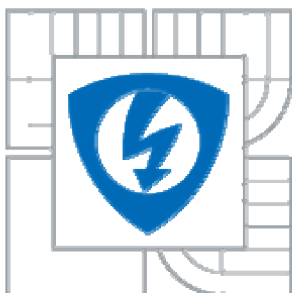




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

HARDWARE A SOFTWARE PRO MOBILNÍ ROBOTY

HARDWARE AND SOFTWARE FOR MOBILE ROBOTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

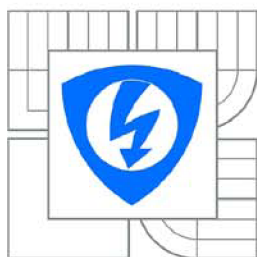
AUTOR PRÁCE
AUTHOR

GABRIEL SÁRKÖZY

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MILOŠ VESELÝ

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Gabriel Sárközy

ID: 106758

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Hardware a software pro mobilní roboty

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte software malých mobilních robotů třídy Micromouse a Path Follower. Dále prostudujte algoritmy pro prohledávání bludiště, především algoritmus Flood-Fill. Získané znalosti použijte pro modifikaci řídicího softwaru těchto robotů tak, aby lépe plnil soutěžní úlohy.

Prostudujte hardware malých mobilních robotů. Navrhněte řešení nedostatků používaného hardware robotů. Modifikujte současný hardware tak, aby dokázal lépe plnit soutěžní úlohy.

DOPORUČENÁ LITERATURA:

Joseph L. Jones, Anita M. Flynn, Bruce A. Seiger: Mobile Robots, Inspiration to Implementation. A K Peters, Ltd., 1999, ISBN 1-56881-097-0

František Šolc, Luděk Žalud: Robotika. Skriptum. VUT Brno 2002

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: Ing. Miloš Veselý

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt:

Cílem projektu je modifikovat hardware a software malých mobilních robotů, aby lépe plnily soutěžní úkoly v třídě Micromouse a Path Follower.

První část projektu zahrnuje oživení a zdokumentování aktuálního stavu robota RUBRIK, následný návrh a realizace vylepšení v mechanické, elektrické a programové části robota.

Druhá část projektu se zaměřuje na návrh a realizaci modifikací pro robota ROBIK. Oba roboti jsou vybaveni mikrokontroléry Atmel a jsou programovány v jazyku C.

Klíčová slova:

Robot, ATmega16, ATmega32, CNY70, SHARP GP2D120, Myš v bludišti, Stopař, diferenciální podvozek, Hardware, Software

Abstract:

The aim of this work is to modify hardware and software of small mobile robots, so they can be better used in Micromouse and Path Follower challenges.

The first part of the project includes debugging and creating of a correct documentation of the existing electric circuits used in the robot called RUBRIK. My work was also to design and realize mechanical, electrical and software upgrades.

The second part includes the design and realization of similar modifications of the robot ROBIK. Each robot has an Atmel AVR microcontroller. Robots are programmed in C programming language.

Key words:

Robot, ATmega16, ATmega32, CNY70, SHARP GP2D120, Micromouse, Linefollower, differential drive, Hardware, Software

Bibliografická citace

SÁRKÖZY, G. *Hardware a software pro mobilní roboty*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 51 s.
Vedoucí bakalářské práce Ing. Miloš Veselý.

P r o h l á š e n í

„Prohlašuji, že svou bakalářskou práci na téma "Hardware a software pro mobilní roboty" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujícího autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **21. květena 2010**

.....
podpis autora

P o d ě k o v á n í

Děkuji vedoucímu bakalářské práce Ing. Milošovi Veselému. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **21. květena 2010**

.....
podpis autora

OBSAH

1. ÚVOD	10
2. POPIS SOUTĚŽNÍCH PRAVIDEL STOPAŘ [4].....	11
2.1 Soutěžní úkol	11
2.2 Dráha.....	11
2.3 Činnost robota	11
2.4 Rozměry a velikost.....	12
2.5 Časové limity	12
3. POPIS SOUTĚŽNÍCH PRAVIDEL MYŠ V BLUDIŠTI [4].....	13
3.1 Soutěžní úkol	13
3.2 Bludiště	13
3.3 ROBOT – Myš.....	13
3.4 Činnost robota	14
3.5 Časové limity	15
4. SEZNÁMENÍ S MIKROKONTROLÉRY	16
4.1 ATMEL AVR ATMEGA16	17
4.2 ATmel AVR atmega32	18
4.3 A/D převodník mikrokontrolérů	18
5. SENZORICKÝ SYSTÉM.....	19
5.1 Snímání čáry	19
5.2 Snímání vzdálenosti	22
6. RUBRIK.....	24
6.1 Popis robota	24
6.2 Oživení řídicí desky	25
6.3 Způsob řízení krokových motorů	27
7. ROBIK.....	28
7.1 Popis robota	28
7.2 Způsob řízení krokových motorů	29
8. POPIS PROGRAMŮ	31
8.1 Popis programu pro bludiště	31

8.1.1	Vývojový diagram pohybu v bludišti	33
8.1.2	Popis funkcí	34
8.2	Popis programu pro sledování čáry	35
8.2.1	Popis funkcí	35
8.2.2	Vývojový diagram sledování čáry	36
8.3	Popis řízení diferenciálního podvozku [13]	37
9.	PROGRAMOVÁNÍ.....	40
10.	ZÁVĚR	41
11.	SEZNAM LITERATURY	42
12.	SEZNAMY	43
12.1	Seznam zkratk	43
12.2	Seznam příloh	44
12.3	Obsah přiloženého CD	44
13.	PŘÍLOHY.....	45

SEZNAM OBRÁZKŮ

Obrázek 1 – Blokové schéma mikrokontroléru ATMEL AVR ATmega16 [8].....	16
Obrázek 2 – Popis vývodů mikrokontroléru ATmega16 [8]	17
Obrázek 3 – Průběh kolektorového proudu na vzdálenosti [5].....	19
Obrázek 4 – Snímání čáry	20
Obrázek 5 – SHARP GP2D120	22
Obrázek 6 – Závislost výstupního napětí na vzdálenosti [6]	23
Obrázek 7 – Robot RUBRIK	24
Obrázek 8 – Blokové schéma robota	25
Obrázek 9 – Robot ROBIK	28
Obrázek 10 – Hodnoty buněk pro bludiště 5x5 beze stěn.....	31
Obrázek 11 – Hodnoty buněk pro příklad bludiště 5x5	32
Obrázek 12 – Příklad buňky se stěnou na severní a západní straně.....	33
Obrázek 13 – Vývojový diagram pohybu v bludišti	33
Obrázek 14 – Vývojový diagram sledování čáry	36
Obrázek 15 – Pohyb kola	37
Obrázek 16 – Principiální schéma robota s diferenciálním podvozkem.....	38
Obrázek 17 – Popis konektoru ISP na řídicích deskách	40

SEZNAM TABULEK

Tabulka 1 - Definované stavy snímačů čáry	21
Tabulka 2 – Popis portu A a portu B mikrokontroléru RUBRIK	26
Tabulka 3 – Popis portu C a portu D mikrokontroléru RUBRIK	26
Tabulka 4 – Popis portu A a portu B mikrokontroléru ROBIK.....	30
Tabulka 5 – Popis portu C a portu D mikrokontroléru ROBIK.....	30
Tabulka 6 – Popis zápisu stěny	33

1. ÚVOD

Cílem bakalářské práce bylo seznámit se s mikrokontroléry typu ATmega16 a ATmega32, které byly použity v malých mobilních robotech, oživit a vylepšit použité plošné spoje a zdokumentovat použité zapojení.

Dále bylo zapotřebí navrhnout rozšiřující plošný spoj pro robota RUBRIK, aby byl využitelný kromě úkolu Micromouse i pro úkol Path Follower. Pro stejného robota bylo zapotřebí změnit, kvůli rozšířené funkci Path Follower, způsob mechanické stabilizace robota v přední části. Kluzný opěrný bod musel být nahrazen dvěma všesměrovými kuličkami, aby prostor zůstal volný pro reflexní optočleny CNY70.

Původní program robota RUBRIK psaný pro mikrokontrolér ATmega8535 musel být přizpůsoben změně zapojení a následně přeportován pro mikrokontrolér ATmega16, aby co nejlépe plnil své úkoly.

Robot ROBIK byl upraven, aby využíval pouze jeden mikrokontrolér ATmega32. Sériová komunikace se již nepoužívá pro získávání informací o stavech snímačů robota, ale může se v budoucnosti použít pro komunikaci s počítačem.

Po mechanické stránce se robot rozšířil o desku se snímači čáry a snímači vzdálenosti. Použil se stejný počet a typ snímačů jako u předchozího robota. Snímače byly připojeny přes vstup/výstupní porty mikrokontroléru a nepoužila se sériová komunikace.

2. POPIS SOUTĚŽNÍCH PRAVIDEL STOPAŘ [4]

2.1 SOUTĚŽNÍ ÚKOL

Sestrojit autonomního mobilního robota, který přejde po určené dráze a v časovém limitu do cíle. Směr a trasa je dána tmavým pruhem, na dráze jsou umístěny rozličné překážky.

2.2 DRÁHA

Podklad dráhy bude světlý (bílý) s tmavou (černou) vodící čarou šířky 15+/- 1 mm. Celková délka dráhy nepřesáhne 20 m. Materiál dráhy je dřevotříska.

Nejmenší poloměr oblouků na trati bude 5 cm. Celkové převýšení nepřesáhne 3 cm, maximální stoupání a klesání je 5%. Na dráze, podobně jako v reálném světě, se můžou vyskytovat náhodné překážky (přerušená vodící čára, překážka na vodící čáře, změna osvětlení čáry a jiné).

Úkolem robota není hledat správnou cestu, ale výhradně sledovat namalovanou čáru. Dráha se v průběhu soutěže nemění, takže robot se může učit a např. rovné úseky při další jízdě přecházet rychleji.

Robot vystartuje z tzv. startovacího pole, které se nachází 30 cm před pomyslnou startovací čarou. Po přerušení světelného paprsku (10 mm nad povrchem dráhy) libovolnou částí robota se začne měřit čas. Dráha bude uzavřena (okruh) a v prostoru startovacího pole (a tedy i cíle) bude přes černou čáru nalepen kolmo proužek dlouhý cca 10 cm, který bude znamenat začátek a konec dráhy. Soutěžící, kterým by proužek vadil, mohou robota na dráhu postavit až za ním.

2.3 ČINNOST ROBOTA

Po odstartování nesmí soutěžící do činnosti robota nijak zasahovat. Robot musí být řízený výlučně vestavěnými elektronickými obvody. Musí být autonomní.

Po odstartování robot nesmí opustit dráhu vyznačenou vodící čarou s výjimkou obcházení překážky a přerušení vodící čáry. Při pohybu nesmí zanechávat žádné stopy a značky. Na dráze nesmí zůstat žádná jeho součást.

Při poškození soutěžní dráhy je robot diskvalifikován.

2.4 ROZMĚRY A VELIKOST

Šířka a výška robota nesmí přesáhnout 20 cm. Délka je bez omezení. Je třeba si uvědomit omezení dané tunelem a poloměrem oblouků na dráze.

Minimální rozměry jsou bez omezení. Hmotnost je také neomezena.

2.5 ČASOVÉ LIMITY

Na překonání dráhy je stanovený časový limit 3 minuty. Po třech minutách se měření času zastaví a pokus se zruší. To však ještě neznamená diskvalifikaci robota.

Zastavení měření času nemá vliv na ostatní pokusy.

3. POPIS SOUTĚŽNÍCH PRAVIDEL MYŠ V BLUDIŠTI [4]

3.1 SOUTĚŽNÍ ÚKOL

Navrhnout a sestrojít mikropočítačem řízený autonomní mobilní robot (myš), který dokáže přejít zadaným bludištěm do cíle v co nejkratším čase.

3.2 BLUDIŠTE

Bludiště bude sestaveno ze sítě (max. 16×16) základních čtverců s rozměry 18×18 cm. Stěny bludiště budou 5 cm vysoké a 1,2 cm silné (+/-5%). Chodbičky budou široké 16,8 cm. Vnější stěna bude uzavírat celé bludiště.

Boční stěny budou bílé, horní strana stěny bude červená. Podlaha bude ze dřeva nebo z podobného materiálu černé matné barvy. Povrch horních a bočních stěn bude odrazet infračervené světlo a povrch podlahy ho bude pohlcovat.

Start bude v jednom ze čtyřech rohů. Ve středu bludiště bude otevřená část tvořená čtyřmi jednotkovými čtverci. Tento centrální čtverec bude cílem. Vchod do tohoto čtverce bude jen jeden. Je možné, že do cíle povede víc než jedna cesta. Je třeba s tím počítat.

V bludišti se dá použít pravidlo pravé nebo levé ruky. Oba směry nemusí být rovnocenné. Určitě to však nebude nejkratší cesta do cíle. Záměrem tohoto zjednodušení bude umožnit účast i začátečníkům.

Na soutěži bude pravděpodobně použito jen bludiště v rozměru 9×9 základních čtverců. Start bude v levém spodním rohu, cíl v pravých horních čtyřech čtvercích. Použití většího bludiště se nevylučuje.

3.3 ROBOT – MYŠ

Myš musí být autonomní. Nesmí používat zdroj energie využívající spalovací proces.

Délka a šířka myši nesmí překročit 25 cm. Když myš mění své rozměry, nikde nesmí přesáhnout 25×25 cm. Výška není omezena.

Myš nesmí během cesty bludištěm nic odložit ani ztratit.

Myš nesmí skákat, překračovat nebo lézt po stěnách, rýt nebo kreslit, poškodit nebo zničit bludiště.

3.4 ČINNOST ROBOTA

Úkolem je projet ze startovacího čtverce do cílového. Tuto cestu nazveme „pokus“ a čas, který zabere, nazveme „čas pokusu“. Cesta zpátky z cílového čtverce do startovacího se nepovažuje za pokus. Měří se i celkový čas strávený v bludišti, který je pro každého soutěžícího omezen na 5 minut.

Když bude myš během jízdy bludištěm vyžadovat zásah soutěžícího, bude zásah považován za „dotknutí“ a čas pokusu bude potrestán třemi sekundami navíc.

Hodnotí se nejkratší čas pokusu ze všech, kterého daná myš dosáhne.

Když myš dosáhne střed bludiště (cíl), může ji soutěžící zdvihnout a restartovat, nebo se může samostatně vrátit na start. Zdvihnutí je samozřejmě považováno za dotyk a soutěžící dostane trojsekundovou pokutu (platnou i pro všechny další pokusy). Po dosažení cíle může myš samostatně pokračovat ve zkoumání bludiště a hledat optimální trasu.

Soutěžící nesmí mít při aktivaci myši možnost volit ani ovlivnit strategii. Po odkrytí bludiště ani při restartu nesmí soutěžící vložit myši žádnou informaci.

Čas pokusu bude měřen od okamžiku, kdy myš opustí startovací čtvereček, po okamžik, kdy vjede do cílového čtverce. Celkový čas v bludišti bude měřen od okamžiku aktivace myši. Myš se nemusí hned po aktivaci začít pohybovat. Musí být však položena do startovního čtverce a být připravena na pokus. Když se myš vrátí na start bez dosažení cíle, tento pokus se zastaví. Po opětovném startu začne měření dalšího pokusu.

Když soutěžící během soutěže vymění libovolnou část myši (baterie, EEPROM,...) nebo když udělá jinou podstatnou úpravu, musí vymazat všechny informace o bludišti, které doposud myš nasbírala. Malé úpravy (nastavení citlivosti snímačů) jsou povoleny pod dohledem porotců. Úpravy rychlosti nebo strategie jsou bez vymazání informací o bludišti zakázány.

Mezi dokončeným pokusem a startem dalšího pokusu musí myš zůstat alespoň 1 sekundu na startovním poli. Během této sekundy nesmí myš zakrývat infrasenzor časomíry.

3.5 ČASOVÉ LIMITY

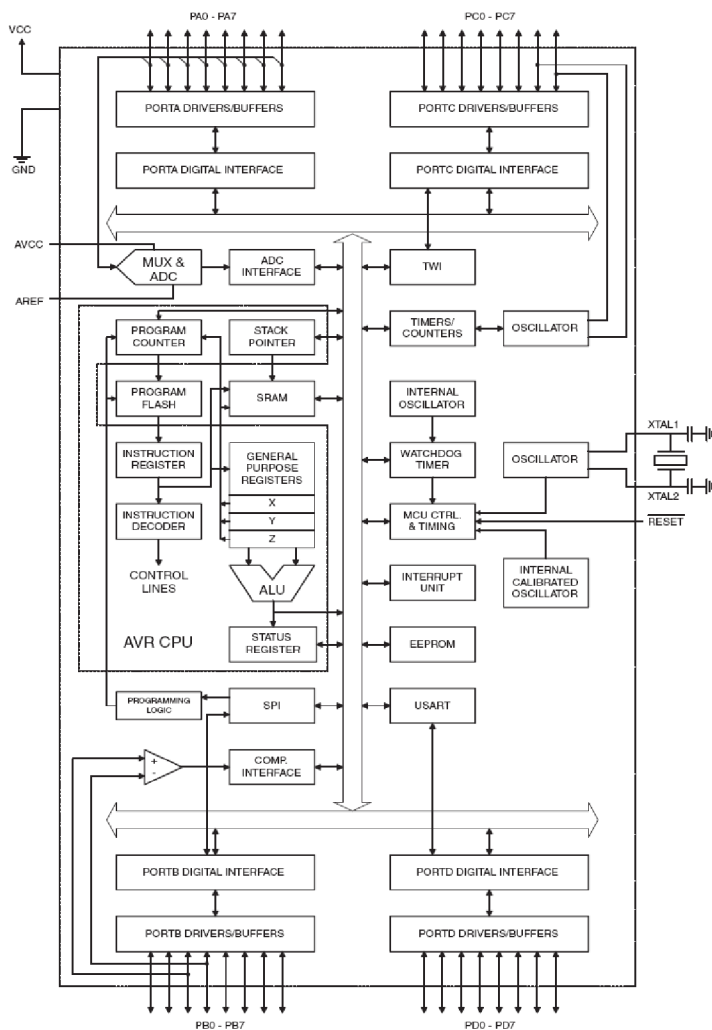
Každá soutěžící myš může strávit v bludišti nejdéle 5 minut, během nichž může absolvovat maximálně 10 pokusů.

4. SEZNÁMENÍ S MIKROKONTROLÉRY

Mikroprocesor je poměrně složitý číslicový obvod, který vykonává sled aritmetických a logických operací, podle nahraného programu. Pomocí programu docílíme toho, že se vykonávají námi požadované funkce.

Mikrokontrolér - jednočipový mikrořadič je spojením mikroprocesoru s pamětí a obvody rozhraní. Čip je opatřen příslušnými vývody, které se mohou používat k přímému řízení připojených periférií.

Mikroprocesory řady ATMEL AVR jsou nízkopříkonové 8 bitové mikrokontroléry založené na AVR RISC architektuře. Blokové schéma mikrokontroléru je znázorněno na obrázku 1.



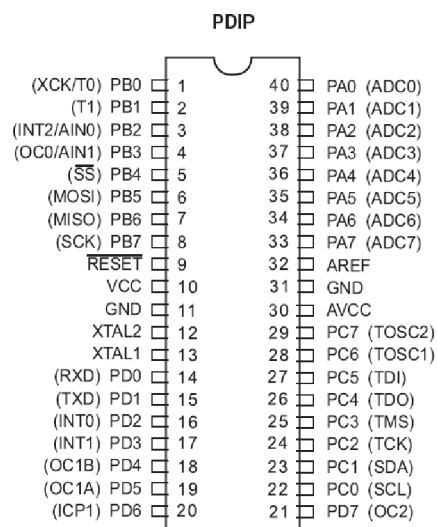
Obrázek 1 – Blokové schéma mikrokontroléru ATMEL AVR ATmega16 [8]

4.1 ATMEL AVR ATMEGA16

Mikroprocesor ATmega16 je osazen do robota RUBRIK. Řídí dva krokové motory použité pro pohyb robota pomocí driver obvodů L297 a H-můstků L298N. K analogově digitálnímu převodníku mikrokontroléru je připojena trojice IR senzorů SHARP GP2D120 a napěťový dělič pro sledování stavu baterie. Výstupy z reflexních optočlenů jsou zpracovány mikrokontrolérem, který slouží ke sledování čáry. Popis vývodů mikrokontroléru je zobrazen na obrázku 2.

Základní charakteristika:

- vnitřní oscilátor, možnost připojení externího oscilátoru s frekvencí do 16MHz
- 16k bytová interní FLASH paměť
- 1k bytů datové paměti SRAM
- 512 bytů paměti EEPROM
- 131 instrukcí
- podpora vnitřních a externích přerušení
- dva 8 bitové časovače/čítače
- jeden 16 bitový časovač/čítač
- čtyři kanály PWM
- 8 kanálový 10 bitový ADC
- pracovní napětí 4,5 V až 5,5 V
- komunikační rozhraní USART, SPI



Obrázek 2 – Popis vývodů mikrokontroléru ATmega16 [8]

4.2 ATMEL AVR ATMEGA32

Mikroprocesor ATmega32 je osazen do robota ROBIK. Řídí dva krokové motory použité pro pohyb robota pomocí driver obvodů MC3479P. K analogově digitálnímu převodníku mikrokontroléru je připojena trojice IR senzorů SHARP GP2D120. Výstupy z reflexních optočlenů CNY70 slouží pro sledování čáry. Popis vývodů mikrokontroléru je stejný jako u ATmega 16, je zobrazen na obrázku 2.

Základní charakteristika:

- vnitřní oscilátor, možnost připojení externího oscilátoru s frekvencí do 16MHz
- 32k bytová interní FLASH paměť
- 2k bytů datové paměti SRAM
- 1024 bytů paměti EEPROM
- 131 instrukcí
- podpora vnitřních a externích přerušení
- dva 8 bitové časovače/čítače
- jeden 16 bitový časovač/čítač
- čtyři kanály PWM
- 8 kanálový 10 bitový ADC
- pracovní napětí 4,5 V až 5,5 V
- komunikační rozhraní USART, SPI

Základní rozdíl mezi použitými mikrokontroléry je ve velikosti paměti. ATmega16 má poloviční velikost paměti vzhledem k ATmega32.

4.3 A/D PŘEVODNÍK MIKROKONTROLÉRŮ

Oba mikrokontroléry obsahují 10 bitový A/D převodník, který je schopen pomocí multiplexoru převádět analogové hodnoty z 8 kanálů. Vstupy multiplexoru jsou všechny piny portu C. Pomocí registru ADMUX se nastavuje kanál pro A/D konverzi, zdroj referenčního napětí a také způsob ukládání výsledku do ADCL a ADCH. Převod analogových hodnot je vyvolán přerušením v programu. Volání probíhá cyklicky s periodou $\text{clk}/128$. ADCSRA registr umožňuje povolit A/D převodník, nastavit způsob a frekvenci volání převodu a také indikuje ukončení konverze.

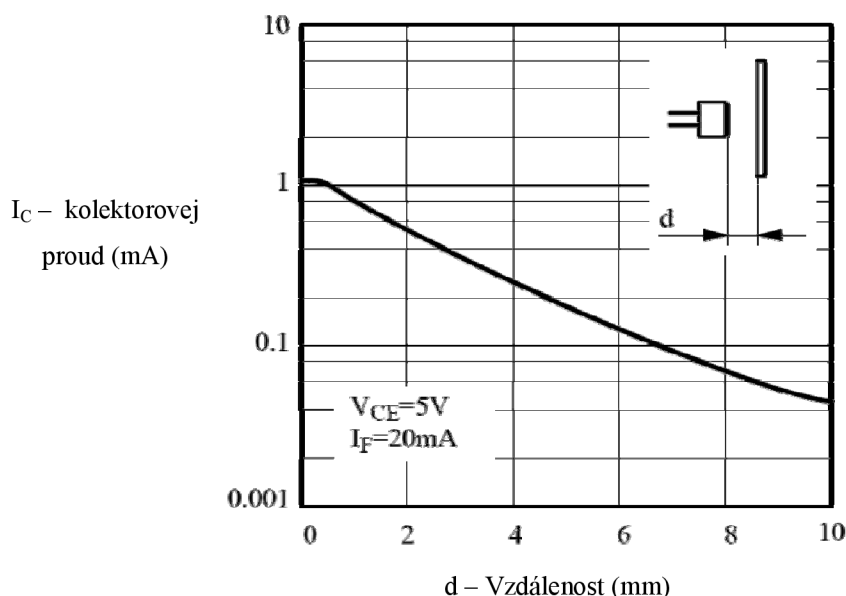
5. SENZORICKÝ SYSTÉM

5.1 SNÍMÁNÍ ČÁRY

Pro snímání čary se dá použít množství různých optických senzorů od nejjednodušších fototranzistorů až po různé CCD snímače. Nejvhodnější volbou pro tento úkol byl CNY70. Jednoduše se připojí k mikrokontroléru a snadno se nastavuje.

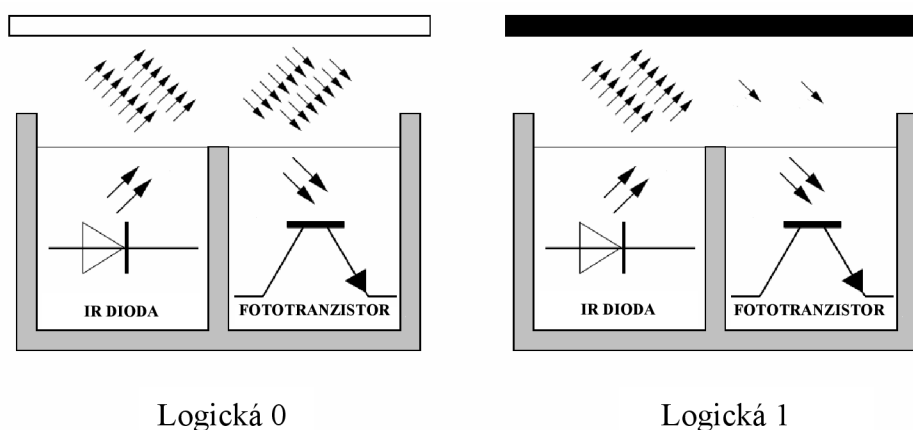
Tyto snímače nepotřebují připojení k A/D převodníku, ale mohou se s použitím minimálního počtu součástek přímo využít jako zdroje logického signálu. Obvody mají kompaktní konstrukci, obsahují infradiodu a fototranzistor. IR dioda i fototranzistor mají stejnou orientaci, tím je usnadněno nastavování snímače. Když se pod snímačem nachází černá čára, emitované světlo vlnové délky 950 nm se z větší části pohltí a fototranzistor zůstává v uzavřeném stavu. V případě, že se pod snímačem nachází pouze bílý podklad dráhy, odrazí se dostačující množství světla, aby se fototranzistor otevřel. Další výhodou odvodů CNY70 je, že mají vysoký signálový výstup, jsou málo závislé na teplotě a jsou od výroby opatřeny optickým filtrem.

Při snímání čary je důležité, aby snímače byly co nejbližší ke snímanému povrchu, což vidíme i na průběhu obrázku 3.



Obrázek 3 – Průběh kolektorového proudu na vzdálenosti [5]

Rozšiřující deska obsahuje 6 snímačů pro dokonalejší snímání. Mikrokontrolér vyhodnocuje napěťové úrovně všech snímačů, které se mění podle stavu fototranzistorů. V uzavřeném stavu je na kolektoru fototranzistoru napěťová úroveň odpovídající logické jedničce. Při otevření tranzistoru se napěťová úroveň sníží a mikrokontrolér tento stav vyhodnotí jako logickou nulu (Obrázek 4).



Obrázek 4 – Snímání čáry

Nasnímané hodnoty polohy čáry jsou mikrokontrolérem vyhodnocovány, podle počtu a polohy zakrytých snímačů. Každému stavu se přiděluje hodnota regulační odchylky. Jednotlivé stavy jsou znázorněny v tabulce 1. Dle hodnoty regulační odchylky se pomocí PD regulátoru určí akční zásah. Na základě velikosti tohoto zásahu se nastaví rychlost motorů. V případech, kdy se čára nachází jen na jednom z krajních snímačů, má regulační odchylka velikost -5 nebo 5. V tomto stavu se musí předejít ztrátě čáry, proto se blokuje jedno z kol. Motory mohou být buzeny jen určitou maximální frekvencí. Překročení maximální rychlosti motoru se testuje. Motoru se vždy přiřadí jen maximální dovolená rychlost. Stejně omezení platí pro minimální rychlost motorů.

Robot se snaží dosáhnout nulové regulační odchylky. Při této hodnotě se čára nachází uprostřed a oba motory mají stejnou rychlost.

Tabulka 1 - Definované stavy snímačů čáry

Regulační odchylka	Senzor 1	Senzor 2	Senzor 3	Senzor 4	Senzor 5	Senzor 6
5	1	0	0	0	0	0
4	1	1	0	0	0	0
3	0	1	0	0	0	0
2	0	1	1	0	0	0
1	0	0	1	0	0	0
0	0	0	1	1	0	0
-1	0	0	0	1	0	0
-2	0	0	0	1	1	0
-3	0	0	0	0	1	0
-4	0	0	0	0	1	1
-5	0	0	0	0	0	1

5.2 SNÍMÁNÍ VZDÁLENOSTI

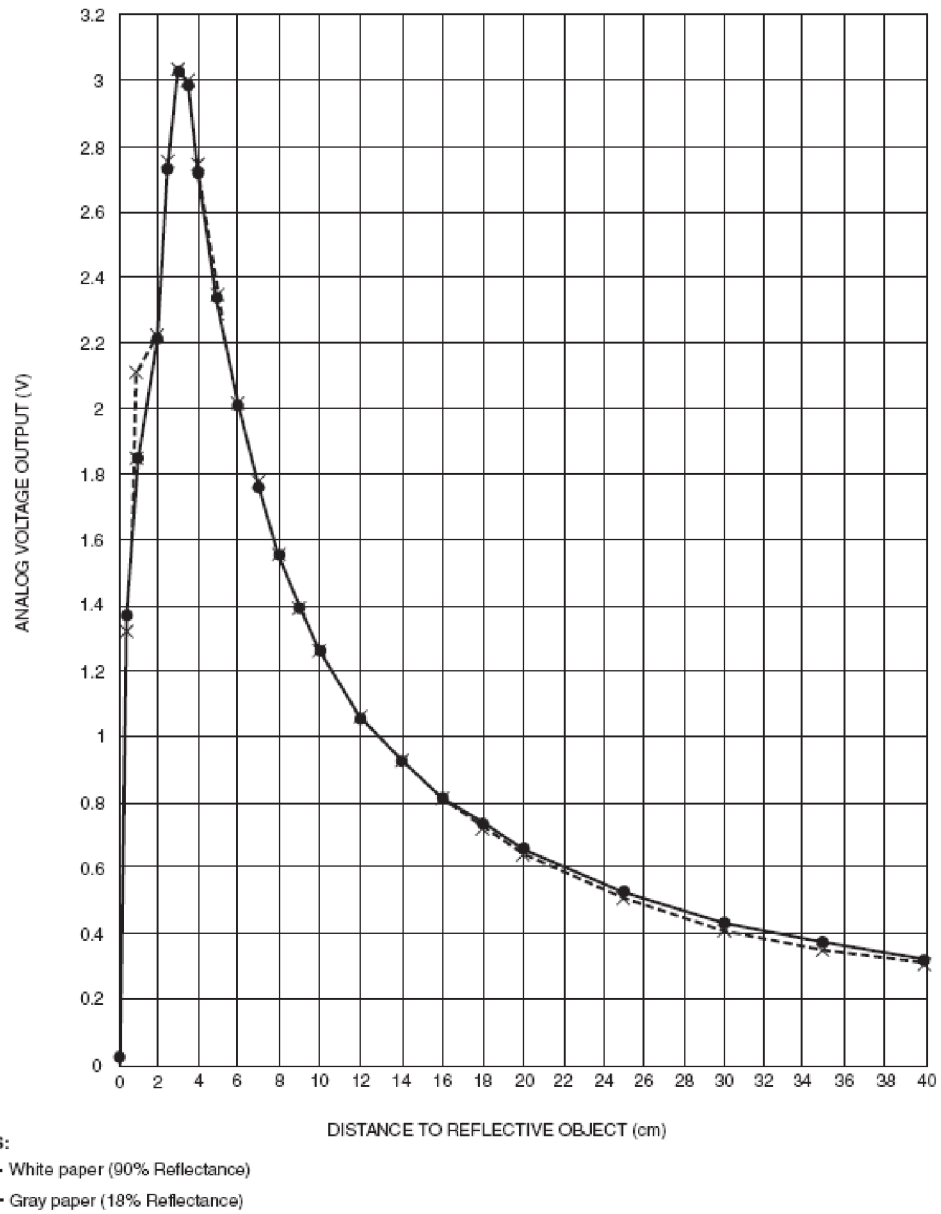
Pro snímání vzdálenosti se používá SHARP GP2D120 (Obrázek 5). Tento snímač vzdálenosti obsahuje integrované signálové zpracování a je opatřen analogovým napěťovým výstupem. Efektivní rozsah je od 4 do 30 cm, což vidíme na závislosti výstupního napětí na vzdálenosti na obrázku 6. Výhodou z hlediska použití v mobilním zařízení s vlastním napájecím zdrojem je nízká spotřeba průměrně 33 mA. Další výhodou je nízká úroveň zkreslení měření vzdálenosti za různých světelných podmínek a při různé odrazivosti měřeného předmětu. Snímače nepotřebují žádné další vnější obvody. Typickým časem odezvy snímače je 39 ms.

Vzdálenost je snímána ve třech směrech vpředu, vpravo a vlevo, proto je třeba použít tři snímače. Vývody snímačů jsou přivedeny na vstupy 10 bitového A/D převodníku. Mikrokontrolér pracuje pouze s 8 bitovou hodnotou A/D konverze, protože nejnižší dva bity jsou zašuměné. Vypovídající schopnost 8 bitové hodnoty je postačující pro plnění úkolů.

Snímače SHARP GP2D120 pracují na principu triangulace. Základem je vyslání pulzu z emitoru IR světla. Když se před snímačem nenachází žádný předmět, světlo se neodráží. V případě, že se světlo odrazí, vytvoří se trojúhelník mezi zdrojem světla, předmětem a PSD. PSD je CCD pole, které vyhodnotí úhel dopadu. Podle velikosti úhlu se zjistí vzdálenost. Větší úhel znamená bližší předmět a menší úhel vzdálenější předmět. Tato metoda je téměř imunní vůči ambientnímu světlu.



Obrázek 5 – SHARP GP2D120



Obrázek 6 – Závislost výstupního napětí na vzdálenosti [6]

6. RUBRIK

6.1 POPIS ROBOTA

Robot RUBRIK byl původně navržen pro úkol Myš v bludišti. Pro potřeby plnění obou soutěžních úkolů se nahradil senzorický systém a řídicí deska. Upravila se mechanická stabilizace robota v přední části. Kluzný opěrný bod musel být nahrazen dvěma všesměrovými kuličkami, aby prostor zůstal volný pro reflexní optočleny.

Rozložení snímačů je patrné z obrázku 7. Snímače vzdálenosti jsou umístěny co nejblíže ke středu osy robota. Rozsah měření pro GP2D120 se udává od 4 do 30 cm. Překážky blíže než 4 cm by se mohly špatně vyhodnotit. Aby se předešlo této chybě, snímače jsou umístěny dále od kraje.



Obrázek 7 – Robot RUBRIK

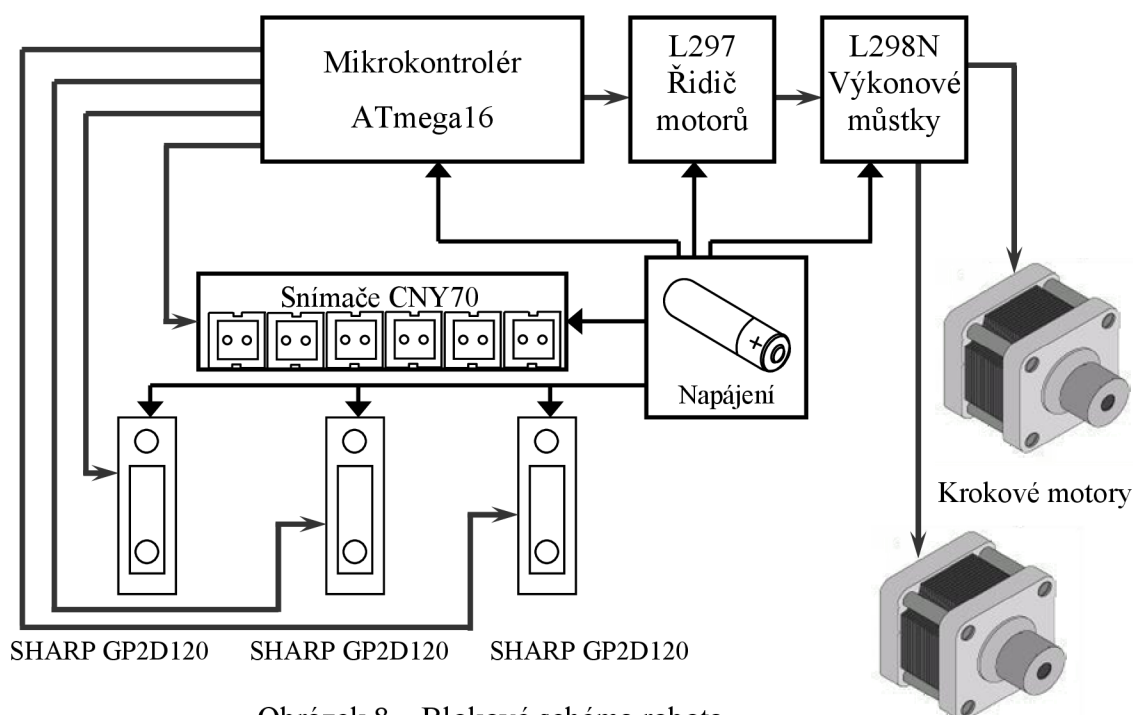
Snímače čáry CNY70 se umístily v přední části pod snímači vzdálenosti. Pro reflexní optočleny je důležité, aby byly co neblíže ke zkoumanému povrchu. Při vyšší vzdálenosti měřeného povrchu nastává problém s vnějším osvětlením, které může mít vliv na množství světla dopadajícího na fototranzistor.

6.2 OŽIVENÍ ŘÍDICÍ DESKY

Řídicí deska robota byla původně navržena s mikrokontrolérem ATmega8535. Z důvodu nedostatku interní paměti se vytvořila úplně nová řídicí deska s novějším mikrokontrolérem ATmega16, který byl již popsán v předchozí kapitole.

Napájení robota zajišťuje 10 AA článku Ni-MH. Řídicí deska má rozvod napájení 5V a 12V, které jsou jistěny pojistkou. Stabilních 5V zajišťuje spínaný zdroj, který je realizován pomocí LM2576. Jako reference analogových vstupů A/D převodníků slouží zapojení s obvodem TL431. Je to stabilizátor napětí s nastavitelným výstupním napětím. Výstupní napětí je nastaveno pomocí napěťového děliče sestávajícího z rezistorů R2 a R3.

Deska je vybavena 2 mikrosvínači a 3 jumpery na portu B využitelnými pro další softwarové zpracování. Pro indikaci je možné použít čtyři LED diody připojené na port A.



Obrázek 8 – Blokové schéma robota

Všechny volné vstupně/výstupní piny jsou vyvedeny na konektory podle níže uvedených tabulek 2 a 3.

Základní spojení a systém, jakým funguje celý robot, jsou znázorněny na obrázku 8. Zde vidíme, že mikrokontrolér na řídicí desce musí vyhodnocovat signály ze snímačů vzdálenosti a také ze snímačů čáry. Podle stavů snímačů se řídí jednotlivé motory tak, aby robot plnil svůj úkol.

Tabulka 2 – Popis portu A a portu B mikrokontroléru RUBRIK

Port A		Port B	
PA0	Levé čidlo SHARP	PB0	Tlačítko 1
PA1	Pravé čidlo SHARP	PB1	Tlačítko 2
PA2	Střední čidlo SHARP	PB2	Jumper
PA3	Měření napájecího napětí přes dělič R4:R9 (47:10)	PB3	Jumper
PA4	LED1	PB4	Jumper
PA5	LED2	PB5	MOSI
PA6	LED3	PB6	MISO
PA7	LED4	PB7	SCK

Tabulka 3 – Popis portu C a portu D mikrokontroléru RUBRIK

Port C		Port D	
PC0	SCL (pro SPI)	PD0	RXD
PC1	SDA (pro SPI)	PD1	TXD
PC2	SENZOR1 (nejvíc vpravo)	PD2	ENABLE
PC3	SENZOR2	PD3	NOT_RESET
Pc4	SENZOR3	PD4	SMER_R
PC5	SENZOR4	PD5	CLOCK_L
PC6	SENZOR5	PD6	SMER_L
PC7	SENZOR6 (nejvíc vlevo)	PD7	CLOCK_R

6.3 ZPŮSOB ŘÍZENÍ KROKOVÝCH MOTORŮ

Krokové motory se řídí pomocí L297 řidičů krokových motorů a L298N duálních plných můstků. Způsob řízení je bipolární a pomocí vstupů HALF/FULL je zvolen režim řízení normální. Tento režim se také nazývá dvoufázový, protože jsou v jeden okamžik buzeny vždy dvě fáze, což zaručuje nepřetržitě moment motoru. Velikost kroku je zvolena plná, to odpovídá $1,8^\circ$.

Pro řízení motorů se využívá port D mikrokontroléru. Pro povolení řízení je potřeba přivést logickou 1 na vývod ENABLE obvodu L297. Při logické hodnotě 0 jsou motory volné, bez napájení. Pomocí vývodů CW/CCW se určuje směr otáčení motoru. Při nulové hodnotě je směr otáčení proti hodinovým ručičkám a při nastavené hodnotě 1 je směr reverzovaný. Správnost otáčení závisí také na způsobu připojení cívek motoru. Směr otáčení se může volit pro každý motor pomocí proměnných SMER_L a SMER_R. Pomocí výstupu RESET se může translátor generující fázové sekvence motoru uvést do výchozí pozice. Pro ovládní rychlosti otáčení kol se používají vývody CLOCK. Přivedené hodinové signály se generují v přerušení zvláště pro pravý a levý motor.

Výhodou oproti řízení motorů robota ROBIK pomocí MC3479P je možnost vypnutí ENABLE při dokončení úkolu.

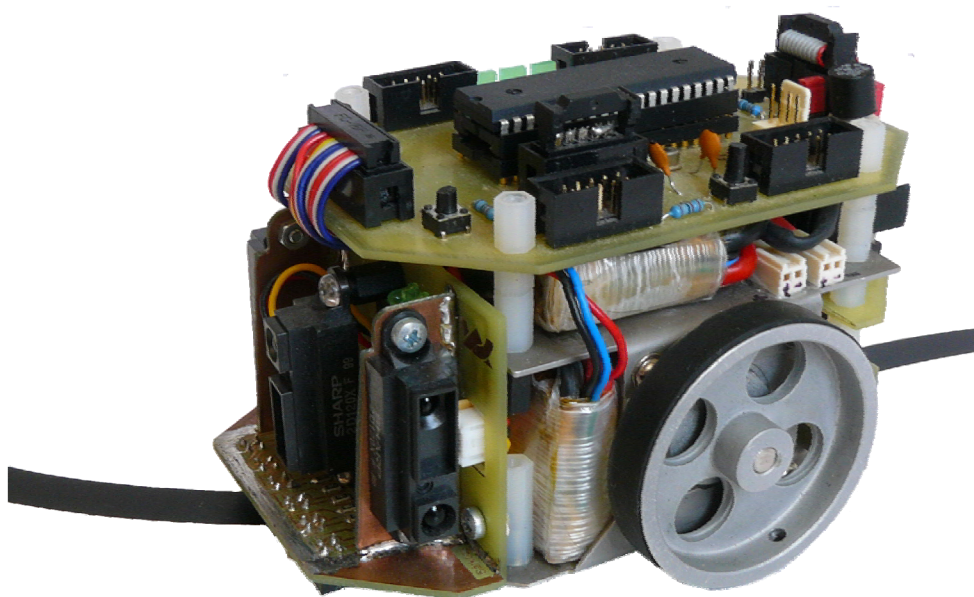
Velikost kol robota RUBRIK je 60 mm, používají se krokové motory s plným krokem $1,8^\circ$. Robot při jednom kroku ujede zhruba 1,9 mm.

7. ROBIK

7.1 POPIS ROBOTA

Robot ROBIK byl navržen pro úkol Stopař. Snímání čáry bylo nespolehlivé a zbytečně komplikované. Proto se tato část nahradila mechanicky robustnějším a jednodušším řešením. Uvolnil se sériový port a snímače čáry se připojily přímo na vstupně/výstupní porty. Pro rozšíření možnosti robota vykonávat úkol Myš v bludišti se přidaly další tři snímače vzdálenosti SHARP GP2D120. Snímače slouží nejen pro zjišťování přítomnosti stěn v bludišti, ale také pro detekci překážek při sledování vodící čáry. Na obrázku 9 je zobrazen robot ROBIK s aktuálním rozložením snímačů.

Na čelní stranu robota se pro jednodušší ladění programu a indikaci stavu snímačů vzdálenosti přidaly tři LED diody. Program zajišťuje rozsvícení jednotlivých diod podle hodnoty A/D převodníku. Diody jsou rozmístěny horizontálně vedle sebe. Při detekci překážky se rozsvítí dioda, která svojí polohou odpovídá aktivovanému snímači.



Obrázek 9 – Robot ROBIK

Blokové schéma je podobné jako u předchozího robota, s tím rozdílem, že pro řízení krokových motorů se používá pouze dvojice obvodů MC3479P. Dalším rozdílem je použitý mikrokontrolér ATmega32.

Použité Li – Pol akumulátory jsou zapojeny do série. O napájení 5V okruhu se stará spínaný stabilizátor LM2576 stejně jako u předchozího robota. Celý okruh napájení je chráněn pojistkou.

Základem robota je diferenciální podvozek. Tento typ podvozku je ideální pro plnění stanovených úkolů. Pomocí dvou nezávisle hnaných kol se robot snáze ovládá. Použití krokových motorů zajišťuje přesnější pohyb. Diferenciální podvozek je nevhodný do terénu a kvůli nízko uloženým snímačům čáry těžko překonává stoupání povrchu.

Po mechanické stránce je tento podvozek výhodnější, protože je užší. Z toho důvodu má větší volnost při pohybu v bludišti. V porovnání s předchozím robotem jsou použita kola menší. Rovněž rozteč kol je menší.

Na řídicí desce je kromě rozhraní USART vyvedeno také rozhraní I²C. Každý port je zpřístupněn přes samostatný konektor pro další možné rozšíření.

7.2 ZPŮSOB ŘÍZENÍ KROKOVÝCH MOTORŮ

Obvody MC3479D slouží pro řízení krokových motorů. Pro řízení dvou motorů je zapotřebí použít dva obvody, pro každý motor jeden. Velikost kroku je nastavena pomocí vývodu Full/Half Step. Je přiveden na GND, tím se zvolil plný krok. Rychlost motoru je ovládána pomocí vývodu Clock. Hodinové signály jsou generovány v přerušeních programu podobně jako u předchozího robota. Volba směru je podobná jako u obvodu L297 vývodem CW/CCW.

Nevýhodou tohoto řešení je, že obvody neposkytují možnost vypnutí motoru. Po přivedení napájecího napětí je motor neustále v záběru. Proto se napájení motorů a řídicí desky zapíná manuálně přepínačem na napájecí částí. Přepínač má tři polohy vypnutí, zapnutí motorů i řídicí desky a zapnutí pouze řídicí desky bez motorů. Poslední volba je vhodná zejména pro ladění programu.

Velikost kol je 50 mm. Z toho vyplývá, že při plném kroku $1,8^\circ$ se robot posune o 1,5 mm na jeden krok. Jedno plné otočení kola je 200 impulzů.

Tabulka 4 – Popis portu A a portu B mikrokontroléru ROBIK

Port A		Port B	
PA0	Pravé čidlo SHARP	PB0	SENZOR5
PA1	Střední čidlo SHARP	PB1	SENZOR6 (nejvíc vlevo)
PA2	Nevyužito	PB2	SENZOR3
PA3	Levé čidlo SHARP	PB3	SENZOR4
PA4	Tlačítko 2	PB4	SENZOR1 (nejvíc vpravo)
PA5	LED pro signalizaci překážky vpravo	PB5	SENZOR2
PA6	Nevyužito	PB6	Nevyužito
PA7	LED1	PB7	Tlačítko 1

Tabulka 5 – Popis portu C a portu D mikrokontroléru ROBIK

Port C		Port D	
PC0	LED pro signalizaci překážky ve střede	PD0	RXD
PC1	LED pro signalizaci překážky vlevo	PD1	TXD
PC2	SMĚR_R	PD2	Nevyužito
PC3	CLOCK_R	PD3	Nevyužito
Pc4	CLOCK_L	PD4	Nevyužito
PC5	SMĚR_L	PD5	Nevyužito
PC6	LED1	PD6	Nevyužito
PC7	LED2	PD7	Nevyužito

8. POPIS PROGRAMŮ

Základem správné činnosti autonomního robota je dobré navržení algoritmu a následně program napsaný dle tohoto algoritmu.

Celý zdrojový kód byl napsán a odladěn v prostředí AVR Studio 4 (IDE). Jedná se o volně stažitelný software ze stránek výrobce námi použitého čipu. Je to prostředí, které je určeno nejen k samotnému psaní a překladač kódu, ale umožňuje také simulaci kódů a zobrazení registrů procesoru během simulace pro lepší orientaci a kontrolu napsaného programu. AVR Studio umožňuje překlad z jazyku C nebo assembler.

8.1 POPIS PROGRAMU PRO BLUDIŠTĚ

Pro soutěžní úkol Robot v bludišti byl vybrán jako nejlepší algoritmus nalezení cílové buňky algoritmus Flood-Fill. Tento algoritmus hledá nejkratší cestu na základě hodnot jednotlivých buněk. Každá buňka má svou hodnotu dle počtu kroků – počet buněk s nižší hodnotou kroku, kterými robot ještě musí přejít, aby se dostal do cíle. Cílová buňka má hodnotu 0 a startovní při rozměru bludiště $N \times N$ buněk je $N-1$. Kdyby buňka, ve které se robot nachází, měla hodnotu 3, robot by v nejlepším případě musel přejít 3 buňky, aby se dostal nejkratší cestou do cíle.

	0	1	2	3	4
0	4	3	2	3	4
1	3	2	1	2	3
2	2	1	0	1	2
3	3	2	1	2	3
4	4	3	2	3	4

Obrázek 10 – Hodnoty buněk pro bludiště 5x5 beze stěn

14	13	2	3	4
15	12	1	4	5
16	11	cíl	5	8
17	10	7	6	7
START	9	8	7	8

Obrázek 11 – Hodnoty buněk pro příklad bludiště 5x5

Základem pro nejrychlejší nalezení cílové buňky je znalost trasy, kterou se robot k požadované buňce dostaneme. Proto je třeba při pohybu robota zaznamenávat polohy stěn, podle kterých se později můžeme rychleji orientovat. Zjištění přítomnosti stěn mají za úkol snímače SHARP. Při projíždění buňkou se ukládají poznatky o přítomnosti stěn do paměti. Každá stěna se projeví ve dvou buňkách, proto se aktualizuje nejen buňka, ve které se robot nachází, ale také sousední buňky.

Zaznamenávají se přítomnost západní, jižní, východní a severní stěny do čtyř bitů podle tabulky 6, logická 1 reprezentuje přítomnost stěny a 0 volný průchod mezi buňkami. Například buňka se stěnou na severní a západní straně (Obrázek 12) by měla hodnotu paměti 0x09 hexadecimálně. Při točení podvozku se posouvá pole čtyř bitů podle otočení, aby se zjistilo, že např. čelní snímač už nesnímá severní stěnu, ale východní (natočení o 90° vpravo).

Po zjištění stavu stěn se aktualizuje pole buněk a vypočte se nejkratší aktuální trasa, po které bude robot pokračovat.

Pro nalezení trasy k cílové buňce není třeba znát mapu celého bludiště. Další výhodou tohoto algoritmu je, že najde cíl i v bludišti s ostrovy. Nevýhodou je potřeba vytváření mapy bludiště.

Flood-Fill algoritmus umí pracovat ve dvou módech. Je to mód hledání cíle a návratový mód. Při hledání cíle je hodnota startovní buňky nejvyšší a hodnota

cílové buňky je rovna nule. Naopak při návratovém módu je robot už v cíli a potřebuje se vrátit na start, proto má startovní buňka, odkud robot začal hledat cíl, teď prioritu cíle, tedy hodnotu nula a původní cílová buňka, ve které se robot nachází, má nejvyšší hodnotu.

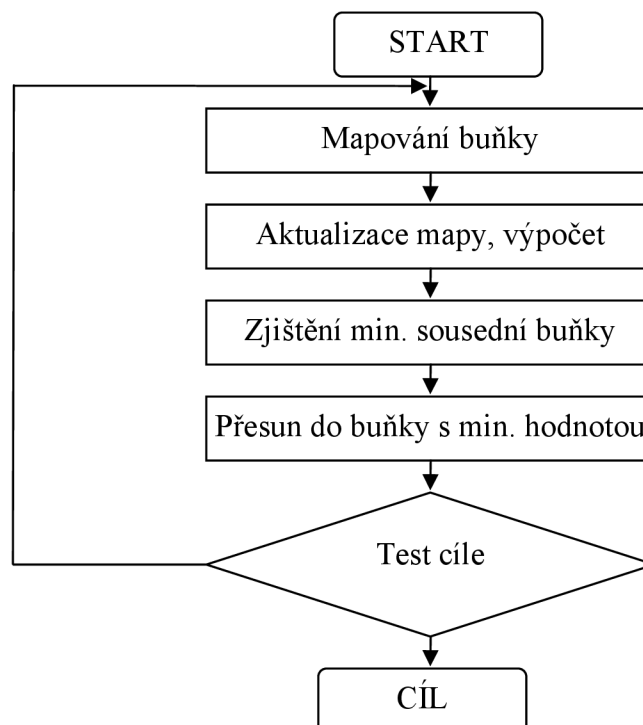
Tabulka 6 – Popis zápisu stěny

Směr	Západ	Jih	Východ	Sever
Pořadí bitu	3	2	1	0



Obrázek 12 – Příklad buňky se stěnou na severní a západní straně

8.1.1 Vývojový diagram pohybu v bludišti



Obrázek 14 – Vývojový diagram pohybu v bludišti

8.1.2 Popis funkcí

Funkce pro pohyb:

- otoc_vlevo()* - otočení vlevo o 90°
- otoc_vpravo()* - otočení vpravo o 90°
- otoc_180()* - otočení vpravo o 180°
- vpred()* - pohyb vpřed o jeden čtverec, pro udržení ve středu buněk se odsud volá regulace
- regulace()*- vypočte periodu_lmotoru a periodu_rmotoru podle regulační odchyšky e , použit je PD regulátor

Funkce pro Flood-Fill:

- init_pole()* - inicializuje hodnoty buněk v poli *pole[][]*
- mapuj()* - aktualizace mapovaných zdí v aktuálních a v sousedních buňkách, pracuje s *maze[][]*
- zapis_zdi(i, j, zdi)* - pro šetření paměťového místa se v lbytu buňky *maze[][]* zapisují dvě hodnoty stěn, do horních čtyř bitů sudé sloupce a do dolních čtyř bitů liché sloupce
- pruzkum_mode()* - v poli *pole[][]* se inicializuje cíl hodnotou buňky = 0, robot jede k cíli
- navrat_mode()* - v poli *pole[][]* se inicializuje startovní buňka hodnotu 0, robot jede ke startu
- minimum(i, j)* - k buňce *pole[i][j]* nalezne sousední volnou buňku s minimální hodnotou, výstupem je poloha a směr k buňce s minimální hodnotou, při shodných vzdálenostech preferuje směr rovně
- floodfill()* - aktualizuje pole *pole[][]* hodnotami pomocí *minimum(i,j)*, přiřazuje nové hodnoty buňkám podle přítomnosti stěn
- pohyb()* - pohne se na sousední volné pole s minimální hodnotou,

Pole maze[][] obsahuje informace o stěnách buněk v bludišti a pole pole[][] obsahuje informace o hodnotě buněk v bludišti.

Informace o stěnách se získávají při přerušení od A/D převodníku. V tomto přerušení SIGNAL(SIG_ADC) se také zabezpečuje přepínání kanálů A/D převodníku a vypočítává se hodnota regulační odchylky e .

Program obsahuje také dvě přerušení od 8bitového čítače/časovače pro nastavování rychlosti otáčení levého a pravého motoru. Pro levý motor slouží SIGNAL(SIG_OVERFLOW0) a pro pravý motor SIGNAL(SIG_OVERFLOW2).

8.2 POPIS PROGRAMU PRO SLEDOVÁNÍ ČÁRY

Robot při úkolu sledování čary (line follower) využívá šesti snímačů CNY70. Podle polohy čary se zjistí regulační odchylka e PD regulátoru, který zpomaluje nebo zrychluje motory. Cílem je udržet čaru na dvou středních snímačích. Když se čara nachází na některém z krajních senzorů, blokuje se motor na jeho straně, aby se zamezilo ztrátě čary.

Robot by měl být schopen detekovat překážku a vyhnout se jí. Následně po jejím objetí by měl nalézt čaru a pokračovat v cestě. Pro případ přerušení čary musí být robot také schopen pokračovat dál a znovu nalézt vodící čaru. Musí se zabezpečit, aby robot po ztrátě čary zastavil a nevyjel z dráhy.

8.2.1 Popis funkcí

Funkce pro pohyb otoc_vlevo(), otoc_vpravo(), otoc_180() a vpred() jsou stejné jako v případě algoritmu pro bludiště.

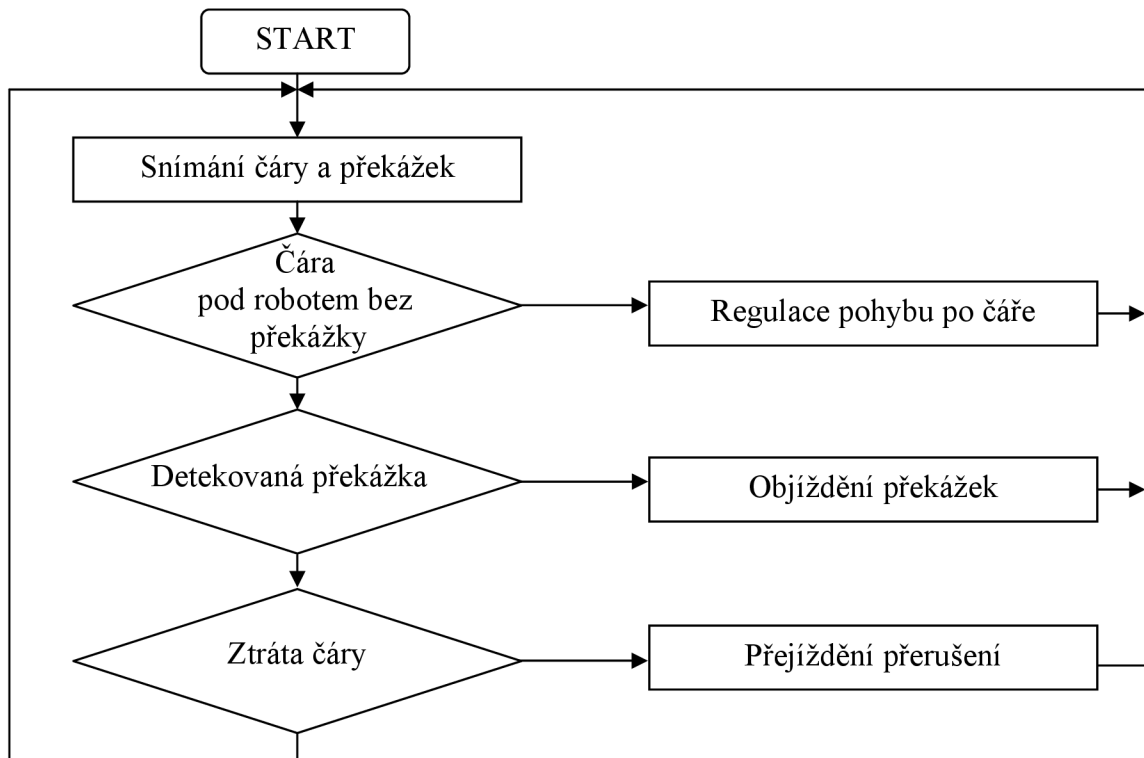
Funkce pro stopování:

- stopuj()* - pohyb vpřed o jeden Step (konstantní počet kroků)
- stopuj_s()* - pomalejší pohyb vpřed, využití při přechodu přerušení a přiblížení k překážce
- precti_cidla()* - zjištění polohy černé čary, snímání překážek
- regulace()* - podle polohy se vypočtou rychlosti obou motorů
- regulace_s()* - stejný výpočet pro pomalejší běh

objed_cihlu() - robot po zjištění překážky zpomalí a následně se otočí vpravo o devadesát stupňů, pokračuje vedle cihly a pak se otočí o devadesát stupňů vlevo a po zadaném počtu kroků vykoná stejné otočení vlevo, pak už jen najde čáru a otočením vpravo se dostane do správné pozice pro pokračování ve stopování

posun_vpřed_tehla(step) – použití při obcházení překážky pro rovné úseky

8.2.2 Vývojový diagram sledování čáry

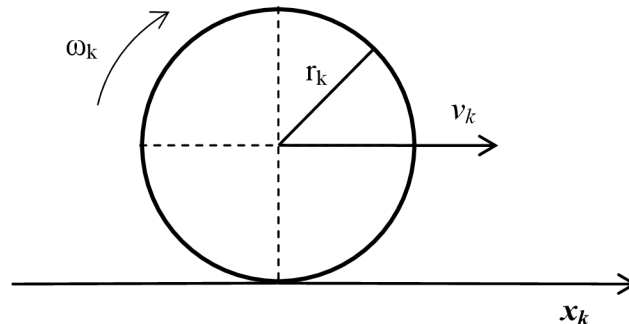


Obrázek 15 – Vývojový diagram sledování čáry

8.3 POPIS ŘÍZENÍ DIFENCIÁLNÍHO PODVOZKU [13]

Rychlost pohybu robota závisí na rychlosti otáčení kol a poloměru kol r_k . Pohyb kola uvažujeme ve směru osy x . Pro převod úhlové rychlosti kola ω_k [rad.s⁻¹] na lineární rychlost v_k [m.s⁻¹] se použije vztah 8.3.1. V rovnici se hodnota úhlové rychlosti násobí poloměrem kola. Rychlost se uvažuje vzhledem k středu kola jako hmotný bod.

$$v_k = \omega_k r_k \quad [\text{m.s}^{-1}] \quad (8.3.1)$$



Obrázek 16 – Pohyb kola

Robot je definován jako hmotný bod v souřadném systému x_r, y_r . Polohu robota určují souřadnice x a y středu robota a úhel natočení robota φ vzhledem k ose x . Principiální schéma robota s diferenciálním podvozkem je uvedeno na obrázku 14. Obvodové rychlosti kol v_1 a v_2 určují polohu robota. Převod úhlové rychlosti kol ω_1 a ω_2 na obvodovou rychlost je určena vztahem 8.3.1.

Když předpokládáme, že se robot pohybuje v rovině x, y bez překluzů kol a bez smyků, tak platí tyto vzorce:

$$v = R\omega \quad (8.3.2)$$

$$v_1 = \left(R + \frac{b}{2} \right) \omega \quad (8.3.3)$$

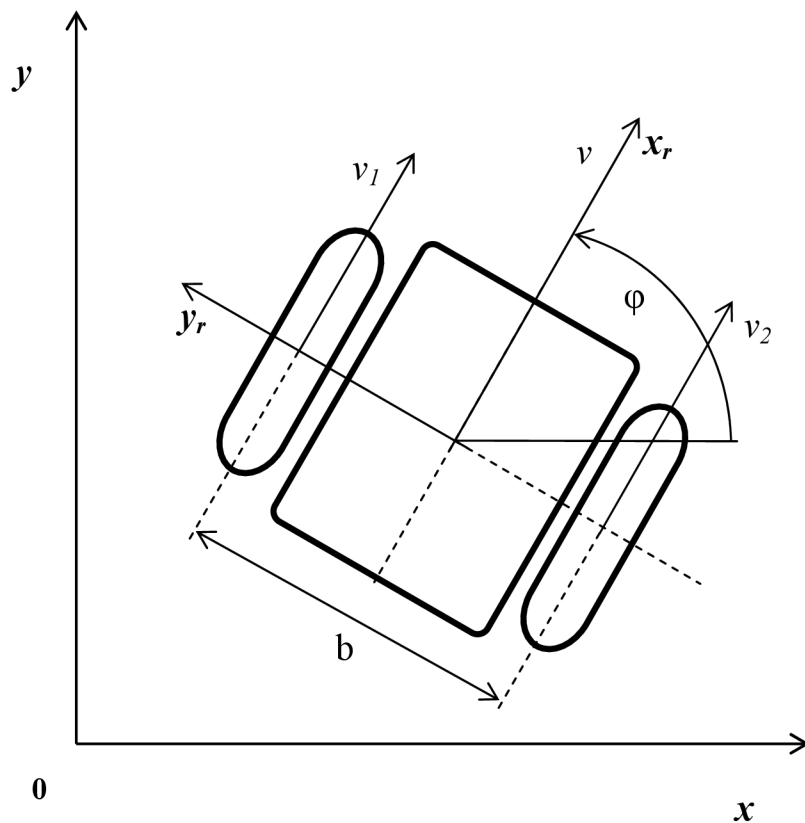
$$v_2 = \left(R - \frac{b}{2} \right) \omega \quad (8.3.4)$$

R je poloměr otáčení robota a b je rozchod kol. Pomocí jednoduchých úprav získáme vztah 8.3.5 pro určení otáčivé rychlosti robota z obvodových rychlostí kol.

$$\omega = \frac{v_1 - v_2}{b} \quad (8.3.5)$$

Podobným postupem můžeme určit vztah pro dopřednou rychlost robota z obvodových rychlostí kol.

$$v = \frac{v_1 + v_2}{2} \quad (8.3.6)$$



Obrázek 17 – Principiální schéma robota s diferenciálním podvozkem

- R poloměr otáčení robota
- b rozchod kol
- v_1 obvodová rychlost jednoho z kol
- v_2 obvodová rychlost druhého z kol

- v dopředná rychlost robota
- φ úhel natočení těla robota
- ω rychlost otáčení robota
- y_r, x_r souřadnicový systém robota
- x, y souřadnicový systém roviny, kde se robot pohybuje

9. PROGRAMOVÁNÍ

Přímé programování robotů umožňuje programovací prostředí AVR Studio. Toto programovací prostředí podporuje jen několik druhů programátorů. U modernějších počítačů se často nevyskytuje paralelní rozhraní, které by bylo možné použít pro programování pomocí STK200. Na přenosných počítačích nenajdeme ani sériové rozhraní, ale mohou se použít převodníky USB na RS232. Pro programování robotů se využíval programátor STK500 připojený přes USB port počítače.

Existují i další programy, které pomocí specifického programátoru umožňují nahrát .hex soubor do mikrokontroléru. Jedním z nejznámějších je PonyProg.

Rozložení pinů konektoru ISP na řídicí desce je stejné pro oba roboty a je znázorněno na obrázku 17.

2	RST				
1	GND	+5V	STK	MISO	MOSI

Obrázek 18 – Popis konektoru ISP na řídicích deskách

10. ZÁVĚR

Výsledkem mojí práce jsou dva roboti, kteří jsou schopni plnit úkol Myš v bludišti a také jsou schopni sledovat čáru. Úspěšně jsem oživil a zdokumentoval oba roboty. Navrhl a vytvořil jsem nový snímací systém, který jsem implementoval do programů. Pro správnou činnost robotů bylo také za potřebí upravit programy pro plnění soutěžních úkolů. Programy se musely přeprogramovat a následně ještě doplnit o snímání vzdálenosti pomocí tří SHARP GP2D120 snímačů.

Za pomoci reflexních optočlenů jsem rozšířil robota RUBRIK o možnost plnit úkol Myš v bludišti. Program robota RUBRIK jsem doplnil ještě o volbu velikosti bludiště a umístění cíle pomocí jumperů. Robot je také schopen detekovat stav baterie a signalizovat nízkou napěťovou úroveň.

Za pomoci snímačů vzdálenosti SHARP jsem rozšířil robota ROBIK o možnost detekce stěn a překážek. Změnil jsem způsob komunikace mikrokontroléru se snímači a tím jsem uvolnil komunikační rozhraní USART.

Pro plynulý pohyb při sledování čáry a také pro udržení robotů ve středu při pohybu v bludišti se používají PD regulátory. Hodnoty jednotlivých složek regulátoru se ještě mohou doladit pro přesnější regulaci pohybu. Dalším vylepšením robotů by mohlo být využití snímání vzdálenosti nejen při zjišťování stěn v bludišti a při regulaci, ale také by se mohla detekovat vzdálenost čelní stěny při pohybu více buňkami. Pro rychlejší pohyb při snímání čáry by se mohl změnit pohon z krokových motorů na jednosměrné motory s vyšší účinností.

11. SEZNAM LITERATURY

- [1] JONES, Joseph L.; FLYNN, Anita M.; SEIGER, Bruce A. *Mobile Robots : Inspiration to Implementation*. A K Peters : Ltd., 1999. 457 s. ISBN 1-56881-097-0.
- [2] CAHA, Luděk. *Caha : AVR Úvod* [online]. 2003, [citováno 29. říjen 2009]. URL:<<http://caha.wz.cz>>
- [3] ŠOLC, František; ŽALUD, Luděk. *Robotika skryptum.*:VUT Brno, 2006. 144 s
- [4] ISTROBOT 2009 [online]. 2009, [citováno 25.říjen 2009]. URL:<<http://www.robotika.sk>>
- [5] *CNY70 : Reflective Optical Sensor with Transistor Output.* : Vishay Telefunken, 2000. 7 s. Document Number 83751.
- [6] *GP2D120 : Optoelectronic Device.* : SHARP Corporation, 2006. 7 s. Reference Code SMA06008.
- [7] *Sharp IR Rangers Information : Demystifying the Sharp IR Rangers.* Colorado : Acroname Inc., 2009. 3 s.
- [8] *ATmega16 : 8-bit Microcontroller with 16K Bytes In-System Programmable Flash.* : Atmel Corporation, 2003. 21 s.
- [9] *ATmega32 : 8-bit Microcontroller with 32K Bytes In-System Programmable Flash.* : Atmel Corporation, 2009. 346 s.
- [10] *L297 : STEPPER MOTOR CONTROLLERS.* : ST Microelectronics, 2001. 11 s.
- [11] *L298N : DUAL FULL-BRIDGE DRIVER.* : ST Microelectronics, 1998. 13 s.
- [12] *MC3479 : Stepper Motor Driver.* : ON Semiconductor, 10 s.
- [13] ŠOLC, František. *Modely kolových robotů.* : VUT Brno, 2007. 39 s.

12. SEZNAMY

12.1 SEZNAM ZKRATEK

A/D	Analogově digitální
CCD	Charge-Coupled Device - zařízení s vázanými náboji.
EEPROM	Electrically Erasable Programmable Read - Only Memory, polovodičová paměť s elektricky mazatelným záznamem dat
IDE	Integrated Development Environment integrované vývojové prostředí
IR	Infrared, Infračervené
ISP	In System Programming, programování přímo v aplikaci
MISO	Master In – Slave Out, sériový vstup do mikroprocesoru
MOSI	Master Out – Slave In, sériový výstup z mikroprocesoru
PD	Proporcionálně Diferenční
PSD	Position sensitive detector - detektor pozice
RISC	Reduced Instruction Set Computer, počítač s redukovanou instrukční sadou
SCK	Serial clock, sériové hodiny pro synchronizaci přenosu dat
SPI	Serial Programming Interface, rozhraní pro sériové programování
SRAM	Static Random Access Memory, statická paměť
USART	Universal synchronous/asynchronous receiver/transmitter - Synchronní a asynchronní sériové rozhraní

12.2 SEZNAM PŘÍLOH

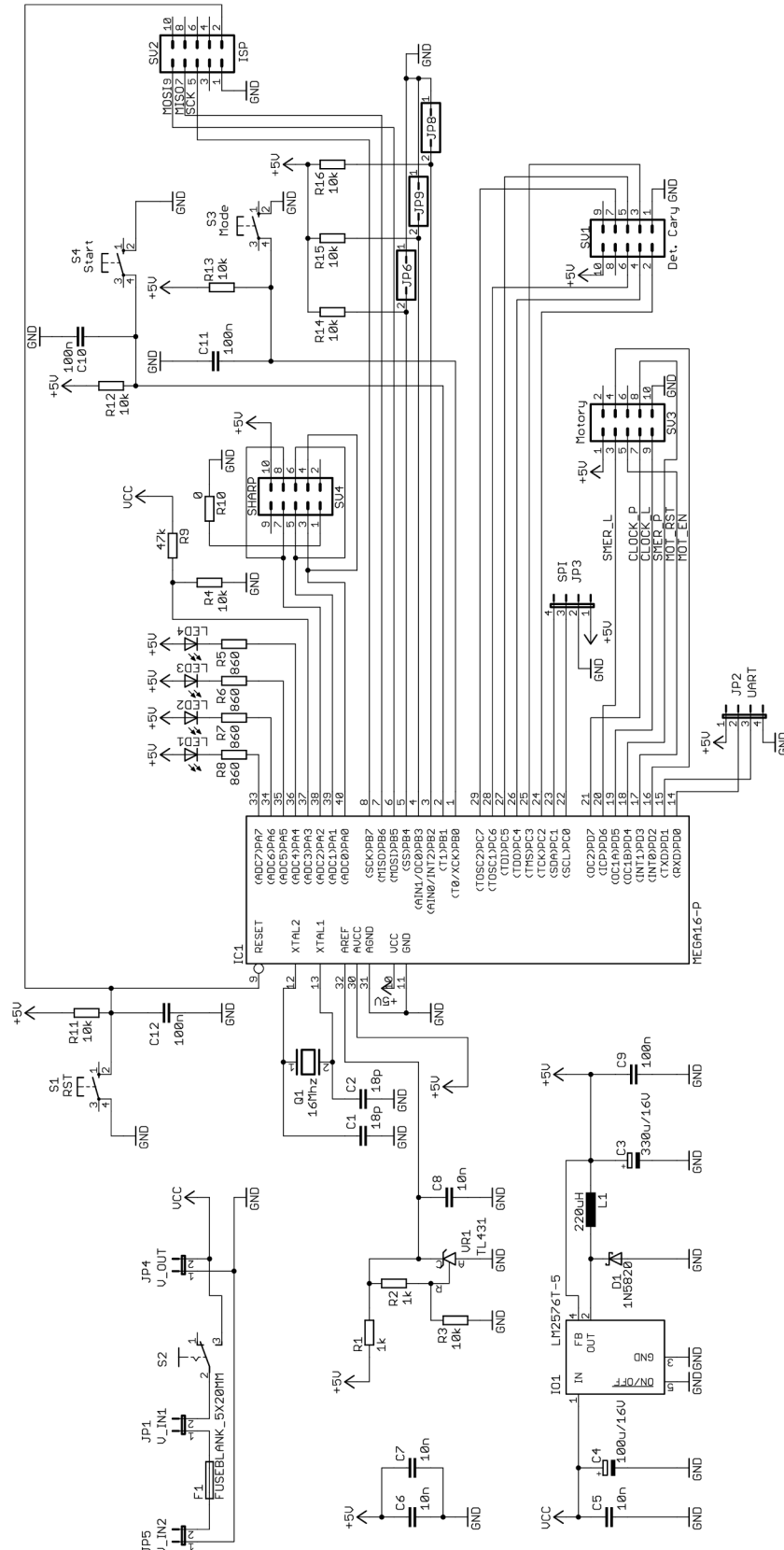
Příloha 1 - Schéma zapojení řídicí desky pro robota RUBIK	45
Příloha 2 - Schéma zapojení desky se snímači CNY70	46
Příloha 3 - Schéma zapojení desky robota ROBIK pro SHARP snímače	47
Příloha 4 - Deska se snímači CNY70 pro robota RUBRIK	48
Příloha 5 - Desky se snímači pro robota ROBIK	48
Příloha 6 – Celkový pohled na robota RUBRIK	50
Příloha 7 – Celkový pohled na robota ROBIK	51

12.3 OBSAH PŘILOŽENÉHO CD

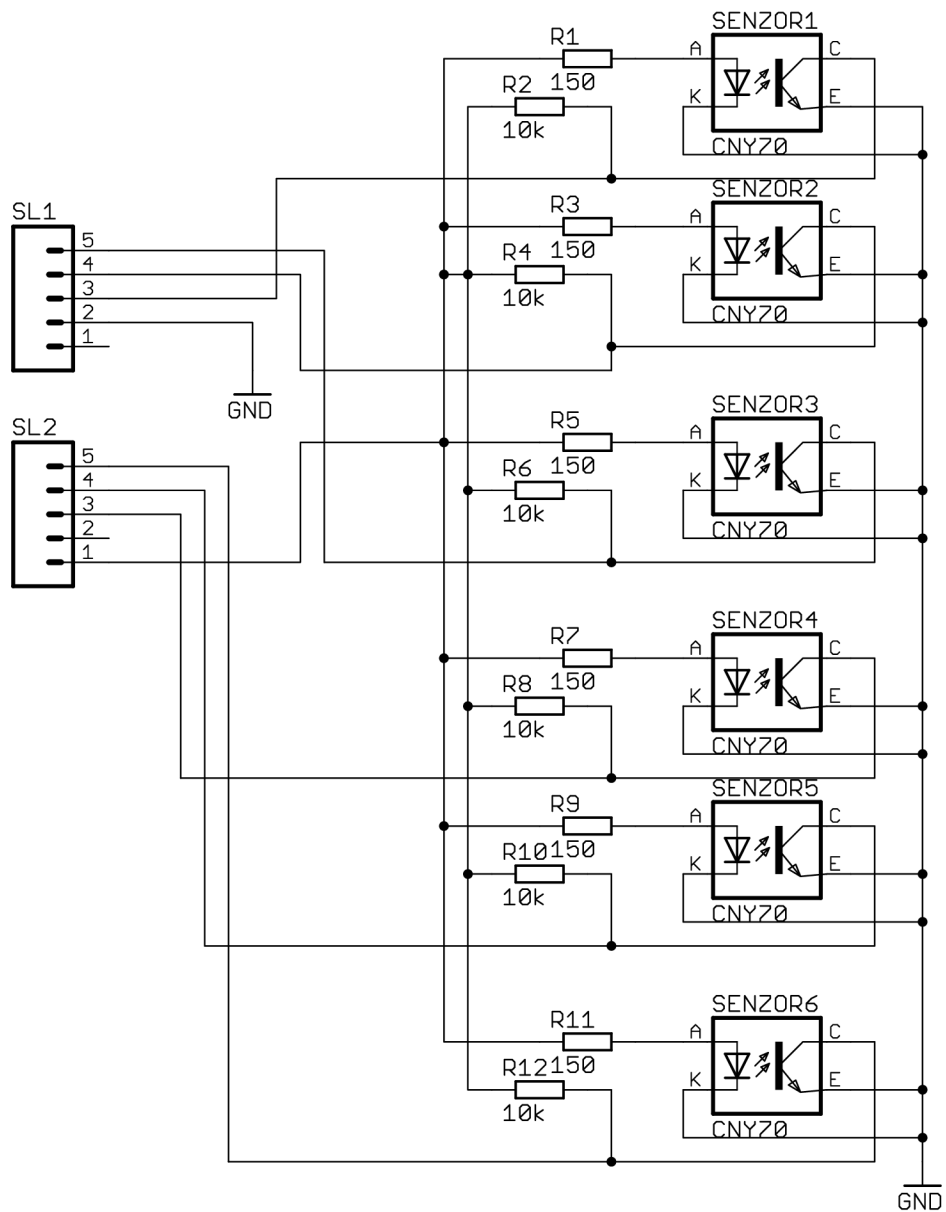
Datové listy	Datové listy použitých součástek a snímačů
Dokumentace	Dokumentace k bakalářské práci
Fotografie	Fotografie robotů a vyrobených plošných spojů
Programy	Firmware pro roboty
Schémata	Schémata a plošné spoje pro oba roboty navržené v návrhovém systému eagle

13. PŘÍLOHY

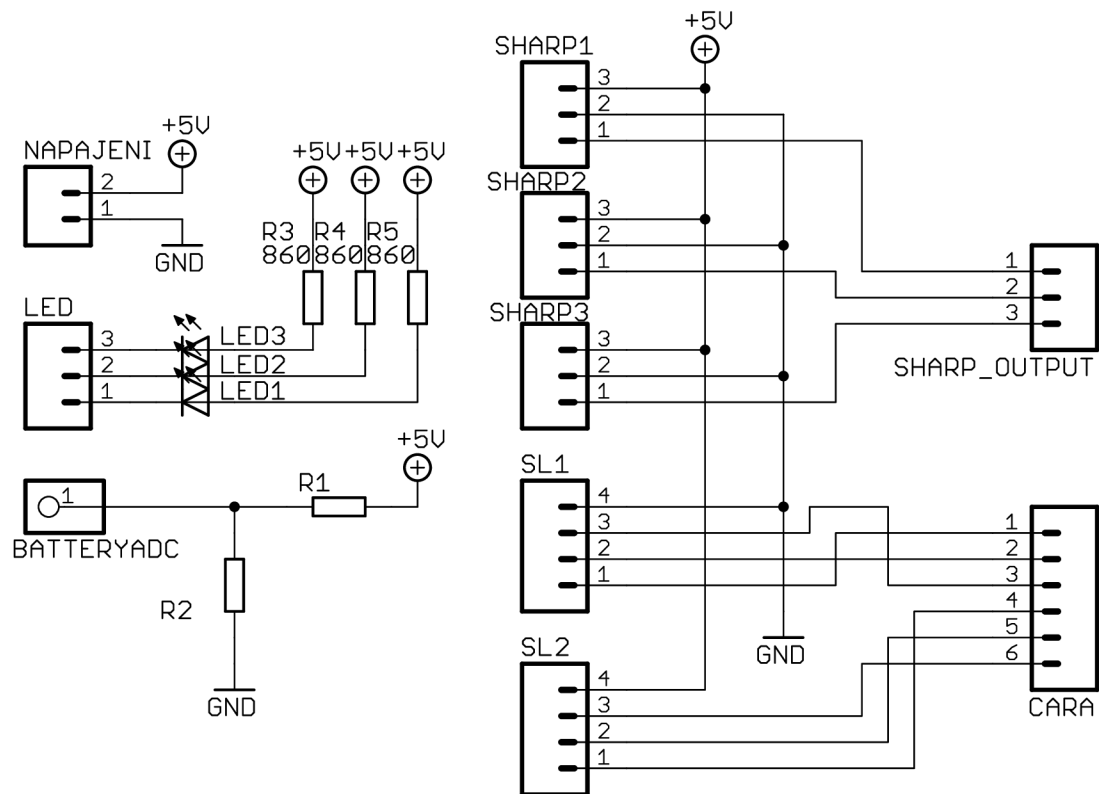
Příloha 1 - Schéma zapojení řídicí desky pro robota RUBIK



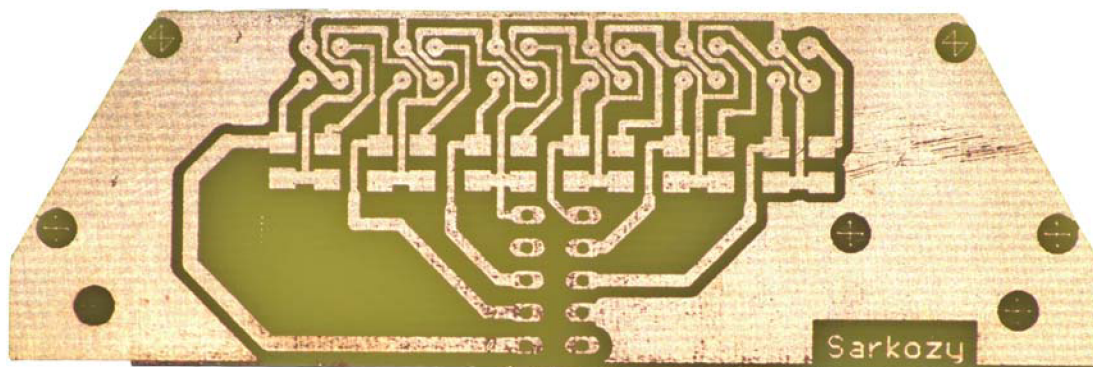
Příloha 2 - Schéma zapojení desky se snímači CNY70



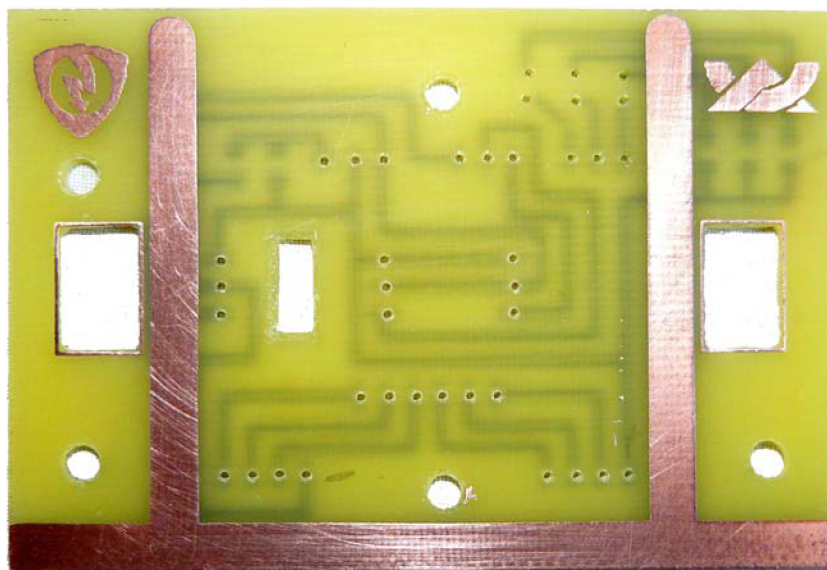
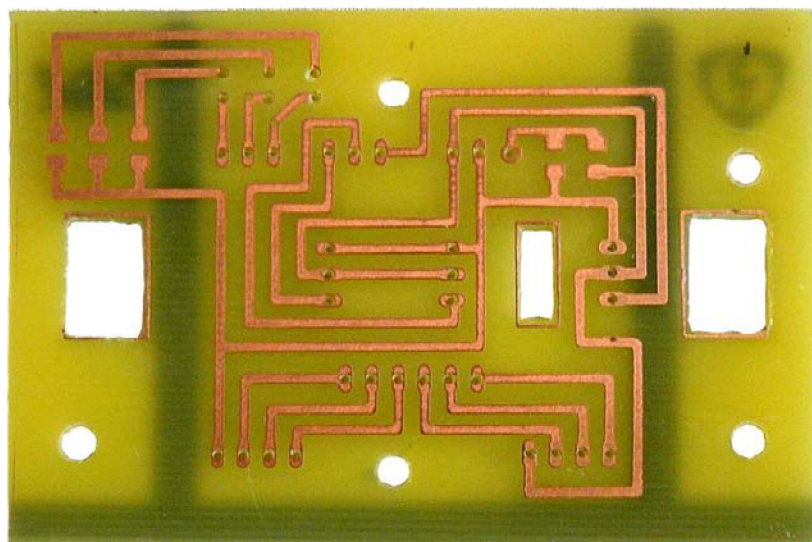
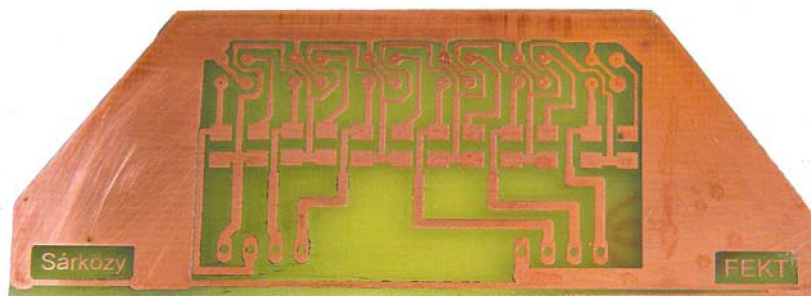
Příloha 3 - Schéma zapojení desky robota ROBIK pro SHARP snímače



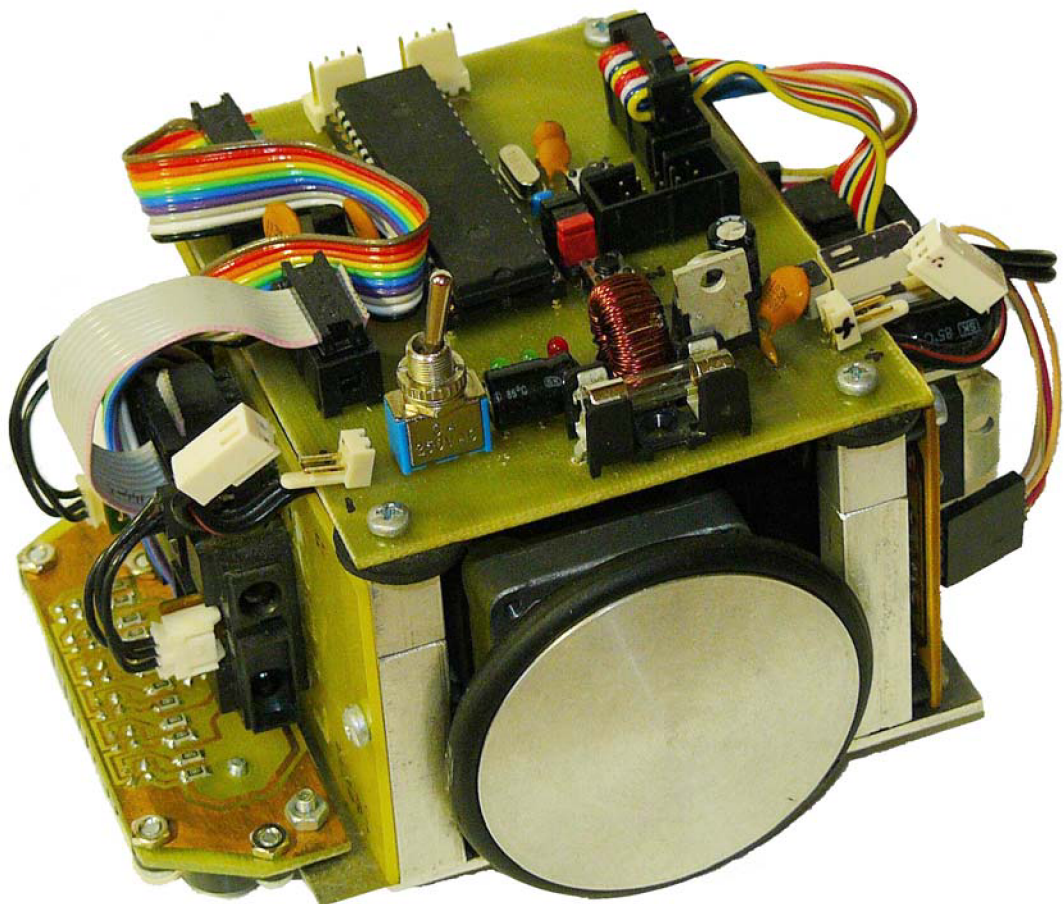
Příloha 4 - Deska se snímači CNY70 pro robota RUBRIK



Příloha 5 - Desky se snímači pro robota ROBIK



Příloha 6 – Celkový pohled na robota RUBRIK



Příloha 7 – Celkový pohled na robota ROBIK

