

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

MAPOVÁNÍ UZAVŘENÉHO PROSTORU

Jiří Jecha

Bakalářská práce

2016

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2015/2016

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří Jecha**  
Osobní číslo: **I13015**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Komunikační a mikroprocesorová technika**  
Název tématu: **Mapování uzavřeného prostoru**  
Zadávající katedra: **Katedra elektrotechniky**

### Z á s a d y p r o v y p r a c o v á n í :

Cílem je návrh autonomního systému pro vytvoření 2D mapy uzavřeného prostoru. Teoretická část práce bude obsahovat možnosti řešení autonomního vozítka, především pohonů, snímání a řízení. Dále budou v teoretické části popsány jednodušší mapovací algoritmy. Praktická část práce bude obsahovat návrh konstrukci a oživení autonomního vozítka pro mapování neznámého prostoru s inteligencí buď přímo na vozítku, nebo s online spojením s pc a aplikací na pc. Dále bude provedena implementace vybraného algoritmu a provedeno testovací mapování. Výsledek bude součástí bakalářské práce.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

- [1] VÁŇA, V. Mikrokontroléry ATMEL AVR: popis procesoru a instrukční soubor. Praha: BEN technická literatura, 2003. 336 s. ISBN 978-80-7300-083-0.
- [2] VÁŇA, V. Mikrokontroléry ATMEL AVR: programování v jazyce C. Praha: BEN technická literatura, 2003. 216 s. ISBN 978-80-7300-102-0.
- [3] VLACH, J. Řízení a vizualizace technologických procesů. Praha: BEN technická literatura, 2002. 160 s. ISBN 978-80-86056-66-X.
- [4] BRTNÍK, B. Základní elektronické obvody. Praha: BEN technická literatura, 2011. 156s. ISBN 978-80-7300-408-8.
- [5] RIPKA, P.; TIPEK, A. Master Book of Sensors. Praha : BEN, 2003. ISBN 0-12-752184.

Vedoucí bakalářské práce:

**Ing. Pavel Rozsival**  
Katedra elektrotechniky

Datum zadání bakalářské práce:

**15. prosince 2015**

Termín odevzdání bakalářské práce:

**13. května 2016**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Zdeněk Němec, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2016

## Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 20. 5. 2016

Jiří Jecha

## **PODĚKOVÁNÍ**

Rád bych poděkoval svému vedoucímu bakalářské práce Ing. Pavlu Rozsivalovi za odborné vedení, za pomoc a rady při zpracování této práce.

## **ANOTACE**

Cílem této práce je návrh a realizace autonomního vozítka, které se bude pohybovat v neznámém prostředí, kde bude snímat okolní prostor a na základě získaných dat vytvoří 2D mapu neznámého prostředí. Teoretická část se věnuje možnostem řešení vozítka. V praktické části je popsána realizace vozítka a postup vytváření inteligence vozítka pro splnění zadaného cíle práce.

## **KLÍČOVÁ SLOVA**

robot, mapování, lokalizace, autonomní vozítko, snímání překážek

## **TITLE**

Mapping Enclosed Space

## **ANNOTATION**

The aim of this work is design and implementation of an autonomous rover that will move in an unknown environment, where it will scan the surrounding area and from acquired data will create 2D map of unknown environment. The theoretical part dedicates the possibilities of solving the rover. The practical part describes the realization of the rover and creation intelligence of the rover to fulfill specified target of this work.

## **KEYWORDS**

robot, mapping, localization, autonomous rover, sensing obstacles

# OBSAH

SEZNAM OBRÁZKŮ.....	9
SEZNAM TABULEK .....	10
SEZNAM ZKRATEK .....	11
ÚVOD.....	12
1    PODVOZEK VOZÍTKA .....	14
1.1    Tříkolové podvozky .....	14
1.2    Čtyřkolové podvozky .....	14
1.3    Pásové podvozky.....	15
2    POHONY.....	16
2.1    Stejnoseměrný motor .....	16
2.2    Krokový motor .....	17
2.3    Servomotor.....	18
2.3.1    Úprava servomotoru HS-311 .....	19
3    LOKALIZACE MR.....	21
4    SNÍMAČE PŘEKÁŽEK.....	22
4.1    Taktilní snímač.....	22
4.2    IR snímač .....	22
4.2.1    IR snímač GP2Y0A21YK0F .....	23
4.3    Ultrazvukový snímač .....	24
4.3.1    Ultrazvukový snímač HC-SR04 .....	24
5    MAPOVÁNÍ.....	26
5.1    Mřížková mapa.....	26
5.1.1    Třístavová mapa.....	26
5.1.2    Pravděpodobnostní mřížka .....	26
5.2    Mapovací algoritmy .....	27
6    ŘÍZENÍ MR.....	28

7	PRAKTICKÁ ČÁST .....	29
7.1	Návrh vozítka .....	29
7.2	Návrh desek plošných spojů (DPS).....	29
7.3	Výsledné vozítko.....	30
7.4	Řídicí aplikace v MCU.....	32
7.5	Řídicí aplikace v PC.....	33
7.5.1	Generování mapy .....	34
7.5.2	Model vozítka .....	36
7.5.3	Pohyb vozítka .....	37
7.5.4	Snímání překážek.....	40
7.5.5	Metody mapování vozítka .....	42
7.6	Testovací mapování a jeho zhodnocení .....	43
7.7	Možné zlepšení snímání změny pozice vozítka .....	45
	ZÁVĚR .....	46
	POUŽITÁ LITERATURA .....	47
	SEZNAM PŘÍLOH.....	49



## SEZNAM OBRÁZKŮ

Obrázek 1 - Tříkolové podvozky a) Ackermanovo řízení, b) diferenční řízení .....	14
Obrázek 2 - Čtyřkolový podvozek (Ackermanovo řízení) .....	15
Obrázek 3 – Příklad stejnosměrných motorků.....	16
Obrázek 4 - Princip funkce stejnosměrného motoru (1).....	17
Obrázek 5 - Vnitřní uspořádání krokového motoru (1) .....	18
Obrázek 6 - Servomotor a) GO-09 (vlevo), b) HS-311 (vpravo) .....	19
Obrázek 7 - Převodový systém servomotoru HS-311 a) úplné převody (vlevo), b) pohled na snímací potenciometr (vpravo dole) .....	19
Obrázek 8 - Úprava převodového kola a) původní převod (vlevo), b) upravený převod (vpravo).....	20
Obrázek 9 - IR snímač GP2Y0A21YK0F (8).....	23
Obrázek 10 - Závislost výstupního napětí na vzdálenosti překážky (8).....	23
Obrázek 11 - Měřicí cyklus (8).....	24
Obrázek 12 - Ultrazvukový senzor HC-SR04 .....	25
Obrázek 13 - Měřicí cyklus snímače HC-SR04 (8) .....	25
Obrázek 14 - Konstrukce vozítka .....	29
Obrázek 15 - Hlavní DPS .....	30
Obrázek 16 - Sestavené vozítko.....	31
Obrázek 17 - Základní kóty vozítka .....	31
Obrázek 18 - Podoba aplikace na PC.....	34
Obrázek 19 - Detail na identifikaci překážek v hranici vozítka .....	40
Obrázek 20 - Detail na princip určování překážky .....	41
Obrázek 21 - Testovací mapování 1 .....	43
Obrázek 22 - Testovací mapování 2 .....	44

## **SEZNAM TABULEK**

Tabulka 1 - Hlavní příkazy .....	32
Tabulka 2 - Vedlejší příkazy.....	32
Tabulka 3 - Označení a význam údajů reprezentující stav dílku .....	35

## **SEZNAM ZKRATEK**

BT	bluetooth
IR	infračervené
MCU	jednočipový počítač
MR	mobilní robot
PC	počítač
PWM	pulsně šířková modulace

## ÚVOD

Schopnost získávat aktuální informace o stavu okolního prostředí autonomních systémů je důležitá pro pohyb vozítka v tomto prostředí případně pro vlastní lokalizaci vůči referenční mapě. Vytvářet si aktuální lokální mapu nejbližšího prostředí vozítka je důležité pro zabránění kolize s objekty ať statickými nebo dynamickými. Dalším důvodem je vlastní lokalizace vozítka, ať už hledání své polohy vůči referenční mapě nebo k vlastní korekci chyby polohy vozítka vůči počátečním souřadnicím.

Tato práce si klade za cíl vytvořit autonomní vozítko, které bude schopné pohybovat se v neznámém prostředí a pomocí vlastních snímačů snímat stav okolního prostředí. Dále pak získaná data vyhodnotit a vytvořit z nich 2D mapu. Požadavek je kladen na kontinuální zobrazování mapy s vlastním modelem vozítka, který by graficky reprezentoval aktuální pozici vozítka od počátečního bodu mapy. V práci je věnována velká pozornost řešení problematiky zamezení kolize vozítka s překážkou.

Prostor, ve kterém se výsledné vozítko bude pohybovat, je omezen na rovnoběžný hladký povrch tvořený z pevného materiálu. Dále se předpokládá, že mapovaný prostor neobsahuje žádné pohyblivé objekty. Jedná se tedy o statický prostor.

Práce je členěna na dvě části. První část se věnuje teoretickému popisu možného řešení dané problematiky. Praktická část se věnuje řešení vozítka a jeho oživení.

První kapitola popisuje základní typy podvozků, které by byly vhodné pro danou aplikaci s ohledem na možnost manévrovatelnosti podvozku v prostoru. V další části jsou vysvětleny principy běžně používaných typů pohonů pro mobilní robotiku. Pozornost je zde hlavně věnována servomotoru a jeho možné aplikaci jako pohonu vozítka. Ve třetí kapitole jsou popsány možnosti zaznamenávání změny polohy vozítka vůči počátečnímu stavu. Čtvrtá kapitola se věnuje popisu tří snímačů vhodných pro snímání okolí vozítka. Jsou zde popsány jejich principy a vlastnosti. U dvou typů je uveden vždy jeden konkrétní příklad snímače a popsány jeho parametry a tvar výstupních dat. V páté kapitole je uveden popis způsobu vytváření mapy na principu čtvercové sítě a stručný popis mapovacích algoritmů. V šesté kapitole je krátký popis možnosti řešení problematiky ovládní, zpracování a grafické interpretace při procesu mapování.

V praktické části práce je nejprve popsán výběr a konstrukce vozítka, výběr pohonů, snímacího zařízení a způsobu komunikace s počítačem. Další část popisuje strukturu desky

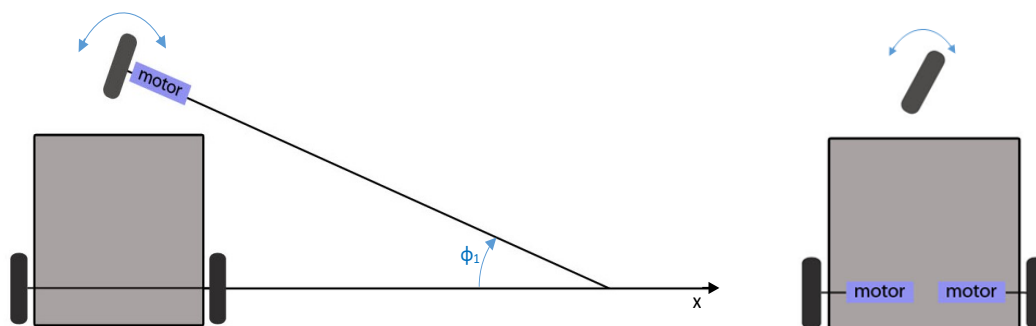
plošných spojů vozítka. Třetí podkapitola reprezentuje výsledné mapovací vozítko. Dále je popsána řídicí aplikace vozítka v mikroprocesoru. Pátá podkapitola je nejširší částí práce. Je zde popsán způsob generování mapy, tvorba modelu vozítka, řešení pohybu vozítka po mapě, metoda snímání překážek a nakonec dva způsoby mapování prostoru. Poslední podkapitola ukazuje výsledky testovacího mapování se zhodnocením získané mapy.

# 1 PODVOZEK VOZÍTKA

Výběr vhodného podvozku se volí podle typu terénu, ve kterém se bude vozítko pohybovat. V následujících podkapitolách jsou popsány základní druhy podvozků vhodné pro pohyb po rovinných plochách z pevného materiálu.

## 1.1 Tříkolové podvozky

Tento typ podvozku se dělí podle typu řízení na Ackermanovo a diferenciální. Ackermanovo řízení, je znázorněno na obrázku 1 - a). Tento podvozek obsahuje přední hnací otočné kolo a zadní hnanou nápravu fixně upevněnou<sup>1</sup>. Natočením řídicího kola začne vozítko opisovat kružnici. Nevýhodou je nemožnost otáčet se kolem své osy. Pro pohyb ve stísněných prostorech by byl tento podvozek značně limitován.



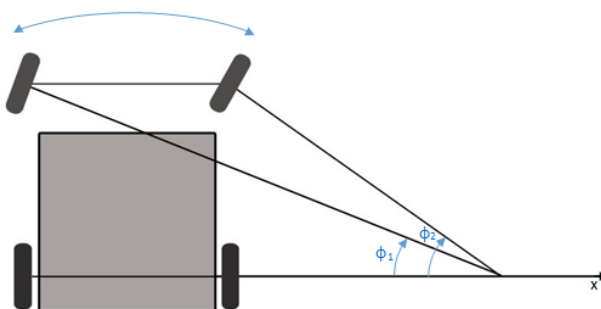
Obrázek 1 - Tříkolové podvozky a) Ackermanovo řízení, b) diferenční řízení

Na obrázku 1 – b) je znázorněná druhá varianta a to diferenční řízení. Zde je přední kolo volně otočné a zadní náprava obsahuje dvě samostatně řízená kola. Různou rychlostí a směrem otáčení obou kol se docílí požadovaného pohybu a natočení. Největší předností tohoto řízení je schopnost otáčet se kolem své osy. (1)

## 1.2 Čtyřkolové podvozky

Tyto podvozky obsahují jednu natáčecí hnací a druhou pasivní hnanou nápravu. Opět tento typ funguje na principu Ackermanova řízení. V případě zatáčení začne podvozek opisovat kružnici. Vnitřní otočné kolo však opisuje kružnici o menším poloměru než vnější, viz obrázek 2. Proto je nutné, aby bylo vnitřní kolo natočeno o větší úhel. Nevýhodou je stejná limitace schopnosti otočení jako u tříkolového podvozku s Ackermannovým řízením.

<sup>1</sup> Možná je i varianta se zadní hnací nápravou (s diferenciálem) a hnaným předním řídicím kolem.



Obrázek 2 - Čtyřkolový podvozek (Ackermanovo řízení)

### 1.3 Pásové podvozky

Pásové podvozky se vyznačují dobrou schopností zdolávat nerovné terény. Řízení funguje na stejném principu jako diferenční řízení. Oproti kolovým podvozkům mají velkou styčnou plochu s povrchem, která způsobuje větší tření a to vede k vyšší energetické náročnosti na pohony. Řízení probíhá pomocí smyku, který může způsobit velkou odchylku předpokládaného pohybu ke skutečnému pohybu. (2)

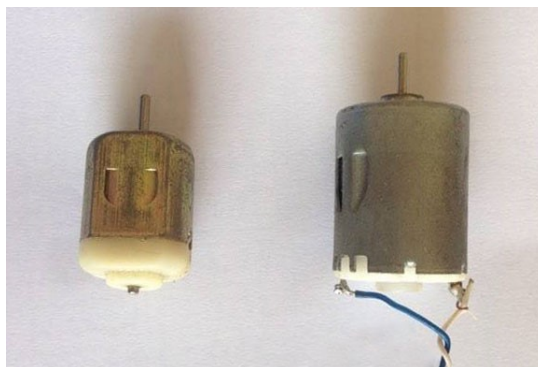
## 2 POHONY

Pohony mobilních robotů (MR) jsou nejčastěji realizovány pomocí rotačních elektromotorů doplněných převodovým systémem. Dále pak krokové motory (KM), u kterých není potřeba převodovka a je možné je přesně polohovat. Jako třetí pohon je modelářský servomotor, který obsahuje zabudovanou převodovku. (1) Ovládání se provádí digitálně pomocí pulsně šířkové modulace (PWM). V základním provedení servomotory mají omezený krouticí moment, který vylučuje jeho použití jako pohonné jednotky. Menším mechanickým zásahem lze tento pohon uzpůsobit k plně rotačnímu pohybu. V kapitole 2.3.1 bude detailně popsána úprava na servu HS-311.

Následující podkapitoly se věnují popisu základních principů výše zmíněných pohonů.

### 2.1 Stejnosměrný motor

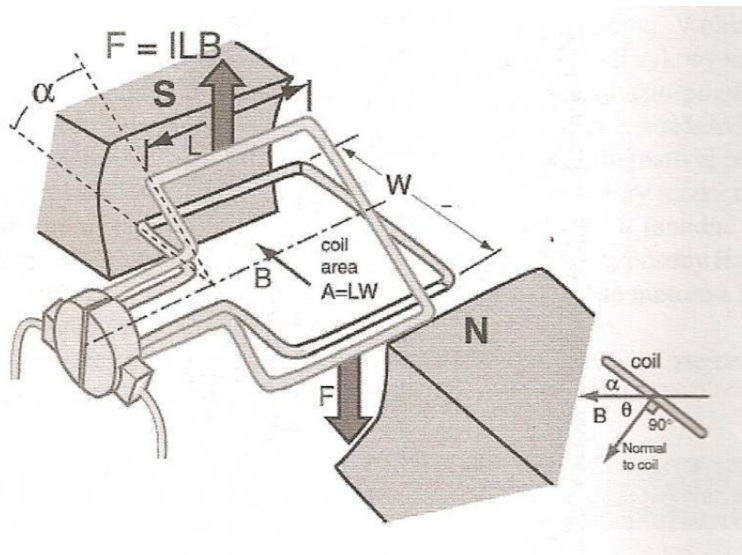
Stejnosměrný motor se skládá z pasivní části (statoru), který obsahuje permanentní magnet. Aktivní, neboli otočnou částí je rotor. Ten se skládá z hřídele a cívky. Příklady stejnosměrných motorů jsou uvedeny na obrázku 3. Vstupní stejnosměrné napětí je na cívku přivedeno pomocí kartáčového držáku.



Obrázek 3 – Příklad stejnosměrných motorků

Protékající proud cívkou rotoru vytvoří ve svém okolí magnetické pole. Pomocí Ampérova pravidla pravé ruky lze určit směr indukčních čar tohoto pole. Na obrázku 4 je znázorněn případ, kdy proud vtéká do vodiče na levé straně. Siločáry permanentního magnetu směřují ze severní (N) strany do jižní (S). Obě tyto pole mezi sebou interagují a v závislosti na směru intenzit se pole v daném místě zesílí nebo zeslabí. Šipky znázorňují směr výsledné Lorentzovy síly působící na vodič. Směr této síly je také směr nejmenší intenzity magnetického pole v daném místě. Velikost Lorentzovy síly lze vypočítat pomocí rovnice 1.





Obrázek 4 - Princip funkce stejnosměrného motoru (1)

$$F = B \cdot I \cdot L \quad (1)$$

kde,

$F$ ... Lorentzova síla [N]

$B$ ... magnetická indukce

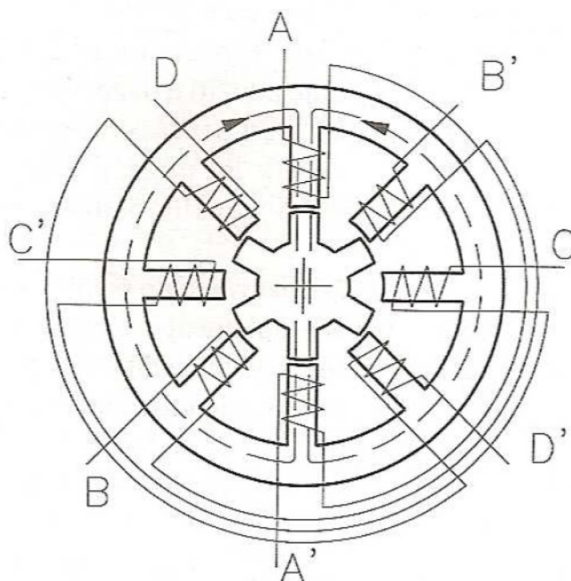
$I$ ... proud protékající vodičem

$L$ ... délka vodiče

Tento motor se vyznačuje jednoduchým řízením. Směr otáčení se změní prohozením polarity vstupního napětí. Rychlost lze regulovat pomocí pulzního řízení. Nevýhodou může být použití u aplikací s požadavkem na přesnost pohybu pohonu, kde je tento nedostatek řešen externími snímači otáček. Příkladem je použití enkodéru.

## 2.2 Krokový motor

Krokové motory obsahují ve statoru několik párů cívek umístěných na kovových výběžcích, vždy proti sobě polohovaných a vzájemně propojených. Každá tato dvojice udává jednu fázi. Rotor obsahuje kovové výběžky o stejné šířce jako stator. (1) Na obrázku 5 je uvedeno principiální schéma. Písmeny A – D jsou označovány jednotlivé fáze. Postupným spínáním cívek se docílí otočného momentu rotoru. Krok rotoru je dán počtem fází statoru, počtem zubů rotoru a způsobem řízení.



**Obrázek 5 - Vnitřní uspořádání krokového motoru (1)**

Tento typ motoru již z principu vyžaduje složitější řízení. Výhodou je přesná regulace natočení motoru.

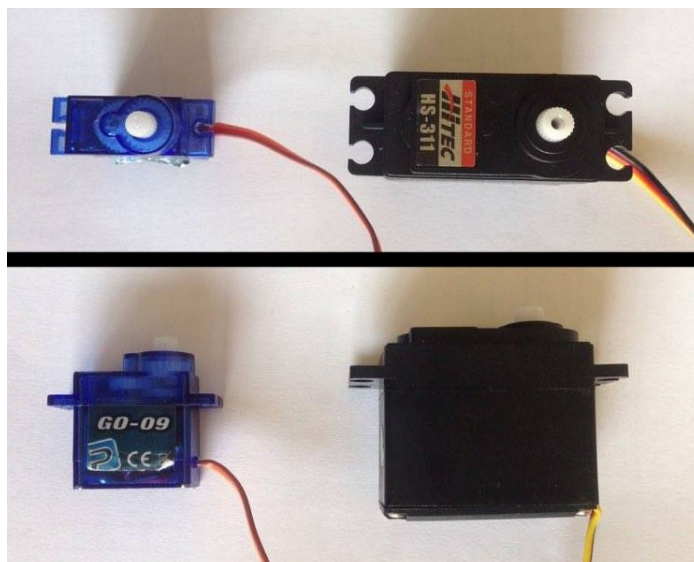
### 2.3 Servomotor

Modelářské servomotory jsou určeny k úhlovému natáčení. Typický rozsah je  $180^\circ$ . Ovládání probíhá pomocí PWM modulace o kmitočtu  $50 \text{ Hz}^2$  a šířkou pulsů od 0,9 - 2,1 ms. (3) Šířka pulsu udává úhel natočení servomotoru. Ukázka dvou typů servomotorů je na obrázku 6.

Servomotor se skládá ze stejnosměrného motoru, převodového systému, snímače natočení a řídicí elektroniky. Jako snímač natočení se využívá potenciometr, který je zabudován do převodového systému.

Principem funkce je ve snaze docílení natočení převodového systému a s ním natočení snímacího potenciometru na požadovanou hodnotu odporu. Šířka vstupních PWM pulsů je úměrná určité hodnotě odporu snímače. Pokud dojde k nastavení žádané hodnoty odporu, motor se zastaví.

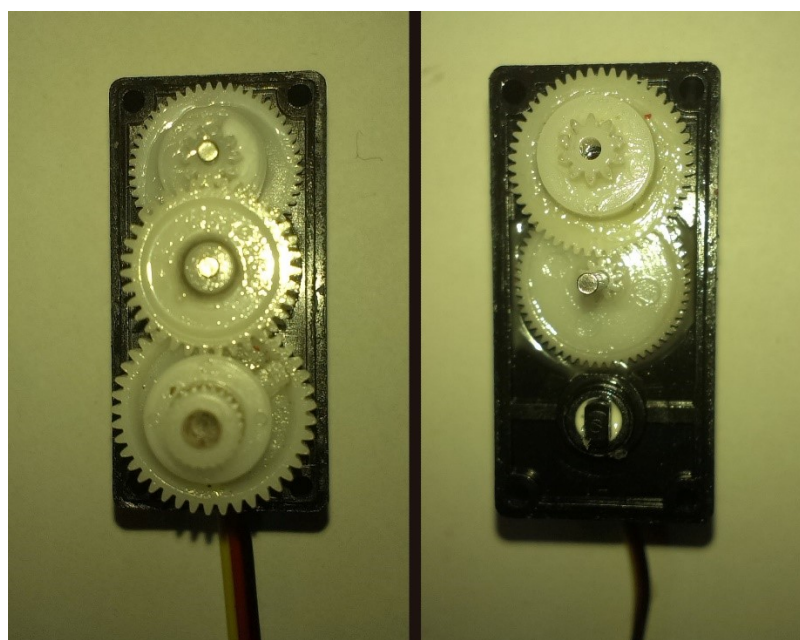
<sup>2</sup> Modelářské servomotory dokáží i pracovat na kmitočtech blízkých 50 Hz.



Obrázek 6 - Servomotor a) GO-09 (vlevo), b) HS-311 (vpravo)

### 2.3.1 Úprava servomotoru HS-311

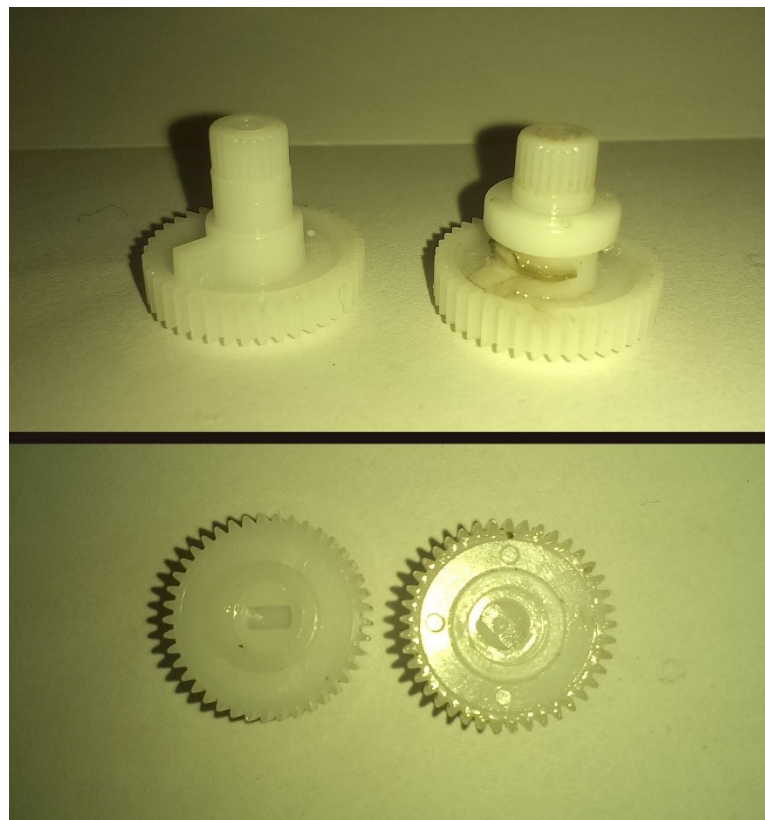
Na obrázku 7 - a) je zachycen pohled na celou převodovou soustavu servomotoru. Spodní převodové kolo je kontaktně připojeno ke snímacímu potenciometru, který je umístěn pod ním. Pohled na potenciometr je na obrázku 7 - b). Tento převod je opatřen zarážkou, která umožňuje pohyb hřídele ve vymezeném prostoru a tím tedy chrání potenciometr před jeho mechanickým poškozením.



Obrázek 7 - Převodový systém servomotoru HS-311 a) úplné převody (vlevo), b) pohled na snímací potenciometr (vpravo dole)

Úprava servomotoru na plně rotační pohyb spočívá v odstranění dané zarážky omezující otáčivý úhel (obrázek 8, nahoře). Dalším krokem je odpojení snímacího potenciometru. Před odpojením je však nutné nastavit pomocí PWM výchozí hodnotu snímacího potenciometru. Tato hodnota odporu bude při řízení reprezentovat zastavený servomotor.

Odpojení snímacího zařízení je možné dvěma způsoby. Prvním je nahrazení potenciometru pevným odporem. Velikost odporu musí být stejná jako nastavená výchozí hodnota potenciometru. Druhou variantou je zvětšení drážky převodu tak, aby nedocházelo ke kontaktu se snímačem. Ve spodní části obrázku 8 je vyobrazena úprava převodu.



**Obrázek 8 - Úprava převodového kola a) původní převod (vlevo), b) upravený převod (vpravo)**

U druhé varianty je žádoucí zakápnout potenciometr lepidlem, aby vlivem vibrací nedocházelo k jeho samovolnému pootočení, které by znamenalo nutnost jiné šířky pulsu řídicího signálu, aby došlo k zastavení servomotoru.

Přivedením pulsů kratších než je výchozí nastavená šířka se servomotor bude otáčet na jednu stranu. Širší impulsy změni orientaci rotace.

### 3 LOKALIZACE MR

Při aplikacích, kde se MR pohybují ve známém nebo neznámém prostředí, je přesné určování polohy MR jedna z nejdůležitějších podmínek.

Elementární určování změny polohy MR je ze známé rychlosti a směru pohybu. Na základě doby trvání pohybu lze určit ujetou dráhu. Vhodnější variantou je použití odometrie, která zpřesňuje předchozí měření díky informaci o skutečném pohybu kol.

Odometrie je proces určování polohy za pomoci měření pohybu MR. Měření se nejčastěji provádí pomocí enkodérů, které jsou připevněny na kolech. Známým geometrickým uspořádáním konstrukce vozítka lze vypočítat změnu pozice od počáteční polohy.

Při každém určování polohy vzniká odchylka od skutečné polohy a polohy naměřené. Tyto chyby se postupně akumulují. Chyby mohou být systematické nebo nesystematické. Systematická chyba vzniká nepřesností snímače otáček nebo odchylkou reálných parametrů vozítka od modelu. Tato chyba se vyskytuje při každém měření. A proto ji lze úpravou algoritmu značně eliminovat. Nesystematická chyba obvykle bývá větší než systematická a má náhodný charakter výskytu. Nejčastěji se jedná o smyk kol. K odstranění této chyby je zapotřebí použití více měřících systémů. (4) (5)

Obě zmíněné metody jsou vhodné pouze pro určování polohy na malé vzdálenosti, kde případná chyba nemá větší charakter.

Třetí variantou je využití akcelerometru pro určení pohybu a gyroskopu, který zajišťuje informaci o úhlu natočení. Použití těchto snímačů je odolnější proti chybám. Na druhou stranu si kladou vyšší nároky na zpracování dat ze snímačů.

Využitím známé mapy okolního prostředí je možné za pomoci snímačů korigovat chybu lokalizace a tím snižovat vzniklou chybu.

Pro odstranění chyby lokalizace existují různé výpočetní metody (6):

- Kalmanův filtr
- Monte Carlo Lokalizace
- Markovova Lokalizace

## 4 SNÍMAČE PŘEKÁŽEK

Při pohybu MR v neznámém prostředí je třeba získávat informace o okolním prostředí. V této kapitole jsou popsány základní snímače pro snímání překážek. Popsány jsou základní cenově dostupné snímače, které nekladou vysoké nároky na zpracování údajů ze snímačů. U infračervených (IR) a ultrazvukových snímačů jsou uvedeny konkrétní příklady a jejich základní popis činnosti.

### 4.1 Taktilní snímač

Nejjednodušší formou detekce překážky, a to bez ohledu na typ materiálu nebo úhlu dané překážky je použití spínače jako snímače překážky. Vhodným umístěním na vozítko vznikne spolehlivá detekce překážek. Snímač udává informaci ve dvoustavové logice.

Nevýhodou tohoto typu snímání je, jak z principu vyplývá, nutnost kontaktního snímání. To může být nevýhoda při nedostatečně rychlé reakční době vozítka na signál o detekci překážky (například doba nutná k úplnému zastavení vozítka) a tím může dojít k přesmykům kol. Tento přesmyk by mohl vést k nemalé chybě lokalizačního algoritmu vozítka vůči fyzické pozici.

### 4.2 IR snímač

Infračervené snímače pracují na principu vyzařování světelného paprsku v infračerveném spektru. Infračervené spektrum je umístěno hned za viditelným spektrem. Jedná se o vlnové délky od 750  $\mu\text{m}$  do 1000  $\mu\text{m}$ . (7) Snímač je opatřen IR vysílačem a IR přijímačem. Vyhodnocení vzdálenosti probíhá z výpočtu doby vyslání světelného paprsku a doby příjmu na přijímači. Vzorec pro výpočet vzdálenosti snímače od překážky je uveden v rovnici 2.

$$s = \frac{v \cdot t}{2} \quad (2)$$

kde,

$s$ ... vzdálenost snímače od překážky [m]

$v$ ... rychlost šíření světla ( $v = 3 \cdot 10^8 \text{ m} \cdot \text{s}^{-1}$ )

$t$ ... doba mezi vysláním paprsku a příjmem

Tento princip není závislý na typu prostředí. Úzký měřicí paprsek umožňuje měřit malé plochy. Nevýhodou je procházení paprsku skrz průhledné materiály a závislost na barvě snímané překážky, to vede k chybě měření. Správnost měření závisí na úhlu dopadu paprsku na snímaný objekt.

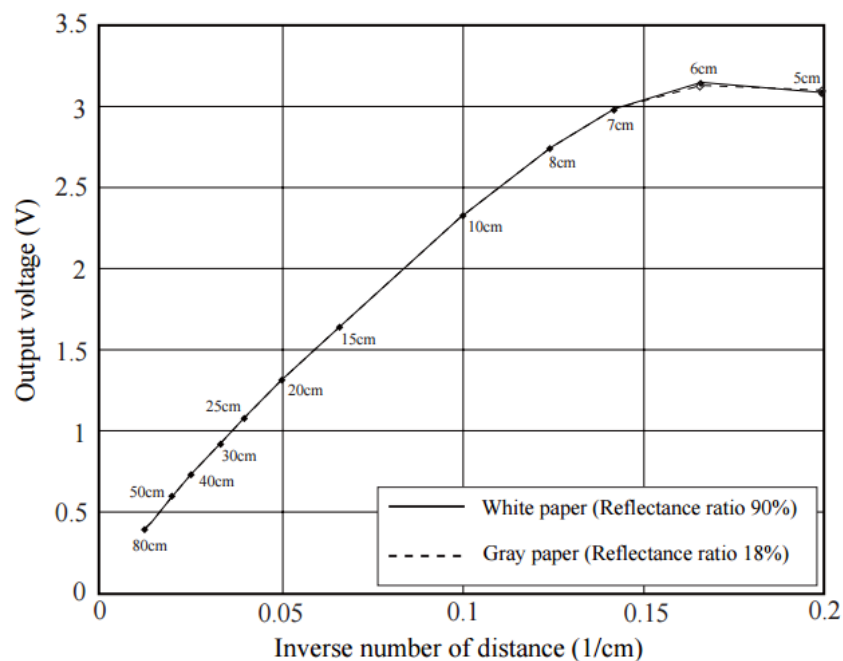
#### 4.2.1 IR snímač GP2Y0A21YK0F

Jako příklad je uveden senzor GP2Y0A21YK0F, viz obrázek 9, který pracuje v rozsahu od 10 do 80 cm.



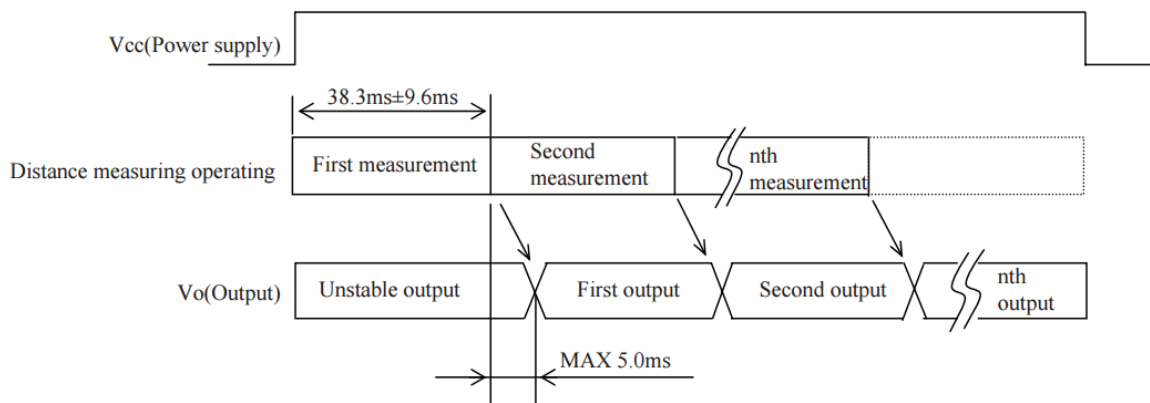
Obrázek 9 - IR snímač GP2Y0A21YK0F (8)

Údaj o naměřené vzdálenosti je na výstupu snímače poskytnut analogovým napětím o rozsahu od 0 do 3,2 V. Nelineární závislost výstupního napětí na vzdálenosti si vyžaduje složitější zpracování signálu. Průběh výstupního napětí je znázorněn na obrázku 10.



Obrázek 10 - Závislost výstupního napětí na vzdálenosti překážky (8)

Tento snímač měří kontinuálně. Snímač pracuje v měřicích cyklech přibližně 40 ms. Po přivedení napájení proběhne první měření, na výstupu po tuto dobu bude nestabilní stav. V dalších cyklech je již na výstupu údaj o vzdálenosti z předchozího cyklu. (8) Průběh měření je znázorněn na obrázku 11.



Obrázek 11 - Měřící cyklus (8)

### 4.3 Ultrazvukový snímač

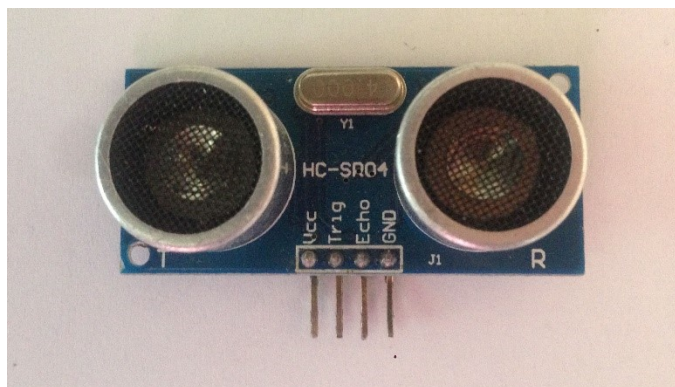
Ultrazvukové snímače pracují na podobném principu jako IR snímače. Rozdílem je použití zvuku jako měřící veličiny. Snímač pracuje na kmitočtech nad 20 kHz. Tedy nad slyšitelným rozsahem člověkem. Princip činnosti je vyslání několika vysokofrekvenčních krátkých impulsů za sebou a z doby přijetí odraženého signálu na přijímači, vypočítá vzdálenost senzoru od překážky. Výpočet vzdálenosti překážky je stejný jako v rovnici 2. Místo rychlosti světla se dosadí rychlost vzduchu, která je v suchém prostředí přibližně  $340 \text{ m} \cdot \text{s}^{-1}$ .

U této metody odpadá závislost nezaznamenání průhledné překážky a závislost na barvě. Oproti světelnému principu je zde větší vyzařovací úhel, který vede k měření větší plochy. U materiálu pohlcující zvuk dojde k chybě měření. Stejně jako u IR snímačů zde hraje roli úhel dopadu zvukového vlnění na překážku. Nevýhodou je, že rychlost zvuku není konstantní.

#### 4.3.1 Ultrazvukový snímač HC-SR04

Na obrázku 12 je zachycena podoba snímače. Pracovní rozsah tohoto senzoru je od 2 do 400 cm. Udávaná přesnost je 3 mm. Obrázek 13 znázorňuje pracovní úrovně pinů při měření. Snímač po přivedení  $10 \mu\text{s}$  impulsu na Trigger pin vyšle 8 ultrazvukových impulsů na kmitočet 40 kHz. Doba šíření signálu od vysílače k přijímači je doba trvání impulsu na Echo pinu. (9)





Obrázek 12 - Ultrazvukový senzor HC-SR04

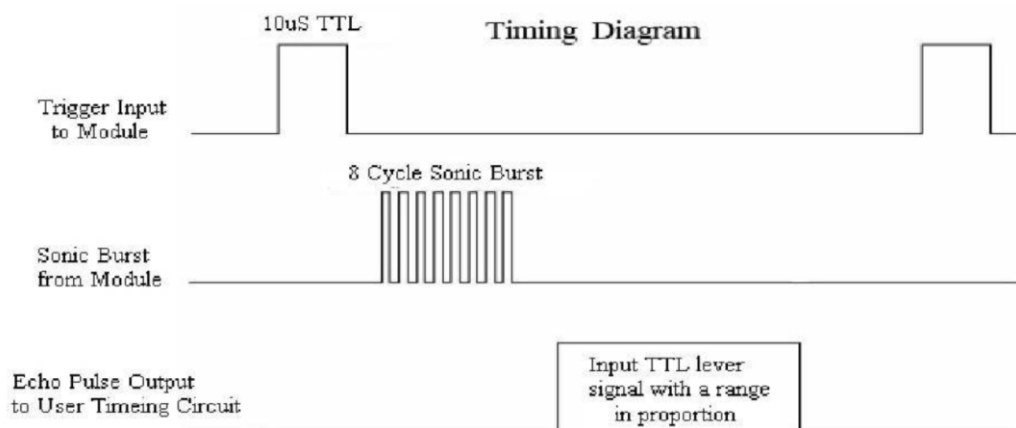
$$s = \frac{t_{high} \cdot v_{sound}}{2} \quad (3)$$

kde,

$s$ ... vzdálenost snímače od překážky [m]

$t_{high}$ ... doba trvání pulsu na Echo pinu

$v_{sound}$ ... rychlost šíření zvuku ( $340 \text{ m} \cdot \text{s}^{-1}$ )



Obrázek 13 - Měřící cyklus snímače HC-SR04 (9)

Z důvodu zamezení odrazům je doporučená doba jednoho měřícího cyklu podle (9) minimálně 60 ms.

## 5 MAPOVÁNÍ

Cílem této práce je grafická reprezentace vytvořené mapy neznámého prostředí. Pro autonomní MR je důležité vytvářet si informaci o stavu prozkoumávaného prostoru. Tato mapa může být například zaslána dalším vozítkům, která se díky známé mapě budou moci pohybovat rychleji a efektivněji. Dalším využitím mapy okolního prostředí je minimalizace chyby lokalizace MR.

V této práci je uvažována pouze mřížková mapa, která je pro mapování malých prostorů vhodná.

### 5.1 Mřížková mapa

Mřížková mapa je založena na principu čtvercové sítě o určité velikosti jednotlivých čtverců. Jejich velikost udává rozlišení výsledné mapy. Velikost je závislá také na rozlišení snímacího zařízení. Další omezení je v náročnosti na paměť. Ze známé polohy vozítka na mapě a známého vektoru snímaného prostoru se nalezne odpovídající čtvereček na mapě a přiřadí se mu příslušná hodnota, která závisí na použité metodě vytváření mapy.

#### 5.1.1 Třístavová mapa

Základní vytváření mapy je pomocí třístavového přiřazení. Na počátku je všem čtvercům přiřazeno číslo reprezentující nezmapovaný prostor. Příkladem může být číslo „0“. Postupným mapováním se jednotlivým čtverečkům přiřadí číslo odpovídající stavu „je překážka“, „není překážka“.

#### 5.1.2 Pravděpodobnostní mřížka

Předchozí způsob nezapočítává nepřesnost měřícího zařízení a proto je vhodnější variantou pravděpodobnostní mřížka. Na počátku je čtvercům přiřazena hodnota reprezentující neutrální pravděpodobnost výskytu překážky či volného prostoru. Postupným snímáním mřížky se u jednotlivých bodů zvyšuje nebo snižuje číslo podle toho, zda v daném bodě je zaznamenána překážka, či ne. Několikanásobným snímáním stejných bodů se zvyšuje pravděpodobnost výskytu překážky v daném prostoru. Výsledná mapa je tedy věrnější než v předchozím případě. (10)

## 5.2 Mapovací algoritmy

Mapovací algoritmy slouží k popsání celého neznámého prostoru o definovaných rozměrech.

Základem mapovacích algoritmů jsou algoritmy hledání cesty v neznámém prostředí. Tyto algoritmy vychází ze znalosti výchozí polohy vozítka a polohy cílového bodu. Mezi elementární algoritmy patří:

- Bug1
- Bug2

Nejprve se vytvoří přímka od souřadnice vozítka k souřadnici cílového bodu. Algoritmus Bug1 se snaží sledovat tuto přímku. V případě, že se objeví překážka, tak se jí pokusí objet kolem do kola k původní pozici. Pokud se v průběhu objíždění překážky vozítko střetne s přímkou určující primární trajektorii, tak si tento bod vyznačí. Při návratu do původního bodu porovná ujeté dráhy z každého směru a vydá se ve směru kratší trajektorie.

Bug2 algoritmus provádí stejné úkony jako Bug1 s tím rozdílem, že pokud při objíždění překážky narazí na přímkou primární trajektorie, tak ukončí objíždění překážky a vydá se po vyznačené dráze. (11)

Principem mapování je, že se po celé mapě rozmístí tzv. **referenční body** (cílové body při hledání cesty). Na souřadnice těchto bodů se postupně MR přemístí a vytvoří informaci o stavu okolí v daném bodě. Po projetí všech bodů vznikne kompletní mapa prostoru.

Rozmístění těchto bodů čistě závisí na parametrech snímačů vozítka. Pokud tedy budou čtyři body tvořit čtvercové rozmístění, tak celkový scan MR ze všech čtyř bodů by měl vyplnit celý vnitřní prostor mezi body.

Po vytvoření referenčních bodů na mapě je potřeba zvolit postup sestavování mapy a vyhodnocování mezních stavů s ním spojeným. Způsob řešení těchto kroků závisí čistě na konkrétní aplikaci.

V případě, že na mapě vzniknou nezmapované úseky, tak po identifikaci stavu překážek v okolí bodů se určí, zda je tyto prostory možné zmapovat. Příkladem může být bod ohraničený ze všech stran překážkou. V těchto případech se předpokládá, že ohraničený prostor je vyplněn překážkou. Pokud není prostor plně ohraničen, tak se naplánuje cesta vozítka do dané lokality. Pohybuje-li se vozítko v již známé mapě, tak lze použít pro plánování cesty algoritmy aplikovatelné pro známé prostředí.

## 6 ŘÍZENÍ MR

Řízení pohybového aparátu a zpracovávání dat ze snímačů je třeba realizovat pomocí vyhodnocovací a řídicí jednotky. Jako ideální možnost se jeví jednočipový mikropočítač (MCU), který obsahuje dostatek periférií, které se dají použít k řízení a zpracovávání akčních a snímacích komponent MR.

MR je tedy možné řídit pouze pomocí MCU. V aplikacích náročných na vysoký objem dat a jejich vyhodnocování je použití MCU nedostatečné. Dalším omezením je ovládání a případný způsob zobrazování, který by se musel řešit dodatečnými komponentami. Pro tento druh aplikací je vhodnější využít kombinace MCU a počítače (PC), kde je možné vytvořit interaktivní a uživatelsky příjemnou aplikaci pro ovládání a případnou grafickou interpretaci zpracovaných dat. Dále pak výpočetní výkon PC je o několik řádů vyšší než klasické jednočipové procesory.

Přenosový řetězec se musí volit s ohledem na typ a množství přenášených dat. Jednou z možností je pomocí přímé pevné sítě, kde je použito vodiče pro komunikaci. Příkladem může být sériová komunikace. Další možností je využití bezdrátové komunikace, kde odpadá závislost omezení volnosti kvůli kabelu. Příkladem je komunikační rozhraní wifi nebo bluetooth (BT).

Výběr daného typu a druhu přenosu dat se musí volit s ohledem na požadovanou rychlost přenosu, množství potřebných přenášených dat a vzdálenost.

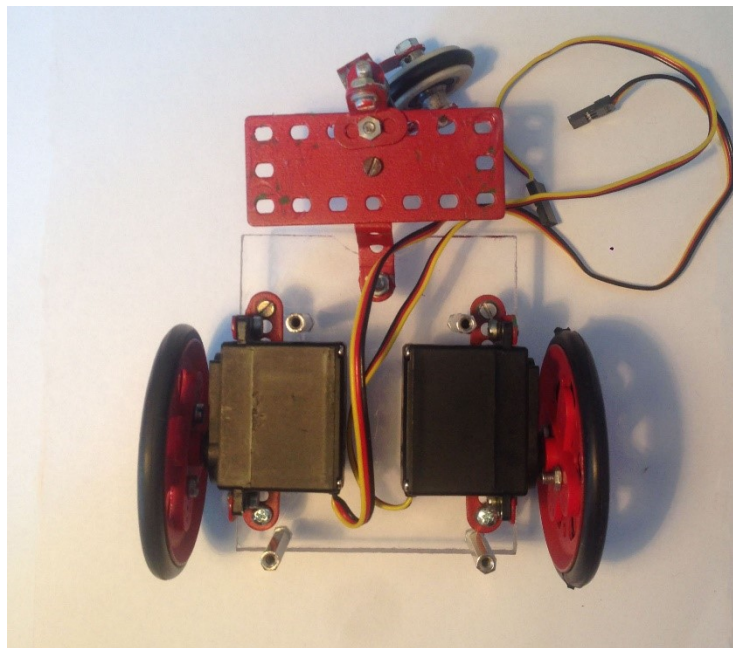
V praktické části této práce je zvolena volba kombinace MCU a PC pomocí bezdrátové přenosové cesty. Předpokládá se vzdálenost vozítka od PC řádově do několika metrů s přímou viditelností. Těmto zadaným parametrům vyhovuje využití komunikace pomocí BT.

Jako vhodný nástroj pro aplikaci na PC se jeví prostředí MATLAB, které je určeno pro zpracování dat a jejich reprezentaci. V praktické části je výsledná aplikace vytvořena právě v tomto prostředí. Alternativní možností je tvorba aplikace v programovacím prostředí Microsoft Visual Studio a jazyku C#.

## 7 PRAKTICKÁ ČÁST

### 7.1 Návrh vozítka

Konstrukce vozítka byla zvolena jako tříkolové s diferenčním řízením, viz obrázek 14. Pohonnou jednotkou jsou upravené modelářské servomotory HS-311. Způsob úpravy těchto pohonů je popsán v kapitole 2.3.1. Při použití upravených servomotorů jako pohonů je třeba brát v úvahu, že jejich rychlost otáčení pro různé úhly není lineární. Proto je nutné při úpravě nastavit odpory obou potenciometrů na stejné hodnoty, aby nedocházelo k nestejnomyšlnému pohybu kol při stejných řídicích impulsích.



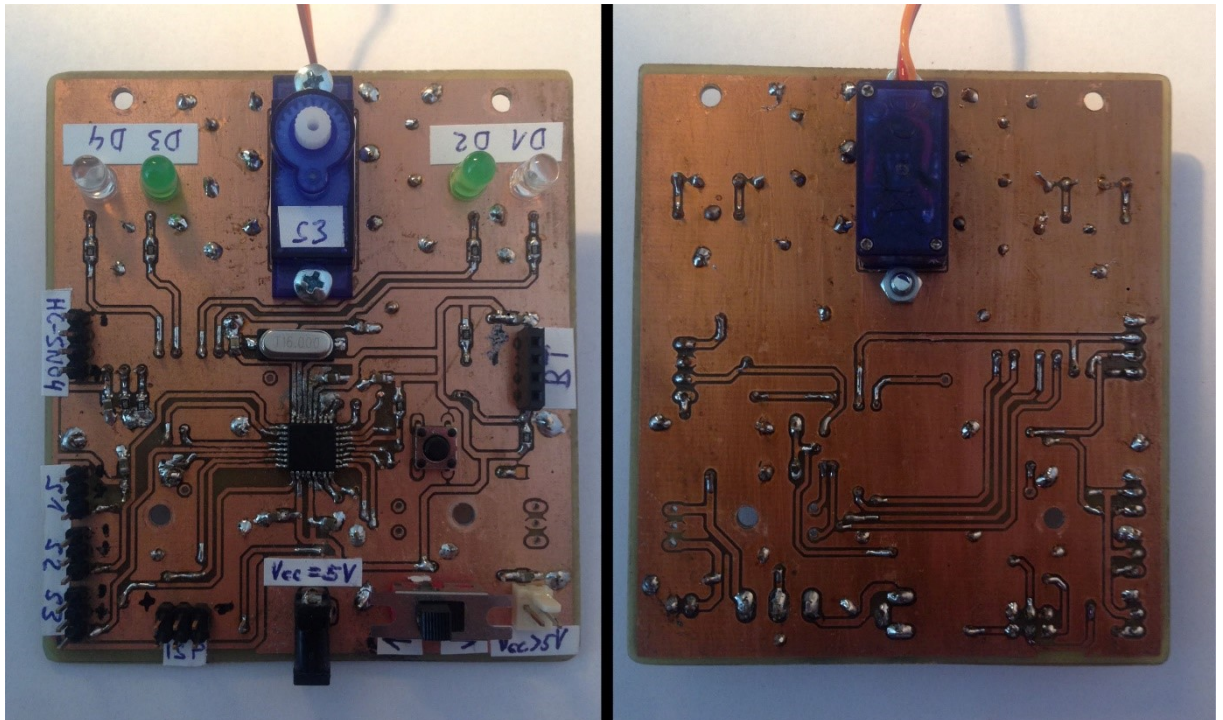
Obrázek 14 - Konstrukce vozítka

Jako snímací zařízení je zvolen ultrazvukový senzor HC-SR04. Při snímání je zde kladen požadavek na široký úhel snímání. Ten je dosažen umístěním snímače na servomotor, který natáčí snímačem v horizontální ose o požadovaném stupni jemnosti.

Ke komunikaci s PC je zvolen BT modul JY-MCU, který komunikuje s MCU pomocí sériové linky. Pro zpracování příkazů je použit MCU ATmega328.

### 7.2 Návrh desek plošných spojů (DPS)

Vozítka se skládá ze dvou DPS. První deska je hlavní řídicí, viz obrázek 15, která napájí všechny komponenty vozítka a propojuje je s MCU. Výrobní data jsou uvedena v příloze A. Druhá deska slouží pro snímač. Obsahuje dva konektory, které slouží k uchycení snímače a pro přívod kabelu. Výrobní data jsou uvedena v příloze B.

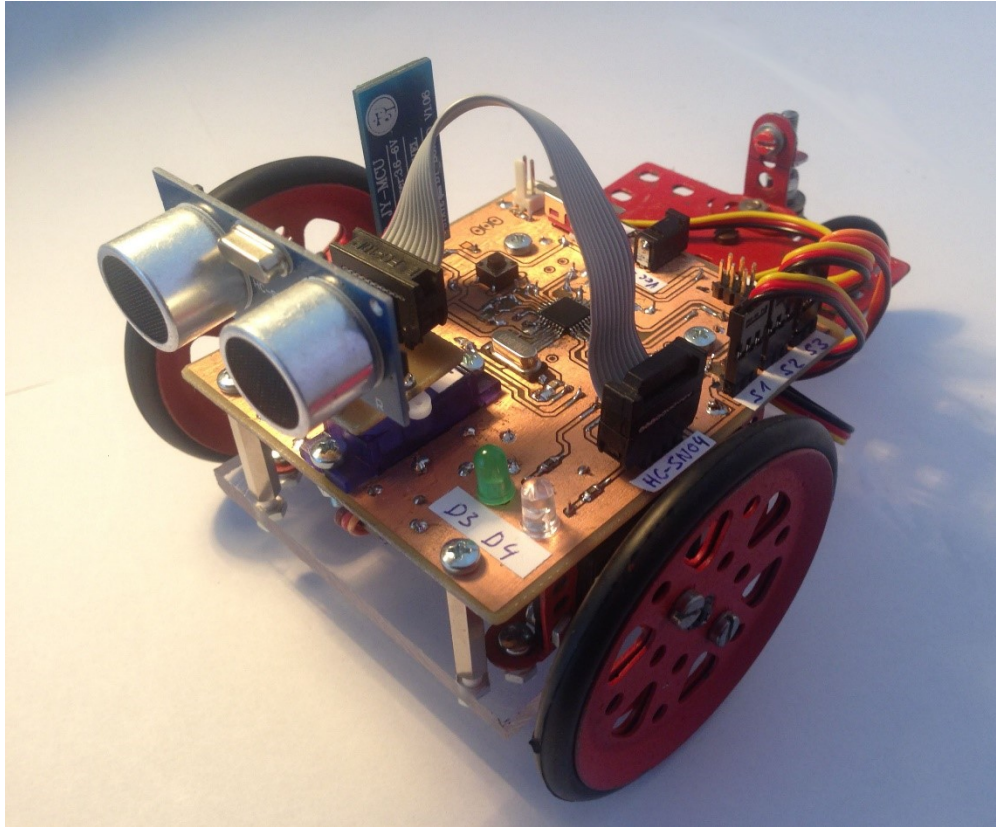


Obrázek 15 - Hlavní DPS

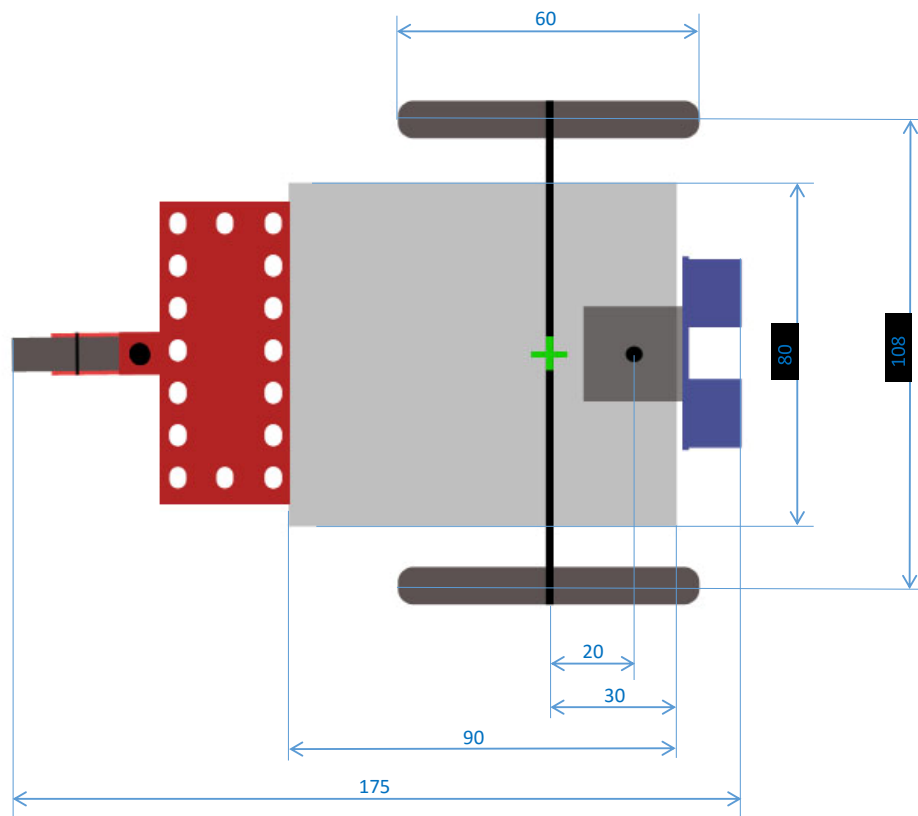
Hlavní deska obsahuje otvor pro upevnění servomotoru snímače. Čtyři led diody jsou určeny pro signalizaci aktuálního stavu procesu. Jejich funkce je specifikována při popisu řídicího programu MCU v kapitole 7.4. Dále jsou zde vyvedeny konektory pro zapojení servomotorů, snímače a modulu BT. K programování je určen ISP konektor. Provozní napětí celé desky je 5 V. Napájení je zde možné dvěma způsoby. Základním předpokladem je, že vozítko se bude napájet z 5 V zdroje a to pomocí dvoulinky nebo baterie. V případě potřeby použít vyšší zdroj je na DPS vyveden konektor pro vyšší napětí, které se následně stabilizuje na provozní napětí. Na desce není napájen stabilizátor z důvodu nevyužití této možnosti v průběhu vývoje a testování. Z nedostatku místa na DPS byl použit pouze jeden přepínač, který přepíná mezi jednotlivými variantami zdroje. Předpokladem je, že vozítko bude napájeno současně pouze z jednoho zdroje. A tedy funkce tlačítka slouží pro odpojení přívodního napětí.

### 7.3 Výsledné vozítko

Na obrázku 16 je zobrazeno výsledné vozítko. Více pohledů na vozítko je uvedeno v příloze C. Na obrázku 17 jsou uvedeny základní rozměry vozítka, které slouží k vytvoření matematického modelu v lokalizačním algoritmu.



Obrázek 16 - Sestavené vozítko



Obrázek 17 - Základní kóty vozítka

## 7.4 Řídicí aplikace v MCU

Aplikace v PC vyše pomocí BT příkazy ve formě definovaných znaků, které přijme MCU a na základě jim přidělené funkce vykoná požadovanou akci. Vozítko má pasivní inteligenci, která čeká na příjem dat a na základě nich vykoná požadovaný úkon. Příkazy se dělí na hlavní a vedlejší. Hlavní příkazy provádějí manipulaci vozítka a měření. Vedlejší příkazy slouží k signalizaci průběhu měření. V tabulce 1 jsou uvedeny hlavní příkazy a v tabulce 2 vedlejší příkazy.

**Tabulka 1 - Hlavní příkazy**

Hlavní příkazy	
Znak	Funkce
w	pohyb vpřed, měření
s	pohyb vzad, měření
a	pohyb vlevo o 45°, měření
d	pohyb vpravo o 45°, měření
t	měření

**Tabulka 2 - Vedlejší příkazy**

Vedlejší příkazy	
Znak	Funkce
p	Rozsvícení led D4 (signalizace startu měření)
e	Rozsvícení led D3 (signalizace konce měření)
r	zhasne všechny led

Program po inicializaci čeká na přerušení způsobené příjmem znaků ze sériové linky od BT modulu. Znaky jsou následně vyhodnoceny. Pokud se jedná o nedefinovaný znak, tak se neprovede žádná akce. Příjmutím vedlejších znaků se pouze provede změna úrovně na příslušných pinech vedoucích k led diodám.

Pokud hlavní příkaz vyžaduje pohyb vozítka, tak se nejprve provede pohyb vozítka a následně se vykoná měřicí sekvence s kontinuálním posíláním dat do PC.

Pohybové vlastnosti vozítka jsou nastaveny na dráhu jednoho kroku 4,2 cm s krokem natočení 45°. Ke snímání uražené vzdálenosti je využito předpokladu, že rychlost pohybu servomotorů je konstantní a tedy po sepnutí pohonů po určitou stanovenou dobu, vozítko urazí opakovaně stejnou vzdálenost. Stanovení časových prodlev bylo zajištěno testováním.

Po ukončení pohybu se zahájí měřicí sekvence. Snímač je zde servomotorem natáčen o  $\pm 60^\circ$ . Krok je volen po  $20^\circ$ . Jeden měřicí cyklus tedy získá 7 údajů o okolním prostředí příslušného



úhlu natočení. Měření se provádí od pravé krajní pozice do levé krajní pozice, viz obrázek 16. Výchozí úhel snímače je  $0^\circ$  v přímém směru. K urychlení měřicího cyklu je proveden pohyb snímače do krajní pozice již při pohybu vozítka. Mezi jednotlivými posuny je dostatečná časová prodleva potřebná k nastavení servomotoru.

Pokud při jednom měření dojde k chybě zaznamenání vzdálenosti, tak se provede opakování měření. Pokud opět dojde k chybě, tak se odešle informace o nezměřené vzdálenosti.

Odesílání údajů o vzdálenosti se provádí ve formátu „<xx>“, kde hranaté závorky reprezentují identifikační znaky začátku a konce předávané informace. Znaky xx je naměřená vzdálenost v rozlišení dvou desetinných míst. V případě chyby měření se odešle místo čísla chybový kód ve formátu „<err>“.

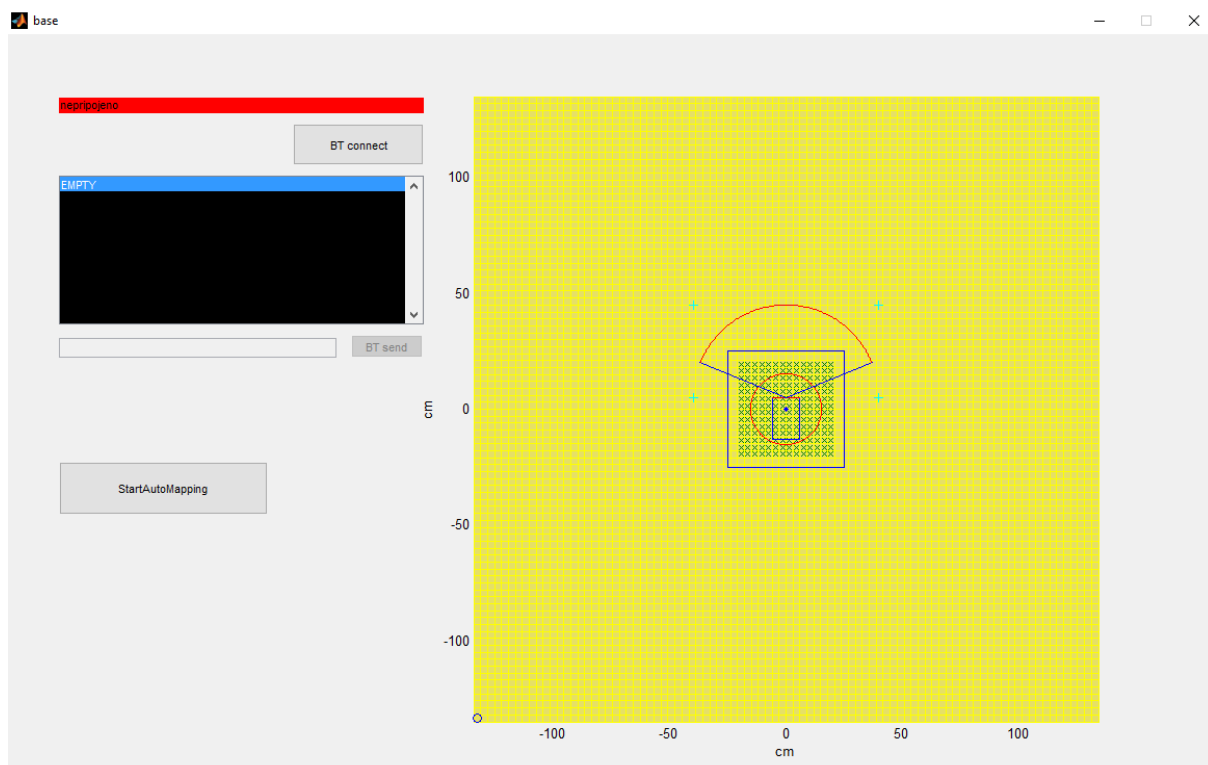
Po každém měření je naměřená hodnota menší nebo rovno 20 cm signalizovaná rozsvícením červené led diody D1. V opačném případě je rozsvícena zelená led dioda D2. Chybu měření reprezentují rozsvícené obě diody D1 a D2.

K naprogramování aplikace v MCU bylo využito (12) a (13). Zdrojové kódy jsou uvedeny v příloze E.

## 7.5 Řídicí aplikace v PC

Aplikace v prostředí Matlab obstarává odesílání příkazů, příjem dat ze snímačů, lokalizaci vozítka, zpracování přijatých dat a jejich vhodnou grafickou interpretaci. Při tvorbě této aplikace byl kladen požadavek na kontinuální zobrazování zpracovaných dat a lokalizaci vozítka. Zdrojové kódy aplikace jsou uvedeny v příloze E.

Na obrázku 18 je zobrazen náhled na výslednou aplikaci. V následujících podkapitolách je popsán princip tvorby mapy, modelu vozítka, způsob pohybu vozítka po mapě. Dále je zde popsán princip omezení pohybu vozítka, aby nevyjelo z mapy nebo nedošlo ke kolizi s překážkou. Následně jsou popsány možnosti způsobu mapování a uveden výsledek testovacího mapování.



Obrázek 18 - Podoba aplikace na PC

### 7.5.1 Generování mapy

Na počátku se definují základní parametry mapy, jako je velikost prozkoumávaného území a velikost jednotlivých dílků, které udávají rozlišení mapy. Pro jednoduchost se vytváří čtvercová mapa o sudé velikosti. Pro získání počtu dílků na jednotlivé osy, se délka strany převede na sudé číslo a následně vydělí velikostí dílku. Umocněním počtu dílků na druhou mocninu se získá počet dílků celé mapy.

Údaje o vzdálenostech a stavech jednotlivých dílků jsou uloženy v několika dvou dimenzionálních maticích. Pro uložení vzdálenosti  $X$  a  $Y$  souřadnice do jedné mapy je využito komplexní reprezentace, která značně ulehčuje práci s daty. Nulové souřadnice  $[x_0, y_0]$  jsou umístěny ve středu mapy, respektive ve středu pole. Proto je nutné vypočítat údaje o vzdálenostech jednotlivých bodů s ohledem na tuto podmínku. Nejprve je nutné určit souřadnice prvního bodu, kterému se přiřadí jedničkové indexy pole mapy.

Výpočet vzdálenosti středu dílku od počátku v ose  $X$  je proveden v rovnici 4. Stejný vzorec je použit pro osu  $Y$ .

$$MAP\_startX = -((MAP\_partsX/2) * MAP\_part) + MAP\_partRadius \quad (4)$$

kde,

*MAP\_startX*... vzdálenost středu dílku v ose X na počátku matice

*MAP\_partsX*... celkový počet dílků na ose X

*MAP\_part*... velikost strany dílku

*MAP\_partRadius*... poloměr dílku

První zmíněné pole, nazvané *MAP\_partsMiddles(mY, mX)*, obsahuje údaje souřadnice středů dílků mapy. Ze známé krajní souřadnice je nyní možné dopočítat všechny středy. Příklad výpočtu je uveden v rovnici 5.

$$MAP\_partsMiddles(mY, mX) = (MAP\_startX + (mX - 1) * MAP\_part) + (MAP\_startY + (mY - 1) * MAP\_part) * i \quad (5)$$

kde,

*MAP\_partsMiddles*... pole mapy obsahující informace o středech dílků

*mY, mX*... indexy pole mapy

*MAP\_part*... délka strany jednoho dílku

Tyto souřadnice jsou využívány pro vyhodnocování umístění překážek. Druhá matice označená *MAP\_corners(mY, mX)* slouží k vykreslení čtverečků mapy. Třetí maticí je matice obsahující informaci o stavu dílku, označená je *MAP\_state(mY, mX)*. Tato matice obsahuje číslo, které charakterizuje stav dílku. Význam jednotlivých čísel je uveden v tabulce 3.

**Tabulka 3 - Označení a význam údajů reprezentující stav dílku**

Označení	Význam
0	„neprozkoumáno“
1	„není překážka“
2	„je překážka“
3	„nelze změřit“

Výchozí stav definuje žlutá barva dílků. Zelený křížek určuje volný prostor, červený křížek překážku a šedý křížek chybně změřený úsek. Použití znaků místo výplně celého dílku má

svůj opodstatněný důvod. Využití funkce pro grafické vykreslování čtverečků je výpočetně náročnější a proto při využití vysokého rozlišení mapy, by překreslování značně zpomalovalo činnost programu. Z toho důvodu byly zvoleny znakové útvary, jako je křížek, jejichž vykreslování nezpomaluje výrazně program.

Při spuštění se automaticky vyhodnotí prostor pod vozítkem a v jeho nejbližším okolí jako bez překážky. Velikost takového prostředí musí být dostatečně velká, aby se v něm mohlo vozítko otočit o 360°. Podmínkou je tedy, aby nejbližší okolí vozítka neobsahovalo žádné překážky.

### 7.5.2 Model vozítka

Na mapě je zobrazen model vozítka, který svojí polohu aktualizuje na základě příkazů. Parametry modelu jsou zadány na počátku programu. Jsou zde uvedeny délky stran vozítka, umístění snímače, pozice střední osy vozítka, podle které se model natáčí. Dalším parametrem je nastavení pozice a úhlu natočení modelu na mapě. Souřadnice [0, 0] umístí vozítko do středu mapy (do jeho počátku). Důležitým parametrem je krok vozítka, který je zadán podle nastavení v programu vozítka. Dále se definuje dosah senzoru, který udává rozsah, po který vozítko mapuje prostor. Krok natáčení vozítka a senzoru je definován v části, kde se zpracovávají data.

Zadané rozměry vozítka je třeba geometricky reprezentovat. Z důvodu, že model se pohybuje a natáčí, je třeba při každém pohybu přepočítávat veškeré souřadnice modelu. V následujícím textu je popsán matematický postup přepočtu pozice jednoho bodu modelu.

Bude uvažován pravý horní bod rámu modelu, viz rovnice 6.  $BOT\_TR$  je nyní komplexní souřadnice bodu vzdálená od osy modelu.

$$BOT\_TR = BOT\_x + (BOT\_y - BOT\_wheel) * i \quad (6)$$

kde,

$BOT\_x, BOT\_y, BOT\_wheel$ ., parametry určující umístění bodu od osy modelu

Následně je třeba zjistit velikost tohoto bodu a úhel vůči ose modelu. K zjištění těchto údajů se použijí funkce **abs** a **angle**. Při pohybu v ose X nebo Y je získání nové souřadnice jednoduché. Pouze se přičte reálná a imaginární složka k původnímu bodu. Situace se komplikuje, pokud se bod má natáčet o určitý úhel vůči ose modelu. Zde je potřeba, ze

znalosti vzdálenosti bodu od osy úhlu a úhlu natočení modelu, vypočítat nové souřadnice pomocí rovnice 7.

$$\begin{aligned} BOT\_TL\_new = & \cosd(BOT\_TL\_angle + BOT\_angle) * BOT\_TL\_abs \\ & + sind(BOT\_TL\_angle + BOT\_angle) * BOT\_TL\_abs * i \end{aligned} \quad (7)$$

kde,

*BOT\_TL\_angle*... původní úhel bodu

*BOT\_angle*... úhel o který je celkový model natočen

*BOT\_TL\_abs*... vzdálenost bodu od osy modelu

Rovnice 7 vypočítá novou souřadnici bodu vůči ose. Přičtením k tomuto bodu souřadnici osy vůči celé mapě, se získá výsledná nová poloha bodu na mapě. Tento princip je aplikován na všechny body modelu.

Pokud je vozítko natočeno pod úhlem, který není kolmý k žádné ose mapy, tak nelze při pohybu přičíst velikost kroku do jedné souřadnice, ale musí se ze známého úhlu natočení vozítka a známé velikosti kroku určit velikost jednotlivých složek posunu ve směru osy X a osy Y.

### 7.5.3 Pohyb vozítka

Pohyb vozítka byl již popsán v kapitole 7.5.2. Nyní je však nutné nějakým způsobem tento stupeň volnosti omezit. Omezení se vztahuje na znemožnění vyjetí vozítka z omezeně velké mapy a druhým zásadním omezením je zakázání vjetí vozítka do prostoru překážky. Nejprve se definuje **oblast hranice bezpečnosti vozítka**. V programu označeno pod názvem *BOT\_safetyRadius*. Jedná se o kruhovou oblast se středem v ose vozítka a poloměrem o velikosti minimálně nejvzdálenějšího bodu vozítka od osy. Jedná se tedy o zónu, kde se za žádných okolností nesmí vyskytnout překážka. Tato oblast deklaruje, že se vozítko nedostane do situace, kde by se nedokázalo otočit o 360°. Tímto je i eliminována možnost zaseknutí vozítka v určitém prostoru. Navzdory usnadnění algoritmu má tento způsob nevýhodu v omezení mapování užších prostorů, kde by fyzicky vozítko bylo schopné vjet, ale pohyb v takovém prostoru by se musel řešit složitějším algoritmem řízení.

Hranice bezpečnosti je tvořena 360 body. Počet bodů je závislý na jemnosti čtverečků mapy. Pro správnou funkci je nezbytné, aby vytvořený obrazec pospojovaných stran čtverečků, ve

kterých se nachází minimálně jeden bod bezpečnosti, tvořil uzavřený mnohoúhelník. Při nesplnění této podmínky nedojde ke 100% vyhodnocení mapy v daném okruhu.

Nyní je potřeba určit konkrétní dílek, který leží na hranici bezpečnosti. Jsou známy dva druhy souřadnic. Prvním druhem jsou souřadnice středového bodu všech dílků mapy a druhým druhem jsou souřadnice bodů hranice bezpečnosti. Určení příslušných relevantních bodů se provede porovnáním velikostí zvláště reálné a imaginární složky (tedy osy X a Y) souřadnice bodu mapy a bodu hranice bezpečnosti. V prvním případě se k reálné souřadnici středu dílku mapy přičte polovina délky dílku a porovná se s velikostí reálné složky bodu hranice bezpečnosti. Ve druhém případě se od středu dílku mapy odečte polovina délky dílku a opět se porovná. Pokud je velikost reálné složky bodu hranice bezpečnosti menší a ve druhém případě větší než velikost bodu dílku, tak reálná část dílku leží mezi vnitřními hranicemi dílku. Provedením stejného ověření u imaginární části se získá informace, zda bod leží mezi vnitřními hranicemi dílku. Pokud bude bod ležet mezi vnitřními hranicemi v reálné a současně i v imaginární ose, tak bod hranice bezpečnosti leží na ploše daného dílku.

Projetím celé mapy a porovnáním všech bodů hranice bezpečnosti s jednotlivými dílky mapy se identifikují konkrétní body na hranici.

Při udělení příkazu na změnu pozice vozítka nejprve vozítko vykoná imaginární pohyb pouze na mapě, kde porovná nové body v hranici bezpečnosti a udělí povolení k vykonání pohybu či ho zakáže.

Při kroku větším než je velikost strany dílku mapy by však došlo k přeskočení několika dílků na mapě a tedy případnému přeskočení bodů obsahující údaj o překážce, který by se nekontroloval. Vozítko by tedy mohlo vjet do již známé překážky.

Řešení tohoto problému se provede tak, že se celkový krok neboli skok vozítka rozdělí na dílčí kroky o velikosti šířky dílku mapy. Ve smyčce se postupně provádí dílčí posun, vyhodnotí se celá mapa a takto pokračuje do posledního kroku. Pokud v dílčích krocích došlo ke konfrontaci s bodem, na kterém se vyskytuje překážka, tak se celkový krok nepovolí.

Při praktické implementaci takového algoritmu se objevil problém u větších velikostí map, větší jemnosti nebo větším kroku. Problém spočíval ve velkém množství matematických operací, které měly zásadní vliv na rychlost vyhodnocení daného úkonu.

K optimalizaci výpočtů je vytvořena v okolí vozítka čtvercová hranice. Poloměr této hranice musí být delší, než je krok vozítka. Na obrázku 18 reprezentuje tuto hranici modrý čtverec.

Hlavní myšlenka spočívá v omezení procházení bodů mapy pouze na body ležící v oblasti čtverce. Stejným způsobem, jako byla popsána identifikace odpovídajícího dílku na hranici bezpečnosti, se provede identifikace rohových dílků čtverce. Identifikací korespondujících dílků se získá informace o indexech položek pole. Procházené pole se tedy omezí na položky ležící v rozmezí získaných indexů.

Tato optimalizace způsobila úplnou eliminaci závislosti na velikosti mapy. Počet porovnávaných položek je konstantní.

Zamezení vyjetí vozítka mimo definované hranice mapy se provede jednodušším ověřením než je zabránění kolize s překážkou. Zde se musí ověřit, zda velikost v absolutní hodnotě souřadnice nějakého bodu hranice bezpečnosti není větší než velikost mapy v absolutní hodnotě<sup>3</sup>. Porovnávají se zvláště reálné a imaginární složky. V případě, že je jedna ze složek větší než velikost mapy, jedná se o překročení hranice mapy.

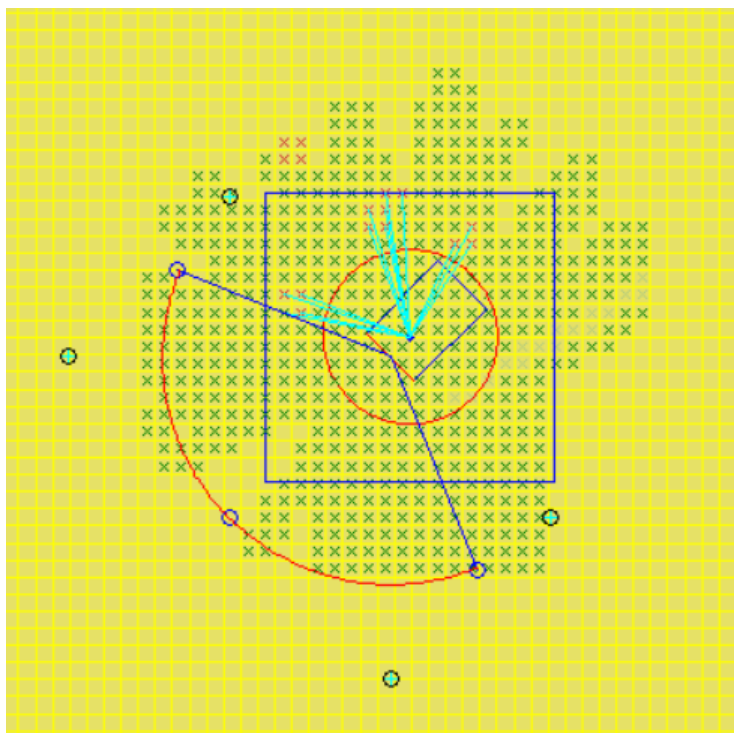
Ověřování všech bodů hranice bezpečnosti, zda některý z nich nepřesáhl mez mapy je vhodné pro případy, že by hranice mapy nebyla čtvercového (případně obdélníkového) tvaru, ale byla by tvořena ostrými mnohoúhelníkovými obrysy, zde je tato metoda nezbytná. Však pro hranice mapy čtvercového (případně obdélníkového) a tedy pravoúhlého tvaru je možné ke kontrole použít pouze čtyři body z hranice bezpečnosti. Úhly bodů by byly  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  a  $270^\circ$ . Jedná se o body kolmé na jednotlivé stěny hranice mapy. Tímto se opět zvýší rychlost ověřování možnosti pohybu.

Diferenční řízení má výhodu, jak již bylo zmíněno, ve schopnosti otáčet se kolem své osy. Za předpokladu, že snímač dokáže obsáhnout úhel, který je širší než šířka vozítka a jeho dosah je mnohonásobně větší než krok vozítka, kde krok vozítka musí být menší než počáteční volná zóna. Zakázáním jízdy pozpátku vozítka se dosáhne situace, kdy vozítko nikdy nevjedou do předem nezmapovaného prostředí. Takový stav snižuje složitost řídicího algoritmu na minimum.

Překážky vyskytující se uvnitř čtvercové hranice jsou zvýrazněny azurovou čarou vedoucí od osy vozítka, viz obrázek 19. Účel čar je pouze poukazovat na překážky v těsné blízkosti vozítka.

---

<sup>3</sup> Velikostí mapy se uvažuje vzdálenost od bodu nula do krajní hodnoty



Obrázek 19 - Detail na identifikaci překážek v hranici vozítka

#### 7.5.4 Snímání překážek

Použitý snímač má vyzařovací úhel  $15^\circ$ . S rostoucí vzdáleností měření se zvětšuje snímáný prostor před vozítkem. Podle vlastností snímání se volí jemnost mřížky. Určením maximálního rozsahu senzoru se získá šířka vyzařovacího paprsku v dané vzdálenosti. Například pro 40 cm to je 10,4 cm. Ze známé šířky se volí jemnost kroku natočení senzoru, aby se dosáhlo plného naskenování prostoru před vozítkem. Na vozítku je zvolen krok  $20^\circ$ . Středové mezní úhly jsou voleny  $\pm 60^\circ$ . Skutečný zabíraný úhel je  $135^\circ$  (musí se přičíst polovina rozptylu senzoru z každé krajní polohy). Snímač se tedy musí natočit na 7 různých pozic a provést měření.

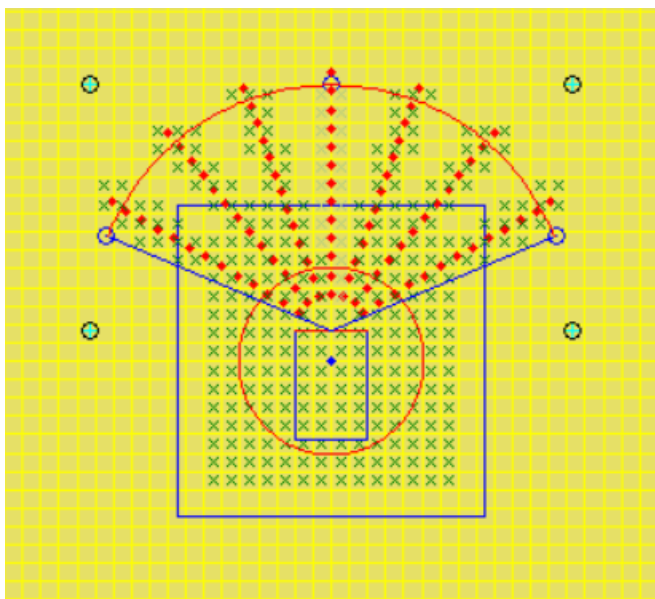
K správné interpretaci dat je nutné nasimulovat v modelu směr vyzařování snímače. Známe-li počáteční bod, úhel natočení, úhel vyzařování a maximální vzdálenost snímaného prostoru, dají se vypočítat konečné středové souřadnice vyzařovacích paprsků. Výpočet pozice bodu je stejný jako v rovnici 7, akorát se do goniometrických funkcí přidá proměnná určující úhel natočení snímače. Opět je potřeba získat souřadnice bodů ležících na přímce mezi počátečním a konečným bodem paprsku snímače, aby jim bylo možné přiřadit odpovídající stav. To je provedeno vytvářením sub rozsahů snímače, kde je zvolen krok o velikosti dílku mapy. Postupným zvětšováním sub rozsahů se popíše jeden snímací paprsek několika body. Na obrázku 20 jsou tyto body zobrazeny červenými tečkami.



V této aplikaci je předpokládáno, že vyzařovací parsek je na celé vzdálenosti stejný. Toto však nijak zvláště neznehodnocuje zaznamenávání dat do mapy. Při malé vzdálenosti, jemného snímacího kroku s použitím velkých dílků mapy by rozdíl obou metod byl zanedbatelný. Aplikací vysoké jemnosti (s ohledem na chybu snímače) a větší vzdálenosti snímání by bylo již vhodné využít věrné zaznamenávání údajů do mapy.

Určování dílků mapy probíhá stejným způsobem, jako bylo popsáno v kapitole 7.5.3. Zde se akorát nastaví velikost oblasti, která bude spadat k danému bodu paprsku. Určování probíhá podle osy X a Y. V případě, že měřicí parsek nebude kolmý na jednu z os, tak zde vzniká jistá nepřesnost určování příslušných bodů. Tato nepřesnost při využití mřížky s mnohonásobně větším poloměrem, než by byl rozdíl odchylky skutečné oblasti a oblasti kolmé, je zanedbatelná.

Jak u kontroly vjetí do překážky, tak i zde je potřeba omezit procházenou oblast dílků mapy. Zde je situace komplikovanější z důvodu umístění mimo osu vozítka. Na obrázku 20 čtyři kulaté body kolem pracovního prostoru snímače reprezentují oblast dílků, které by se měly při každém zpracování údajů procházet. V předchozím případě se čtvercová oblast neotáčela a proto určení minimálních a maximálních indexů polí bylo snadné. V tomto případě, kde se celá oblast natáčí s vozítkem, nelze hned označit souřadnice např. levého dolního bodu (na obrázku 20) za počáteční index daného sub pole. Proto je třeba získat minimální a maximální indexy tohoto pole. To je provedeno smyčkou, kde se postupně porovnají všechny čtyři body a určí se body s minimálním a maximálním indexem.



Obrázek 20 - Detail na princip určování překážky

Pole je procházeno od minimálního po maximální index pole. Pokud vozítko je v kolmé pozici na jednu z os, tak je procházen přesně obdélníkový prostor vymezený danými body. V případě natočení vozítka se pořád prochází obdélníková oblast, ale již nekopíruje hranice vytyčené body. Je vždy kolmá na hlavní osy. To způsobí zvětšení procházející oblasti při vyhodnocování. Příčinou je použitý algoritmus procházení pole. Pokud není jemnost dílků kriticky malá a velikost této oblasti podstatně velká, tak to má zanedbatelný vliv na zpoždění.

### **7.5.5 Metody mapování vozítka**

Mapování prostoru lze provádět dvěma způsoby. První je manuální a druhý autonomní, za pomoci vloženého mapovacího algoritmu. Manuální režim je zde hlavně z důvodu vývoje a odladování aplikace, kde bylo potřeba řídit pohyby vozítka po krocích.

Pro manuální řízení je zde vytvořen terminál. Aplikace obsahuje textové pole, kde se zadávají příkazy. Funkce vyhodnotí, zda se jedná o platný znak (je mu přidělena funkce). Při vyhodnocení platného znaku se provede kontrola možnosti pohybu. Je popsáno v kapitole 7.5.3. Pokud se v prostoru pohybu nenachází žádná překážka, je pohyb povolen. Po komunikačním řetězci se odešle zadaný znak do vozítka a aplikace čeká na příjem dat z vozítka. V tuto chvíli není možné zadávat žádné další příkazy. V případě zadání požadavku k další akci se tyto příkazy ignorují. Přijetím dat z vozítka se vyvolá funkce, která provede parsování přijatého řetězce. Řetězec musí obsahovat počáteční a ukončovací identifikační znaky, které mají tvar „<“ a „>“. Informace mezi těmito znaky se extrahuje a porovná se, zda se jedná o chybovou sekvenci. V případě, že ne, je vyčleněný řetězec převeden na číslo. Pořadí přijímání dat má definované pořadí, které musí souhlasit s úhlem snímače na vozítku i modelu.

Pokud přijatá informace obsahuje chybu měření, tak prostor od počátku snímače po jeho maximální dosah se označí šedou barvou. Tento stav je uvažován jako „není překážka“. V druhém případě, kde je přijata informace o vzdálenosti, se provede vyhodnocení hodnoty. Pokud hodnota je vyšší než maximální dosah, znamená to, že v daném rozsahu není překážka. Přijetím údaje menšího než je maximální dosah snímače, musí být provedena kontrola, zda naměřená překážka neleží uvnitř hranice bezpečnosti vozítka. Jelikož se uvažuje statické prostředí, tak tento stav není možný, aby nastal. Důvod nemožnosti výskytu překážky v hranici bezpečnosti je vysvětlen v kapitole 7.5.4. Jedná se o chybu snímače a tento stav není vyhodnocen jako překážka. Třetí případ je vzdálenost větší než hranice bezpečnosti a menší než uvažovaný dosah senzoru. Tento případ se už vyhodnocuje jako překážka. Oblast kratší

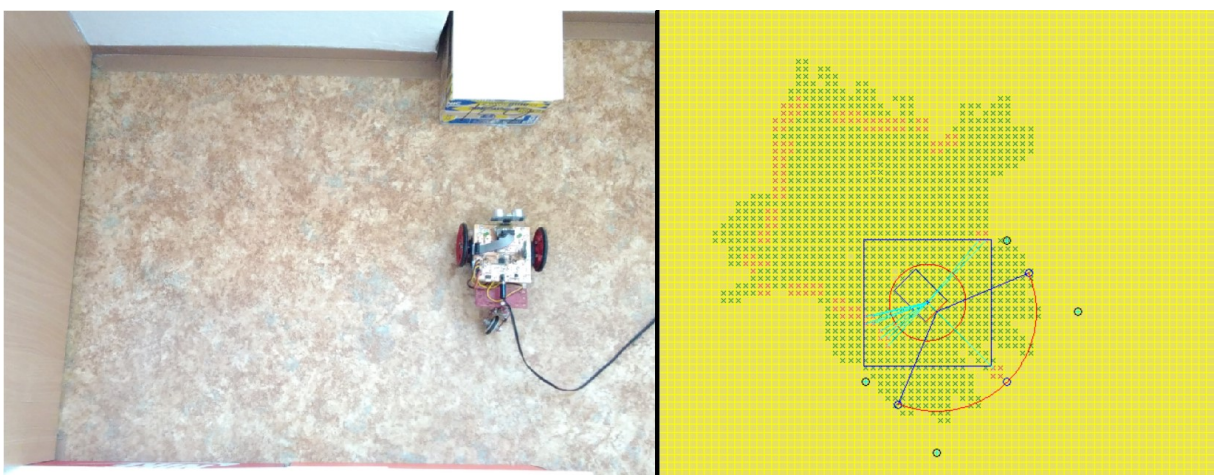
než je naměřená překážka se označí jako „není překážka“ a oblast kolem bodu v dané vzdálenosti se označí stavem „je překážka“.

V této práci je využito mapování bez paměti. To znamená, že pokud snímač znovu snímá již v minulosti snímanou oblast a tedy známou oblast, tak se původní data přepíše aktualizovanými údaji. Tento princip vede k degradaci věrnosti mapy vlivem velké chyby odometrie vytvořeného vozítka. Daný problém je zmíněn v kapitole 7.6.

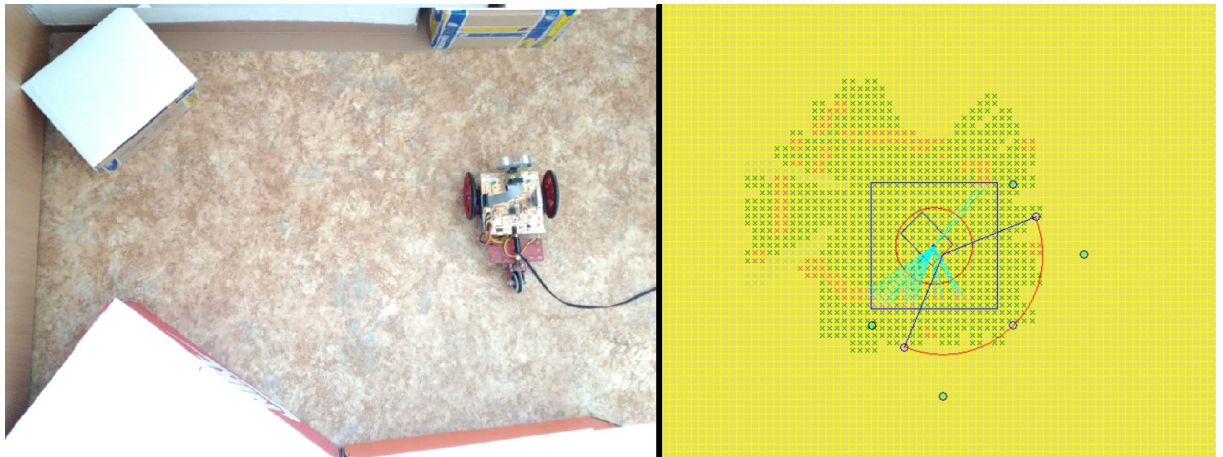
V automatickém režimu byl zaimplementován primitivní autonomní mapovací algoritmus. Vývojový diagram je uveden v příloze D. Vozítko se snaží jet rovně a mapovat prostor před sebou. V případě nemožnosti jet vpřed se vozítko natočí doleva a pokusí se provést dva kroky vpřed. Poté se natočí doprava a pokračuje vpřed. Vznikne-li nemožnost provést dva kroky vpřed po změně směru, tak se provede opět natočení vlevo. A postup se opakuje. V tomto případě se vozítko už nevrátí do výchozího úhlu natočení. Z toho vyplývá, že algoritmus není schopen systematicky zmapovat celý prostor mapy. V podstatě se snaží najít možnou cestu jízdy a při ní skenuje okolní prostor. Zaimplementováním sofistikovanější inteligence vozítka se může vytvořit úplné zmapování prostoru.

## 7.6 Testovací mapování a jeho zhodnocení

Testovací mapování bylo provedeno pomocí použité triviální inteligence, která je popsána v kapitole 7.5.5. Nejprve byl vytvořen zkušební prostor, do něj se umístilo vozítko a spustil se samo mapovací algoritmus. Testovací mapování je zaznamenáno na obrázcích 21 a 22, kde na levé straně je mapovaný prostor a na pravé straně vytvořená mapa.



Obrázek 21 - Testovací mapování 1



Obrázek 22 - Testovací mapování 2

Jak je z výsledných obrázků zřejmé, tak výsledná mapa obsahuje odchylky od reálného půdorysu prostoru. Použitý princip lokalizace vozítka předpokládá ideální pohyb reálného vozítka, tedy, že zadaný krok a úhel natočení bude vždy stejný. V reálném případě však použité snímání ujeté dráhy, které je založeno na předpokladu konstantní rychlosti pohybu pohonu, obsahuje nezanedbatelnou chybu. Použité upravené servomotory se nepohybují pokaždé stejnou rychlostí. Při řízení dvou upravených servomotorů je rychlost otáčení na stejné řídicí impulsy (i při identickém nastavení snímacích potenciometrů) odlišná. Jedná se o chybu metody, která je závislá na stejnorodém chování pohonů a tato podmínka u použitého typu pohonu není splněna. Chyba postupně exponenciálně narůstá s počtem kroků mapování.

Důsledkem je, že reálná pozice vozítka a pozice modelu na mapě není správná. Při větší chybě lokalizace dojde k nesprávnému vytváření mapy. Příkladem je obrázek 21, kde spodní stěna je zaznamenána s velkou chybou způsobenou právě touto odchylkou polohy vozítka a modelu.

Pohyb vozítka je značně zpomalován velkým počtem měřících cyklů vozítka. Při vývoji byla vyzkoušena varianta měření pouze ve třech úhlech, kde se prostor omezil pouze na větší objekty. Po aplikaci se sice měřící cyklus dvojnásobně urychlil, ale za cenu ještě větší degradace měření. Ultrazvukový snímač je závislý na úhlu snímání vůči úhlu objektu v prostoru. Použitím pouze tří snímků prostoru docházelo k velkému množství chyb měření a nezaznamenávaly se fyzické objekty do mapy. Proto použitím snímání okolí o malém úhlu kroku je dosaženo značného snížení této chyby.

## **7.7 Možné zlepšení snímání změny pozice vozítka**

Z celkové problematiky vytváření mapy v této praktické části vyplynulo, že chyba vzniká pouze na straně vozítka. Konkrétně na použité metodě snímání pohybu. Minimalizovat náhodný charakter chyby změny pozice vozítka s ponecháním stejné metody snímání uražené vzdálenosti, lze použitím jiného druhu pohonu. Z pohonů popsaných v teoretické části se jeví jako nejvhodnější krokový motor. Ponecháním stávajícího pohonu je nutné vozítko opatřit snímačem otáček, například enkodérem nebo akcelerometrem, či gyroskopem. Opatřením vozítka přesným zaznamenáváním změny pozice, by se odstranila zásadní chyba celého mapování.

Samozřejmě se v celé problematice neuvažuje situace smyku kol nebo umělá změna polohy vozítka. V této práci popsané principy lokalizace vozítka nedokáží tyto případy zpracovat.

## ZÁVĚR

Cílem této práce bylo navrhnout autonomní model vozítka pro vytváření 2D mapy uzavřeného prostoru. V úvodu práce byly stanoveny základní podmínky na prostředí, ve kterém by se vozítko mělo pohybovat.

V teoretické části byly popsány možnosti konstrukce podvozku s ohledem na jeho pohybové vlastnosti. Dále byly popsány tři druhy pohonů a principy měření pohybu vozítka v prostoru. V kapitole zabývající se snímači byly popsány 3 základní druhy snímání okolního prostředí vozítka, s uvedením konkrétních příkladů reálných snímačů a jejich popis. Poté byl popsán jeden ze způsobů tvorby mapy a to pomocí mřížkové mapy. Stručně byly zmíněny základní mapovací algoritmy. Poslední kapitola teoretické části se zabývala možnostmi řešení řízení a zpracování dat MR.

Praktická část se zabývá problematikou návrhu, vývoje a následného oživení vozítka. Nejprve bylo popsáno konstrukční řešení vozítka, návrh řídicí desky, způsob řešení ovládání a popis algoritmu v mikroprocesoru. V kapitole 7.5 byl detailně probrán postup návrhu řídicí aplikace vozítka. Popsán zde byl způsob tvorby mapy a modelu vozítka. Dále bylo popsáno řešení pohybu modelu vozítka po mapě. Byla řešena problematika stanovení omezení vozítka při pohybu po mapě, které se týkalo zamezení vyjetí vozítka z prostoru mapy a nemožnost vjetí vozítka do prostoru překážky. V další části bylo vysvětleno manuální mapování vozítka a byl zde popsán jednoduchý autonomní mapovací algoritmus.

Nakonec bylo provedeno testovací mapování na dvou zkušebních prostorech. Výsledek testu byl následně okomentován. Jako zásadní nedostatek vozítka se projevila nepřesná metoda snímání pohybu vozítka. Možnosti řešení tohoto problému jsou v práci zmíněny.

Odstraněním chyby určování pohybu vozítka by se odstranila degradace vytvořené mapy s počtem kroků cyklu mapování. Další problematikou je chyba snímací metody. Tuto chybu lze minimalizovat použitím kombinace s infračerveným snímačem vzdálenosti a využití pravděpodobnostní mřížky při vytváření mapy. Pro úplné mapování zadaného prostoru je nutno doplnit vytvořený program o komplexní mapovací algoritmus.

Na závěr lze konstatovat, že všechny body práce byly splněny. Bylo vytvořeno funkční vozítko schopné autonomního pohybu v neznámém prostoru a vytvářet kontinuální 2D mapu s aktuálním lokalizováním vozítka ve formě modelu vozítka na mapě.

## POUŽITÁ LITERATURA

1. NOVÁK, Petr, Mobilní roboty. 1. vyd. Praha: BEN - technická literatura, 2005. 243 s. ISBN 80-7300-141-1.
2. CHLUD, Michal, HYBRIDNÍ ROBOTICKÉ PLATFORMY A JEJICH POUŽITÍ V AREÁLU FSI VUT - bakalářská práce. Brno: Vysoké Učení Technické v Brně, Fakulta strojního inženýrství ústavu výrobních strojů, systémů a robotiky, 2008, 30 s. 0 příl. Vedoucí bakalářské práce Ing. Tomáš Kubela
3. BALADA, Radek. Řízení serva. [online]. [cit. 2016-04-8]. Dostupné z: <http://vyvoj.hw.cz/novinky/rizeni-serva.html>
4. WINKLER, Zbyněk, Odometrie. [online]. [cit. 2016-02-18]. Dostupné z: <http://robotika.cz/guide/odometry/en>
5. Prostředky relativní lokalizace. [online]. [cit. 2016-02-18]. Dostupné z: <http://marek.sk.sweb.cz/lokalizace/kapitola3.html>
6. BLEJKA, Jan, NÁVRH A REALIZACE METODY MAPOVÁNÍ OKOLÍ PRO MOBILNÍ ROBOT - diplomová práce. Brno: Vysoké Učení Technické v Brně, Fakulta strojního inženýrství ústav automatizace a informatiky, 2007, 68 s. 0 příl. Vedoucí diplomové práce Ing. Stanislav Věchet, Ph.D.
7. NAVE, R. Infrared. [online]. [cit. 2016-04-2]. Dostupné z: <http://hyperphysics.phy-astr.gsu.edu/hbase/ems3.html>
8. Datasheet GP2Y0A21YK0F. [online]. [cit. 2016-03-02]. Dostupné z: [http://www.sharpsma.com/webfm\\_send/1489](http://www.sharpsma.com/webfm_send/1489)
9. Datasheet Ultrasonic Ranging Module HC - SR04. [online]. [cit. 2016-01-08]. Dostupné z: <http://www.micropik.com/PDF/HCSR04>.
10. WINKLER, Zbyněk, Odometrie. [online]. [cit. 2016-02-20]. Dostupné z: <http://robotika.cz/guide/gridplan/en>
11. DLOUHÝ, Martin, Bug algoritmy. [online]. [cit. 2016-05-5]. Dostupné z: <http://robotika.cz/guide/bug-alg/cs>
12. VÁŇA, V. *Mikrokontroléry ATMEL AVR: programování v jazyce C*. Praha: BEN-technická literatura, 2003. 216 s. ISBN 978-80-7300-102-0.

13. Datasheet ATmega48A/PA/88A/PA/168A/PA/328/P. [online]. [cit. 2016-01-10].  
Dostupné z: [http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p\\_datasheet\\_complete.pdf](http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_complete.pdf)



## **SEZNAM PŘÍLOH**

Příloha A – *Hlavní DPS*

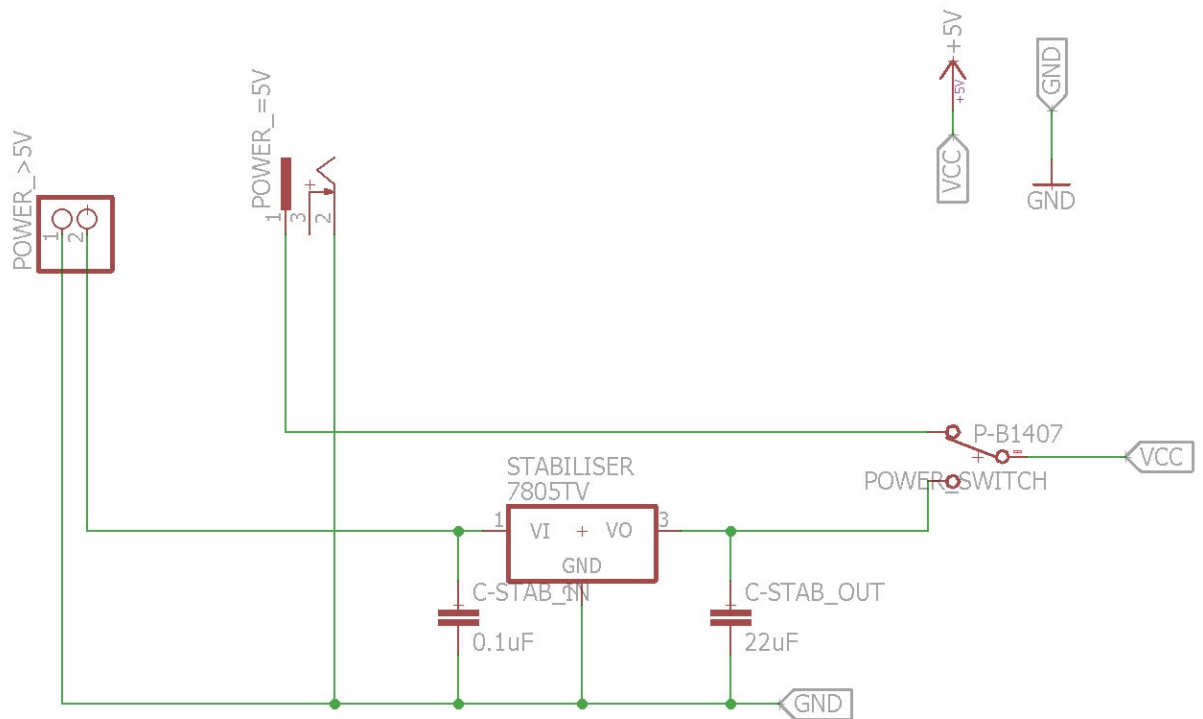
Příloha B – *Návrh DPS pro snímač*

Příloha C – *Výsledné vozítko*

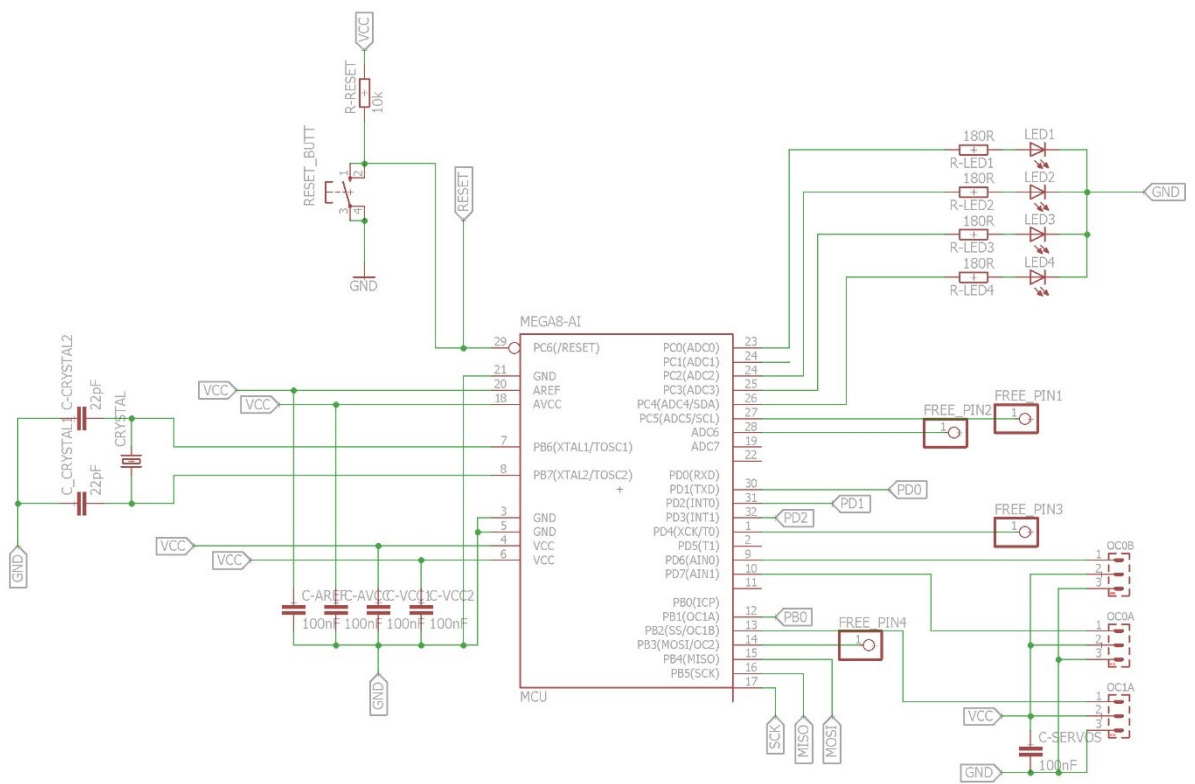
Příloha D – *Blokový diagram samo mapovacího algoritmu*

Příloha E – *CD*

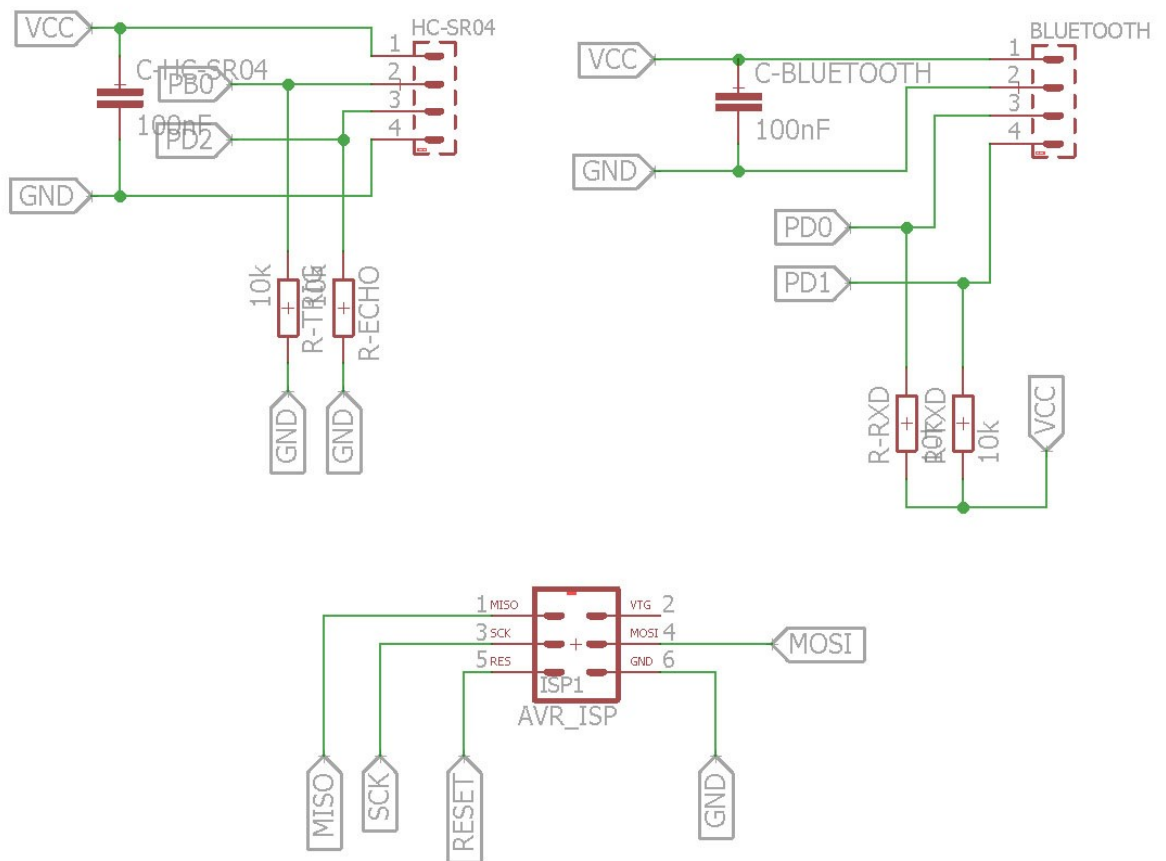
Příloha A – Hlavní DPS



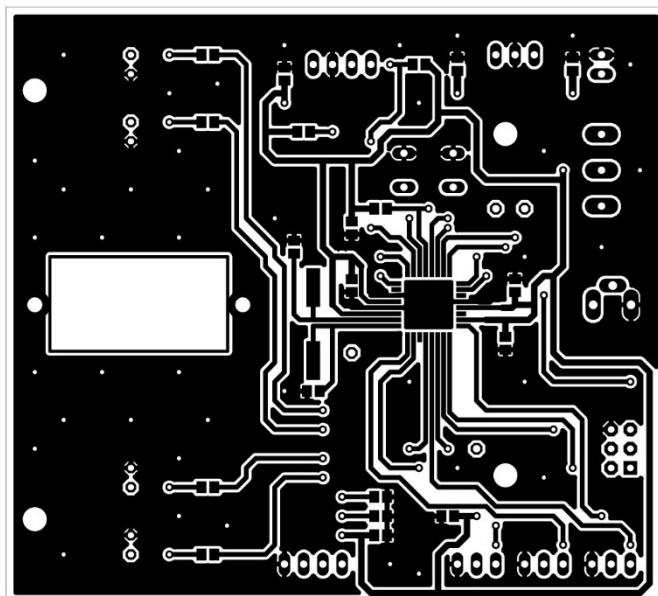
Obrázek 1 - Schéma hlavní DPS část 1/3



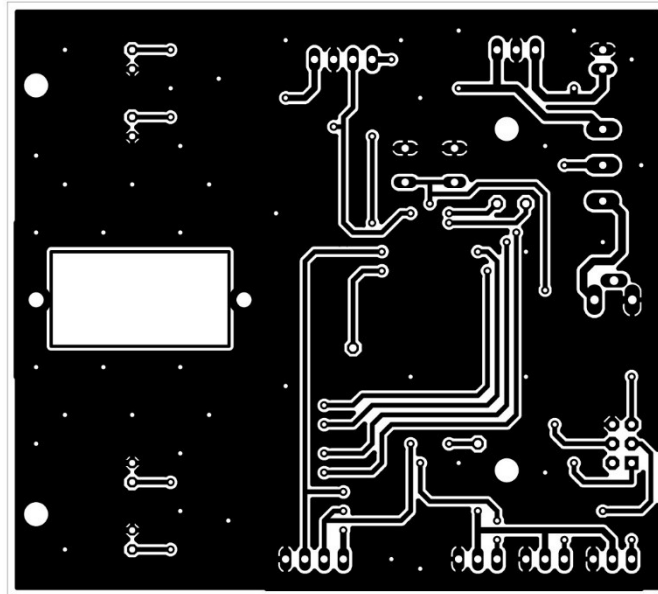
Obrázek 2 - Schéma hlavní DPS část 2/3



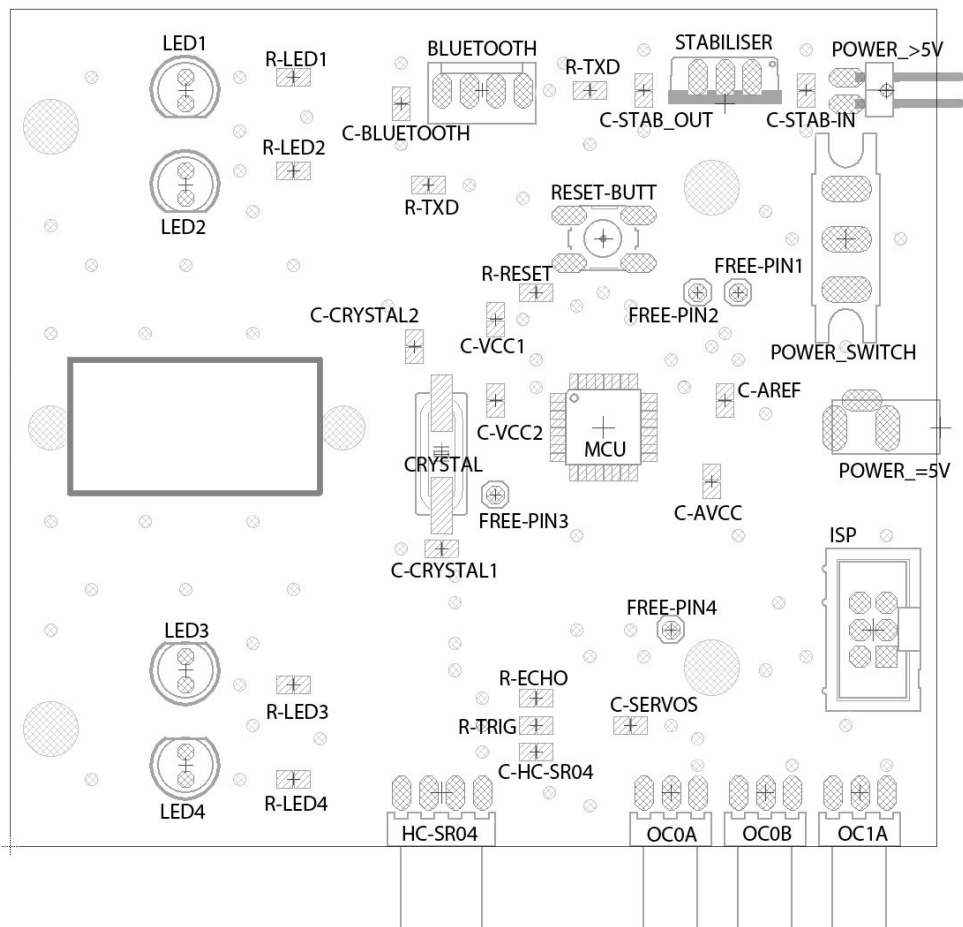
Obrázek 3 - Schéma hlavní DPS část 3/3



Obrázek 4 - Návrh hlavní DPS, vrchní strana



Obrázek 5 - Návrh hlavní DPS, spodní strana



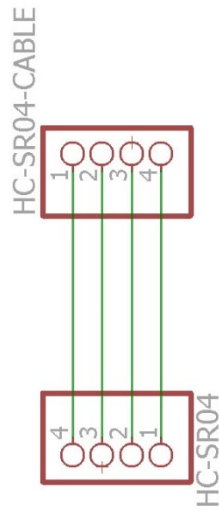
Obrázek 6 - Rozmístění součástek hlavní DPS

Tabulka 1 - Seznam součástek hlavní DPS

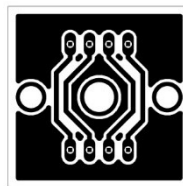
Pořadové číslo	Označení součástky na DPS	Název/hodnota	Pouzdro	Poznámka
1	POWER_>5V	PSH02-02PG		Konektor se zámkem do DPS, 2 pin, 2,54 mm
2	POWER_=5V	PC-GK1.3		Napájecí konektor do DPS 1,3 mm
3	STABILIZER	LM2940CT-5	TO220	
4	C-STAB_IN	100 nF	SMD, 0805	
5	C-STAB_OUT	22 $\mu$ F	SMD, 0805	
6	POWER_SWITCH	P-B1407		
7	RESET_BUTT	TC-0104		mikrospínač do DPS
8	R-RESET	10 k $\Omega$		
9	CRYSTAL	HC49,16 MHz	US-SMD	
10	C-CRYSTAL1	22 pF	SMD, 0805	
11	C-CRYSTAL2	22 pF	SMD, 0805	
12	C-AREF	100 nF	SMD, 0805	
13	C-AVC	100 nF	SMD, 0805	
14	C-VCC1	100 nF	SMD, 0805	
15	C-VCC2	100 nF	SMD, 0805	
16	MCU	ATmega328-AU	TQFP	
17	R-LED1	180 $\Omega$	SMD, 0805	
18	R-LED2	180 $\Omega$	SMD, 0805	
19	R-LED3	180 $\Omega$	SMD, 0805	
20	R-LED4	180 $\Omega$	SMD, 0805	
21	LED1	LED 5 mm		$U_f = 2,2$ V, (červená)
22	LED2	LED 5 mm		$U_f = 2,2$ V, (zelená)
23	LED3	LED 5 mm		$U_f = 2,2$ V, (červená)
24	LED4	LED 5 mm		$U_f = 2,2$ V, (zelená)
25	FREE_PIN1	S1G01		oboustranný kolík, 1pin, 2,54 mm
26	FREE_PIN2	S1G01		oboustranný kolík, 1pin, 2,54 mm

Pořadové číslo	Označení součástky na DPS	Název/hodnota	Pouzdro	Poznámka
27	FREE_PIN3	S1G01		oboustranný kolík, 1pin 2,54 mm
28	FREE_PIN4	S1G01		oboustranný kolík, 1pin 2,54 mm
29	HC-SR04	S1G04		oboustranný kolík, 4pin, 2,54 mm
30	C-HC-SR04	100 nF	SMD, 0805	
31	R-TRIG	10 k $\Omega$	SMD, 0805	
32	R-ECHO	10 k $\Omega$	SMD, 0805	
33	BLUETOOTH	BL04G		dutinková lišta, 4 pin, 2,54 mm
34	C-BLUETOOTH	100 nF	SMD, 0805	
35	R-RXD	10 k $\Omega$	SMD, 0805	
36	R-TXD	10 k $\Omega$	SMD, 0805	
37	ISP	ZL202-6G		oboustranný kolík, 2x3 pin, 2,54mm
38	OC0A	BL03G		dutinková lišta, 3 pin, 2,54 mm
39	OC0B	BL03G		dutinková lišta, 3 pin, 2,54 mm
40	OC1A	BL03G		dutinková lišta, 3 pin, 2,54 mm
41	C-SERVOS	100 nF	SMD, 0805	

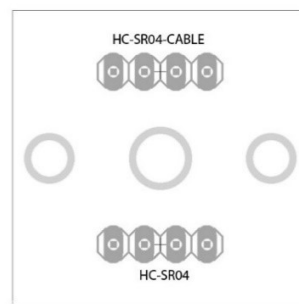
Příloha B – *Návrh DPS pro snímač*



Obrázek 1 – Schéma DPS pro snímač



Obrázek 2 - Návrh DPS pro snímač

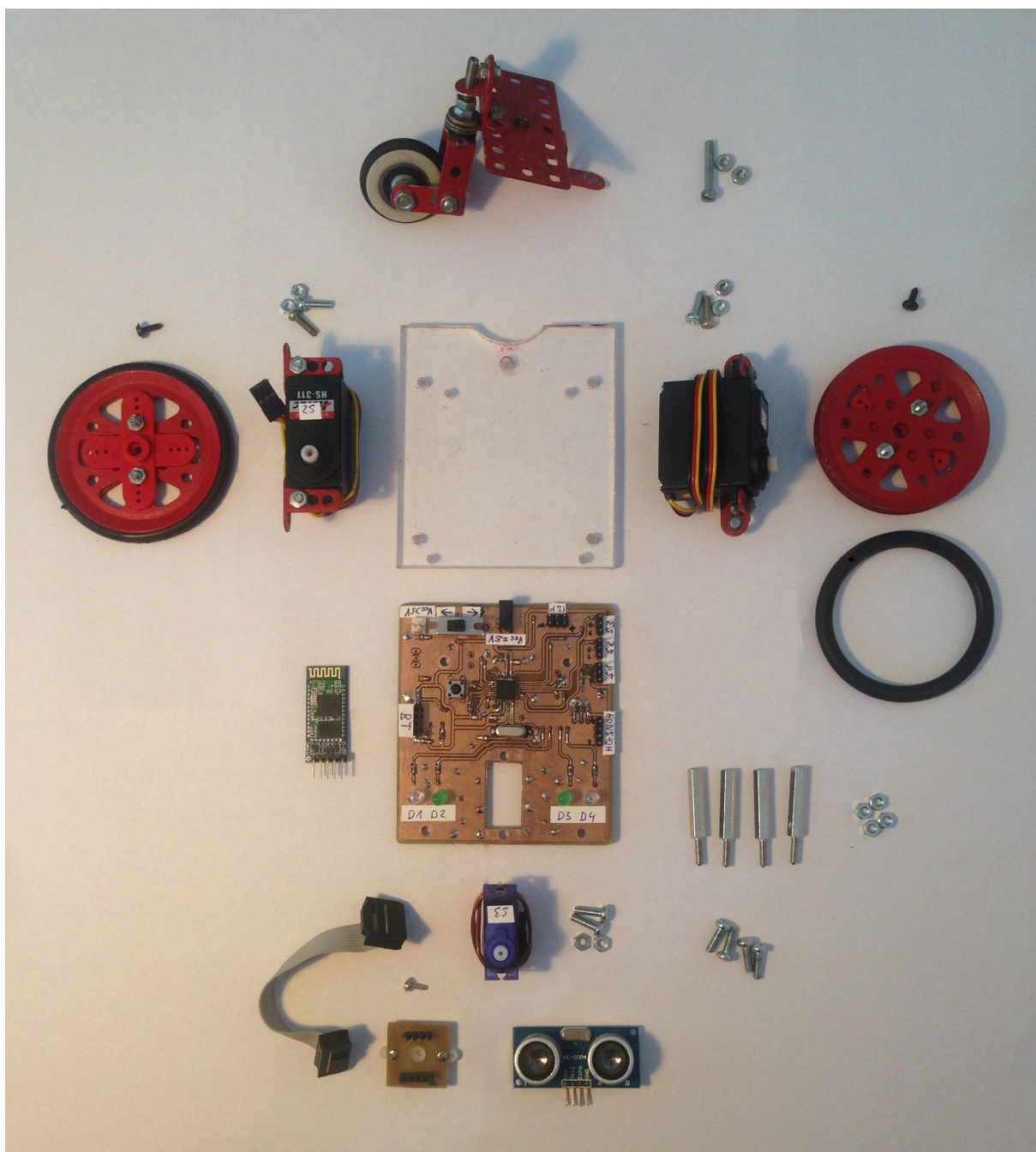


Obrázek 3 - Rozmístění součástek DPS snímače

Tabulka 1 - Seznam součástek pro DPS snímače

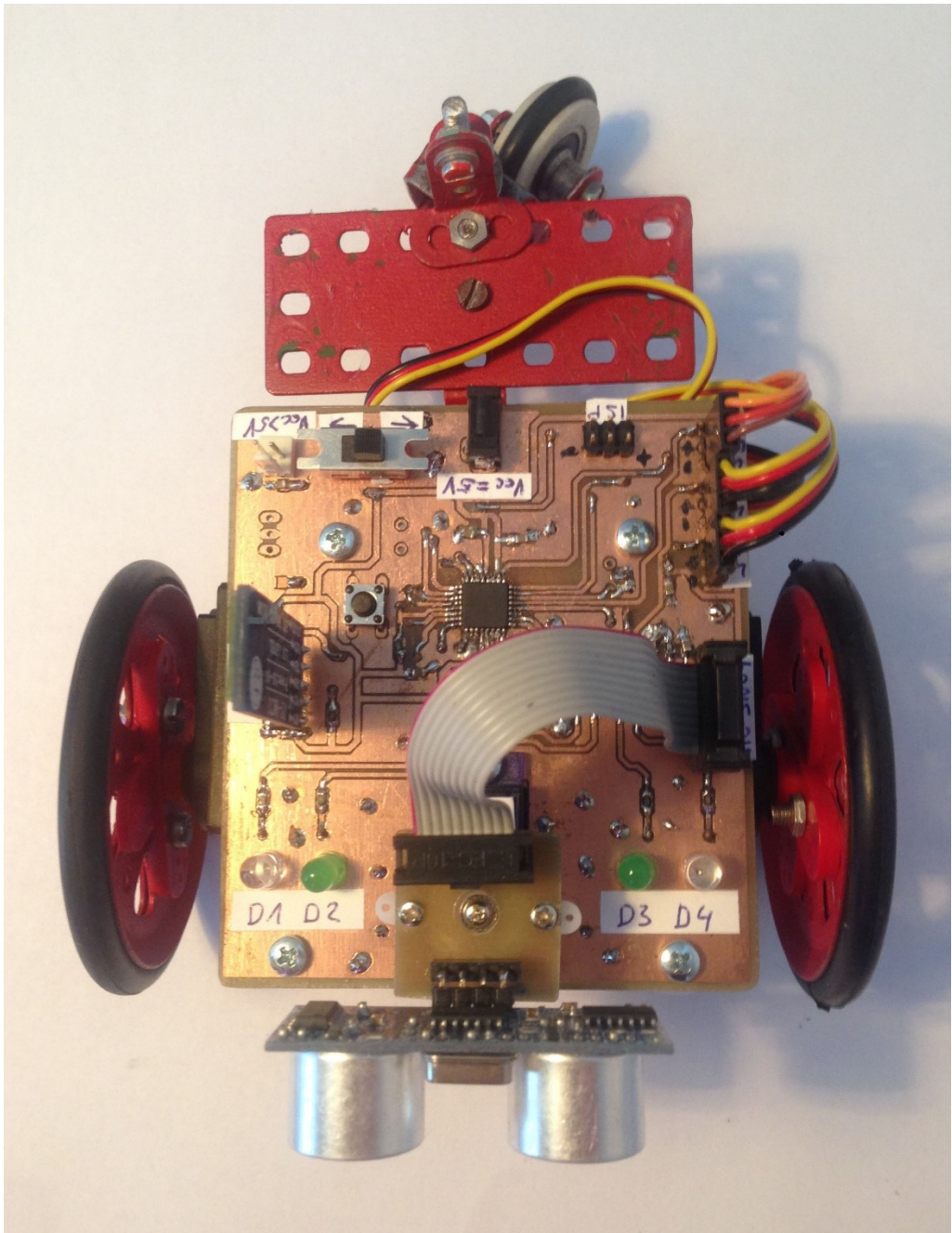
Pořadové číslo	Označení součástky na DPS	Název/hodnota	Pouzdro	Poznámka
1	HC-SR04-CABLE	S1G04		oboustranný kolík, 4 pin, 2,54 mm
2	HC-SR04	BL04G		dutinková lišta, 4 pin, 2,54 mm

Příloha C – Výsledné vozítko

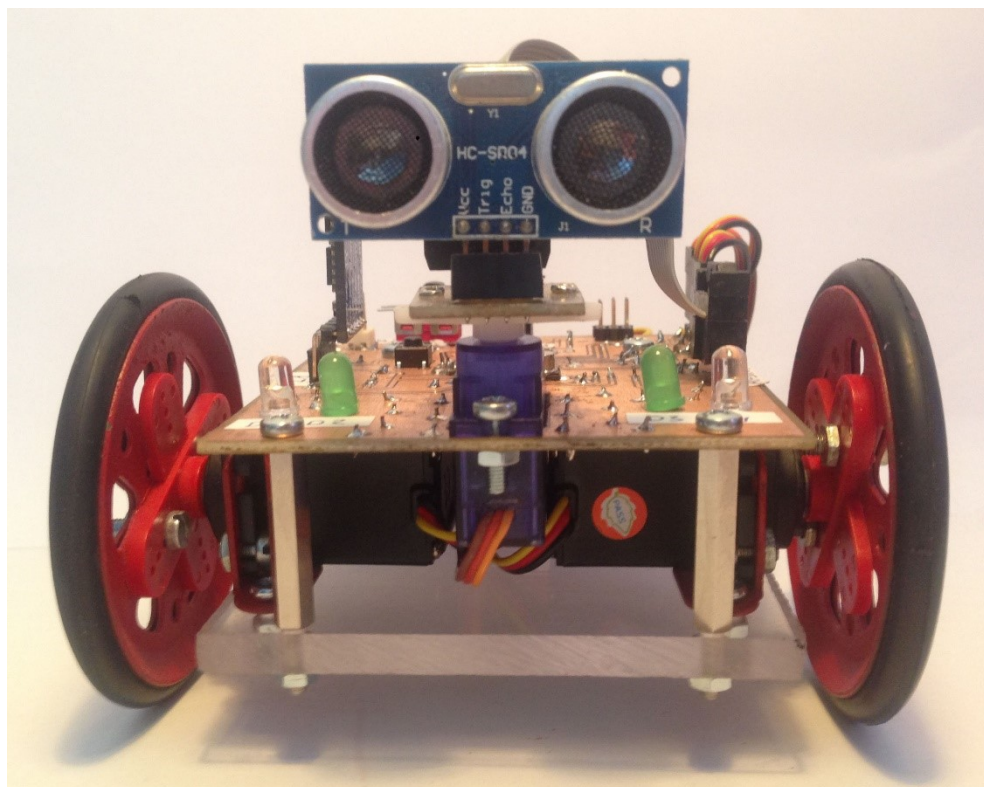


Obrázek 1 - Pohled na všechny části demontovaného vozítka

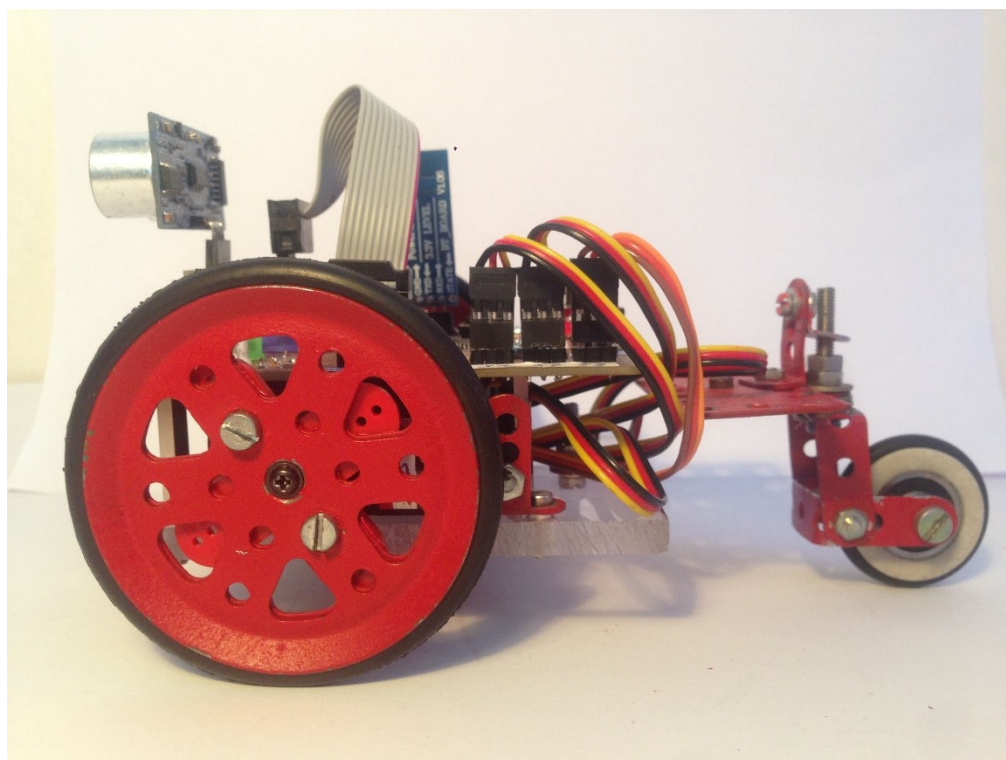




Obrázek 2 - Sestavené vozítko, pohled zvrchu

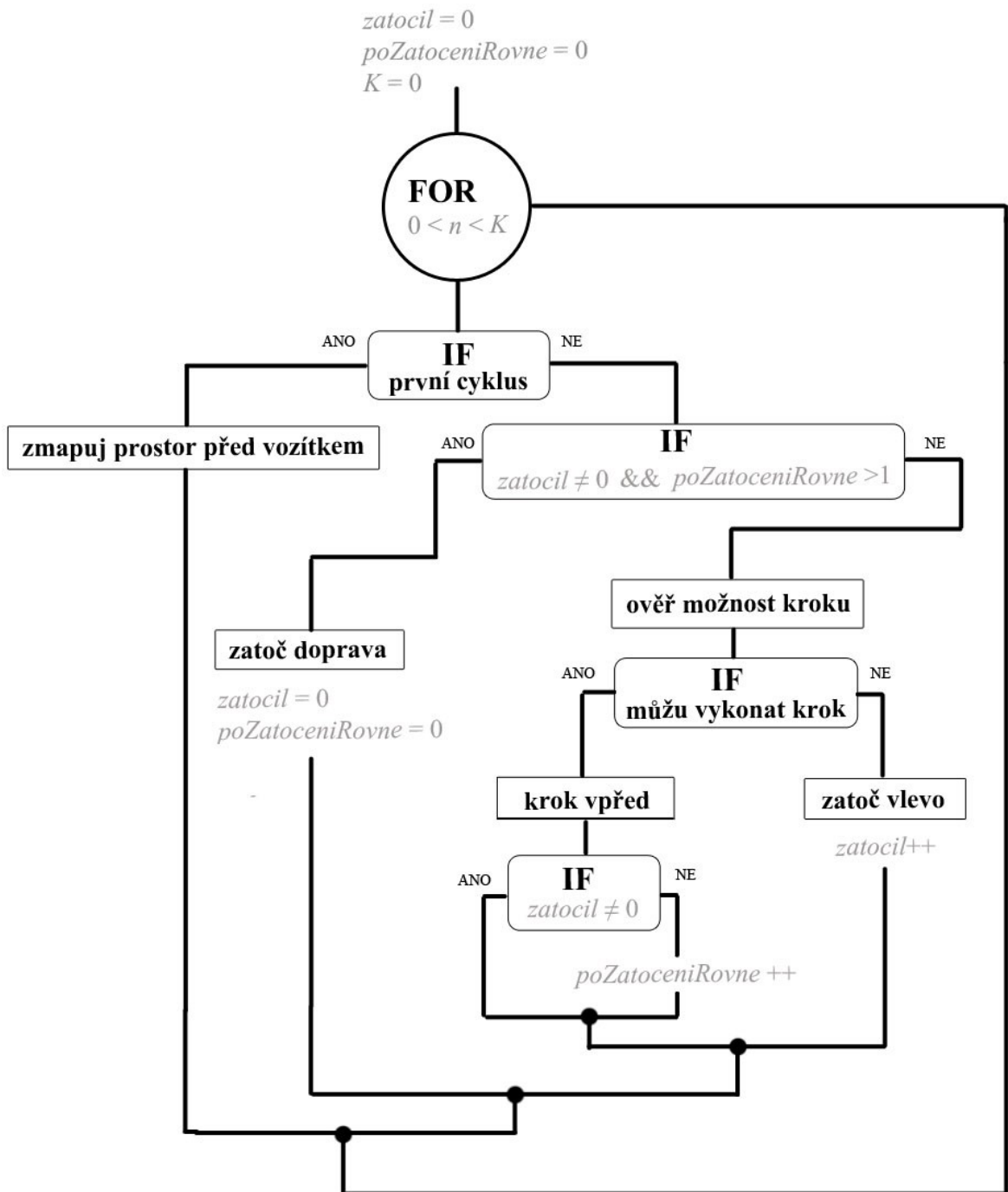


**Obrázek 3 - Sestavené vozítko, pohled zředu**



**Obrázek 4 - Sestavené vozítko, pohled z boku**

Příloha D – Blokový diagram samo mapovacího algoritmu



Obrázek 1 - Blokový diagram samo mapovacího algoritmu

Příloha E – *CD*

Obsah CD:

**DPS\_BP** – Návrhy DPS

**MCU** – Zdrojové kódy pro MCU

**MATLAB** – Zdrojové kódy PC aplikace

**JechaJ\_MapovaniUzavreneho\_PR\_2016.pdf** – Vlastní text práce