

Výuková pomůcka pro předmět mikropočítače – robot

Educational module for microcontrollers course - simple robot

Jakub Henzelý

Bakalářská práce
2009



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub HENZELÝ**

Studijní program: **B 3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Téma práce: **Výuková pomůcka pro předmět mikropočítače --
robot - student Jakub Henzelý**

Zásady pro vypracování:

1. Seznamte se s vývojovým kitem M68EVB908GB60, který se používá pro výuku programování mikropočítačů na naší fakultě.
2. Navrhněte a realizujte model robota řízeného pomocí mikropočítače HCS08 prostřednictvím uvedeného vývojového kitu.
3. Vytvořte podpůrné softwarové vybavení k ovládání robota pro daný mikropočítač.
4. Zpracujte technickou dokumentaci k vytvořenému modelu.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. MC9S08GB/GT Data Sheet, Rev.2.3. Freescale Semiconductor, 2004. 290 s.
2. HCS08 Family Reference Manual, Rev.1. Freescale Semiconductor, 2003.
3. CPU08 Central Processor Unit Reference manual. Motorola, 2001.
4. NOVÁK P.: Mobilní roboty – pohony, senzory, řízení. Praha: BEN -- technická literatura, 2005. 256 s. ISBN 80-7300-141-1.
5. VÁŇA V.: Začínáme s mikrokontroléry Motorola HC08 Nitron. Praha: BEN -- technická literatura, 2003. 96 s. ISBN 80-7300-124-1.
6. LADMAN J.: Elektronické konstrukce pro začátečníky. Praha: BEN - technická literatura 2002. 144 s. ISBN 80-7300-015-6.
7. MANN B.: C pro Mikrokontroléry. Praha: BEN -- technická literatura 2004. 280 s. ISBN: 80-7300-077-6.

Vedoucí bakalářské práce:

Ing. Jan Dolinay

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

20. února 2009


Termín odevzdání bakalářské práce:

1. června 2009

Ve Zlíně dne 13. února 2009



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem této práce je navrhnout a realizovat model robota řízeného pomocí mikropočítače HCS08 prostřednictvím uvedeného vývojového kitu M68EVB908GB60, který se používá pro výuku programování mikropočítačů na naší fakultě.

Klíčová slova: robot, mikropočítač, servo, CodeWarrior

ABSTRACT

The aim of my bachelor project is to design and build a robot model controlled by a HCS08 microcontroller using the development kit M68EVB908GB60 which is used for microcontrollers programming on our faculty.

Key words: robot, microcomputer, servo, CodeWarrior

Na tomto místě bych rád poděkoval svému vedoucímu bakalářské práce, panu Ing. Janu Dolinayovi za vedení při práci. Dále děkuji rodičům a přátelům za jejich podporu při studiu.

Zmínka prof. Chudoby o tom, jak se podle svědectví Oxfordského slovníku ujalo slovo robot a jeho odvozeniny v angličtině, mne upomíná na starý dluh. To slovo totiž nevymyslel autor hry RUR, nýbrž toliko je uvedl v život. Bylo to tak: v jedné nestřežené chvíli napadla řečeného autora látka na tu hru. I běžel s tím zatepla na svého bratra Josefa, malíře, který zrovna stál u štafle a maloval po plátně, až to šustělo.

"Ty, Josef," začal autor, "já bych měl myšlenku na hru."

"Jakou," bručel malíř (opravdu bručel, neboť držel přitom v ústech štětec).

Autor mu to řekl tak stručně, jak to šlo.

"Tak to napiš," děl malíř, aniž vyndal štětec z úst a přestal natírat plátno. Bylo to až urážlivě lhostejné.

"Ale já nevím," řekl autor, "jak mám ty umělé dělníky nazvat. Řekl bych jim laboři, ale připadá mně to nějak papírové."

"Tak jim řekni roboti," mumlal malíř se štětcem v ústech a maloval dál. A bylo to. Tím způsobem se tedy zrodilo slovo - robot; budiž tímto přičeno svému skutečnému původci.

Karel Čapek, Lidové noviny, 24. 12. 1933 [16]

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.
V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 ROBOTY	11
1.1 HISTORIE.....	11
1.2 DĚLENÍ	13
1.2.1 Stacionární roboty	14
1.2.1.1 Podle využití ve výrobě:	14
1.2.1.2 Podle způsobu programování:	14
1.2.1.3 Zadávání pozice ramena:	14
1.2.2 Mobilní roboty	15
1.2.2.1 Podle prostředí, ve kterém se má robot pohybovat, dělíme mobilní roboty:	15
1.2.2.2 Podle účelu nasazení je lze dělit:	15
1.2.2.3 Mobilní roboty pohybující se po souši, můžeme dále dělit podle typu pohybového subsystému:	15
1.2.3 Kolové roboty.....	16
1.2.4 Pásové roboty	19
1.2.5 Kráčeující roboty.....	19
2 MIKROPOČÍTAČE	20
2.1 OBECNĚ.....	20
2.1.1 Mikropočítače okolo roku 1980	20
2.1.2 Dnešní mikropočítače.....	21
2.2 EMBEDDED SYSTÉM	21
2.3 JEDNOČIPOVÝ POČÍTAČ	22
2.3.1 Rozdělení dle architektur	23
2.3.1.1 Von Neumannova architektura	23
2.3.1.2 Hardwarská architektura	24
2.3.2 Rozdělení dle instrukčních souborů	24
2.3.2.1 CISC.....	24
2.3.2.2 RISC.....	24
2.3.3 Základní struktura	24
II PRAKTICKÁ ČÁST	26
3 POPIS JEDNOTLIVÝCH SOUČÁSTÍ	27
3.1 VÝVOJOVÝ KIT FREESCALE M68EVB908GB60	27
3.1.1 Mikropočítač M9S08GB60	28
Základní vlastnosti:	28
3.1.2 Napojení periférií na porty	28
3.1.3 Paměťová mapa.....	29
3.2 MODELÁŘSKÁ SERVA	29
3.2.1 Popis serva	31
3.2.2 Napájení serv	32
3.2.3 Kompatibilita serv	32
4 MECHANICKÁ ČÁST	33

4.1	POPIS ROBOTA	34
4.2	PARAMETRY ROBOTA	35
5	ELEKTRONICKÁ ČÁST	36
5.1	NAPÁJENÍ A AKUMULÁTORY	36
5.2	PŘÍDAVNÝ TIŠTĚNÝ SPOJ	36
5.3	PROPOJENÍ TIŠTĚNÉHO SPOJE S VÝVOJOVÝM KITEM	37
5.3.1	Ovládání servomotorů (PWM).....	37
5.3.2	Ovládání diod	37
5.3.3	Snímání foto-odporů	38
5.4	UVEDENÍ ROBOTA DO PROVOZU	39
6	SOFTWAREOVÁ ČÁST	40
6.1	PŘEHLED FUNKCÍ OVLADAČE	40
6.2	UKÁZKOVÝ PROGRAM	41
6.2.1	Popis programu	41
6.2.2	Ukázka práce s funkcemi ovladače	42
	ZÁVĚR	44
	ZÁVĚR V ANGLIČTINĚ.....	45
	SEZNAM POUŽITÉ LITERATURY.....	46
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	48
	SEZNAM OBRÁZKŮ	49
	SEZNAM TABULEK.....	50
	SEZNAM PŘÍLOH.....	51

ÚVOD

Robot je samostatně pracující stroj vykonávající předem dané úkoly. Slovo „robot“ pochází ze slova „robota“, které bylo známé již v 17. století, ve významu otrocká práce poddaných. Pozměněný název „robot“ byl poprvé použit v roce 1920 v divadelní hře Karla Čapka R.U.R. Což je zkratka Rossum's Universal Robot. Karel Čapek ale není vynálezcem tohoto slova. To mu poradil jeho bratr Josef. Karel chtěl původně použít slovo Labor.

Roboti v historii prošli určitým vývojem. Jejich původní záměr měl být v užitečnosti a zjednodušení práce člověka, který přešel časem k zábavě a nyní se pohybuje někde mezi oběma odvětvími.

Trendy současného světového vývoje v oblasti mobilních a servisních robotů ukazují na značně široké spektrum možností jejich uplatnění. Na rozdíl od průmyslových robotů nacházejí využití především v nestrojírenských oblastech. Je to zřejmé už z toho, že servisní roboty jsou definovány jako technická zařízení, která se podílejí na nevýrobních činnostech. To znamená, že jsou nasazovány především do oblasti služeb a uspokojování nejrůznějších potřeb lidí nebo firem.

Různorodost a kvalita prováděných úloh (činností) se odráží ve velkém rozptýlu požadavků na servisní roboty, které jsou na ně kladeny v souvislosti s prováděním těchto úloh. To vše klade velké nároky na jejich design, vybavenost potřebnými pohony, senzory, způsob řízení, komunikaci s okolním prostředím apod.

Servisní roboty nacházejí uplatnění zejména všude tam, kde je prostředí pro člověka nebezpečné, těžko dostupné či dokonce nedostupné. Jako příklad lze uvést monitorování zamořeného prostředí radiací nebo chemickými látkami, monitorování kráterů sopek, podmořského dna, povrchu jiných planet apod. Mobilní servisní roboty určené k provádění monitorování, inspekčních činností, manipulace s předměty apod. jsou často vybaveny manipulačními nástavbami. Mohou tak provádět např. manipulační činnosti, transport předmětů nebo provádět různé technologické operace apod.

Cílem této práce je vytvořit robota, který bude používán jako pomůcka při výuce předmětu mikropočítače na naší fakultě. Robot je tříkolový, dvě kola jsou poháněná servy a jedno je všesměrové. Robot je dále vybaven foto-odpory pro snímání intenzity světla a diodami.

I. TEORETICKÁ ČÁST

1 ROBOTY

1.1 HISTORIE

Roboti pod jinými názvy byli známí již 200 let př. n. l. V Alexandrii bylo možné vidět samo-otevírající se těžká bronzová vrata chrámu, ptáci na nádvoří prozpěvovali a sochy kropily věřící svícenou vodou. To vše bylo dílem Hérona z Alexandrie, jež k těmto mechanickým „zázrakům“ potřeboval pouze vodu a páru. Ze starověku, středověku ani z pozdějších dob se žádné konkrétní mechanické automaty nedochovaly. Dochovaly se pouze písemné zmínky. Například Leonardo da Vinci konstruoval automaty - v nákresech se dochoval rytíř.

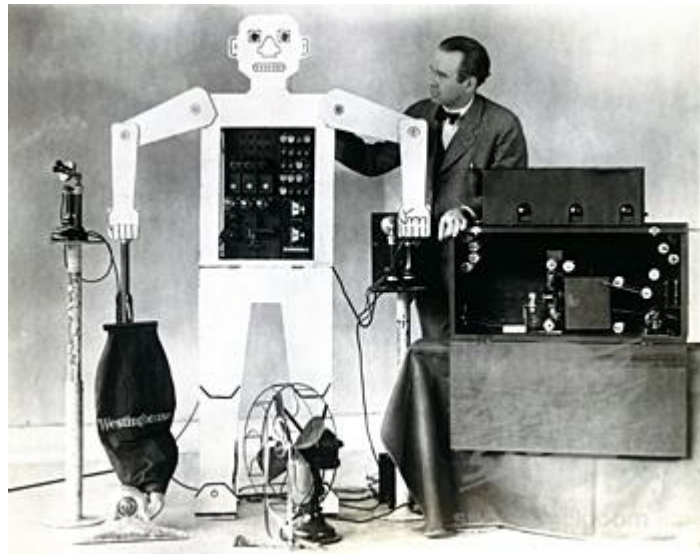
Zřejmě nejstaršími dochovanými mechanickými automaty – androidy jsou písař, muzikantka a kreslič od švýcarského hodináře jménem Pierre Jaquet-Droz (1721-1790). Konstruoval je se svým synem Henri-Louisem v letech 1768-1774. Jednalo se o figurky pro zábavu sestavených z tisíců dílů. Kreslič byl sestaven z 2000 dílů, muzikantka z 2500 dílů a nejdokonalejší je písař, který je z 6000 dílů. Písař – android skutečně psal a kreslil, protože jeho tvůrce převedl elementární pohyby ruky do „programovatelné paměti“, která byla tvořena složitou soustavou vaček. [15]



Obrázek 1.1 První android spisovatel (vlevo) a jeho vnitřní struktura (vpravo)

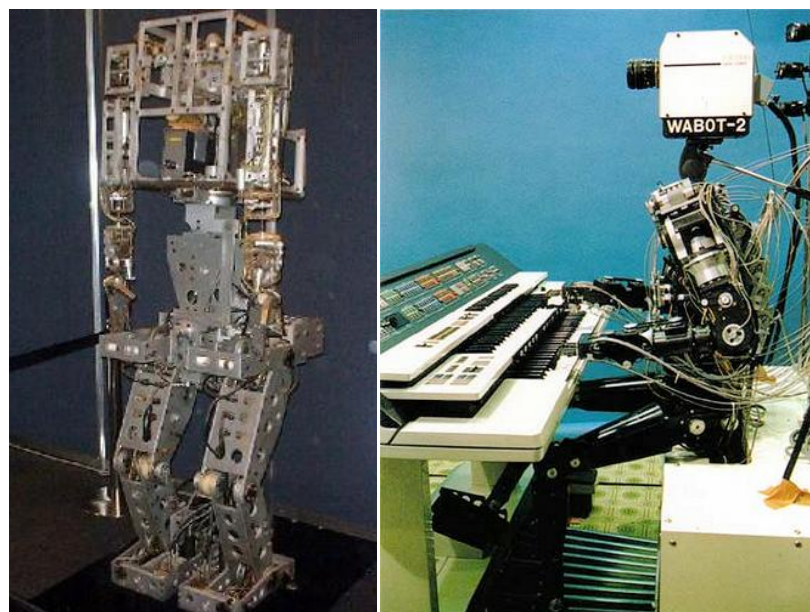
Velkým průlomem byl první robot, který plně odpovídal definici robota – Televox. Vytvořil ho americký inženýr R. J. Wensley v roce 1927. Útroby měl plné relé a vykonával

vše, co jeho později vytvoření, dokonalejší následovníci – vysával, otevíral dveře a zvedal telefon.



Obrázek 1.2 R. J. Wensley představuje Televox

Až o mnoho let později v roce 1973 japonští vědci z tokijské Waseda University představili světu WABOT-1, první elektronický chodící robot, z kterého se v roce 1984 vyvinul WABOT-2 - klavírista, který kromě své vědecké úlohy i bavil diváky svojí hrou.



Obrázek 1.3 WABOT-1 (vlevo) a WABOT-2 (vpravo)

A to už se dostáváme do současnosti, kde se budeme zabývat jednotlivými typy robotů zvláště. Ještě stojí za zmínku americký spisovatel a biochemik Isaac Asimov, který je tvůrcem zákonů robotiky. Popisují, jakými pravidly se mají roboti řídit, a byly uveřejněny ve sbírce povídek Já, robot v roce 1942. Tyto tři zákony robotiky zní:

1. *Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby mu bylo ublíženo.*
2. *Robot musí poslechnout člověka, kromě případů, kdy je to v rozporu s prvním zákonem.*
3. *Robot se musí chránit před poškozením, kromě případů, kdy je to v rozporu s prvním nebo druhým zákonem.*

Postupem času byl přidán nultý zákon:

4. *Robot nesmí ublížit lidstvu nebo svou nečinností dopustit, aby mu bylo ublíženo.*

Postupem doby byly pokusy zákony doplnit či pozměnit. Za zmínku stojí bulharští sci-fi autoři jako Lyuben Dilov či Nikola Kesarovski, kteří vytvořili další dva zákony robotiky:

4. *Robot se musí vždy prokazovat jako robot.*
5. *Robot musí vědět to, že je robot.* [17]

1.2 DĚLENÍ

Možnosti dělení robotů jsou rozsáhlé, takže začneme od toho nejjobecnějšího:

- **Kuchyňský robot** - kombinace mixéru, hnětače a dalších kuchyňských strojů, obvykle provedený jako motorová jednotka s nastavci
- **Droid** – jakýkoliv inteligentní a samočinný robot
- **Android** - robot podobný člověku – obvykle se očekává biologické složení. Roboti v R.U.R. byli podle tohoto dělení androidi
- **Humanoid** - robot podobný člověku principiální stavbou těla a zejména způsobem pohybu. (za humanoidy se považují i mytická stvoření)
- **Kyborg (kybernetický organismus)** - živá bytost obohacená o mechanické či elektronické součástky

Další dělení je dle jejich schopnosti se pohybovat:

- **Stacionární** – nemohou se pohybovat z místa na místo (průmyslové manipulátory)
- **Mobilní** – mohou se pohybovat

1.2.1 Stacionární roboty

1.2.1.1 Podle využití ve výrobě:

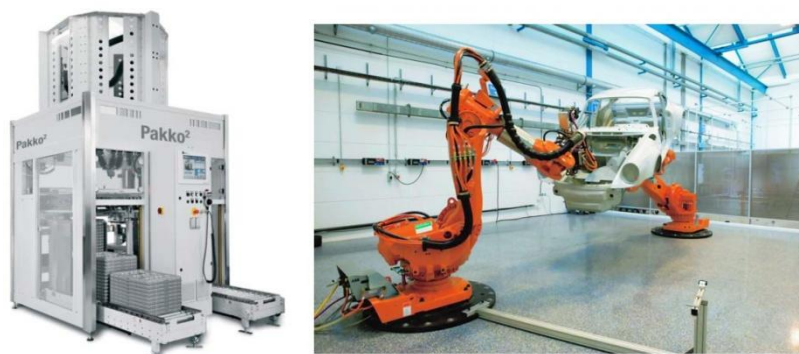
- **Bodové sváření** – sváření karoserie automobilu
- **ARC welding** - souvislé sváření
- **Montáž** - instalace a kompletace
- **Aplikace** - nanášení lepidel, těsniv
- **Lakování** - stříkání tekutých a práškových barev, laků
- **Manipulace** - překládání, nakládání a vykládání pro dopravníky, simulace pohybu modelu v aerodynamickém tunelu
- **Paletizace** - skládání nebo vykládání výrobků na paletách
- **Kontrola** - měření pomocí kamer, laseru a čidel

1.2.1.2 Podle způsobu programování:

- **Přímé programování**
 - vedením robotova ramena
 - zadáváním povelů z ovládacího panelu
- **Nepřímé programování** (off-line)
 - zadáváme prostorové křivky (získané z výkresů)
- **Přímé plánování** (on-line)
 - obdobně jako nepřímé, ale robot se přizpůsobuje měnícím se vnějším podmínkám (pomocí čidel)

1.2.1.3 Zadávání pozice ramena:

- **Spojité trasa** – zadání přesné pozice ramena ve všech okamžicích činnosti robotu (vedení ramena, nepřímé programování, přímé plánování)
- **Bod po bodu** – zadání konkrétních pozic, v nichž se rameno musí v konkrétních časech činnosti nacházet



Obrázek 1.4 Paletovací stroj Pakko (vpravo)
a příklad průmyslového robota (vlevo)

1.2.2 Mobilní roboty

Mobilní roboty je možné také dělit podle řady kritérií. Základní rozdělení je na dva typy: [4]

- **Autonomní** - předpokládá se schopnost samostatně vykonávat zadanou úlohu, např.: Sledování čáry na podlaze a schopnosti reagovat na eventuální překážku, zastavit se, popřípadě se jí vyhnout a vrátit se na značku a pokračovat v jízdě. Nebo umět se pohybovat v neznámém prostředí, dokázat ho zmapovat a orientovat se v něm.
- **Dálkově řízené** - jsou řízeny operátorem, který má – zpravidla vizuální- informaci o pracovním okolí robotu. Ale i takový robot by měl být vybaven určitou schopností autonomního chování. Např.: Když se přeruší komunikace s operátorem, tak by měl robot uvolnit cestu a přemístit se k nejbližší stěně.

1.2.2.1 Podle prostředí, ve kterém se má robot pohybovat, dělíme mobilní roboty:

- Pohyb na souši (terrestrial)
- Vnitřní/vnější prostředí (indoor/outdoor)
- Ve vodě (aquatic)
- Ve vzduchu (airbone)
- Ve vesmírném prostoru (space)
- Hybridní

1.2.2.2 Podle účelu nasazení je lze dělit:

- Manipulační
- Montážní
- Servisní
- Inspekční
- Průzkumové
- Vojenské
- Zdravotnické
- Určené pro zábavu

1.2.2.3 Mobilní roboty pohybující se po souši, můžeme dále dělit podle typu pohybového subsystému:

- Kolové
- Pásové
- Kráčeující
- Plazivé

- Šplhající
- Skákající
- Hybridní

Z nich nejrozšířenější jsou kolové, pásové a kráčející. Dále se budeme zabývat hlavně kolovými roboty, protože vytvořený robot patří do této kategorie. [14] [4]

1.2.3 Kolové roboty

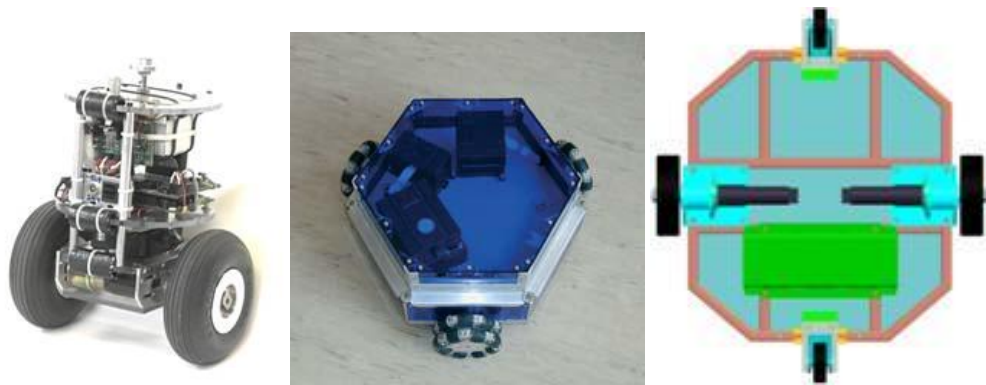
Kola u kolových robotů mohou být aktivní – hnaná nebo pasivní – vlečná. Z konstrukčního hlediska je pak možno využít různé typy kol – standardní, všesměrové, Weinsteinovy, článkové a netradiční. [4]

Dalším důležitým parametrem je počet stupňů volnosti jejich pohybu. Běžně používaná jich mají jeden nebo dva. Kola s jedním stupněm volnosti se mohou pouze odvalovat po povrchu podél jedné osy (hřídele). V případě kola s více stupni volnosti se kolo pohybuje kolem dvou os, které mohou být rovnoběžné s povrchem země (analogie kupačky mechanické počítačové myši), nebo jedna osa rovnoběžná a druhá kolmá (přední kolo automobilu). Klasickým příkladem kolového robotu, je mobilní průzkumný robot „Sojourner“, který má všech šest kol hnaných, přičemž přední a zadní pár je směrový. [4]



Obrázek 1.5 Mobilní robot Sojourner

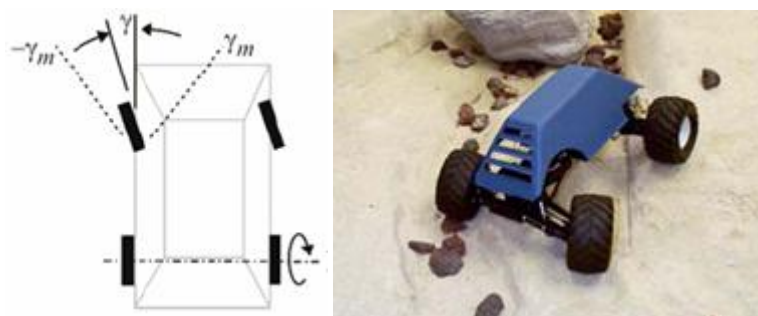
Podle počtu kol lze roboty dělit na jednokolové, dvoukolové, tříkolové, čtyřkolové, šestikolové, osmikolové a zvláštní konstrukce. [4]



Obrázek 1.6 Dvojkolový robot (vlevo), trojkolový robot (uprostřed), čtyřkolový robot s diferenciály (vpravo)

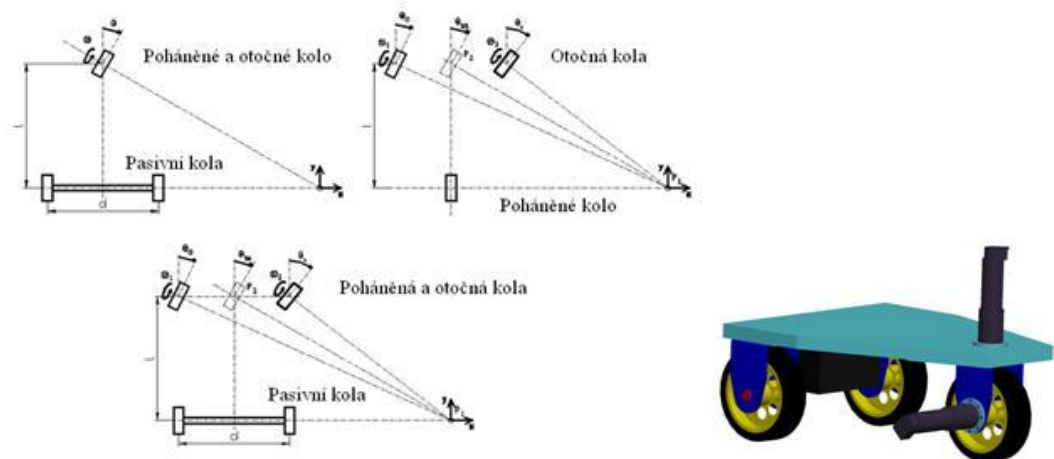
Nejjednodušším typem je tříkolové uspořádání podvozku s diferenčně řízenými koly. Toto uspořádání využívá dvě nezávisle poháněná kola a jedno volně otočné nepoháněné směrové kolo vpředu nebo vzadu. Při čtyřkolovém uspořádání podvozku jsou použita dvě volně otočná kola. Výhodou této koncepce je manévrovatelnost – robot je schopen otáčet se na místě kolem osy, dále snadnost řízení – zatáčení změnou rychlosti otáčení kol. Nevýhodou je pak nízká prostupnost terénem. Tento typ podvozku je často používán v praxi i experimentální oblasti. [4]

Často používaným typem podvozku je tzv. Ackermanův podvozek. V případě čtyřkolového uspořádání jde o typ známý u automobilů. U mobilních robotů je nejpoužívanější konfigurace s předními koly řízeně natáčenými a zadními hnanými. [4]



Obrázek 1.7 Čtyřkolové pohybové ústrojí řízené Ackermanovým způsobem

U tříkolového pohybového ústrojí řízené Ackermanovým způsobem je možné se setkat s variantami zadní kola hnaná a přední otočné, nebo zadní kola volně otočná a přední kolo poháněné. Nevýhodou je neschopnost robota otáčet se na místě. [4]



Obrázek 1.8 Tříkolové pohybové ústrojí řízené Ackermanovým způsobem

Zvláštní skupinou pohybových subsystémů tvoří tříkolové a čtyřkolové podvozky se všesměrovými koly. Tato pohybová ústrojí umožňují robotu pohybovat se ve všech směrech a zatáčet či rotovat na velmi malém prostoru. Umožňují taky okamžitou změnu směru s nulovým poloměrem zatáčení. Jejich nevýhodou je vyšší cena, daná požadavkem na precizní výrobu, dále pak malá schopnost překonávání překážek. Tento systém se čtyřkolovým podvozkem se v praxi používá například na některých amerických bitevních lodích na přepravu a manipulaci s hlavicemi raket. [4]



Obrázek 1.9 Detail všesměrového kola (vlevo)
a tříkolový robot se všesměrovými koly (vpravo)

1.2.4 Pásové roboty

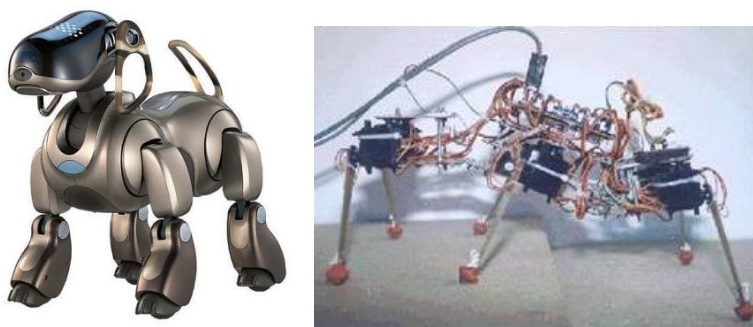
Kinematika pásových podvozků je podobná kinematice diferenciálního podvozku. Je možné si představit, že kola jsou o pásy rozšířena. Díky tomu se zlepšuje prostupnost terénem a stoupavost. Nevýhodou je ale větší energetická náročnost díky pohybu „smykem“. Ve vnějších prostředích bývá často použit u buldozerů a vojenských nosičů, bezpečnostních složek (odminování, průzkum a manipulace s nebezpečnými předměty). Ve vnitřních prostředích bývá použit pro jízdu ze schodů a do schodů. [4]



Obrázek 1.10 Robot „RATLER™“

1.2.5 Kráčejíci roboty

Kráčejíci roboty nacházejí uplatnění pro terény náročné na průchodnost. Svým vzhledem a způsobem pohybu mohou napodobovat člověka, zvířata nebo hmyz. Nejčastěji se kráčejíci roboty rozdělují podle počtu noh a jejich stupni volnosti. Podle tohoto rozdělení je nejrozšířenější šestinohá skupina. Mezi nevýhody patří vyšší počet řízených os a tedy i akčních členů (pohony, převody, senzory atd.), složitější řízení, u dvounohých je třeba řídit stabilitu, konstrukční složitost, výrobní náklady apod. [4]



Obrázek 1.11 Robot Sony AIBO (vlevo)
a robot IOAN překonávající schody (vpravo)

2 MIKROPOČÍTAČE

2.1 Obecně

Mikropočítač je pojem, jehož význam se postupem času mění. Přibližně v 60. - 80. letech 20. století se jednalo v podstatě o protivníka „velkých počítačů“, v současné době se mikropočítačem obvykle myslí zařízení výrazně menší než je stolní osobní počítač. [18]

2.1.1 Mikropočítače okolo roku 1980

Mikropočítače obsahovaly mikroprocesor, řadič, paměť (velmi malou) a aritmeticko-logickou jednotku s registry, do kterých se nahrával buď operandy aritmetických operací, nebo logického porovnání. Mikroprocesory jsou 8bitové, dražší 16bitové nebo pro náročnější úlohy, jako např. překlad jazyků, se používají mikroprocesory 32bitové. Pro zrychlení mikroprocesoru se používá koprocesor, např. pro matematické operace. [18]

Dále mikropočítače obsahují zdroj hodinového signálu o taktu 4 - 40MHz. Zaváděcí program je uložený v paměti ROM. Paměť RAM slouží k zpracování dat a ukládání mezivýsledků. Polovodičové paměti se vyrábějí 16, 64kB ale i 4MB.

Mikropočítače jsou většinou stavěny tak, aby byly všechny součástky umístěny v jedné „bedně“. Obecně by se měl mikropočítač vejít na psací stůl, ostatní počítače jako minipočítače nebo mainframe jsou o hodně větší, někdy zabírají i celou místnost. [18]

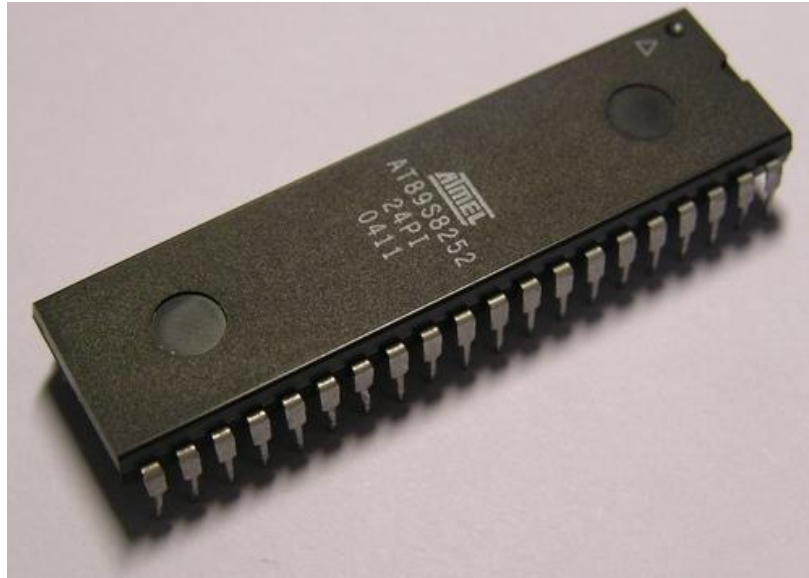


Obrázek 2.1 Mikropočítač Commodore 64 (1982)

2.1.2 Dnešní mikropočítače

Dnešní mikropočítače se rozdělují na dva základní typy:

- *Embedded systém*
- *Jednočipový počítač*

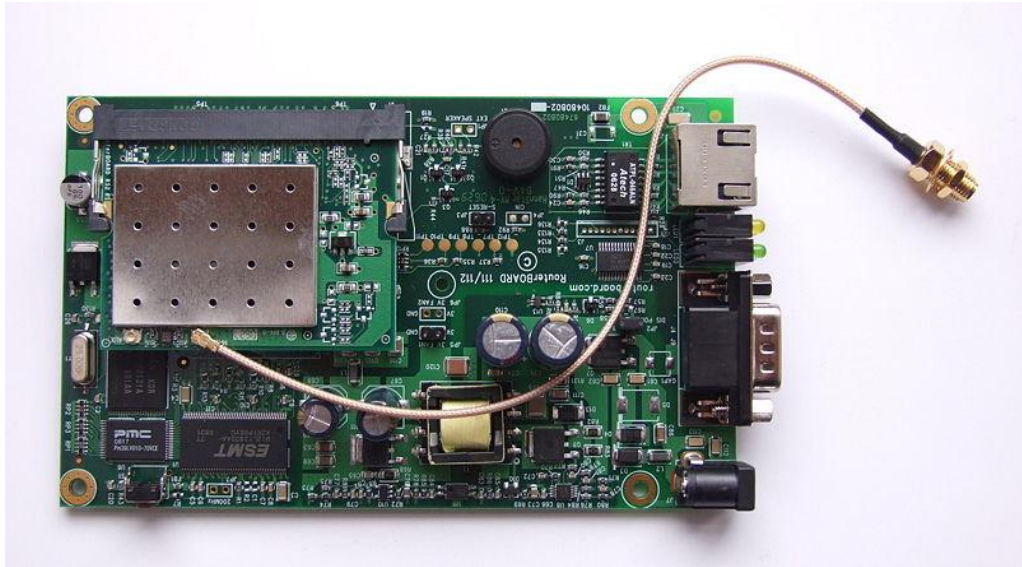


Obrázek 2.2 Mikropočítač Atmel 8051

2.2 Embedded systém

Embedded systém (vestavěný systém, zabudovaný systém) je jednoúčelový systém, ve kterém je řídicí počítač zcela zabudován do zařízení, které ovládá. Na rozdíl od univerzálních počítačů jako jsou osobní počítače, embedded (zabudované) počítače jsou většinou jednoúčelové, určené pro předem definované činnosti. Vzhledem k tomu, že systém je určen pro konkrétní účel, mohou tvůrci systému při návrhu optimalizovat pro konkrétní aplikaci a tak snížit cenu výrobku. Embedded systémy jsou často vyráběny sériově ve velkém množství, takže úspora bývá vynásobena velkým počtem vyrobených kusů. [13]

Počítače do dlaně (PDA), mobilní digitální pomocníci (MDA) a inteligentní mobilní telefony jsou také často označovány jako embedded zařízení vzhledem k vlastnostem hardware i přes to, že z hlediska software jsou rozšiřitelné a všeobecně použitelné podobně jako osobní počítače. S rozvojem těchto zařízení se stírá rozdíl mezi embedded zařízeními a osobními počítači. [13]



Obrázek 2.3 Embedded systém v podobě RouterBoardu 112 s U.FL-RSMA

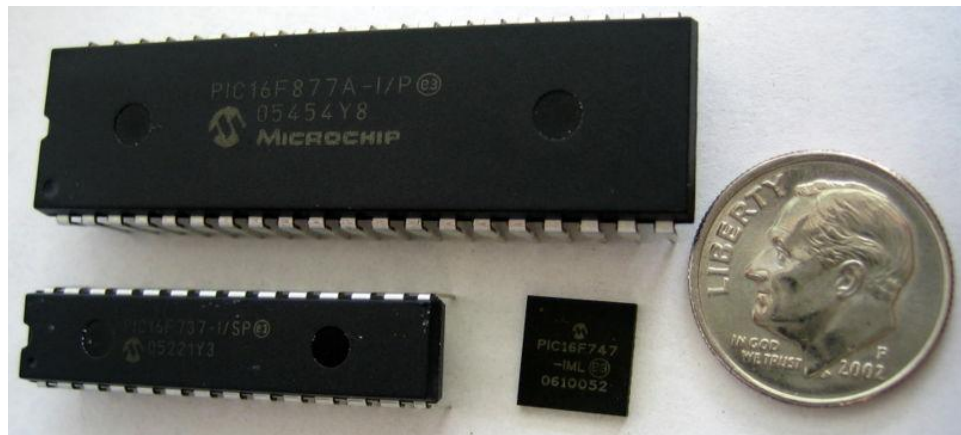
2.3 Jednočipový počítač

Jednočipový počítač nebo také anglicky Microcontroller (Mikrokontrolér, MCU, μC) je většinou monolitický integrovaný obvod obsahující kompletní mikropočítač. Jednočipové počítače se vyznačují velkou spolehlivostí a kompaktností, proto jsou určeny především pro jednoúčelové aplikace jako je řízení, regulace a pod. Často jsou jednočipové počítače součástí embedded systémů. [12]

Jednočipový počítač je integrovaný obvod, který v sobě zahrnuje jádro mikroprocesoru společně s nevolatelní, vestavěnou pamětí (ROM, FLASH, EEPROM), pamětí RAM a periferními obvody (logické vstupy/výstupy, komunikační linky, PWM, A/D převodníky, čítače, časovače, apod.), takže může obsáhnout celou aplikaci, aniž by potřeboval složité podpůrné obvody. [12]

Existují základní dvě architektury mikroprocesorů – von Neumannova a Harvardská, každá má svoje výhody i nevýhody. Při současném stupni integrace se zřejmě častěji využívá Harvardská architektura, vysoký stupeň integrace dovoluje připojit různé bloky paměti pomocí vlastních sběrnic. Dělení na „Harvardskou“ a „Von Neumannovu“ architekturu je však při dnešním stupni integrace již poněkud akademické. U moderních struktur se často uživateli adresový prostor jeví navenek jako lineární (Von Neumannovský), zatímco fyzicky jsou paměti k jádru připojeny pomocí několika nezávislých sběrnic (např. jedna sběrnice pro FLASH/ROM (paměť programu), druhá pro

uživatelskou vnitřní RAM a zásobník, třetí pro připojení integrovaných paměťově mapovaných periférií, další pro připojení externí RAM). [12]

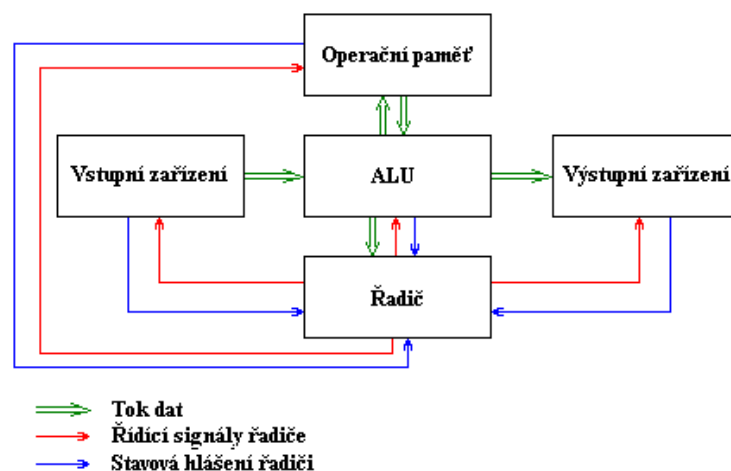


Obrázek 2.4 Mikropočítače PIC ve srovnání s mincí

2.3.1 Rozdělení dle architektur

2.3.1.1 Von Neumannova architektura

Von Neumannova architektura je architektura, pro kterou je typická společná paměť pro data i program. Toto uspořádání má výhody v tom, že nepotřebujeme rozlišovat instrukce pro přístup k paměti dat a paměti programu, což vede k zjednodušení vlastního čipu. Další výhodou je, že je potřeba pouze jedna datová sběrnice, po které se přenáší oba typy dat, což je výhodné v případě použití externích pamětí, kdy se redukuje potřebný počet nutných vstupů a výstupů. Nevýhodou je, že přenos obou typů dat po jedné sběrnici je pomalejší, než při oddělených sběrnicích pro paměti dat a programu. [12]



Obrázek 2.5 Von Neumannova architektura

2.3.1.2 *Harvardská architektura*

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

2.3.2 Rozdělení dle instrukčních souborů

2.3.2.1 *CISC*

CISC označuje procesor se "složitým instrukčním souborem". Procesor podporuje mnoho formátů a druhů instrukcí. Na jednu stranu to znamená úsporu místa v programové paměti (vyšší hustotu kódu), na druhé straně to však znamená komplikovanější dekodér instrukcí ve vlastním mikrokontroléru a pomalejší zpracování instrukcí. [12]

2.3.2.2 *RISC*

RISC označuje procesor s redukováným instrukčním souborem. Základní myšlenkou je omezení počtu a zjednodušení kódování instrukcí, což vede ke zjednodušení instrukčního dekodéru. Hlavní výhodou tohoto přístupu je rychlost, na stejné ploše čipu může být místo 16 bitového procesoru CISC 32 bitový procesor RISC. Nevýhodou je, že pro zakódování instrukce je potřeba více místa, někdy musíme použít dvě instrukce místo jedné, takže klesá hustota kódu. [12]

2.3.3 Základní struktura

Procesor - taktovací kmitočet bývá jednotky až stovky MHz, šířka slova procesoru 4 – 64 bitů

Operační paměť - paměť typu RAM, velikost od jednotek byte do desítek kB

Paměť programu - paměť typu ROM, EPROM, EEPROM nebo flash obsahující program a data, velikost řádově desítky až stovky kB

Vstupně výstupní rozhraní - paralelní a sériové porty, porty komunikačních sběrnic, A/D a D/A převodníky, PWM výstupy, atd.

Další periferie – řadič přerušení, časovače, čítače, watchdog timer (hlídací pes – resetuje systém při jeho zaseknutí), řadič displeje, řadič klávesnice, programovatelné hradlové pole

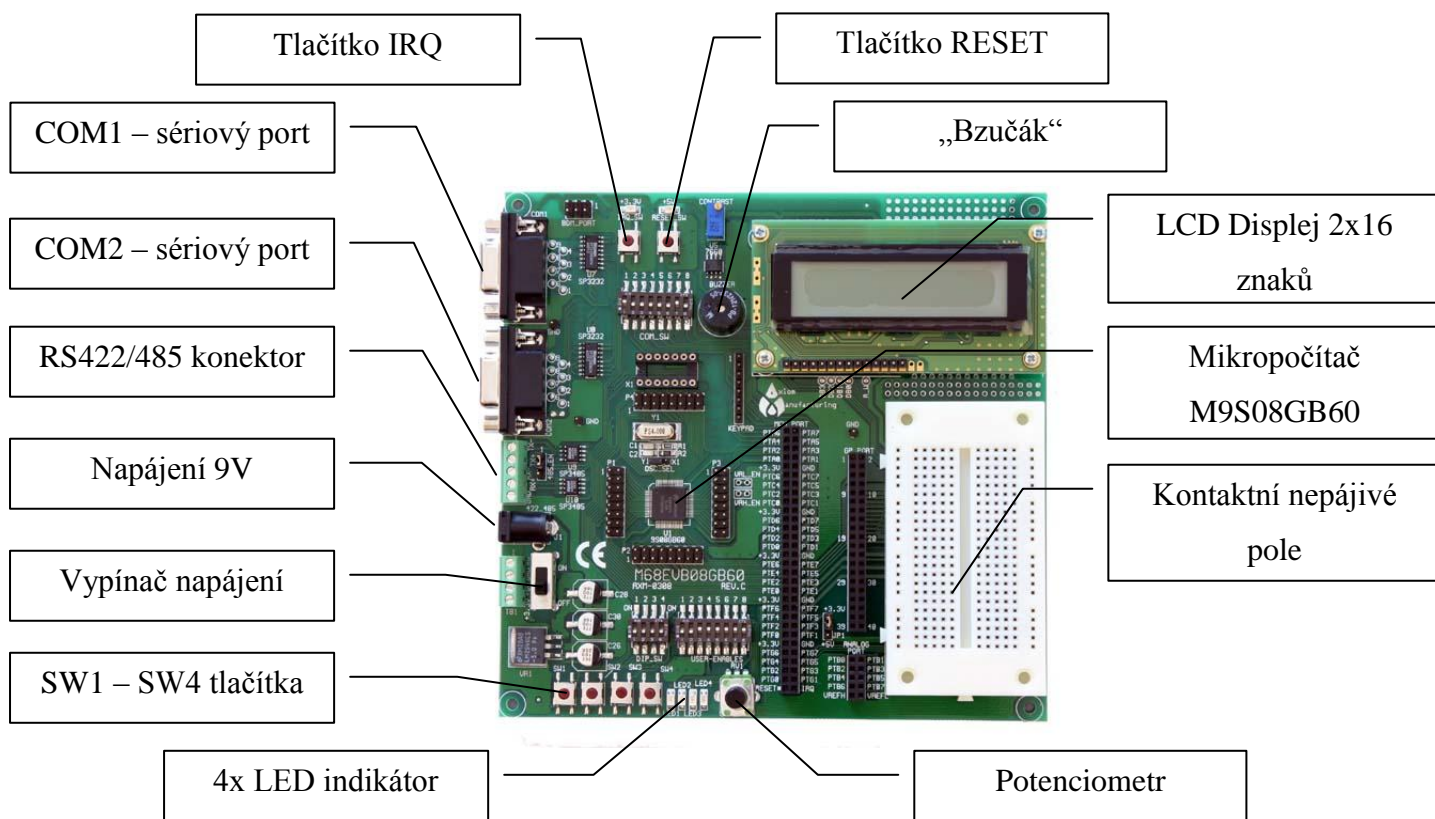
II. PRAKTICKÁ ČÁST

3 POPIS JEDNOTLIVÝCH SOUČÁSTÍ

Tato kapitola obsahuje popis některých důležitých součástí, ze kterých se robot skládá.

3.1 Vývojový kit Freescale M68EVB908GB60

Vývojový kit Freescale M68EVB908GB60 se používá pro programování mikropočítačů na naší fakultě. Jeho hlavní prvek je mikropočítač HCS08, který obsahuje například čítače, časovače, vstupně/výstupní linky. Na kitu lze najít také například bzučák, potenciometr, displej a další periferie. Kit se propojuje přes sériové rozhraní například s PC. Pro jeho programování se na naší fakultě používá program CodeWarrior, který byl také použit při programování robota.



Obrázek 3.1 Popis vývojového kitu M68EVB908GB60

3.1.1 Mikropočítač M9S08GB60

Základní vlastnosti:

- 8 bitová centrální procesní jednotka HCS08, taktovací frekvence max. 40MHz
- 60KB FLASH paměti
- 4KB RAM paměti
- 56 vstupně/výstupních linek na 7 portech (porty A-G)
- 5 kanálový TPM2 časovač
- 3 kanálový TPM1 časovač
- 1x Synchronní sériové periferní rozhraní (SPI)
- 1x I2C rozhraní
- 2x Asynchronní sériové komunikační rozhraní (SCI)
- Interní generátor hodinového kmitočtu s FLL obvodem (32KHz – 20MHz frekvence sběrnice)
- 8 kanálový, 10 bitový A/D převodník
- COP watchdog systém s nastavitelnou časovou prodlevou na 2^{18} nebo 2^{13} cyklů sběrnice
- Systém kontroly napájecího napětí umožňující detekovat pokles napětí pod stanovenou mez
- Podpora režimů se sníženou spotřebou umožňující nasazení v bateriově napájených aplikacích
- BDM rozhraní pro pokročilé ladění a programování aplikací přímo v aplikaci

3.1.2 Napojení periférií na porty

- LED indikátory – port F
- LED1 – PTF0, LED2 – PTF1, LED3 – PTF2, LED4 – PTF3
- Tlačítka – port A
- SW1 – PTA4, SW2 – PTA5, SW3 – PTA6, SW4 – PTA7
- Bzučák – port D
- BUZZER – PTD0
- LCD modul – port G a port E
- PTG3-7 a PTE6-7
- 4 pólový DIP spínač – port B
- PTB4-7
- Potenciometr – port B
- RV1 – PTB0/AD0

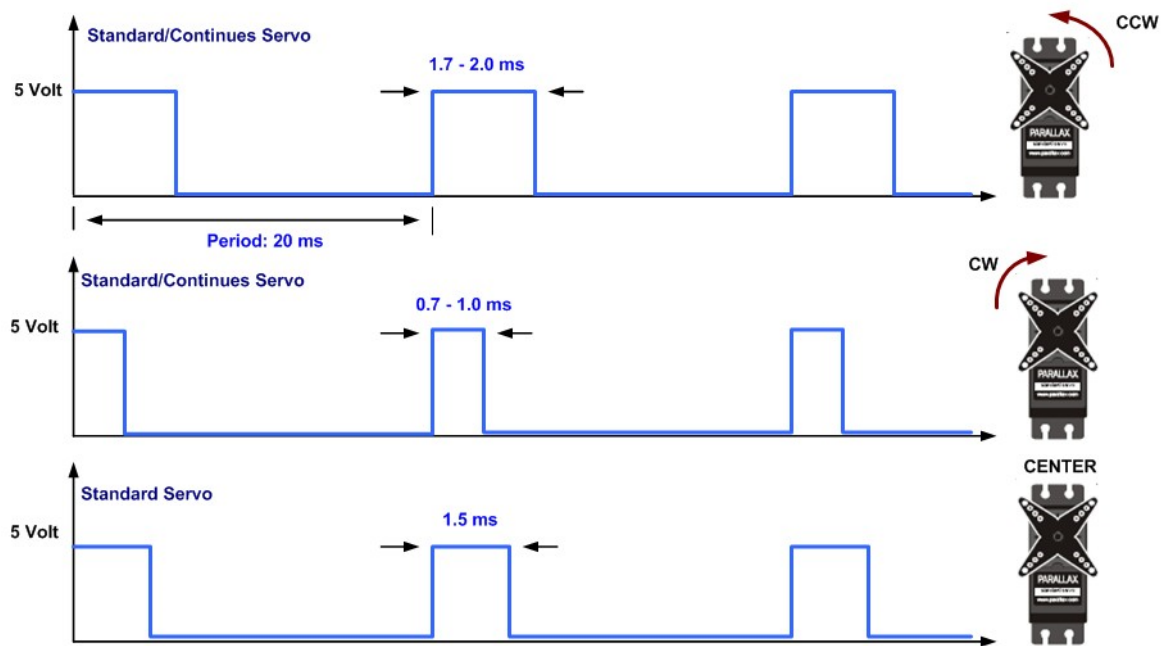
3.1.3 Paměťová mapa

Tabulka 3.1 Adresy pamětí

Paměti	Počáteční adresa	Koncová adresa	Poznámky
Registry periferních obvodů	\$0000	\$007F	
Interní paměť RAM	\$0080	\$107F	4KB
Flash paměť	\$1080	\$17FF	1920B
Registry periferních obvodů	\$1800	\$182B	
Uživatelská flash paměť	\$182C	\$FBCB	57KB
Uživatelské vektory přeršení	\$FBCC	\$FBFF	
Chráněná FLASH s programem „Monitor“	\$FC00	\$FFFF	

3.2 Modelářská serva

Serva převádějí digitální signál z přijímače na mechanický pohyb (otáčení výstupního kotouče nebo páky). Současná serva se připojují tří-žilovým kabelem, v němž dva vodiče slouží pro napájení serva (plus a minus), po třetím se přenáší řídicí signál. Ten má podobu obdélníkových pulsů různé šířky (0,9 - 2,1 ms) o stálé frekvenci. Šířka impulsu určuje polohu výstupní páky. Z toho také vyplývá, že směr pohybu páky serva nemůžete obrátit přepólováním napájení (to by vedlo ke zničení serva). Poloze ovladače ve středu odpovídá neutrální poloha serva (signál o šířce 1,51 ms, u starších RC souprav a serv Multiplex 1,61 ms). [10]



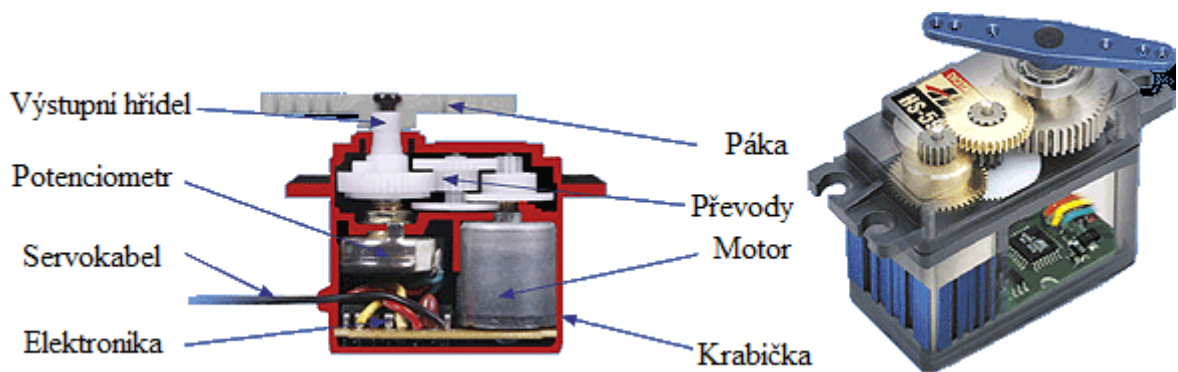
Obrázek 3.2 Průběhy PWM pro servomotory

Pro zabudování serva do kolového robota je nutné servo upravit pro otáčení o 360° . Tuto úpravu dovoluje většina serv, ale je nutné se podívat do technických parametrů serva. Úprava spočívá ve vyjmutí nebo odstranění některých částí převodů tak, aby se vstupní hřídel mohla otáčet o 360° (odstranění mechanického dorazu) a odstranilo se spojení výstupní hřídele s potenciometrem, který poskytuje řídicí elektronice informaci o poloze výstupní hřídele. Potenciometr se zafixuje ve střední poloze, takže pro elektroniku serva je výstupní páka trvale ve střední poloze. Přivedení signálu s hodnotou jinou než odpovídá střední poloze páky, pak způsobí trvalé otáčení výstupu serva. [10]



Obrázek 3.3 Modelářské servo

3.2.1 Popis serva



Obrázek 3.4 Popis serva

Motor

Tah („síla“) serva je zhruba úměrný velikosti motoru.

Deska elektroniky

Dnes většinou zhotovena technikou plošné montáže (SMD).

Převody

Převodový poměr určuje poměr mezi rychlostí a tahem serva. Běžné je, že se serva vyrábějí ve dvojicích, které mají stejný motor a elektroniku, ale liší se převody - např. HS-625 a HS-645. Jedno z nich je potom silnější, ale pomalejší, druhé má zvýšenou rychlost na úkor tahu. Převody jsou nejčastěji plastové, pro větší zatížení a větší spolehlivost pak kovové. [10]

Potenciometr

Snímá polohu výstupního hřídele a uzavírá tak zpětnou vazbu zajišťující správnou činnost serva. U menších a levnějších serv je potenciometr připojen přímo na výstupní hřídel, u serv vyšších kategorií je připojen přes zvláštní převod (tzv. nepřímý náhon), který velmi účinně chrání před přenosem vibrací. Poškození odporové dráhy potenciometru se může projevit jako „zaškubávání“ serva v určité poloze, nebo chvění v neutrální poloze. [10]

Výstupní hřídel

Přenáší pohyb mechanismu serva na ovládací páku. U serv menších a nižších kategorií je uložen v plastovém pouzdru, serva vyšších kategorií mají pro přesnější chod a vyšší životnost kuličková ložiska (jedno nebo ještě lépe dvě). [10]

Krabička

Je vyrobena z houževnatého plastu a je opatřena patkami pro montáž serva do modelu.

Přívodní kabel

Je to tří-žilový kabel se silikonovou izolací s velkou ohebností. U serv větších rozměrů bývá vyroben z krouceného vodiče pro potlačení indukce rušivých napětí při vysokých proudech. Zapojení kabelu: bílý (žlutý, oranžový) vodič: řídicí signál, červený vodič: kladný pól napájení („plus“), černý (hnědý) vodič: záporný pól napájení („minus“, „zem“). [10]

3.2.2 Napájení serv

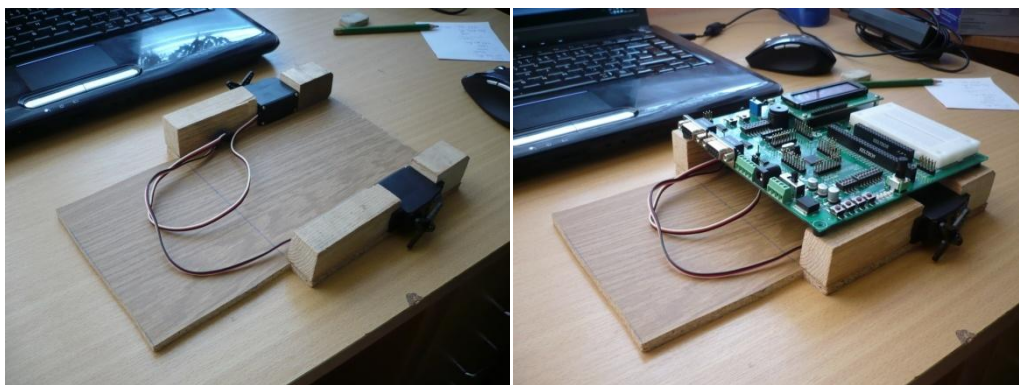
Provozní napětí serv bývá udáváno v rozmezí 4,8 až 6 V. To odpovídá nominálnímu napětí čtyř, respektive pěti-člankové sady NiCd nebo NiMH akumulátorů. Vyšší napájecí napětí přináší vyšší rychlost a vyšší tah serva, což ale také přináší zvýšené opotřebení především motoru serva. Použití pěti-članků je běžné u obřích modelů, kde zvýšené napětí pomáhá odstranit problémy spojené s úbytkem napětí na dlouhých kabelech k servům. [10]

3.2.3 Kompatibilita serv

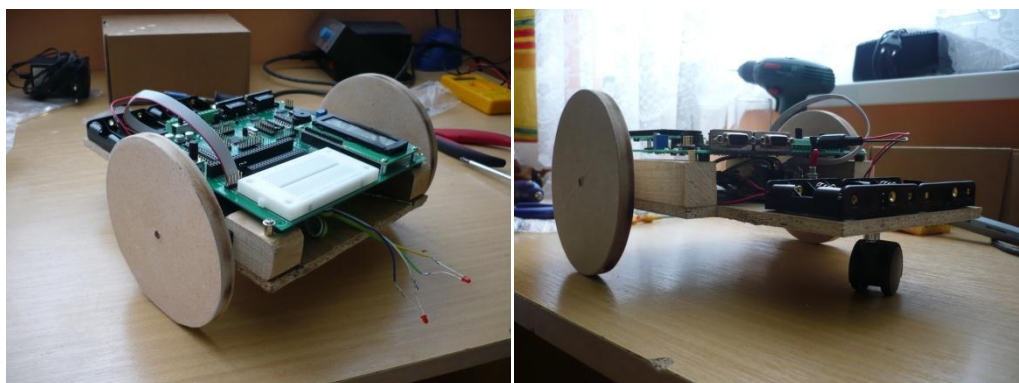
Díky používání podobných elektronických obvodů (tzv. standardní logika TTL) je možné skoro libovolně kombinovat serva a přijímače různých značek při napájení ze čtyř-člankového akumulátoru (4,8 V). Bohužel vzhledem k tomu, že prahové úrovně, při nichž se elektronika serva rozhoduje, zda obdržení signál je logická „1“ nebo „0“, nejsou u všech výrobců úplně stejné a navíc tyto prahové úrovně se mění v závislosti na napájecím napětí, může se při napájení z pěti-članku (zvláště plně na-bitého) stát, že servo není schopno signál správně detekovat. Projevuje se to jako chaotické škubání nebo chvění páky serva. Pokud se napájecí napětí sníží, normální funkce serva se obnoví. (Obdobná situace může nastat i při silném poklesu napájecího napětí.) [10]

4 MECHANICKÁ ČÁST

Mechanická část robota je vytvořena převážně ze dřeva. Spodní a zároveň hlavní část je tvořena deskou o velikosti 17x17 cm, což vychází z velikosti vývojového kitu. Deska je zároveň prodloužena o užší a kratší část 10x14cm, která je určena pro držáky na akumulátory. Po bocích hlavní desky robota jsou navrtány vzpěry, do kterých se připevní serva, a zároveň je na nich připevněn vývojový kit. Střed je ponechán volný pro elektroniku a kabely. Ve přední části robota je pouze tenký panel, který nám zakrývá kabely, a na něm jsou vyvedeny diody a foto-odpory. Na panelu je ještě přiděláno stínítko, aby byly od sebe foto-odpory dobře odstíněny, a tudíž mohl program lépe vyhodnocovat intenzitu osvětlení. Na spodní straně robota se nachází všesměrové nehnané kolo. K servům jsou připevněna dřevěná kola o průměru 12,5 cm a šířce 1 cm.



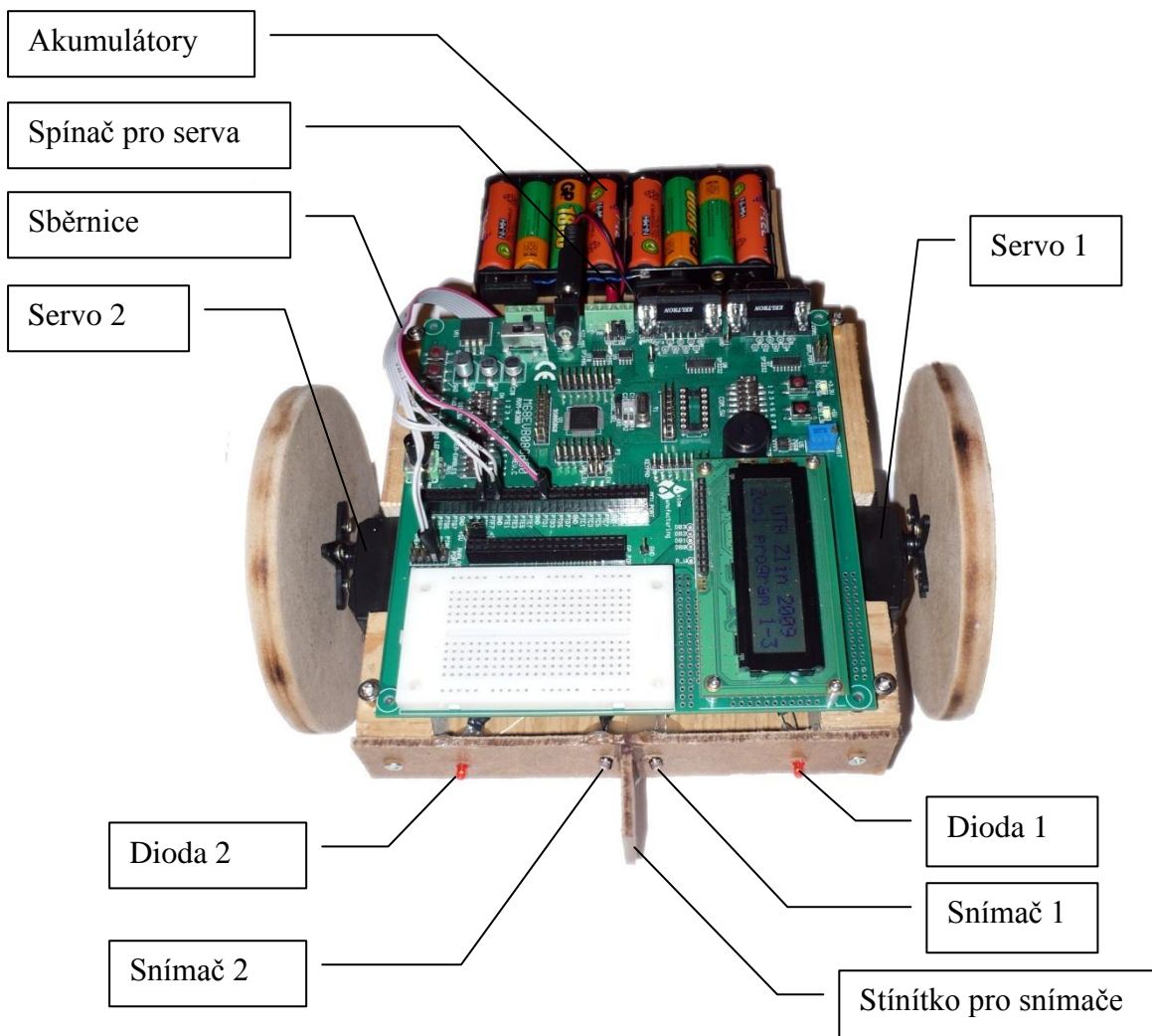
Obrázek 4.1 Podvozek robota, podvozek robota s vývojovým kitem



Obrázek 4.2 Robot s koly

4.1 Popis robota

Vývojový kit je připevněn na vrchní část robota čtyřmi šrouby. (Obrázek 4.3) Snímače a diody jsou v přední části robota, kde je také umístěno stínítko. To slouží pro světelné oddělení snímačů. Pokud bychom stínítko mezi snímače neumístili, tak by výstupní hodnoty byly stále téměř stejné. Po bocích robota jsou velká hnaná kola, připevněná na serva. Podvozek je doplněn třetím kolem, které je pouze vlečné a všesměrové a nachází se v zadní části robota pod akumulátory. Pro provoz robota je potřeba 8 tužkových AA akumulátorů.



Obrázek 4.3 Popis robota

4.2 Parametry robota

Šířka:	170 mm
Délka:	300 mm
Výška:	130 mm
Rozpětí předních kol:	230 mm
Výška podvozku:	45 mm
Váha:	1 kg
Rychlost:	0,4 m/s

5 ELEKTRONICKÁ ČÁST

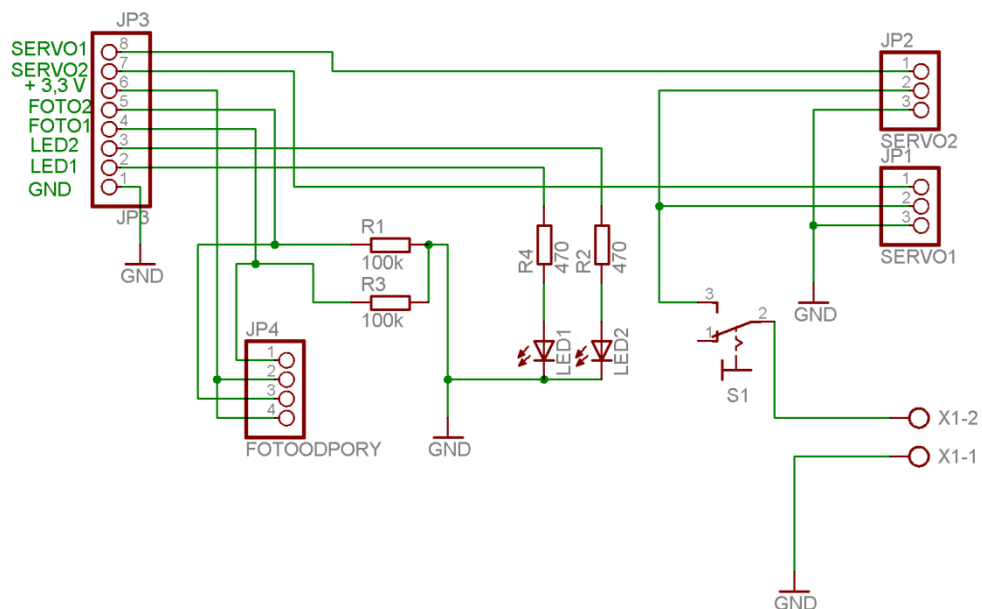
5.1 Napájení a akumulátory

Robot bude operovat s dvěma napětími. Jedno napájí vývojový kit a druhé servomotory. Napětí pro vývojový kit je předpokládáno kolem 9V. K tomu použijeme 8 kusů sériově zapojených akumulátorů AA, které mají napětí 1,2V. Což nám vyjde na 9,6V. Pro napájení servomotorů je potřeba napětí kolem 5V. K tomu použijeme 4x akumulátor AA, což nám dá 4,8V. Abychom ale ušetřili místo a redukovali váhu robota, tak k napájení servomotorů použijeme 4 akumulátory, které napájí vývojový kit.

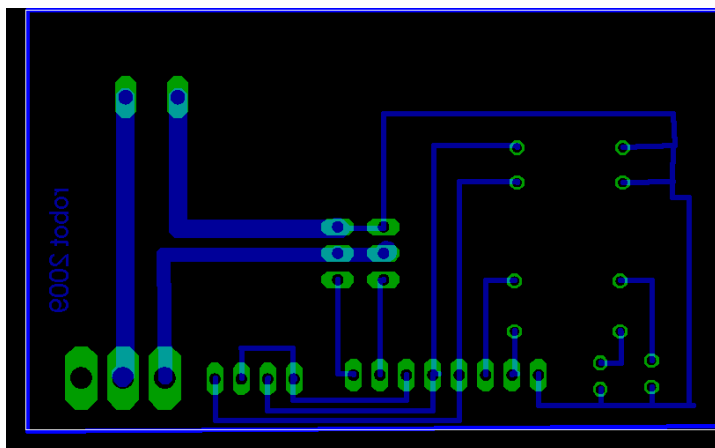
Další periferie jako foto-odpory a diody jsou napájeny napětím 3,3V, které přivedeme z vývojového kitu.

5.2 Přídavný tištěný spoj

Napájení pro servomotory je přivedeno do přídavného tištěného spoje. Do něj jsou také připojeny přívodní kabely servomotorů, foto-odporů a diod. Z tohoto spoje vyvádíme osmi žilovou sběrnici, po které jdou řídicí signály pro servomotory a diody, a přivádíme po ní analogový signál z foto-odporů. Touto sběrnici nám také jde napětí 3,3V pro diody a propojení uzemnění. Tištěný spoj má také přepínač, který nám spíná napětí 4,8V pro serva.



Obrázek 5.1 Schéma zapojení přídavného tištěného spoje



Obrázek 5.2 Návrh tištěného spoje v programu Eagle

5.3 Propojení tištěného spoje s vývojovým kitem

Kromě propojení diod, foto-odporů a servo-motorů, které jsou popsány níže, je na přídatný tištěný spoj přivedeno také napětí 3,3V (pin 39) a uzemnění GND (pin 40)

5.3.1 Ovládání servomotorů (PWM)

Serva jsou ovládána signálem pulzně-šířkovou modulací. Tento signál je generován časovačem mikropočítače a je dostupný na vývodech PTD0 a PTD1 mikropočítače. Serva připojujeme na MCU_PORT, což je konektor na vývojovém kitu. (Obrázek 5.3)

Servomotor1: PTD0=TMP1CH0 (pin 27)

Servomotor2: PTD1=TMP1CH1 (pin 28)

5.3.2 Ovládání diod

Diody připojujeme taktéž na MCU_PORT. Začínáme od PTF4, protože první čtyři bity jsou registrovány pro diody, které má vývojový kit přímo na sobě.

Dioda1: PTF4 (pin 43)

Dioda2: PTF5 (pin 44)

PTA6/KBD6	1	2	PTA7/KBD7
PTA4/KBD4	3	4	PTA5/KBD5
PTA2/KBD2	5	6	PTA3/KBD3
PTA0/KBD0	7	8	PTA1/KBD1
3.3V	9	10	GND
PTC6	11	12	PTC7
PTC4/CLKOUT	13	14	PTC5
PTC2/SDA	15	16	PTC3/SCL
PTC0/TXD2	17	18	PTC1/RXD2
3.3V	19	20	GND
PTD6/TPM2CH3	21	22	PTD7/TPM2CH4
PTD4/TPM2CH1	23	24	PTD5/TPM2CH2
PTD2/TPM1CH2	25	26	PTD3/TPM2CH0
PTD0/TPM1CH0	27	28	PTD1/TPM1CH1
3.3V	29	30	GND
PTE6	31	32	PTE7
PTE4/MOSI	33	34	PTE5/SPSCK
PTE2/SS*	35	36	PTE3/MISO
PTE0/TXD1	37	38	PTE1/RXD1
3.3V	39	40	GND
PTF6	41	42	PTF7
PTF4	43	44	PTF5
PTF2	45	46	PTF3
PTF0	47	48	PTF1
3.3V	49	50	GND
PTG6	51	52	PTG7
PTG4	53	54	PTG5
PTG2/EXTAL	55	56	PTG3
PTG0/BGND/MS	57	58	PTG1/XTAL
RESET*	59	60	IRQ

Obrázek 5.3 Schema zapojení

konektoru MCU_PORT

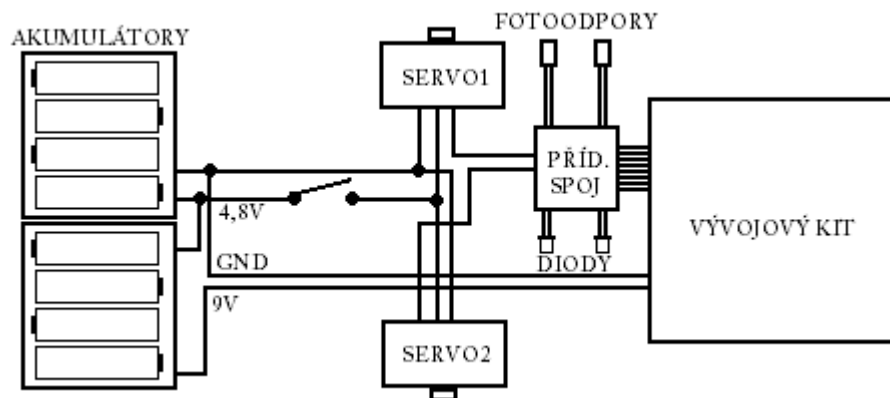
5.3.3 Snímání foto-odporů

Připojujeme na ANALOG_PORT (Obrázek 5.4), protože nám z foto-odporů přichází spojitá informace, kterou A/D převodníkem digitalizujeme. Snímače připojujeme od PTB2, protože první bit je rezervován potenciometrem integrovaným na plošném spoji.

Snímač1: PTB3 (pin 4), Snímač2: PTB2 (pin 3)

PTB0	1	2	PTB1
PTB2	3	4	PTB3
PTB4	5	6	PTB5
PTB6	7	8	PTB7
VREFH	9	10	VREFL

Obrázek 5.4 Schéma analogového konektoru



Obrázek 5.5 Schéma zapojení robota

5.4 Uvedení robota do provozu

Mechanicky připevníme na robota vývojový kit (čtyři šroubky). Dáme si přitom pozor na sběrnici, kterou vedeme bokem kolem serva 1. Nyní následuje připojení sběrnice do správných pinů. Vývody sběrnice máme spárovány podle periférií, ke kterým vedou. Nejprve zapojíme kabely od foto-odporů (s popiskem FOTOODPORY) do analogového portu (ANALOG_PORT) a to tak, že použijeme PTB2 a PTB3, tudíž piny 3 a 4. Všechny ostatní kabely se budou zapojovat do MCU_PORT-u. Zapojíme kabely označené DIODY do pinů PTF4 a PTF5 (pin 43 a 44). Dále zapojíme kabely s popiskem SERVA do pinů PTD0 a PTD1 (pin 27 a 28). Nakonec zapojíme kabel označený 3,3V+GND do pinů 39 a 40. Pokud by se nám při programování stalo, že máme přehozené strany u některých zařízení, přehodíme mezi sebou piny.

Pro provoz robota potřebujeme 8 tužkových AA baterií. Čtyři jsou použity pro napájení serv a všech osm je použito pro napájení kitu. Stabilizace není potřebná, protože serva mají toleranci napětí a kit má integrovaný stabilizátor. Před spuštěním robota je potřeba přepnout také spínač na přídavném plošném spoji do polohy ON.

Program se nahrává přes port COM1 vývojového kitu. Pro programování robota je potřeba softwarový ovladač, jehož sestavením se zabýváme v softwarové části práce.

6 SOFTWAREVÁ ČÁST

Samotná programová část robota se řeší pomocí vývojového kitu s mikropočítačem HCS08. Přes sériový port se do kitu mikropočítače nahraje program, který poté ovládá servo-motory, fotorezistory, diody a další periférie, které má vývojový kit přímo na sobě jako například displej a tlačítka. K robotu je vytvořen ovladač v jazyce symbolických adres. V něm jsou funkce pro ovládání a jejich popis, které jsou velké zjednodušení pro programátory, kteří budou s robotem pracovat. Tento ovladač zastupují soubory `robot_gb60.asm` a `robot_gb60.inc`. Tyto soubory se implementují do programu pro vytvoření hlavního programu a poté se jen podprogramy z ovladače volají.

6.1 Přehled funkcí ovladače

robotinit – Funkce nastavuje časovač pro PWM modulaci, porty pro servomotory, diody, foto-odpory, nastavuje A/D převodník. Funkce také mění frekvenci CPU mikropočítače na 20 MHz, aby servo nezajelo do krajní polohy po odpojení debuggeru. Modulace řídicí serva je totiž nastavena pro 20 MHz, ale standardní frekvence po odpojení debuggeru je 8 MHz. Tato funkce by se měla vždy volat na začátku programu, který bude s robotem pracovat.

servo1 – Funkce očekává vstupní parametr v registru HX a to je šířka impulzu v tisícinách milisekundy. Pro servo1 je pro pohyb otáčení dopředu hodnota 2500 a dozadu 1250, pro zastavení se dá do HX hodnota 0. Funkce zároveň ošetřuje vstupní parametry mimo rozsah. (Př.: LDHX 1250)

servo2 – Funkce shodná s funkcí servo1. Je pouze potřeba si uvědomit, že motorčky jsou k sobě otočeny o 180°, což má za následek opačné vstupní hodnoty. (dopředu LDHX 1250, dozadu LDHX 2500)

robot_go – Funkce nastaví servům takovou šířku řídicího impulzu, aby jel robot dopředu. Zavolání této funkce rozjede robota směrem dopředu.

robot_back – Funkce nastaví servům takovou šířku řídicího impulzu, aby jel robot dozadu. Zavolání této funkce rozjede robota směrem dozadu.

robot_stop – Funkce nastaví servům nulovou šířku řídicího impulzu. Zavolání této funkce robota zastaví.

robot_l – Funkce nastaví oběma servům stejnou šířku řídicího impulsu, což znamená, že se servo1 točí dozadu a servo2 dopředu. To má za následek otočení o zhruba 90°, kdy se robot zastaví. Zavolání této funkce otočí robota směrem doleva.

robot_r – Funkce nastaví oběma servům stejnou šířku řídicího impulsu, což znamená, že se servo1 točí dopředu a servo2 dozadu. To má za následek otočení o zhruba 90°, kdy se robot zastaví. Zavolání této funkce otočí robota směrem doprava.

snimac1 – Funkce nemá žádný vstupní parametr. Po zavolání této funkce se pomocí A/D převodníku zjistí hodnota intenzity osvětlení na foto-odporu1 v rozsahu 0-255. Hodnota 0 odpovídá absolutní tmě a hodnota 255 velmi silnému osvětlení. Tato hodnota je po zavolání funkce uložena v registru A.

snimac2 – Funkce nemá žádný vstupní parametr. Po zavolání této funkce se pomocí A/D převodníku zjistí hodnota intenzity osvětlení na foto-odporu2 v rozsahu 0-255. Hodnota 0 odpovídá absolutní tmě a hodnota 255 velmi silnému osvětlení. Tato hodnota je po zavolání funkce uložena v registru A.

6.2 Ukázkový program

Pro otestování funkčnosti robota a ovladače a pro demonstraci jeho možností byl vytvořen ukázkový program.

6.2.1 Popis programu

Po spuštění programu se nám na displeji zobrazí úvodní informace a text „Zvol program 1-3“. To je výzva pro uživatele, aby pomocí tlačítek SW1-3 zvolil variantu demo programu. Tlačítko SW4 je nastaveno tak, že jeho stiskem vyskakujeme z jakéhokoliv z demo programů do hlavní smyčky. Po stisku tlačítka má uživatel 5 sekund, aby robota položil na zem. Po tuto dobu je na displeji zobrazeno „STARTUJU“.

Po stisku tlačítka SW1 se spustí program DEMO1. V tom je ukázka jednoduchého programu pro nastavení konstantní dráhy robota. Robot je tady ovládán pomocí funkcí ovladače robot_go, robot_back, robot_l, apod. Robot bude jezdit vytvořenou trasu až do stisku tlačítka SW4, kdy dojede cyklus a vrátí se do hlavní smyčky.

Po stisku tlačítka SW2 se spustí program DEMO2. Zde robot porovnává hodnotu nastavenou na potenciometru vývojového kitu s hodnotami snímačů. Pokud je hodnota snímačů vyšší než hodnota potenciometru, robot začne točit koly. Pokud přesáhne hodnota

levého snímače hodnotu potenciometru, roztočí se levé kolo a analogicky pro pravý snímač. To má za následek, že robot sleduje světlo. Jde vyzkoušet, například když na robota posvítíte baterkou. Hodnota potenciometru je vlastně citlivost robota na přijímané světlo.

Po stisku tlačítka SW3 se spustí program DEMO3. Tento režim by se dal nazvat jako „úprk před světlem“. Robot porovnává obě hodnoty potenciometrů a začne točit tím kolem, kde hodnota snímače je větší. To znamená, že robot utíká před světlem. Zde je také možnost nastavit hodnotu potenciometru, která robotovi říká, jaká intenzita světla už je dostačující pro zastavení. Dalo by se říct, že robot hledá v okolí takovou intenzitu světla, která je nižší než hodnota potenciometru. Jakmile ji robot najde, zastaví se.

Pro zjednodušení programu je samozřejmě potřeba do projektu vložit ovladače robota a displeje na kitu.

6.2.2 Ukázka práce s funkcemi ovladače

Následující kód je ukázka základních funkcí pohybu robota. Robot se po spuštění rozjede, po chvíli zatočí vlevo a tento cyklus čtyřikrát opakuje. Na začátku programu je vidět volání inicializace robota. Ta se provede jen jednou.

```

        INCLUDE 'derivative.inc'
        INCLUDE 'robot_gb60.inc'

; export symbols
        XDEF _Startup, main
        XREF __SEG_END_SSTACK

MY_ZEROPAGE: SECTION  SHORT
test rmb 1

MyCode:   SECTION
main:
_Startup:
        LDHX    #__SEG_END_SSTACK
        TXS
        CLI           ; povolení přerušeni
        NOP
        jsr robotinit ; inicializace robota
        mov #0, test  ; nulování proměnné test
mainLoop: jsr robot_go ; volání fce pro rozjezd
          jsr cekej   ; volání fce čekej
          jsr cekej
          jsr robot_l ; volání fce pro zatočení vlevo
          jsr robot_go
          jsr cekej
          jsr cekej
          inc test    ; přičtení 1 k proměnné test
          lda test    ; načtení test do registru A

```

```
        cmp #3                ; porovnání s 3
        beq konec            ; pokud se rovná skoč na konec

        feed_watchdog       ; reset watchdogu
        BRA    mainLoop      ; skok na začátek programu

konec   jsr robot_stop       ; volání fce pro zastavení
        jsr cekej            ;
        nop                  ;
        bra konec            ; skok na návěští konec

cekej   ; podprogram cekej
        PSHH
        PSHX
        PSHA
        LDA #5
n2      ldhx  #$ffff
n1      AIX #-1
        feed_watchdog
        CPHX #0
        BNE  n1
        DECA
        BNE  n2
        PULA
        PULX
        PULH
        RTS
```

ZÁVĚR

Tato bakalářská práce měla za cíl vytvořit modul robota, který se bude využívat jako učební pomůcka pro programování mikropočítačů. Jako základní prvek jsme použili vývojový kit M68EVB908GB60, který se na naší fakultě používá. Díky kitu byla práce přehlednější a jednodušší, než kdybychom robota sestavovali pouze s mikropočítačem a další komponenty přiřazovali zvlášť.

V teoretické části je nejprve popis robotů jako takových, jejich historie, rozdělení dle určitých kritérií. Dále jsou podrobněji popsány možnosti kolových robotů a to proto, že právě kolového robota máme za cíl vytvořit. Další kapitola se zabývá mikropočítači. Jejich stručnou historií, popisem a rozdělením. Podkapitoly tvoří systém embedded a jednočipový počítač.

V první kapitole praktické části se zabýváme součástmi, které budeme pro vytvoření robota potřebovat. Podrobně se věnujeme vývojovému kitu M68EVB908GB60 a tím pádem i mikropočítači HCS08. Důležitá je podkapitola, která se zabývá servo-motory. V ní je vysvětlen způsob řízení servo-motoru PWM modulací, hlavně pak nastavení správné šířky impulsu. Další kapitoly jsou už věnovány čistě výrobě robota. Nejdříve se zabýváme mechanickou částí, sestavením dílů. Následuje elektronická část, kde je popsána podrobně výroba přídavného plošného spoje. Poslední kapitola se zabývá softwarovým vybavením robota, hlavně ovladače. Vytvořený demo-program dobře demonstrovuje možnosti robota.

Na přiloženém CD se nachází text bakalářské práce, zdrojové soubory k ovladači a k demo-programu.

ZÁVĚR V ANGLIČTINĚ

The goal of this bachelor project was to build a robot, which will be used as a teaching tool for the programming of microcontrollers. As an essential element, I used the development kit M68EVB908GB60, which is used on our faculty. With the kit, the work was easier and simpler than if the robot building only with a microcontroller and other components are involved separately.

The theoretical part consists of the description of the robots themselves, their history and their division into groups according to different criteria. Further there are detailed descriptions of the possibilities of wheel robots, because our goal is to create one of these wheel robots. The following chapter is about microcontrollers, their brief history, description and their division. Subchapters include the system embedded and a single chip computer.

In the first chapter of the practical section I mention the parts, which we will need for the construction of the robot. We take a close look at the development kit M68EVB908GB60 and therefore the HCS08 microcontroller too. Important is also the subchapter aimed at the servos. In this subchapter there is a description of the way of controlling the servo through PWM modulation, especially the right adjustment of the width of the impulse. Other chapters are about the construction of the robot. Firstly we take a look at the mechanical part and the construction of the parts. Secondly we take a look at the electronic part, where there is a detailed description of the construction of the additional printed circuit. The last chapter is about the software features of the robot, mainly the controls. The demo-program clearly demonstrates the abilities of the robot.

On the enclosed CD you will find the text of my bachelor dissertation, source filesto the controller and the demo-program.

SEZNAM POUŽITÉ LITERATURY

- [1] SPLÍTKOVÁ, J. *Tištěné ABC*. [Online] [Citace: 22. 4 2009.] Dostupný z WWW: <http://www.iabc.cz/scripts/detail.php?id=10636>.
- [2] LADMAN, J. *Elektronické konstrukce pro začátečníky*. Praha : BEN - technická literatura, 2001. 96 s. ISBN 80-7300-015-6.
- [3] VÁŇA, V. *Začínáme s mikrokontroléry MOTOROLA HC08 NITRON*. Praha : BEN - Technická literatura, 2003. 144 s. ISBN 80-7300-124-1.
- [4] NOVÁK, P. *Mobilní roboty - pohony, senzory, řízení*. Praha : BEN - technická literatura, 2005. 256 s. ISBN 80-7300-141-1.
- [5] Freescale Semiconductor. *MC9S08GB/GT Data Sheet, Rev. 2.3*. 2004.
- [6] Freescale Semiconductor. *HCS08 Family Reference Manual, Rev. 1*. 2003.
- [7] Motorola. *CPU08 Central Processor Unit Reference Manual*. 2001.
- [8] MANN, B. *C pro Mikrokontroléry*. Praha : autor neznámý, 2004. 280 s. ISBN 80-7300-077-6.
- [9] *Wikipedie: Mikrokontrolér PIC*. [Online] [Citace: 22. 04 2009.] Dostupný z WWW: http://cs.wikipedia.org/wiki/Mikrokontrol%C3%A9r_PIC.
- [10] PELIKÁN, D. *RCM Pelikán*. [Online] [Citace: 22. 4 2009.] Dostupný z WWW: <http://www.rcm-pelikan.cz/index.php?sec=1>.
- [11] KÁRNÍK, L. *Multimediální texty*. [Online] [Citace: 22. 04 2009.] Dostupný z WWW: http://robot2.vsb.cz/elekskripta/servisni_roboty/index.htm.
- [12] *Wikipedie: Jednočipový počítač*. [Online] [Citace: 22. 04 2009.] Dostupný z WWW: http://cs.wikipedia.org/wiki/Jednočipový_počítač.
- [13] *Wikipedie: Embedded systém*. [Online] [Citace: 22. 04 2009.] Dostupný z WWW: http://cs.wikipedia.org/wiki/Embedded_systém.
- [14] *Wikipedie: Robot*. [Online] [Citace: 23. 4 2009.] Dostupný z WWW: <http://cs.wikipedia.org/wiki/Robot>.
- [15] *History of the Jaquet-Droz androids, three extraordinary automatons of the XVIIIth century*. [Online] [Citace: 22. 4 2009.] Dostupný z WWW: http://www.automates-anciens.com/english_version/automatons-music-boxes/jaquet-droz-androids.php.

- [16] *Karel Čapek - O původu slova robot.* [Online] [Citace: 22. 4 2009.] Dostupný z WWW:
<http://capek.misto.cz/robot.html>.
- [17] *Wikipedie: Základy robotiky.* [Online] [Citace: 23. 4 2009.] Dostupný z WWW:
http://cs.wikipedia.org/wiki/Zákony_robotiky.
- [18] *Wikipedie: Mikropočítač.* [Online] [Citace: 23. 4 2009.] Dostupný z WWW:
<http://cs.wikipedia.org/wiki/Mikropočítač>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PWM	Pulse Width Modulation, pulzně-šířková modulace
RAM	Random-Access Memory, paměť s náhodným přístupem
ROM	Read-Only Memory, paměť pouze pro čtení
EPROM	Erasable Programmable Read-Only Memory, mazatelná programovací paměť pouze pro čtení
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektronicky mazatelná programovací paměť pouze pro čtení
GND	Ground, uzemnění

SEZNAM OBRÁZKŮ

Obrázek 1.1 První android spisovatel (vlevo) a jeho vnitřní struktura (vpravo)	11
Obrázek 1.2 R. J. Wensley představuje Televox	12
Obrázek 1.3 WABOT-1 (vlevo) a WABOT-2 (vpravo).....	12
Obrázek 1.4 Paletovací stroj Pakko (vpravo)	14
Obrázek 1.5 Mobilní robot Sojourner.....	16
Obrázek 1.6 Dvojkolový robot (vlevo), trojkolový robot (uprostřed), čtyřkolový robot s diferenciály (vpravo)	17
Obrázek 1.7 Čtyřkolové pohybové ústrojí řízené Ackermanovým způsobem	17
Obrázek 1.8 Tříkolové pohybové ústrojí řízené Ackermanovým způsobem	18
Obrázek 1.9 Detail všesměrového kola (vlevo).....	18
Obrázek 1.10 Robot „RATLER™“	19
Obrázek 1.11 Robot Sony AIBO (vlevo) a robot IOAN překonávající schody (vpravo).....	19
Obrázek 2.1 Mikropočítač Commodore 64 (1982).....	20
Obrázek 2.2 Mikropočítač Atmel 8051	21
Obrázek 2.3 Embedded systém v podobě RouterBoardu 112 s U.FL-RSMA	22
Obrázek 2.4 Mikropočítače PIC ve srovnání s mincí	23
Obrázek 2.5 Von Neumannova architektura.....	23
Obrázek 3.1 Popis vývojového kitu M68EVB908GB60.....	27
Obrázek 3.2 Průběhy PWM pro servomotory	30
Obrázek 3.3 Modelářské servo	30
Obrázek 3.4 Popis serva.....	31
Obrázek 4.1 Podvozek robota, podvozek robota s vývojovým kitem	33
Obrázek 4.2 Robot s koly	33
Obrázek 4.3 Popis robota.....	34
Obrázek 5.1 Schéma zapojení přídatného tištěného spoje.....	36
Obrázek 5.2 Návrh tištěného spoje v programu Eagle 4.16	37
Obrázek 5.3 Schéma zapojení.....	38
Obrázek 5.4 Schéma analogového konektoru.....	38
Obrázek 5.5 Schéma zapojení robota	39

SEZNAM TABULEK

Tabulka 3.1 Adresy paměti	29
---------------------------------	----

SEZNAM PŘÍLOH

PŘÍLOHA 1 – Ovladač robota robot_gb60.asm

PŘÍLOHA 2 – Ovladač robota robot_gb60.inc

PŘÍLOHA 1: OVLADAČ ROBOTA ROBOT_GB60.ASM

```
XDEF
robotinit, servo1, servo2, snimac1, snimac2, robot_go, robot_stop, robot_back, ro
bot_l, robot_r           ; zpristupneni funkci ovladace okoli

INCLUDE 'derivative.inc'

MY_ZEROPAGE: SECTION SHORT           ; Insert here your data
definition

MyCode: SECTION

robotinit
    LDA #$7A
    STA ICGC1
    LDA #$30
    STA ICGC2
srvil
    LDA ICGS1
    AND #%00001000
    BEQ srvil

    MOV #$ff,PTFD                ; inic. datoveho reg. portu F
    MOV #0,PTFPE                 ; odpojime pull-up rezistory
    MOV #%00110000,PTFDD        ; přepnutí diod jako výstup
    mov #%01001100,TPM1SC       ; inicializace TPM1, fsource=busclk,
                                ; prescaler=128
    ldhx #25000                  ; obsah modulo registru pro ttof=0.25s
    sthx TPM1MOD                 ; uložíme do modulo registru
    MOV #%00101000, TPM1C1SC    ; inicializace pro servo na kanálu 1
    MOV #%00101000, TPM1C0SC    ; inicializace pro servo na kanálu 0
    mov #%00001101, ATD1PE      ; pin PTB0,2,3 prepnut do
                                ; režimu vstup AD převodníku
    mov #%11100100,ATD1C        ; inicializace převodníku (AD zap,
                                ; 10bit, zarovnání vpravo, ATD clock 2MHz)
    RTS

servo1
    CPHX #0                      ; porovnáme hodnotu v HX s 0
    BEQ konecsrv1a              ; pokud je nula, skok na konecsrv1a
    CPHX #1250                  ; porovnáme hodnotu HX s 1250
    BLO konecsrv1              ; pokud je menší, skok na konec
    CPHX #2500                  ; porovnáme hodnotu HX s 2500
    BHI konecsrv1              ; pokud je větší, tak skoč na konec
konecsrv1a STHX TPM1C0V         ; zapíšeme hodnotu z HX do registru TPM1C0V
konecsrv1  RTS

servo2
    CPHX #0                      ; porovnáme hodnotu v HX s 0
    BEQ konecsrv2a              ; pokud je nula, skok na konecsrv1a
    CPHX #1250                  ; porovnáme hodnotu HX s 1250
    BLO konecsrv2              ; pokud je menší, skok na konec
    CPHX #2500                  ; porovnáme hodnotu HX s 2500
    BHI konecsrv2              ; pokud je větší, tak skoč na konec
konecsrv2a STHX TPM1C1V         ; zapíšeme hodnotu z HX do registru TPM1C1V
konecsrv2  RTS

robot_go
```

```

LDHX #2500      ; vezmu hodnotu 2500 do registru HX
jsr servol     ; volám podprogram servol
LDHX #1250     ; vezmu hodnotu 1250 do registru HX
jsr servo2     ; volám podprogram servo2
RTS

robot_back
LDHX #1250     ; vezmu hodnotu 1250 do registru HX
jsr servol     ; volám podprogram servol
LDHX #2500     ; vezmu hodnotu 2500 do registru HX
jsr servo2     ; volám podprogram servo2
RTS

robot_stop
LDHX #0        ; zapíšu hodnotu 0 do HX
jsr servol     ; volám podprogram servol
LDHX #0        ; zapíšu hodnotu 0 do HX
jsr servo2     ; volám podprogram servo2
RTS

robot_l
LDHX #1250     ; zapíšu hodnotu 1250 do HX
jsr servol     ; volám podprogram servol
LDHX #1250     ; zapíšu hodnotu 1250 do HX
jsr servo2     ; volám podprogram servo2
jsr cekej     ; volám podprogram cekej
jsr cekej     ; volám podprogram cekej
jsr cekej     ; volám podprogram cekej
jsr robot_stop ; volám podprogram robot_stop
RTS

robot_r
LDHX #2500     ; zapíšu hodnotu 2500 do HX
jsr servol     ; volám podprogram servol
LDHX #2500     ; zapíšu hodnotu 2500 do HX
jsr servo2     ; volám podprogram servo2
jsr cekej     ; volám podprogram cekej
jsr cekej     ; volám podprogram cekej
jsr cekej     ; volám podprogram cekej
jsr robot_stop ; volám podprogram robot_stop
RTS

snimac1
nav0 mov #00000011,ATD1SC ; nastavím hodnotu ATD1SC na PTB3
lda   ATD1SC             ; cekej na dokončení převodu
and #10000000           ; logický součin s %10000000
beq nav0                 ; pokud se rovná skoč na nav0
lda ATD1RH               ; zapíšu do registru A jako výstup podprogramu
rts

snimac2
nav1 mov #00000010,ATD1SC ; nastavím hodnotu ATD1SC na PTB2
lda   ATD1SC             ; cekej na dokončení převodu
and #10000000           ; logický součin s %10000000
beq nav1                 ; pokud se rovná skoč na nav0
lda ATD1RH               ; zapíšu do registru A jako výstup podprogramu
rts

cekej
PSHH
PSHX

```

```

        PSHA
        LDA #5
n2     ldhx  #$ffff
n1     AIX #-1
        feed_watchdog
        CPHX #0
        BNE  n1
        DECA
        BNE  n2
        PULA
        PULX
        PULH
        RTS

TOF_int    pshh
           lda      TPM1SC          ; precteme stav TPM1
           bclr    7,TPM1SC        ; vynulujeme priznak preruseni
           lda      PTFD            ; nacteme stav portu F
           eor     #%00110011      ; negujeme PTFD bit0
           sta     PTFD            ; zapiseme novy stav na port F
           pulh
           rti

; nasmerovani vektoru preruseni TOF od TPM1 na nasi obsluhu

        org  $ffee          ; adresa vektoru preruseni TOF od TPM1
        dc.w TOF_int        ; vyplneni vektoru adresou nasi obsluhy

```

PŘÍLOHA 2: OVLADAČ ROBOTA ROBOT_GB60.INC

xref

robotinit, servo1, servo2, snimac1, snimac2, robot_go, robot_stop, robot_back, robot_l, robot_r