



## **Středoškolská technika 2014**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

## **ROBOT PRO STOLNÍ SOUTĚŽ**

**Jakub Novotný**

SPŠ A VOŠ PÍSEK  
Karla Čapka 402, 397 11 Písek

## **Anotace**

Tato práce se zabývá teoretickým rozborem možných řešení robota stopaře jeho výrobou oživení a naprogramování. Popisuje pravidla a parametry nutné k soutěžení. Zabývá se vysvětlením PID regulace pro dosažení co nejmenší ujeté dráhy. Nabízí ověřený popis realizace robota pro sledování čáry.

**Klíčová slova:** Robot, Arduino, Pathfollowing, programování, PID, mobilní robotika

## **Abstract**

This paper deals with the theoretical analysis of possible solutions to the robot tracker boost, its production and programming. It describes the rules and parameters necessary to compete. It deals with the explanation of PID control to achieve the smallest possible traveled distance. It serves the validated description of the implementation of the robot tracking lines.

**Keywords:** Robot, Arduino, Pathfollowing, programming, PID, mobile robotics

## OBSAH

1. Úvod do dlouhodobé maturitní práce .....	4
2. Teoretický rozbor.....	4
2.1 Robotika .....	4
2.2 Robotické soutěžní disciplíny.....	5
2.3 Pohyb robota .....	6
2.4 Senzory .....	9
2.5 Elektromotor .....	9
2.6 MCU.....	14
3. Praktické řešení .....	19
3.1 Mechanická konstrukce .....	19
3.2 Stavba robota.....	20
3.3 Pohon robota.....	20
3.4 Elektronika.....	20
4. Program Robota.....	28
4.1 Arduino studio.....	28
4.2 Tvorba programu .....	29
5. Závěr.....	32
6. Citace.....	33
7. Obsah CD.....	33
8. Přílohy .....	34

# 1. ÚVOD DO DLOUHODOBÉ MATURITNÍ PRÁCE

Tématem maturitní práce je realizace mobilního robota, který se pohybuje po čáře a může se účastnit robotické disciplíny Pathfollower. Návrh robota byl vytvořen na kroužku Robotiky KRA Písek. Pro výrobu jsem použil podvozek z plechu a polykarbonátu domácí výroby, kde jsou uchyceny motory, akumulátory, řídicí elektronika, senzory a kola. K řízení byl použit 16-ti bitový mikroprocesor ATmega328 s bootloaderem Arduino. Celá základová deska je koncipovaná jako Arduino UNO. V této desce je zasazen Motorshield, kde je I2C expandér, který obsluhuje sedmisegmentový display, senzory a motory jsou řízeny pomocí H-můstku.

## 2. TEORETICKÝ ROZBOR

### 2.1 ROBOTIKA

Co to je robotika ?

Je chápána jako disciplína o vytváření inteligentních strojů integrující několik vědeckých a inženýrských oblastí.

**Dělení robotiky :**

**Teoretická robotika:** hledá principy, možnosti a omezení (biologie, psychologie, etologie, matematika, fyzika).

**Experimentální robotika:** ověřuje principy, staví hračky (kybernetika, umělá inteligence, inženýrské disciplíny).

**Průmyslová robotika:** Navrhuje, staví a používá průmyslové roboty (teorie a instrumentace řízení, elektronika, strojírenské technologie a automatizace)

**Různá aplikovaná robotika:** Navrhuje různé inteligentní stroje pro průmysl i jina. ( stroje pro kontrolu kvality ve výrobě často vybaveny schopností vidět, **mobilní roboty**) [1]

## 2.2 ROBOTICKÉ SOUTĚŽNÍ DISCIPLÍNY

### Robosumo

V ringu soutěží vždy jen dvojice robotů, kteří se položí za startovní čáry. Po startu se snaží vytlačit jeden druhého mimo ring. Pokud se jeden z robotů dotkne jakoukoli svou částí plochy mimo ring, prohrává. Ring je kruhová deska černého podkladu s bílým okrajem, kde jsou hnědé startovací čáry. Kolem ringu by měl být 2m prostor kde nesmí nic být kvůli ultrazvukovým senzorům. Robot, který bude soutěžit v této kategorii tak musí být 10x10cm dlouhý a široký. Výška není nijak omezena, jen robot nesmí být těžší než 500g. [2]

### Robotem rovně

Ač se název nebo zadání úkolu zdá být jednoduché, je tato disciplína velmi podceňována. Úkolem je, aby autonomní robot bez jakéhokoliv dálkové ovládní musí ujet co nejdelší trasu. V této soutěži se zachází do detailů, jako je třeba GPS navigace nebo i odometrie. [2]

### Stopař

Robot by měl být autonomní, realizovaný tak, aby projel dráhu, která je vyznačená čarou. Dráha má čistě bílý podklad obvykle dýhovaná dřevotříska, může na ní být svíceno. Čára je tmavý vodící pruh široký 15mm. Pro sledování čáry se obvykle využívá reflexních vlastností bílého podkladu, zatímco černá čára světlo pohlcuje. Na dráze může a nemusí být několik překážek (přerušená čára, cihla, tunel nebo i třeba most). **Přerušená** čára znamená vyzkoušet schopnost robota navázat na čáru, která může být přerušena až na 7cm. **Cihla** je druh překážky, který osvědčí zda robot vnímá okolí a zda se cihle dokáže vyhnout. **Tunel** může mít minimální průlez 20x20cm. Kategorie stopař má však i své omezení a to takové, že robot musí splňovat určité parametry. Nesmí být širší a vyšší než 20cm. Délka není limitována, avšak je třeba brát v potaz, že se zde může nacházet tunel nebo most. Minimální rozměry nejsou omezeny ani hmotnost robota není omezena. [2]

## 2.3 POHYB ROBOTA

Obecně dělíme podle **pohybu** :

### Stacionární

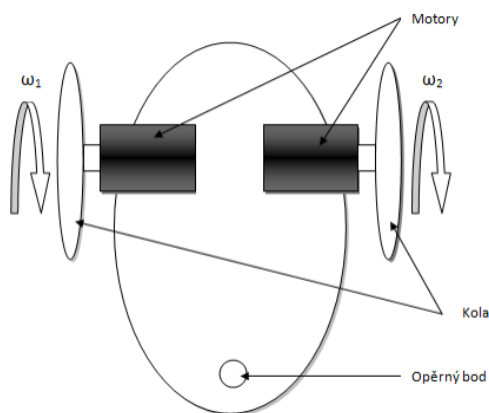
Na rozdíl od mobilních se nepohybují ale stále stojí na stejném místě v praxi se můžeme nejčastěji setkat s průmyslovými roboty a manipulátory.

### Mobilní

Tyto roboti jsou většinou autonomní. Mají snímače a jiné prvky určené k poskytnutí informací o poloze, směru a rychlosti pohybu robota. Další možnost řízení operátorem na stejném principu, jako je auto na dálkové ovládní. Ale jejich hlavní vlastností je fakt, že mají vlastní podvozek, pohon a napájení.

Podvozků je několik **druhů**:

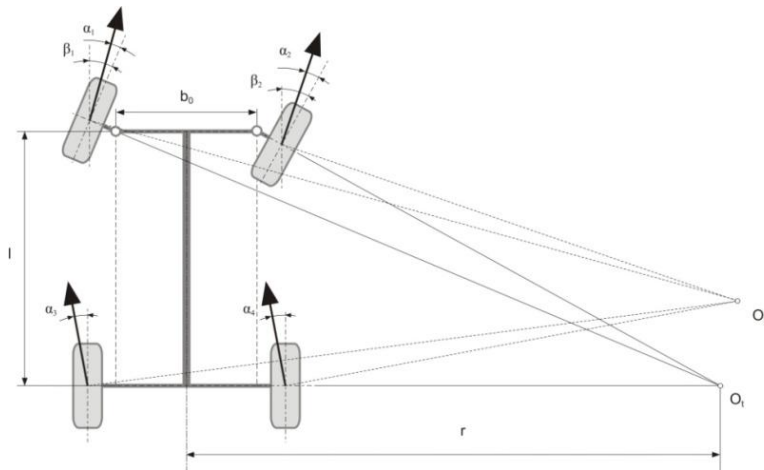
- **Diferenciální podvozek** – Princip diferenciálního podvozku spočívá v tom, že používá dvě nezávisle poháněná kola a jedno volně otočné. Obrázek 2.1 Výhoda této koncepce je manévrovatelnost - robot je schopen se otáčet kolem osy, dále snadné řízení-změnou rychlosti otáčení kol. Nevýhoda je použití v členitém terénu. [3]



### 2.1 Diferenciální řízení

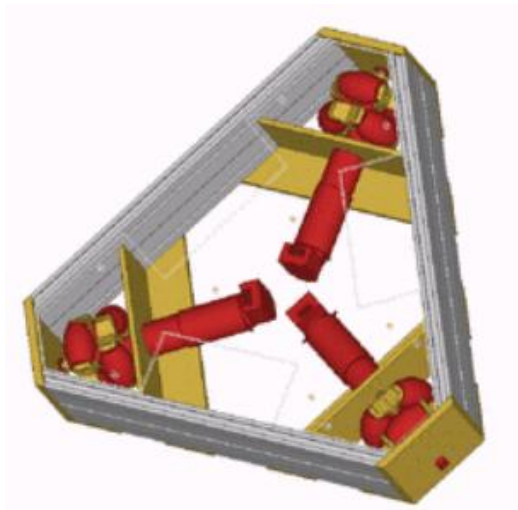
- **Ackermanův podvozek** - jedná se o nejznámější typ podvozku. Variantu se čtyřmi koly můžeme vidět například u automobilů, kde se používají dvě varianty, přední

kola natáčecí a zároveň hnaná, nebo přední kola natáčecí a zadní hnaná. Obrázek 2.2  
Tříkolová varianta je známá jako dětská tříkolka, kde zadní kola jsou hnaná, a přední  
kolo je natáčecí. Ovšem přední kolo může být natáčecí a zároveň hnané. [3]



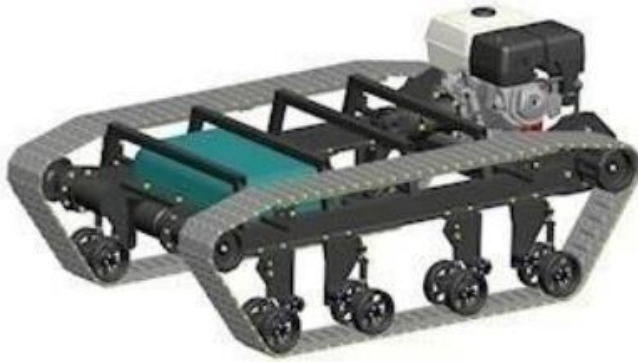
Obrázek 2.2 Ackermanův podvozek

- **Všesměrový podvozek** - Tento typ podvozku je založen na principu všesměrových kol, která umožňují pohyb ve dvou osách a zatáčet s nulovým poloměrem. Nejčastěji tři kolové nebo čtyřkolové ústrojí. Obrázek 2.3 Nevýhodou je vyšší cena. [3]



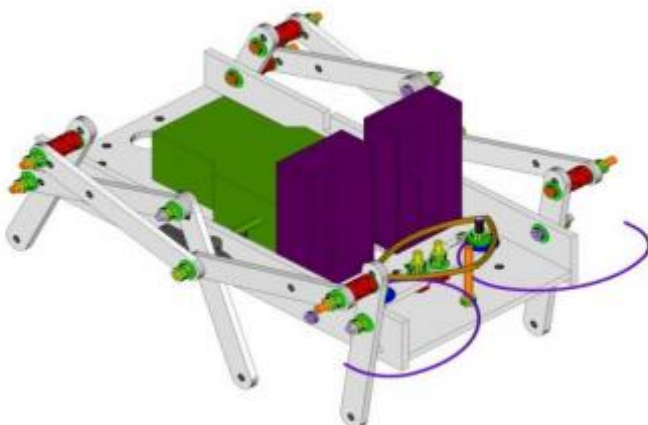
Obrázek 2.3 všesměrový podvozek

- **Pásový podvozek** - Princip řízení je téměř totožný s diferenciálním podvozkem. Manévrovací schopnosti, jako je otáčení, se neobejde bez smyku. Obrázek 2.4 Pro pásové podvozky platí, že rozvor mezi pásy a jejich délka mají přímý vliv na manévrovací schopnosti. Výhodou toho typu podvozku je vysoká průchodnost terénem, robustnost a spolehlivost. [3]



Obrázek 2.4 pásový podvozek

- **Kráčející podvozek** - Zvláštní skupina mobilních robotů je založena na kráčejícím podvozku. Obrázek 2.5 Roboti se objevují od dvounohých až po osminohé konstrukce. Svým pohybem napodobují např. pohyb člověka, nebo hmyzu. [3]



Obrázek 2.5 kráčející podvozek



## 2.4 SENZORY

Snímač je zařízení, které je též známo jako čidlo nebo snímač. Je určeno pro snímání a detekci různých fyzikálních veličin, vlastností látek nebo technických stavů. Snímače se používají v mnoha oblastech lidské činnosti.

Podle oblasti a měřené veličiny je lze rozdělit následovně:

- **ultrazvukový snímač**

Senzor pracující se zvukem při 40kHz. Princip je jednoduchý. Senzor vysílá zvukovou vlnu asi 5 $\mu$ s. Ta se odrazí od měřeného objektu a letí zpět, kde je přijata a vyhodnocovací elektronika senzoru vypočítá časový interval mezi vysílaným a přijímaným signálem. Z toho lze vypočítat vzdálenost.

- **odrazový infrasenzor**

Princip reflexních vlastností bílé a černé barvy.

- **snímač čárového kódu**

CCD snímač.

- **teplotní snímač**

Senzor snímající teplotu. Zde je hlavní součást termistor. Používá se v teploměrech, automobilech, PC apod.

- **snímač polohy – optický**

Snímač polohy se používá v servomotorech a různých enkodérech. Je to obdoba optoreflexní závory.

- **snímač barvy a osvětlení**

Pracuje na principu odrazu světla od různých barev.

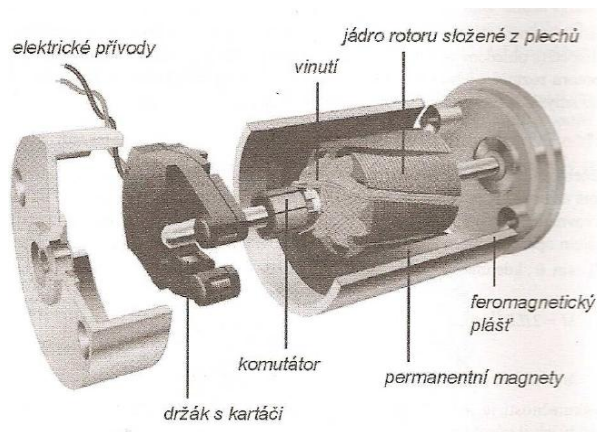
## 2.5 ELEKTROMOTOR

Je to takový stroj, který převádí elektrickou energii na mechanickou. V praxi se můžeme setkat se dvěma druhy motorů. Motory na stejnosměrný proud, nebo střídavý proud. Každý má své výhody a nevýhody.

Hlavní konstrukce:

**Stator** (pevná část) - je u většiny typů prakticky stejný. Je složen z nosné kostry motoru, svazku statorových plechů a statorového vinutí. Obrázek 2.6

**Rotor** (pohyblivá část) - hřídel s nalisovanými rotorovými (elektrotechnickými) plechy s drážkami, do kterých se vkládají měděné tyče, které jsou na obou stranách spojeny mosaznými kruhy. [3]

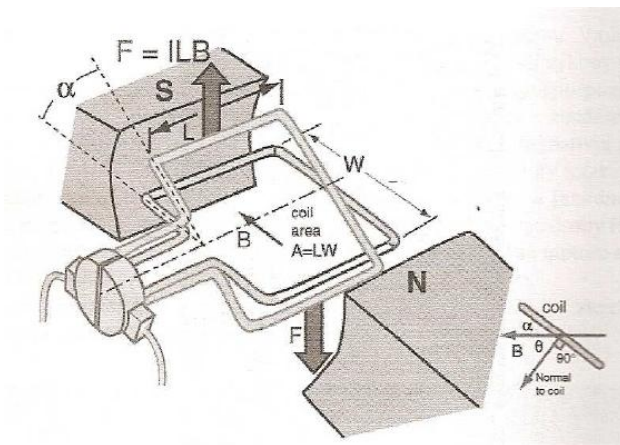


Obrázek 2.6 Rotor stator

## Stejnoseměrné elektromotory

### Stejnoseměrný motor

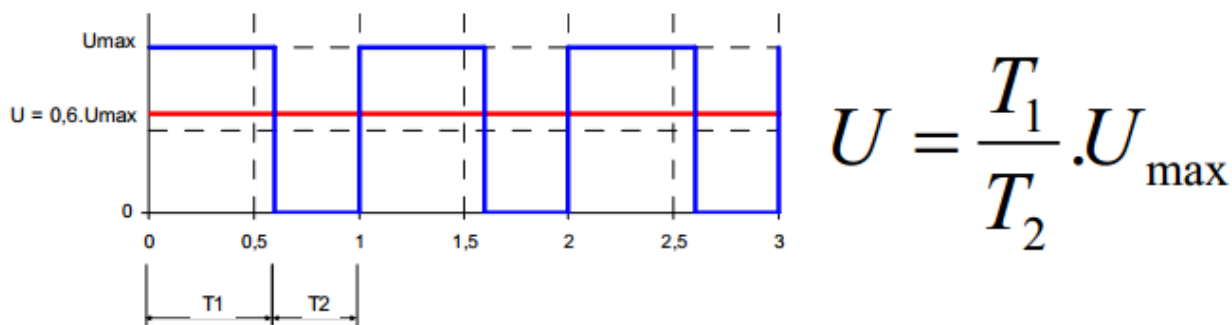
Stejnoseměrný motor nebo také často označovaný DC motor je točivý elektromotor, napájený stejnosměrným proudem. Proud protékající smyčkou se chová stejně jako permanentní magnet, který se může otáčet. Díky komutátoru se navíc dvakrát za otáčku změni jeho polarita. Jelikož souhlasné póly magnetů se odpuzují, tak pokud je komutátor nastavený tak, že se polarita magnetů změni v okamžiku, kdy jsou si jejich opačné póly nejbližší, tak budou na pohyblivý magnet neustále působit síly, které ho budou nutit k pohybu. Obrázek 2.7 Výhody jsou snadné řízení - změnou budícího napětí na rotoru a lineární charakteristika v závislosti počtu otáček na napětí. [4]



Obrázek 2.7 řez motorem-princip činnosti

**Řízení DC motoru:** Dva druhy řízení.

- 1. Řízení rychlosti / otáček motoru.** Jelikož otáčky DC motoru jsou přímo úměrné napájecímu napětí, můžeme motor řídit změnou napětí. Proto se používá PWM řízení - pulzní šířková modulace. Princip spočívá v rychlém spínání a vypínání napětí. Obrázek 2.8 Díky vysoké frekvenci spínání, rotor nestačí tyto změny sledovat a chová se, jako by byl napájen středním napětím, která je dána dobou sepnutí a rozepnutí. [5]

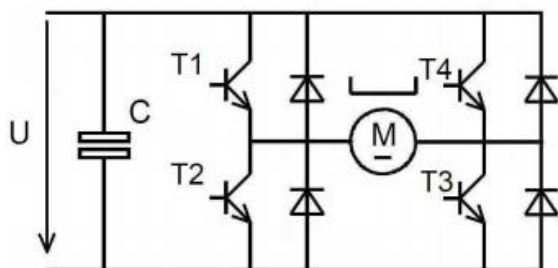


Obrázek 2.8 PWM - vypočet středního napětí

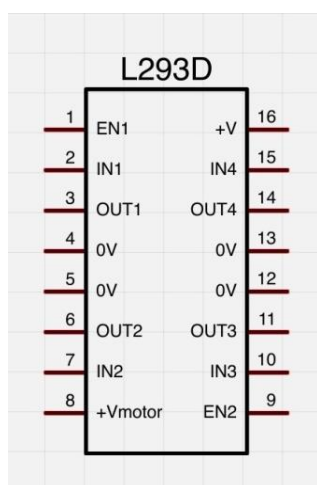
- 2. Řízení směru otáčení.** Realizujeme změnou polaritu napájení zapojením v H-můstku.

H-můstek je čtveřice spínacích prvku nejčastěji bipolární tranzistory nebo MOS-fety. Obrázek 2.9 Výběr mezi nimi záleží na velikosti proudu, který bude spínán.

H-můstek se řídí impulzy, které otevírají nebo zavírají tranzistory. Nejpoužívanější H-můstek je L293D. Obrázek 2.10



Obrázek 2.9 H-můstek



Obrázek 2.10 L293D

Elektroda EN1 a EN2 slouží k zapnutí spínacího prvku k výstupu OUT 1,2,3,4. IN1-4 zde posíláme řídicí impulzy. V+ je napájení obvodu a V motor je napájení motorů. Viz příloha 1 - Datasheet L293D

## Servomotor

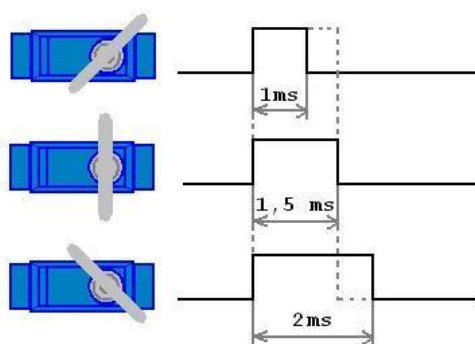
Servomotor je motor pro pohony, u kterých je potřeba nastavit přesnou polohu natočení osy. Požívá se v průmyslu u CNC strojů, ale také v modelářské technice. Servomotor má většinou rozsah od 0°- 180° a je zde zpětná vazba v podobě potenciometru, který je buď přímo nebo nepřímo připojený na motor. Poznává v jaké je

poloze. Servomotor se skládá ze tří hlavních částí. Obrázek 2.11 **DC motor** se stará o pohon serva. **Převodovka** slouží k nastavení parametrů síla/rychlost a převádí úhel natočení z motoru na zpětnovazební potenciometr. **Řídící elektronika** zpracovává vstupní PWM signál ze, kterého vyhodnocuje úhel natočení a ten porovnává s potenciometrem.



Obrázek 2.11 řez-serva

**Řízení servomotoru :** Z každého serva vedou tři kablíky červený, černý a žlutý. Červený je napájení +5V, černý je země a žlutý je řídicí. Na žlutý řídicí přivádíme 1 - 2 ms dlouhý pulz s frekvencí 50Hz. Šířka pulzu je přímo úměrná úhlu natočení. Obrázek 2.12 Budeme li generovat pulz o šířce 1,5ms, servo se natočí do střední polohy. [6]

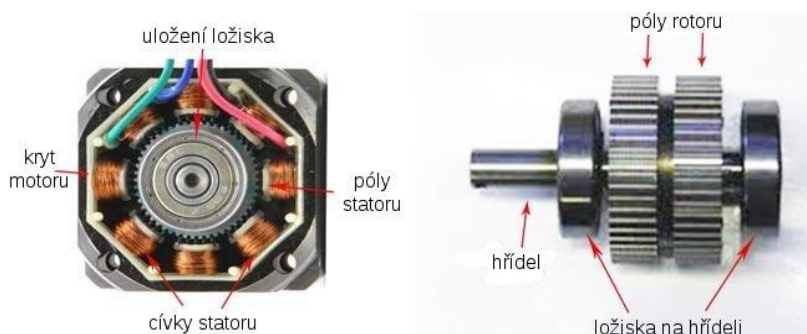


Obrázek 2.12 řízení serva

### Krokový motor

Krokový motor je točivý stroj, kde je magnetické pole generováno postupným napájením pólových dvojic. Proud procházející cívkou statoru vytvoří magnetické pole, které

přitáhne opačný pól magnetu rotoru. Vhodným zapojováním cívek dosáhneme vytvoření rotujícího magnetického pole, které otáčí rotorem. Obrázek 2.13 Krok je reakce rotoru krokového motoru na řídicí impuls. Rotor se při jednom kroku pootočí z výchozí polohy (klidového stavu) do nejbližší magnetické klidové polohy.



Obrázek 2.13 krokový motor

**Řízení krokového motoru:** Řízení krokového motoru spočívá v tom, že na vodiče, které jsou připojeny k cívkám motoru, jsou posílány pulzy "1" a "0". Po odeslání jedné sekvence se motor otočí o jeden krok. Obrázek 2.14 Je tedy nutné na vodiče posílat impulzy s určitou frekvencí. Celý motor je možné řídit dvěma H-můstky, přitom každý má dvě cívky. Nastává zde nevýhoda s vyšším počtem řídicích linek, což se dá vyřešit přídatnou logikou. [7]

	Krok 1	Krok 2	Krok 3	Krok 4
Vstup 1	1	1	0	0
Vstup 2	1	0	0	1
Animace				
Cívka L1	1	0	0	1
Cívka L2	1	1	0	0
Cívka L3	0	1	1	0
Cívka L4	0	0	1	1

Obrázek 2.14 řízení krok. motoru

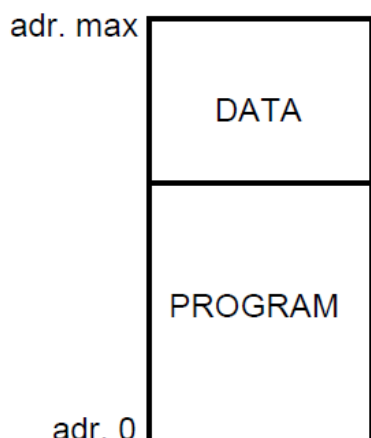
## 2.6 MCU

MCU nebo také mikrořadič je programovatelný logický automat schopný samostatné a smysluplné práce. Mikrořadič je složen z mikroprocesoru, paměť, obvody

vstupu a výstupu. Princip mikrořadiče je postupné čtení instrukcí z paměti. Existují základní dvě architektury mikroprocesorů – von Neumannova a Harvardská. Každá architektura má svoje klady a zápory. Při současném stupni integrace se častěji využívá Harvardská architektura. [8]

### Von Neumannova architektura

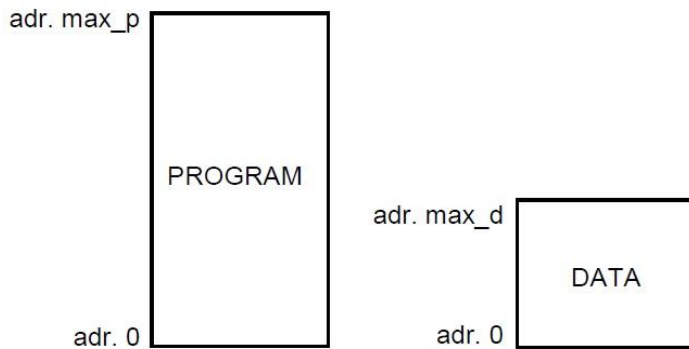
Von Neumannova architektura popisuje počítač se společnou pamětí pro instrukce i data. To znamená, že část paměťového prostoru je vymezena pro program a část pro data Obrázek 2.15. Procesor počítače se skládá z řídicí a výkonné (aritmeticko-logické) jednotky. Řídicí jednotka zpracovává jednotlivé instrukce uložené v paměti, přičemž jejich vlastní provádění nad daty má na starosti aritmeticko-logická jednotka. Nevýhodou je, že přenos obou typů dat po jedné sběrnici je pomalejší, než při oddělených sběrnících pro paměti dat a programu. [8]



Obrázek 2.15 paměť Von Neumann arch.

### Harvardská architektura

Harvardská architektura je typická oddělením paměti programu a paměti dat. Obrázek 2.16 Hlavní nevýhodou této architektury je větší technologická náročnost kvůli nutnosti vytvořit dvě sběrnice. Za hlavní výhodu lze považovat možnost jiné šířky programové a datové sběrnice. Této možnosti se široce využívá, takže najdeme osmibitové mikrořadiče s programovou sběrnicí širokou 12, 14 i 16 bitů. Mezi další výhody harvardské architektury patří rychlost vykonávání instrukcí, protože instrukci i potřebná data lze číst v jeden okamžik. [8]



Obrázek 2.16 paměť Harvardská arch.

## Instrukční soubory

Další rozdělení mikrokontrolerů je podle použitého instrukčního souboru. V oblasti jednočipových počítačů se běžně používají instrukční soubory typu CISC, RISC i DSP.

### CISC

CISC označuje procesor s kompletním instrukčním souborem. Procesor podporuje mnoho formátů a druhů instrukcí.

### RISC

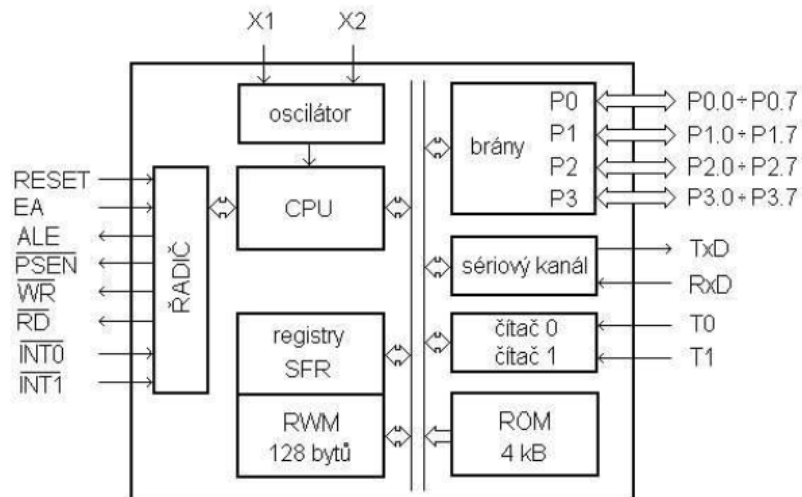
RISC označuje procesor se zmenšeným instrukčním souborem. Základní myšlenkou je omezení počtu a zjednodušení kódování instrukcí, což vede ke zjednodušení instrukčního dekodéru. [8]

### Základní struktura

- Procesor - taktovací kmitočet může být jednotky až stovky MHz, šířka slova 4 až 64 bitů. Obrázek 2.17
- operační paměť – paměť typu RAM, velikost od jednotek byte po desítky kB
- paměť programu – paměť typu ROM, EPROM, EEPROM nebo flash obsahující program a data, velikost řádově desítky až stovky kB



- Vstupně/výstupní rozhraní – takzvané porty



Obrázek 2.17 struktura MCU

## Periferie

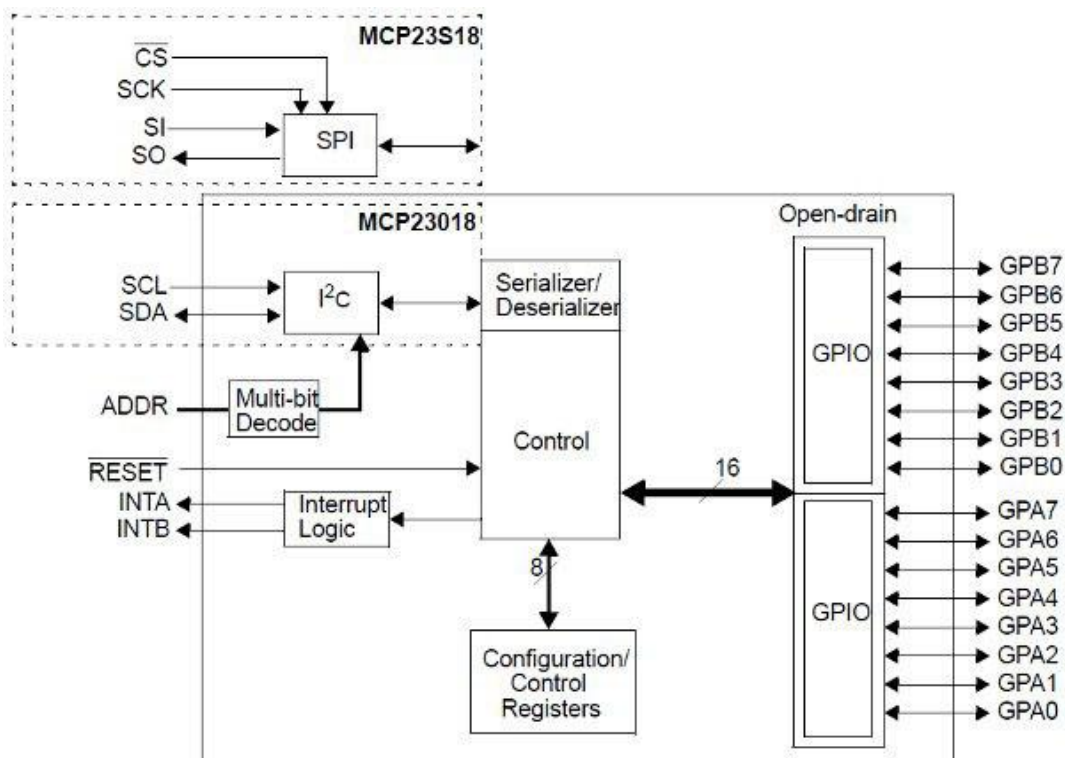
- Jednočipový počítač může kromě základních součástí obsahovat další periferie:
- řadič přerušení,
- časovače
- čítače
- řadič displeje

## Vstupně/výstupní rozhraní

V závislosti na složitosti a počtu pinů může pro komunikaci s okolím používat různá vstupní nebo výstupní zařízení. Obrázek 2.18

- paralelní porty (až desítky pinů)
- séřiové porty (asynchronní, synchronní, séřiový kanál)

- porty komunikačních sběrnic (CAN-BUS, Ethernet)
- A/D převodníky
- D/A převodníky
- PWM výstupy (pulzně šířková modulace)
- vstupy pro práci s časem – časovače, snímače impulzů
- aplikačně zaměřené porty (např. vstupy pro čtení čidel polohy rotoru, budiče LCD, apod.) [8]



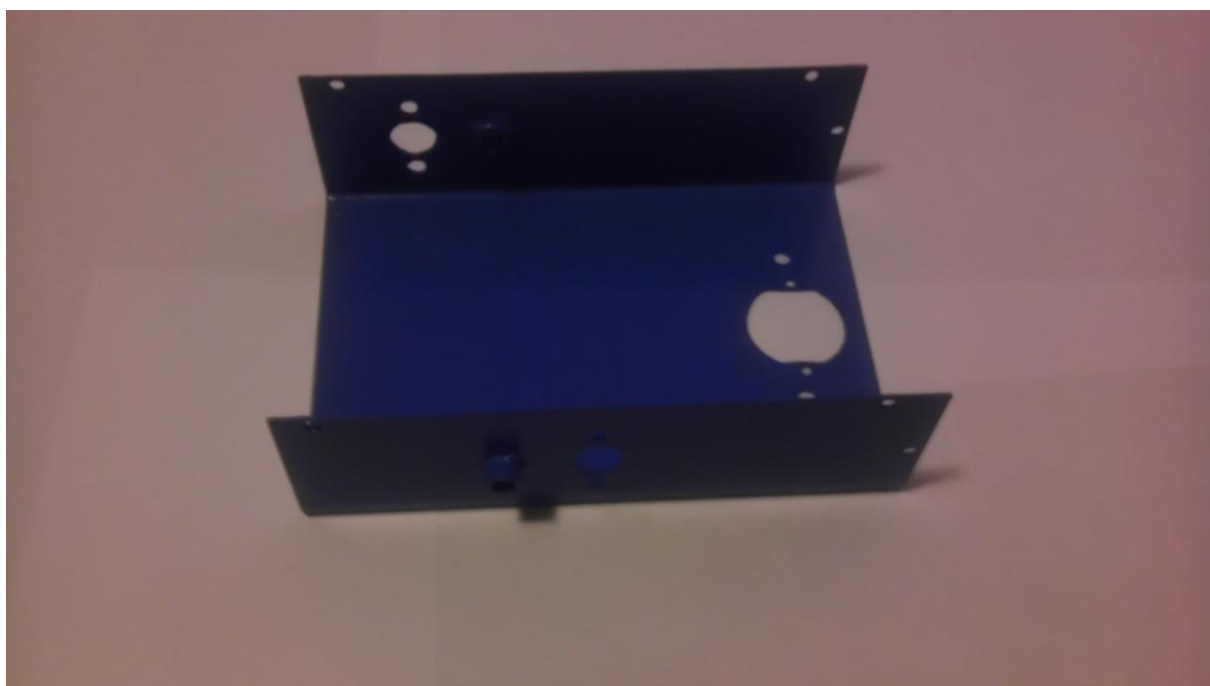
Obrázek 2.18 Vstupní/Výstupní

### 3. PRAKTICKÉ ŘEŠENÍ

#### 3.1 MECHANICKÁ KONSTRUKCE

Viz příloha 2 - podrobné foto

Jako základ byl použit jednoduchý podvozek z plechu. Obrázek 3.1 Na tento kus plechu je jako horní kryt přišroubován kus plexiskla. Na plexisklu je na distančních podložkách umístěna deska plošných spojů. Jedná se o podvozek, který je diferenciólně řízený. Je tedy osazený dvěma motory, které přes jednoduchou převodovku otáčejí kola. Kola jsou vyrobena z plastové cívky<sup>1</sup>, na které je nasazeno ozubené kolo převodovky. Jako opěrný bod je použito kulové ložisko, které je umístěno v přední části. Uvnitř přední části podvozku se nachází senzorová lišta pro sledování čáry a akumulátory.



Obrázek 3.1 Podvozek

---

<sup>2</sup> plastová cívka(špulka) kde byl namotaný cín

### **3.2 STAVBA ROBOTA**

Vezmeme plechový podvozek. V přední části se nachází kulatý otvor do, kterého umístíme kulové ložisko. To zajistíme podložkami. Poté upevníme motory k bokům podvozku. Do otvoru, ve kterém jsou bronzová pouzdra, zasuneme osu kol. Kolo je též domácí výroby. Z vnitřní části je ozubené kolo s trapézovým vzorem. Kola jsou zajištěna podložkou z každé strany osy. Poté umístíme senzor čáry do přední části podvozku kde jsou pro tuto úlohu připraveny otvory. Díky distančním podložkám je možné nastavit si výšku senzorů. Poté umístíme akupack do přední části pro lepší rozložení váhy. Na horní část polykarbonátu umístíme DPS na distančních podložkách. Ten přišroubujeme k horní hraně plechové části. Do přední části umístíme poslední kus polykarbonátu. Poté zapojíme elektroniku. K celému postupu je fotodokumentace na CD.

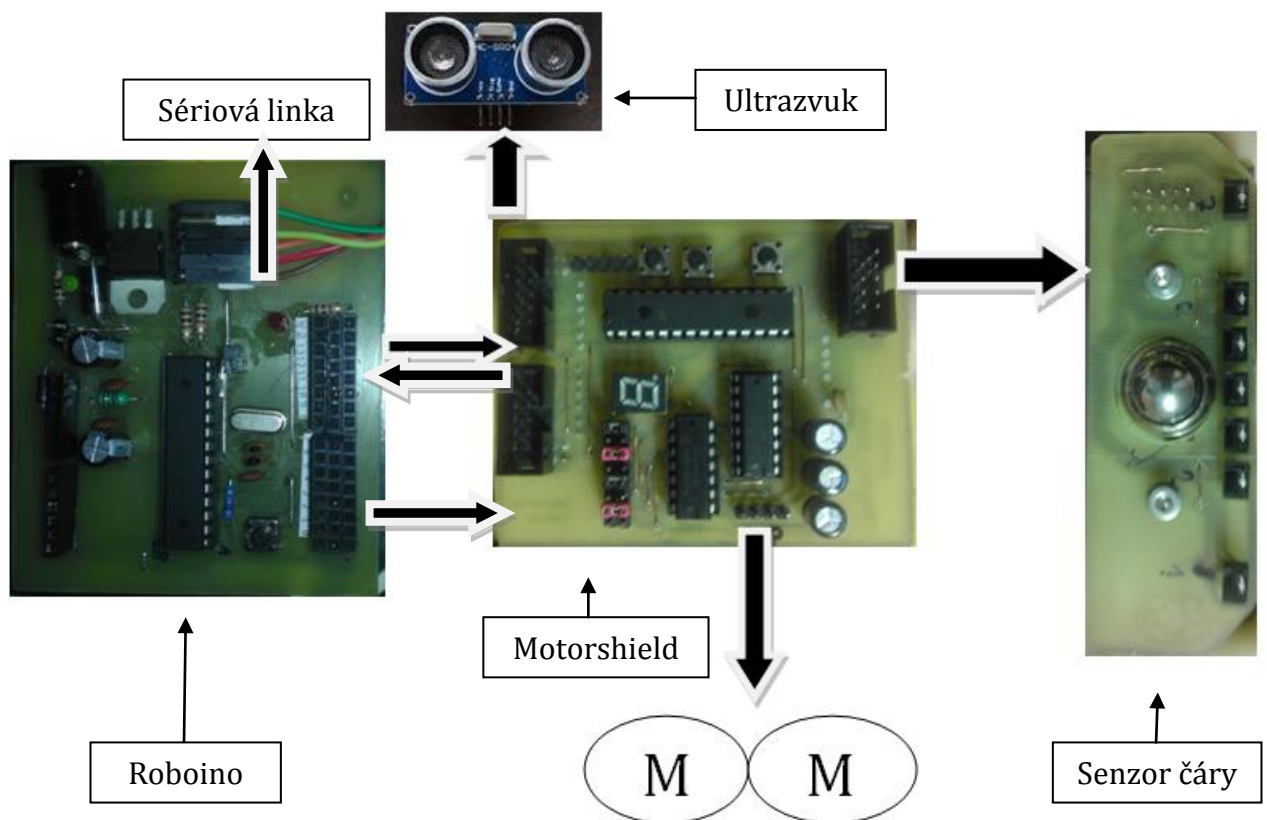
### **3.3 POHON ROBOTA**

O pohon celého robota se starají dva stejnosměrné motory. Motory jsou z průmyslové tiskárny. Výhodou je, že jsou osazeny ze zadní části velmi přesnými enkodéry. Enkodéry by se daly použít pro zrychlení robota na dráze. Převodový poměr je 50:1 . Motory jsou připojeny na motorshield, kde je budící obvod L293D, který se stará o jejich řízení.

### **3.4 ELEKTRONIKA**

Viz příloha 3 - schéma a DPS

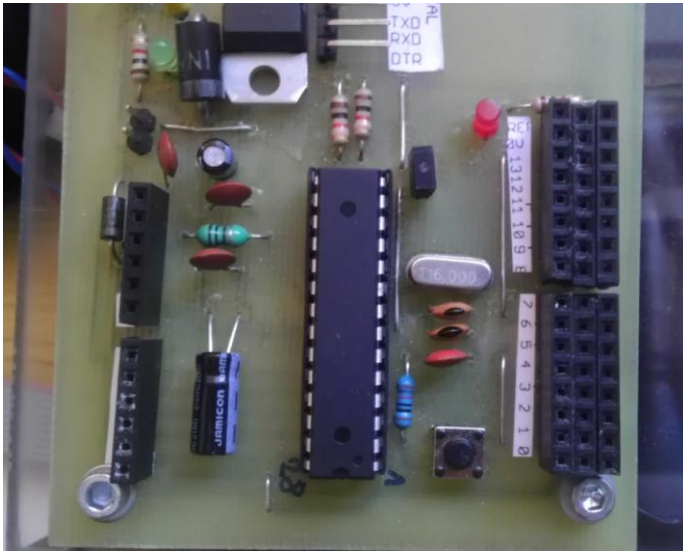
Elektronika celého robota je koncipovaná rozměry Arduina. Je zde řídicí deska nazývaná jako Roboino. Na Roboinu je umístěn motorshield, ke kterému je připojen senzor čáry, ultrazvuk a motory. Obrázek 3.2 Návrh veškerých schémat a DPS byl proveden na kroužku Robotiky KRA Písek. Já jsem provedl pouze drobné kosmetické úpravy.



Obrázek 3.2 Elektronika schéma

### Roboino - základová deska

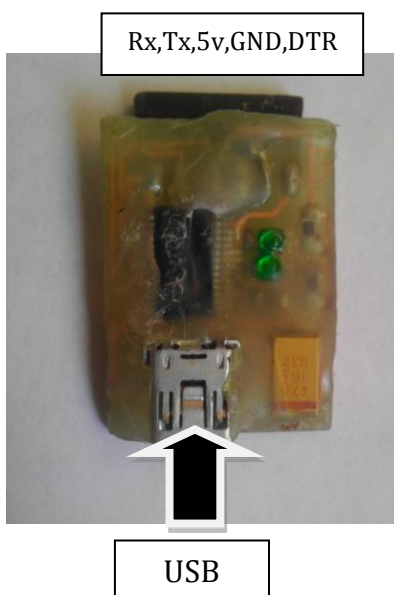
Roboino je základová deska celého robota. Je koncipována jako Arduino, takže má stejné rozmístění vstupních a výstupních pinů. Obrázek 3.3 O stabilizaci napájení se stará lineární stabilizátor LM7805. K němu jsou paralelně zapojeny kondenzátory jako filtry napájení. Jako ochrana stabilizátoru proti otočení napájení je zde ochranná dioda 1N5821. O výpočetní výkon se stará AT-Mega 328. Je taktována 16MHz krystalem a je v ní bootloader Arduino. Reset MCU provádíme stisknutím tlačítka S1.



Obrázek 3.3 Roboino

### Komunikace Roboino - PC

Programování a komunikace probíhá přes programátor USB to TTL, který je osazen převodníkem FTDI RS232RL. Obrázek 3.5 U sériové linky používáme piny Rx, Tx, +5v, GND a DTR. Pro bezdrátovou komunikaci můžeme použít bluetooth modul. Obrázek 3.4 Díky bluetooth modulu se povedlo odladit potřebné algoritmy a sledovat chování robota přímo na dráze.



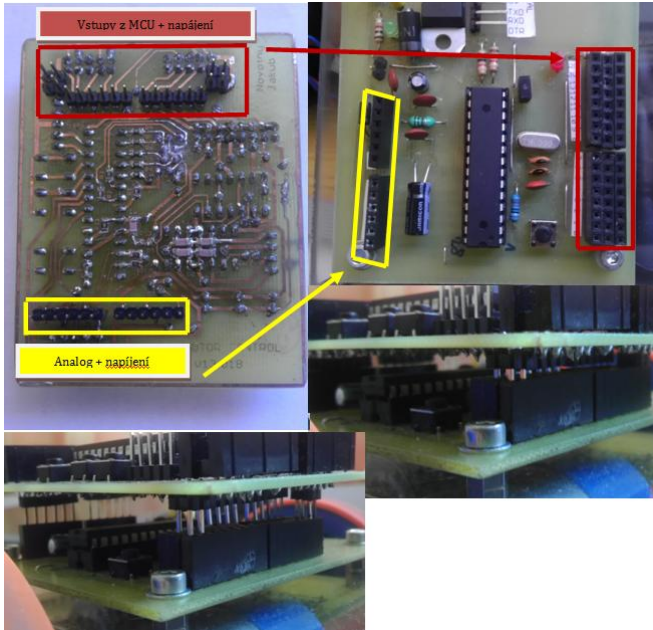
Obrázek 3.5 USB převodník



Obrázek 3.4 BT modul

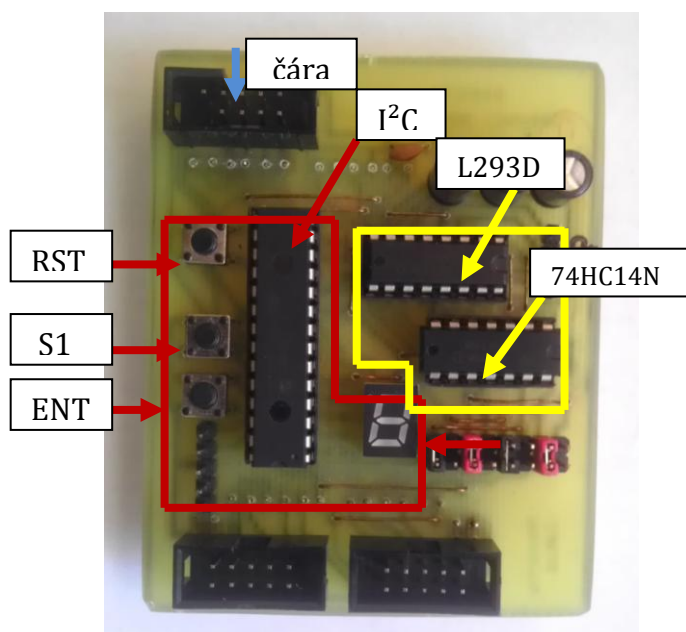
## Motorshield

Motorshield je koncipován jako shield k Arduino Obrázek 3.6 a je osazen jehlovou lištou ze spodní strany desky. Rozměry jehlové lišty jsou totožné s rozměry výstupních a vstupních pinů Roboina. Stačí pouze motorshield nasadit na Arduino.



Obrázek 3.6 Motorshield umístění

Hlavní součástí motorshieldu je I<sup>2</sup>C expander MCP 23017-E, který obsluhuje hned několik dalších periférií. Expander komunikuje s LED displejem, který slouží k snadnějšímu ovládní robota i ladění programu. Obrázek 3.7 Dále ho řídí tři senzory, které se chovají jako digitální. K expanderu jsou připojeny tři tlačítka. Reset tlačítko je připojeno na pin RESET a slouží k měkkému restartu celé elektroniky. Resetuje se pouze expander, ale ne MCU. Dalším tlačítkem je S1, kterým volíme program, a tlačítkem ENTER ho potvrdíme. I<sup>2</sup>C expander slouží pro nedostatek výstupních pinů z MCU a chová se jako posuvný registr.



Obrázek 3.7 Motorshield

Pro spínání dvou DC motorů je použit integrovaný H-můstek L293D. Tento obvod umožňuje řízení směru otáčení a PWM. K motoru je připojen integrovaný obvod 74HC14N, který obsahuje Schmittův klopný obvod s logickými obvody. Díky nim je možné řízení směru jednoho motoru jedním vodičem. Maximální proud, který je schopen dodat, je 0,5A. Motory jsou napájeny z nestabilizovaného zdroje napětí. Motory se připojují k jehlové liště vedle obvodu. Sensory jsou připojeny ke konektorům.

### Napájení

O napájení celého robota se starají dva články Samsung INR18650. Jmenovité napětí je 3,6V a kapacita je 1300mAh. Články jsou spojeny do série a jsou připojeny k stabilizátoru u Roboina a k obvodu pro řízení motorů. V nejbližší době bych rád doplnil odpojovač zátěže kvůli ochraně článků.

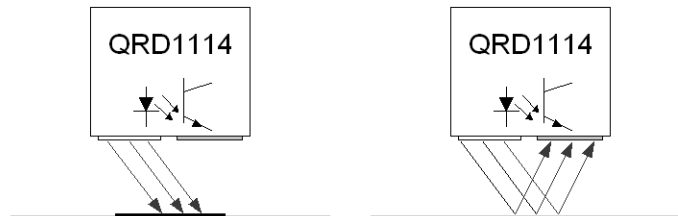
### Senzory

Robot má dva druhy senzorů. První důležitější je senzor čáry a druhý je ultrazvukový senzor překážek. V robotu jsou umístěny také interní senzory na zadní straně elektromotoru. Bohužel nejsou zapojeny a v dohledné době nebudou, jelikož jejich použití není klíčové k soutěži Stopař.



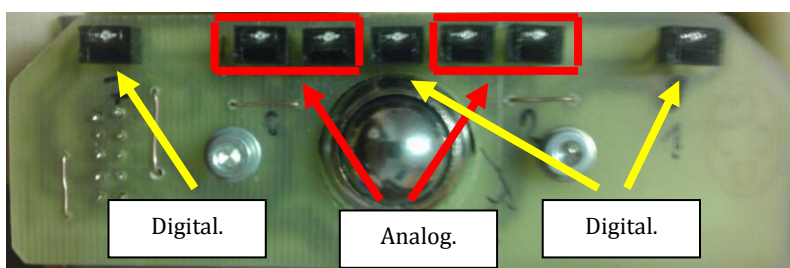
## Senzor čáry

Ke snímání čáry slouží sedm snímačů **QRD1114** vyráběných firmou Fairchild Semiconductors. Jedná se o infračervenou LED diodu a fototranzistor umístěný ve společném pouzdru.



Obrázek 3.8 Princip funkce

**Černá čára** tvořena izolační páskou, infračervené světlo pohltí, takže paprsek na fototranzistor nedopadne. Pokud se pod čidlem objeví **bílá plocha**, odrazí se paprsek do fototranzistoru. Obrázek 3.8 Šířka senzoru čáry odpovídá šířce přední části robota. Senzor čáry je osazen sedmi snímači. Čtyři jsou zapojeny na analogový vstup MCU přes motorshield. Obrázek 3.9 Zbylé tři se chovají jako digitální a jsou zapojeny na I<sup>2</sup>C expander. Pro zjištění funkčnosti čidel se použije program, který zobrazuje jednotlivé senzory jako jeden segment na led displeji.



Obrázek 3.9 Senzor čáry

Několikrát se mi stalo, že po zapnutí robota senzory nefungovaly jak měly. Často byla chyba v konektoru. Jednou odešel senzor sám a několikrát SMD součástka. Proto jsem napsal program, který testuje jednotlivé čidla a zobrazuje je na sedmsegmentovém displeji. Je to jednoduchý kód, kdy to proměně byte předdefinovanou funkci uložíme aktuální obsah čidel, který necháme zobrazit na displeji. Podle toho to jednoduchého kódu jsem schopný zjistit, jaký senzor nefunguje.

```
1. void Test_IR()
2. {
3.   byte cara = Robot.Cara();
4.   Robot.Displej_data(cara);
5.
6. }
```

## Senzor překážky

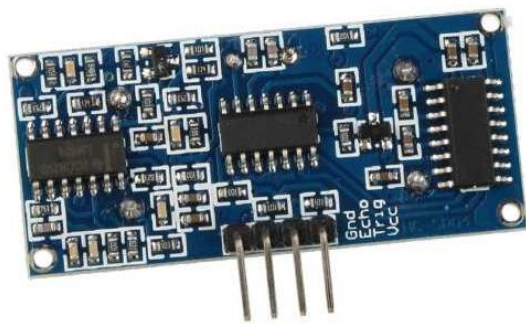
Externí snímač pro detekci předmětu, resp. cihly. Snímač je založen na hotovém produktu v podobě ultrazvukového dálkoměru HC-SR04. Obrázek 3.10 Jedná se o kompaktní modul pro měření vzdáleností od 3cm do 4m. Prakticky bylo zjištěno, že se dá měřit i kolem 1 cm. Robot senzor používá na detekci cihly, ale v praktických podmínkách je velké rušení např. lidé kolem dráhy, startovní brána, okraje apod.



Obrázek 3.10 Ultrazvuk senzor

Na robotovi je použito připojení pomocí dvou vodičů a to Trigger a Echo (pin Mode právě volí možnost připojení).  
právě volí možnost připojení).

Obrázek 3.11 Trigger znázorňuje vstup, a impuls minimální délky  $10\mu\text{s}$  způsobí, že modul HC-SR04 zahájí měření vzdálenosti. Naměřená vzdálenost je pak v podobě impulsu na výstupu Echo. Délka impulsu pak odpovídá časové délce echa, která trvá od vyslání signálů k jeho opětovnému zachycení.



GND, Echo, Trig, Ucc

Obrázek 3.11 rozmístění pinů

Program pro obsluhu ultrazvukového čidla je komplikovaný na odladění. Jelikož je na dráze velké rušení, je možnost, že bude reagovat na startovací bránu nebo okraj dráhy. Proto použijeme program, který po zachycení překážky inkrementuje proměnnou fear o 1. Pak stačí dát podmínku, když se fear rovná např. 5, tak zapni program pro objetí. Tento program je jednoduchý, ale velmi účinný co se týče potlačení chybné detekce.

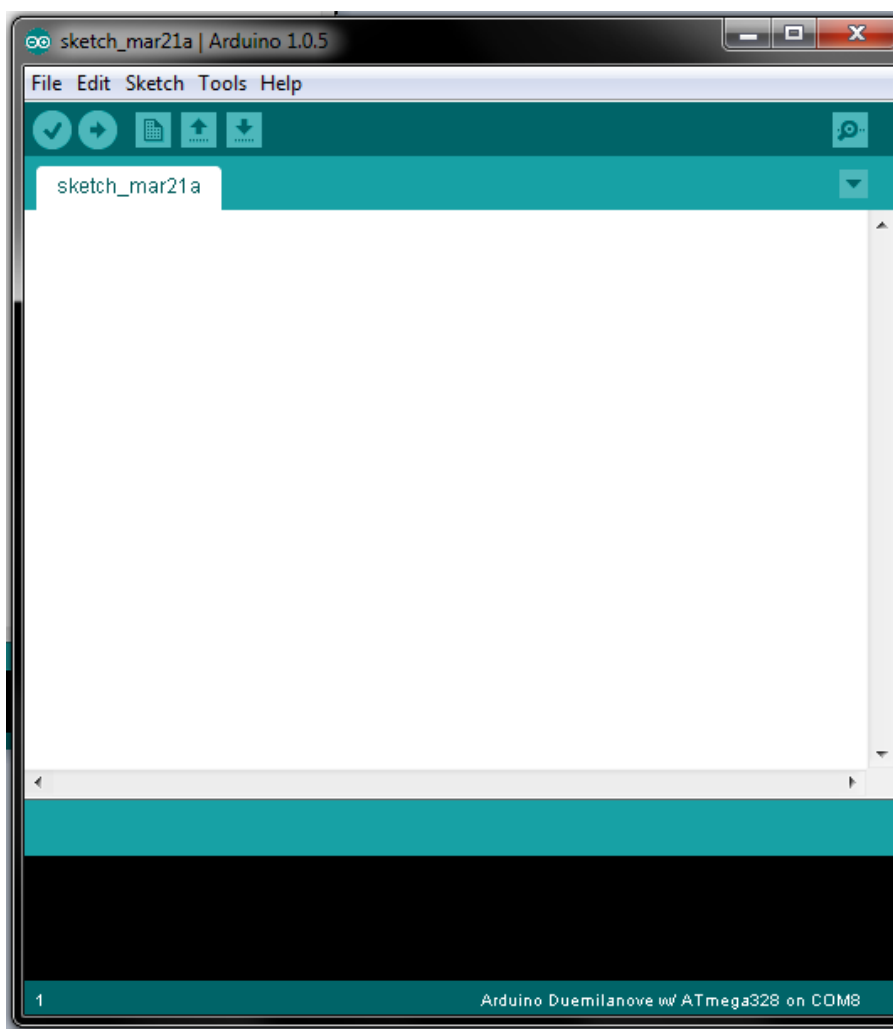
```
1. void loop()
2. {
3.   digitalWrite(11,HIGH);
4.   delayMicroseconds(5);
5.   digitalWrite(11, LOW);
6.   long cas = pulseIn(10,HIGH);
7.   long cm = (cas/2)/29;
8.   //Serial.println(cm);
9.   svit(cm/3);
10.  delay(50);
11. }
12. int fear = 0
13. void detekuj(int pocet)
14. {
15.   int i ;
16.
17.   for(i=0;i<=7;i++)
18.     {
19.       if(detekuj == 1)
20.         fear ++;
21.     }
22. }
23. if(fear == 5)
24. objeti();
```

## 4. PROGRAM ROBOTA

### 4.1 ARDUINO STUDIO

viz příloha 4 - zdrojový kód

Program je psaný pro Arduino, který má vlastní programovací jazyk Wiring. Je to obdoba jazyka C. Bootloader používáme od Arduina MCU, lze programovat v Arduino studiu. Obrázek 4.1



Obrázek 4.1 Arduino studio

Ovládání Arduino studia je velice jednoduché. V nástrojové liště je několik velmi důležitých tlačítek.

První z nich je *Verify* připomíná fajfku. Tímto tlačítkem zkontrolujeme program, zda je správná syntaxe. V případě že není, v dolní části uvidíme chybu i s číslem řádku.

Tlačítko je *Upload* , kterým nahrajeme do MCU program. A asi nejdůležitější funkce programu je *Serial Monitor* díky kterému, můžeme sledovat, co se v robotu děje komunikovat s ním nebo odladovat program.

## 4.2 TVORBA PROGRAMU

Každý program začíná vložením knihoven nebo definicí proměnných. Naše knihovna, která ovládá motorshield a zajišťuje komunikaci mezi ostatními periferiemi. Knihovnu pro ovládání jednotlivých částí robota napsal František Cifreund. Veškeré ostatní řídicí algoritmy jsem vytvořil sám. Program se dá rozdělit do dvě části. První je jednoduchý výběr programu, který provádíme tlačítky připojené k expanderu a zobrazené na displeji.

### Úvodní program

Zde jsou dvě hlavní funkce, které musí být. Funkce *Setup()* kde se inicializují proměnné, jejich pomocí určíme zda bude pin vstupní nebo výstupní. A funkce *Loop()*, která je hlavní funkcí každého programu. Běh programu má následující postup. Přidáme knihovny, definujeme proměnné, provedeme funkci *Setup()*, kde nastavíme mód pinu a funkce *Loop()*, která se opakuje do nekonečna, začne vykonávat svůj program.

Náš program začíná vložením potřebných knihoven inicializací proměnných. V knihovnách se definují piny a funkce. Poté přecházíme k hlavnímu programu, který vidíme níže. Zde si volíme program, který budeme chtít spustit. Výběr provádíme pomocí tlačítka na motorshieldu S1 a tlačítkem Enter ho potvrdíme.

```
1.  switch (Program)
2.  {
3.      case P0: Sleduj(); break;
4.      case P1: sleduj_digi(); break;
5.      case P2: test_cara();break;
6.      case P3: sleduj_automat ();break;
7.      case P4: Test_IR (); break;
8.      case P5: test_motory (); break;
9.      default :
10.         Program = Robot.Vyber_Programu(); // Dokud nenĀ zvolen pouĀlitĀ program
11.         break;                               // spoustej Vyber_progr
12.         amu()
13.     }
```

## Program pro sledování čáry

Druhá nejdůležitější část programu je část s algoritmem pro sledování čáry.

Program pro sledování čáry se dá napsat dvěma způsoby. První je jednoduchý program plný podmínek, které se dotazují, zda je pod čidlem čára, či není. Například je-li čára pod středním čidlem oběma motorům dej stejnou rychlost. Je-li pod krajním pravým levý motor naplno a pravý zpětný chod. Výhoda tohoto programu je v jednoduchosti na odladění a napsání. Nevýhodou je menší rychlost a kmitání. Kmitání znamená zbytečně ujetá dráha navíc. Obrázek 4.2



Obrázek 4.2 kmitavý pohyb

### Program-podmínky

```
1. if (Cara & Sensor_4) Robot.Motory_PWM(90,90);
2. if (Cara & Sensor_3) Robot.Motory_PWM(70,90);
3. if (Cara & Sensor_5) Robot.Motory_PWM(90,70);
4. if (Cara & Sensor_2) Robot.Motory_PWM(50,90);
5. if (Cara & Sensor_6) Robot.Motory_PWM(90,50);
6. if (Cara & Sensor_1) Robot.Motory_PWM(-20,127);
7. if (Cara & Sensor_7) Robot.Motory_PWM(127,-20);
```

A druhý způsob je použití regulátoru. V úvahu připadá **PID** regulace. Hlavní výhodou je vyšší rychlost plynulost a nekmitání. Obrázek 4.3



Obrázek 4.3 nekmitavý pohyb

PID regulace spočívá v tom, že MCU počítá s odchylkou minulou, budoucí a přítomnou. P-proporcionální funguje podobně jako zesilovač - regulátor přenásobí odchylku výstupní hodnoty od požadované hodnoty danou konstantou. Dále přidáme I- integrační složku, která sčítá odchylky dohromady. D- derivační složkou, která počítá s minulou

odchylkou a aktuální odchylkou. Složka P se snaží odstraňovat chyby aktuální, složka I chyby minulé a složka D chyby budoucí.

Odchylku má každé analogové čidlo přiřazené pomocí váženého průměru. Pokud není na analogových čidlech je zde podmínka pro dvě postraní.

Výpočet PID:  **$pid = kp*error + ki*sum\_e + kd*diff;$**

Koeficienty kp, kd a ki postupně nastavujeme.

Program : PID

```
1. error = pozadovana - prum;
2. P = error*Kp;
3. D = (error - oldError)*Kd;
4. turn = P+D; // vypocet PID
5. SpeedL = Speed - turn;
6. SpeedR = Speed + turn;
7. oldError = error;
8. Robot.Motory_PWM(SpeedL, SpeedR);
```

Odladění PID regulace je mnohdy dlouhodobá práce. Díky BT modulu se mi to povedlo asi za týden. Nejdříve jsem napsal program pro regulaci P. Je jednoduchý, ale složitý na odladění. Musíme nastavovat koeficient P, dokud robot nekmitá. V zásadě to jsou malé hodnoty. U koeficientu D se naopak používají velké hodnoty. A poslední koeficient I se mi nepovedlo přes všechnu snahu odladit.

## 5. ZÁVĚR

Povedlo se mi, zkonstruovat a naprogramovat plně funkčního mobilního robota, konkurenceschopného na robotické soutěži Stopař. Jeho stavba mi pomohla se zdokonalit v elektronice, mikroprocesorové technice a programování. Tato práce pro měla velký informační i praktický přínos. Tuto zkušenost bych všem doporučil.

Všechny elektronické součásti, kterými robot disponuje, fungují správně. Díky podobné koncepci s Arduinem je možná jejich obměna. Mechanická konstrukce byla pojata v jednoduchém stylu. Hlavní část je tvořena z plechu, který dodává robotu potřebnou váhu, aby lépe seděl na dráze. Horní a přední kryt je z průhledného polykarbonátu, díky kterému dostává pěkný vzhled.

Z programové stránky se povedlo odladit PD regulaci, díky níž robot dosahuje plynulé a nekmitavé jízdy, na což jsem hrdý, jelikož nastavení PD regulace je mnohdy neskutčný a zdlouhavý úkol. Velmi mi pomohl bluetooth modul, díky němuž jsem mohl sledovat chování robota přímo na dráze.

Robotické soutěže jsem se zatím nezúčastnil, ale mám to v plánu. Chtěl bych vyzkoušet svého robota proti silné konkurenci. Na zkušební dráze se robot pohyboval plynule za což vděčím PD regulaci. Složku I se mi nepovedlo přes velké úsilí odladit. Je zde potřebné velmi přesné nastavení.

Možností, jak zrychlit robota na dráze, není mnoho. Jednou z možností je lepší přilnavost kol, kterou jsem vyřešil gumovým náramkem místo pláště, který se skvěle osvědčil. A neméně důležité je dobré rozmístění váhy mezi přední a zadní částí. Často se u menších špatně navržených robotech stávalo, že při rozjezdu se přední část zvedla kvůli velké akceleraci. Další možnost je zvýšení počtu napájecích článků, a tím zvýšení napájecího napětí, ale to znamená chlazení stabilizátoru a výměna budicího obvodu L293D za L298N, který má možnost chlazení a zvládá větší proudy kolem 2A. Další vyzkoušená možnost je předsunutí čidel čáry trochu dopředu, aby robot mohl rychleji reagovat. A poslední možnost je použití odometrie neboli odměřování, aby by si robot zkušebně projel dráhu a zapamatoval rovné části, kde by mohl jet naplno. Tato metoda je však velmi složitá proto jsem se o ní ani nepokoušel. Poslední část, kterou bych v



nejbližší době rád doplnil, by byl odpojovač zátěže pro ochranu Li-ion článků, které jsou velmi náchylné na podbití.

Ještě jednou bych chtěl velmi poděkovat panu Mgr. Milanu Janouškovi za pomoc, rady a usměrnění. Dále panu Bc. Josefu Pajerovi za pomoc při osazování SMD součástek a svému otci za pomoc při stavbě podvozku.

## 6. CITACE

[1] ÚVOD DO ROBOTIKY. HLAVÁČ, Václav. *Robotika* [online]. [cit. 2014-03-21]. Dostupné z: <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/51Robotika/01UvodRobotika.pdf>

[2] SEINER, Antonín. PRAVIDLA KATEGORIE STOPAŘ. [online]. [cit. 2014-03-21]. Dostupné z: <http://kufr.cz/view.php?nazevclanku=pravidla-kategorie-stopar&cislocclanku=2010080004>

[3] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. Vyd. 1. Praha: BEN - technická literatura, 2004, 247 s. ISBN 80-730-0141-1.

[4] Elektromotor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-03-21]. Dostupné z: [http://cs.wikipedia.org/wiki/Elektromotor#St.C5.99.C3.ADdav.C3.BD\\_stroj](http://cs.wikipedia.org/wiki/Elektromotor#St.C5.99.C3.ADdav.C3.BD_stroj)

[5] KAKÁČ, Petr. Bezsenzorové řízení rychlosti a polohy stejnosměrného motoru. [online]. [cit. 2014-03-21]. DOI: řízení stejnosměrného motoru.

[6] Řízení serva. *Serva.cz* [online]. [cit. 2014-03-21]. Dostupné z: <http://www.serva.cz/rizeni-serva-teorie/>

[7] ŘEZÁČ, Kamil. Krokové motory. *Robotika.cz* [online]. New York: ACM Press, c2001 [cit. 2014-03-21]. Dostupné z: <http://robotika.cz/articles/steppers/cs>

[8] Jednočipový počítač. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-03-22]. Dostupné z: <http://cs.wikipedia.org/wiki/Jedno%20Dipov%20po%20ADta%204>

## 7. OBSAH CD

- 1) Příloha Datasheet L293D
- 2) Příloha foto robota
- 3) Příloha schéma a DPS elektroniky

4) Příloha program

5) Příloha video robota na dráze

## 8. PŘÍLOHY

