro sériovou linku 2 a 3. Následně jsou pro jednotlivé názvy servomotorů přiřazeny odpovídající piny, kam jsou servomotory připojeny. Funkcí *pinMode* je nastavena úroveň jednotlivých portů. Úroveň portů mohou nabývat hodnot *High* - logická 1 nebo *Low* - logická 0. Poslední sekvence instrukcí obsahuje zapnutí LCD displeje, nastavení kurzoru, výpis znaků a po časové prodlevě smazání údajů na LCD display. Následuje krok, do hlavní smyčky *loop*, který je popsán v další části níže.

```
void setup()  
{ Serial3.begin(9600);  
  Serial2.begin(9600);  
  servo1.attach(7);  
  servo2.attach(4);  
  servo3.attach(6);  
  servo4.attach(5);  
  servo5.attach(3);  
  pinMode(LED, OUTPUT);  
  pinMode(BT, OUTPUT);  
  pinMode(TLAC1, INPUT);  
  pinMode(TLAC2, INPUT);  
  pinMode(TLAC3, INPUT);  
  pinMode(TLAC4, INPUT);  
  digitalWrite(BT, LOW);  
  bts=0;
```



```
lcd.init();  
lcd.backlight();  
lcd.setCursor(2,0);  
lcd.print("ROBOT SKORY");  
delay(5000);  
lcd.clear(); }
```

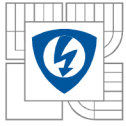
5.1.2 Základní menu

Tato část programu umožňuje uživateli buď vstoupit do dílčího menu *nastavení*, nebo program *spustit*. V tomto kódu jsou 3 cykly s podmíněnou funkcí *if*, každý s odlišnou vstupní podmínkou. První slouží k výpisu na LCD displej. Tento cyklus proběhne pouze jednou. Na konci cyklu se v proměnné *l* změní hodnota z *0* na *1* a poté cyklus nemůže znovu proběhnout, protože nesplňuje vstupní podmínku $l=0$. Další dva cykly jsou už velmi podobné, jen reagují na stisk odlišného tlačítka. Jednotlivý cyklus se skládá ze čtyř funkcí. První funkce *delay* slouží k pozastavení programu na *200 milisekund*. Tato prodleva je nezbytná a byla určena v průběhu testování. Tlačítko potřebuje určitou dobu, aby se stihlo vrátit do výchozí pozice. Pokud by doba prodlení nebyla nastavena nebo by její hodnota byla nízká, tlačítko by stihlo reagovat i v další podprogramu, což by způsobilo kolizi. Následně se v cyklu smaže LCD displej, do proměnné *l* pro výpis na LCD displej se zapíše opět *0*, aby se v dalším podprogramu mohl spustit cyklus na výpis LCD displeje. Poslední funkcí v cyklu je už samotný skok do dalšího podprogramu.

```
void loop() //Menu Spustit  
{ if (l==0)  
  {lcd.setCursor(4,0);  
   lcd.print("SPUSTIT");  
   lcd.setCursor(0,1);  
   lcd.print("  -> ENT  ");  
   l=l+1; }  
  if (digitalRead(TLAC2) == 0)  
  {delay(200);  
   lcd.clear();  
   l=0;  
   nastaveni(); }  
  else if (digitalRead(TLAC3) == 0)  
  {delay(200);  
   lcd.clear();  
   l=0;  
   demo(); }  
  loop(); }
```

5.1.3 Nastavení

Sekce *nastavení* je přístupná z hlavního menu a je zde k dispozici základní ovládání bluetooth modulu, tedy zapnutí nebo vypnutí. Výchozí nastavení při zapnutí robota uchovává hodnotu "Vypnuto" pro modul bluetooth. Při párování je viditelný pod názvem HC05 a přístupovým kódem 1234.



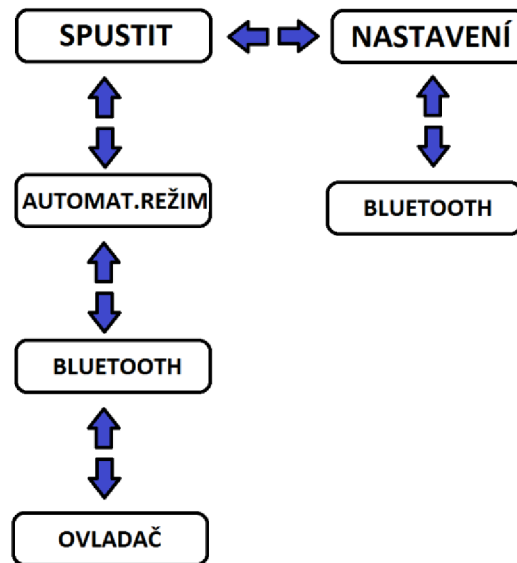
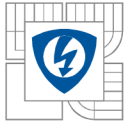
Samotný program začíná porovnávacím cyklem, který testuje, zda je modul v zapnutém stavu nebo je stále ještě vypnutý. Tedy pokud bude vypnutý tak v proměnné *bts* bude hodnota 0 a současně v proměnné *l* pro LCD bude rovněž hodnota 0, údaj o stavu “vypnutý“ se vypíše na LCD. Po zobrazení hodnoty se do proměnné pro LCD zapíše 1. Tím se v podprogramu vyřadí první dva zobrazovací cykly a fungují jen ostatní na tlačítka. Všechny cykly končí stejně a to zapsáním nuly do proměnné LCD, aby se povolil opět zápis na začátku dalších podprogramů. Při stisku tlačítka následuje opět prodlení, jak bylo již vysvětleno v předchozím podprogramu. Následuje změna úrovně signálu na povolovacím pinu bluetooth modulu. V posledním kroku se zapíše do proměnné *bts* 1 nebo 0, podle předchozího stavu.

```
void nastaveni() //Menu Nastaveni
{if (l==0)
  {lcd.setCursor(3,0);
  lcd.print("NASTAVENI");
  lcd.setCursor(0,1);
  lcd.print("<- -> ENT  ");
  l=l+1; }
if (digitalRead(TLAC1) == 0)
  {delay(200);
  lcd.clear();
  l=0;
  loop(); }
else if (digitalRead(TLAC3) == 0)
  {delay(200);
  lcd.clear();
  l=0;
  blue(); }
nastaveni(); }
```

5.1.4 Výběr módu

Základní funkcí této části programu je spouštět jiné podprogramy. Při spuštění robota se program nastaví do výchozího stavu - na pozici *SPUSTIT*. Z výchozí pozice lze volit další podpůrné funkce výběrem z příslušného dílčího menu. V nabídce menu a tedy z dostupných funkcí lze volit *AUTOMATICKÝ REŽIM* pro činnost robota řízenou programem, *BLUETOOTH*, kdy je robot ovládán externím zařízením připojeným přes bluetooth rozhraní nebo *OVLADAČ*, kdy je robot řízen manuálně připojeným ovladačem. Druhá varianta výběru je režim *NASTAVENÍ*. V rámci aktuální konfigurace obsahuje mód nastavení jediné dílčí menu *BLUETOOTH*. Blokové schéma je zřejmé z obr 21.

Aktuální konfigurace je výsledkem zásadní přestavby původního řešení. Je zde zřetelný nárůst dostupných funkcí, rozhraní a programových variant. Na druhou stranu je zřejmé, že tato konfigurace není konečným řešením a umožňuje další rozšíření. Jedná se zejména o nové možnosti nastavení, stejně tak bude možné rozšířit vlastní funkcionalitu robota o nové nebo kombinované režimy ovládání. V neposlední řadě bude předmětem rozšíření využití softwarové platformy, popř. její rozšíření



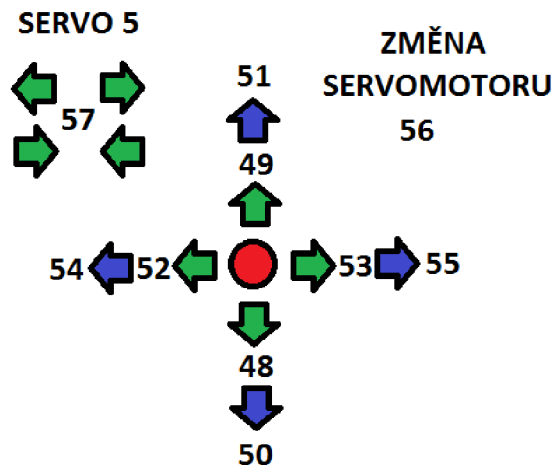
Obrázek 21: Struktura menu

Prioritní funkcí tohoto programu je testování resp. zjišťování stavu tlačítka, tedy stisknuto – nestisknuto a zápis údajů na LCD displej. V prvním cyklu se na základě aktuálních hodnot proměnných vypisují názvy dílčích menu na LCD displej. Na konci cyklu se aktualizuje hodnota proměnné l pro LCD displej. Tímto způsobem je zajištěno, aby se cyklus znovu neopakoval. Následují dva cykly s tlačítky, tedy detekce stavu tlačítek. Při stisknutí tlačítka se vždy uplatní časová prodleva, jejíž důvod a hodnota byly popsány v předchozím textu. Následuje vynulování hodnoty proměnné l a odskok do dalšího podprogramu, který byl plánován na základě struktury na obr 21.

```
void blue() //Menu Nastaveni -> Bluetooth
{if(l==0)
  {lcd.setCursor(3,0);
  lcd.print("BLUETOOTH");
  lcd.setCursor(0,1);
  lcd.print("< -> ENT ESC");
  l=l+1; }
if (digitalRead(TLAC4) == 0)
  {delay(200);
  lcd.clear();
  l=0;
  nastaveni(); }
else if (digitalRead(TLAC3) == 0)
  {delay(200);
  lcd.clear();
  l=0;
  bluen(); }
blue(); }
```

5.1.5 Program ovladače

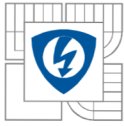
Program pro dálkové ovládání nabízí oproti obslužnému programu bluetooth modulu širší stupeň řízení. Přibylo zobrazování servomotorů na LCD displeji, možnost rychlejšího a pomalejšího pohybu ramene. Hodnoty, které se přijímají ze sériové linky, jsou naznačeny na obr 22. Zde je vidět, že se využívá dvou stupňů náklonů v ose x a y.



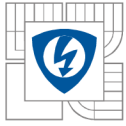
Obrázek 22: Hodnoty z akcelerometru

Servomotor 5 má pouze dvě polohy, a to buď rozevřeno, nebo sevřeno a využívá se hodnota 57. Hodnoty 52-55 jsou využívány pouze pro servomotor 1. Jinak je to u hodnot 48-51, které jsou společné pro servomotor 2, 3, 4 a přepíná se mezi nimi hodnotou 56. Podrobněji je to vysvětleno v následujícím odstavci, kde jsou i části použitého kódu.

Na začátku programu je cyklus pro tlačítko, který zajistí opuštění podprogramu. Navíc je tu příkaz `Serial2.print`, který po sériové lince zašle znak "I". Pro program v dálkovém ovladači to znamená přerušení posílání hodnot z akcelerometru a opustit svůj podprogram. Bude vysvětleno u konkrétního programu u dálkového ovladače. Následuje další cyklus, který začíná přerušením ze sériové linky 2 a přesunutí hodnoty do proměnné s . V dalším kroku se porovná hodnota v proměnné s , s počátečními podmínkami. U servomotoru 1 je to celkem jednoduché, dle obrázku 22 je nastaven velký nebo malý krok a směr. U servomotorů 2 je to podobné s tím, že přibylo ještě jedna vstupní podmínka v podobě proměnné t . Ta, může nabývat hodnot 1, 2 nebo 3. Hodnota určuje, který servomotor budou ovlivňovat hodnoty 48-51. Hodnotu v proměnné t měníme hodnotou 56, která odpovídá stisknutí tlačítka na ovladači. Podrobněji bude vysvětleno v programu k ovladači. Tuto volbu zobrazuje LCD displej s informací, který servomotor máme zvolen. Cyklus pro servomotor 3 a 4 je téměř stejný jako pro servomotor 2 jen s odlišným zápisem na servomotor. Poslední částí je ovládání servomotoru 5. Jelikož tento servomotor je opět ovládán pouze jedním tlačítkem, stačí jen jedna hodnota, v tomto případě 57. Ve vstupní podmínce je nová proměnná p . Ta určuje, zda s novou příchozí hodnotou 57 budeme servomotor stahovat nebo roztahovat. Je to provedeno tak, že na konci tohoto cyklu je zmenšení nebo zvětšení hodnoty v proměnné p .



```
void ovlpr() //Čtení hodnot a převod na serva z Serial2
{if (digitalRead(TLAC4) == 0)
{ lcd.clear();
  Serial2.print("1");
  delay(200);
  ovladac(); }
if ( Serial2.available())
{ s=(Serial2.read());
  //SERVO1
  if (s == 53)
  { s1=s1+1;
    servo1.write(s1); }
  else if (s == 52)
  { s1=s1-1;
    servo1.write(s1); }
  if (s == 55)
  { s1=s1+3;
    servo1.write(s1); }
  else if (s == 54)
  { s1=s1-3;
    servo1.write(s1); }
  //SERVO2
  if (s == 49 && t==1)
  { s2=s2+1;
    servo2.write(s2); }
  else if (s == 48 && t==1)
  { s2=s2-1;
    servo2.write(s2); }
  else if (s == 50 && t==1)
  { s2=s2-3;
    servo2.write(s2); }
  else if (s == 51 && t==1)
  { s2=s2+3;
    servo2.write(s2); }
  else if (s == 56 && t<2)
  { t=t+1;
    lcd.clear();
    lcd.print("OVLADANI: SERVO2");
    lcd.setCursor (0,1);
    lcd.print("      ESC");}
  else if (s == 56 && t<3)
  { t=t+1;
    lcd.clear();
    lcd.print("OVLADANI: SERVO3");
    lcd.setCursor (0,1);
    lcd.print("      ESC"); }
  else if (s == 56 && t==3)
  { t=1;
```



```
lcd.clear();  
lcd.print("OVLADANI: SERVO1");  
lcd.setCursor(0,1);  
lcd.print("      ESC"); }  
  //SERVO5  
  if (s == 57 && p==0)  
  { p=p+1;  
    servo5.write(150); }  
  else if (s == 57 && p==1)  
  { p=p-1;  
    servo5.write(40); }  
}  
ovlpr();}
```

5.1.6 Obslužný program bluetooth modulu

Tato část podprogramu je z hlediska možností funkcionality značně omezena. Jsou zde standardní přednastavené hodnoty, u kterých se očekává, že je program obdrží od bluetooth zařízení. Jinými slovy je třeba vyhledávat mobilní aplikace, které umožní nastavení posílaných hodnot, které zařízení posílá například pro pohyb doleva “zaslat 0“, “doprava 1“,... V první části podprogramu se ověří, zda je bluetooth modul zapnutý. Pokud ano, cyklus pokračuje ve čtení hodnot z bluetooth, pokud ne, podprogram se přeruší a vrátí se do menu.

Počátek programu byl vysvětlen u předchozích. V podprogramu je zařazen cyklus pro tlačítko, které ukončí program. Hlavní část programu obsahuje příkaz *Serial3.available*, který reaguje na přerušení ze sériové linky. V dalším kroku proběhne přečtení hodnoty a její uložení do proměnné *s*. Tato proměnná se porovnává s několika čísly, která odpovídají určitému pohybu. Pro příklad: po sériové lince pošleme číslo “7“, to vyvolá přerušení a zároveň se číslo 7 zapíše do proměnné *s*. Následuje porovnávací cyklus až do doby, dokud se číslo neshodne se vstupní podmínkou. Spustí se daný cyklus, který na počátku odečte z proměnné *s4* hodnotu “1“ a novou hodnotu zapíše do proměnné *s4*. Následně se tato hodnota zapíše na servo4. Uvnitř knihovny se tato hodnota převede na šířku pulzu a pokračuje smyčka běhu programu.

```
void blueprog() //Čtení hodnot a převod na serva z Bluetooth  
{if ( bts==0 && l==0 )  
  { lcd.setCursor(4,0);  
    lcd.print("ZAPNETE");  
    lcd.setCursor(3,1);  
    lcd.print("BLUETOOTH");  
    delay (2000);  
    lcd.clear();  
    l=0;  
    bluepr(); }  
else if ( bts==1 && l==0 )  
  { lcd.setCursor(0,1);  
    lcd.print("      ESC");  
    l=l+1;}  
if (digitalRead(TLAC4) == 0)
```

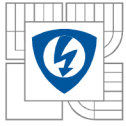


```
{ delay (200);  
  lcd.clear();  
  l=0;  
  bluepr(); }  
if ( Serial3.available() )  
{ s=(Serial3.read())- '0';  
  if (s == 0)  
  { s1=s1+1;  
    servo1.write(s1); }  
  else if (s == 1)  
  { s1=s1-1;  
    servo1.write(s1); }  
  else if (s == 2)  
  { s2=s2+1;  
    servo2.write(s2); }  
  else if (s == 3)  
  { s2=s2-1;  
    servo2.write(s2); }  
  else if (s == 4)  
  { s3=s3+1;  
    servo3.write(s3); }  
  else if (s == 5)  
  { s3=s3-1;  
    servo3.write(s3); }  
  else if (s == 6)  
  { s4=s4+1;  
    servo4.write(s4); }  
  else if (s == 7)  
  { s4=s4-1;  
    servo4.write(s4); }  
  else if (s == 8)  
  { servo5.write(40); }  
  else if (s == 9)  
  { servo5.write(150); }  
}  
blueprog(); }
```

5.1.7 Automatický režim

Automatický režim a programová podpora tohoto režimu je oproti ostatním programům relativně jednoduchá. Začátek programu je stejný jako u ostatních, tedy zápis na LCD displej a testování stavu tlačítka pro ukončení podprogramu. Z hlediska jádra programu se pak jedná o posílání hodnot na jednotlivé servomotory. Hodnoty se postupně zvyšují nebo snižují a aktuální hodnota odpovídá skutečnému úhlu natočení servomotoru.

Automatický režim umožňuje širokou škálu programových změn. Změna programového kódu vychází z požadavků na požadovanou funkcionalitu robota, tedy opakované činnosti realizované na základě časového a dispozičního schématu. Robot může zajišťovat funkce, které odpovídají elementárním logistickým procesům. Vzhledem



ke konstrukčnímu provedení robota se jedná zejména o manipulační činnosti, např. opakovaný přesun různých předmětů z místa A do místa B. Složitější řešení může odpovídat činností spojených např. s pásovým dopravníkem resp. technologickým nebo výrobním procesem. V tomto případě musí být součástí algoritmu programu nejen sekvence pro manipulační činnosti, ale musí být zohledněno i načasování. Pokud bychom uvažovali nasazení ekvivalentního robota ve výrobním prostředí, může zajistit např. přísun výrobků nebo polotovarů do zpracování na prvním pracovišti a po ukončení zpracování jejich odebrání a přesun na další pracoviště. Jedná se např. o procesy postupného lisování polotovarů, myčka polotovarů, dílčí operace (kompletace, balení) na výrobních páscech. V obou zmíněných případech se v podstatě jedná o jednocelový automat. Logickým požadavkem je rozměrová dispozice pracoviště a možnost dosahu ramen robota.

5.2 Program řídicí jednotky (uvnitř ovladače)

Program řídicí jednotky uvnitř ovladače obsahuje sekci inicializace a zaznamenávání hodnot akcelerometrem. Na rozdíl od programu řídicí jednotky robota nejsou k dispozici možnosti rozšíření, lze uvažovat pouze o případném použití jiného ovladače.

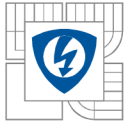
5.2.1 Inicializace

Stejně jako u robota, řídicí jednotka vyžaduje provedení inicializace. V tomto případě je inicializace poměrně omezenější než u jednotky robota. V prvních dvou krocích se definují pomocné proměnné pro akcelerometr, gyroskop a pro ostatní cykly. Jelikož se jedná o 3 osy zařízení, je potřeba nastavit 6 pomocných proměnných. V dalších dvou krocích se propojují názvy *TLAC1* a *TLAC2* s piny. Pro tlačítka jsou použité piny 3 a 2.

Následuje smyčka *main*, která obsahuje čtveřici příkazů. Pomocí příkazu *pinMode(TLAC1, INPUT_PULLUP)* se nastaví integrovaný “pullup“ rezistor na konkrétní pin. Tímto způsobem je zajištěno, že pokud nebude sepnuté tlačítko, tedy vstup uzemněný, bude pin připojen na vysokou úroveň. Další příkaz vytvoří softwarovou sériovou linku na analogových pinech 2 a 6. Posledním příkazem ve smyčce *main* je nastavení rychlosti na sériové lince.

Smyčka *loop* testuje stav přerušeni buď ze sériové linky, nebo změny úrovně tlačítka. Pokud znak pošle sériová linka, řídicí jednotka otestuje komunikaci s chipem akcelerometru. V případě úspěšně proběhlé komunikace odpoví na sériovou linku písmeno “Y“. V případě, kdy komunikace neproběhne korektně, odpoví na sériovou linku písmenem “N“. Při změně úrovně tlačítka, program skočí na cyklus záznam, který je popsán níže.

```
void setup()
{pinMode(TLAC1, INPUT_PULLUP);
pinMode(TLAC2, INPUT_PULLUP);
Wire.requestFrom(2, 6);
Serial.begin(9600);}
void loop()
{if (digitalRead(TLAC2) == 0)
{ delay(400);
zaznam(); }
if ( Serial.available())
{ a=(Serial.read());
```

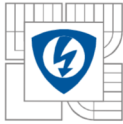


```
    delay(30);  
    if (a==48)  
    { accelgyro.initialize();  
      Serial.write(accelgyro.testConnection() ? "Y" : "N");  
      delay(500);}  
    }  
    loop();  
}
```

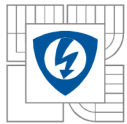
5.2.2 Zaznamenávání akcelerometrem

Prvotním a základním příkazem je *accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz)*. Slouží k tomu, aby trvale četl aktuální hodnoty jak z akcelerometru, tak gyroskopu a ukládal je do předem nadefinovaných proměnných. V další části programu se porovnávají jednotlivé načtené hodnoty v několika podmínkových cyklech *if*. Vstupní podmínky tvoří dvě čísla, horní a dolní hranice. Definice cyklů je schematicky uvedena níže. První číslo u prvního stupně tvoří tzv. minimální hranici, kdy se začne hodnota odesílat. Toto číslo musí mít dostatečně vysokou hodnotu, aby tvořilo tzv. mrtvý bod, neboli oblast, kde ovladač nereaguje. Druhé číslo je zároveň prvním číslem druhého stupně. Symbolizuje tedy konec prvního stupně a zároveň větší naklonění ovladače a posílání hodnoty, která zajistí rychlejší pohyb robota. Jedná se celkem o 8 podmíněných cyklů *if* na porovnávání pohybu v souřadnicových osách *x* a *y*. Posledním příkazem je čtení přerušeni ze sériové linky, které je vyvoláno poslaným znakem z řídicí jednotky v robotu. Je dáno ukončením programu pro ovládání pomocí ovladače a program v dálkovém ovladači přeruší zaznamenávání hodnot a přejde do smyčky *loop*.

```
void zaznam()  
{ accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);  
  if(ax>4500 && ax<8000 && t==1)  
  {Serial.write("0");  
   delay(40);}  
  else if(ax<-5500 && ax>-9000 && t==1)  
  {Serial.write("1");  
   delay(40);}  
  else if(ax>8000 && ax<15000 && t==1)  
  {Serial.write("2");  
   delay(40);}  
  else if(ax<-9000 && ax>-15000 && t==1)  
  {Serial.write("3");  
   delay(40);}  
  else if(ay>4500 && ay<8000)  
  {Serial.write("4");  
   delay(40);}  
  else if(ay<-5500 && ay>-9000)  
  {Serial.write("5");  
   delay(40);}  
  else if(ay>8000 && ay<15000)  
  {Serial.write("6");  
   delay(40);}
```

```
else if(ay<-9000 && ay>-15000)
{Serial.write("7");
  delay(40);}
if (digitalRead(TLAC2) == 0)
  {Serial.write("8");
    delay(400);}
else if (digitalRead(TLAC1) == 0)
  {Serial.write("9");
    delay(400);}
else if ( Serial.available())
  {a=(Serial.read());
    if (a==49)
      {delay(40);
        loop();}
  }
zaznam();
}
```

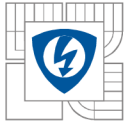


6 Závěr

Cílem této práce bylo zdokonalení původního elektronického řídicího systému robota a použití akcelerometru s gyroskopem pro manuální řízení. Původní řídicí systém byl navržen a realizován s konkrétním cílem a s ohledem na momentální možnosti a dostupné technické prostředky. Robot simuloval manipulační činnosti ve výrobním procesu. Jakékoli změny v programu řídicího systému vyžadovaly demontáž částí základny robota, aby bylo možné připojit se na programovací konektory. Kromě programové sekvence byl k dispozici i manuální mód. I ten měl řadu omezení a při jeho používání byl chod robota nestabilní.

V rámci realizovaného řešení byly konstrukční dispozice robota zachovány. Na tělo robota byl doplněn USB konektor pro připojení k programovacímu bloku a konektor externího ovladače. Zásadním způsobem se změnil řídicí systém. Jádrem řídicího systému je nyní multifunkční kit Arduino. Tento kit obsahuje celou řadu kompatibilních modulů, pro současné řešení je významná aplikace modulů bluetooth, gyroskopu, akcelerometru a LCD displeje. Pro manuální mód se nyní využívají moduly gyroskop a akcelerometr, robot podporuje použití externího ovladače, který nahradil tlačítkové ovládání umístěné na těle robota. Pro ovládání robota bylo rovněž otestováno bluetooth rozhraní a ovládání pohybu robota mobilním telefonem. Na straně softwarového řešení došlo ke změně programovacího jazyka. Jednou z významných předností řešení je podpora rozšiřitelnosti a multifunkčnosti.

Cíle stanovené pro tento projekt byly úspěšně splněny. Všechny nedostatky, které vykazoval původní systém, byly spolehlivě odstraněny. Navíc se podařilo implementovat rozšířené funkce pro ovládání, např. aktivní bluetooth modul. Nutnou podmínkou pro využití této varianty ovládání robota je instalace podpůrné mobilní aplikace. Využití nové konfigurace s modulem Arduino se podařilo dosáhnout zřetelně vyšší stability oproti původnímu systému. Stejně tak obsluha robota dálkovým ovladačem se výrazně zjednodušila oproti tlačítkovému systému.



Seznam použitých zdrojů

- [1] Serva a jejich ovládání. JEŽERSKÝ, Jiří. [online]. [cit. 2014-12-14]. Dostupné z: <http://m.serva.webnode.cz/rizeni-serva-teorie/>.
- [2] Root.com: Arduino jak pro něj začít programovat [online]. [cit. 2014-12-14]. Dostupné z: <http://www.root.cz/clanky/arduino-jak-pro-nej-zacit-programovat/>.
- [3] Arduino: Arduino Mega 2560. [online]. [cit. 2014-12-16]. Dostupné z: <http://arduino.cc/en/Main/ArduinoBoardMega2560>
- [4] JURÁNEK, Martin. *Prostředky automatického řízení: Akcelerometry*. Dostupné z: http://homel.vsb.cz/~jur286/prostredky_aut_rizeni/preklad.htm. VŠB Technická univerzita Ostrava.
- [5] Omegaeng: Akcelerometry. [online]. [cit. 2014-12-16]. Dostupné z: <http://www.omegaeng.cz/prodinfo/Accelerometers.html>
- [6] Arduino: Lekce 25 - Arduino a akcelerometr s MPU-6050 Více zde: <http://arduino8.webnode.cz/news/lekce-25-arduino-a-akcelerometr-s-mpu-6050/>. [online]. [cit. 2014-12-16]. Dostupné z: <http://arduino8.webnode.cz/news/lekce-25-arduino-a-akcelerometr-s-mpu-6050/>
- [7] Wikipedie: Bluetooth. [online]. [cit. 2014-12-16]. Dostupné z: <http://cs.wikipedia.org/wiki/Bluetooth>
- [8] Arduino: Arduino Nano. [online]. [cit. 2014-12-16]. Dostupné z: <http://arduino.cc/en/Main/ArduinoBoardNano>