

# **Výukový modul pro předmět mikropočítače: ovládání serva**

Educational module for microcontrollers course: servo control

Josef Migdal

---

Bakalářská práce  
2007



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2006/2007

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Josef MIGDAL**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Téma práce: **Výukový modul pro předmět mikropočítače:  
ovládání serva**

Zásady pro vypracování:

1. Seznamte se s vývojovým kitem M68EVB908GB60, který se používá pro výuku programování mikropočítačů na naší fakultě.
2. Navrhněte strukturu a zapojení modulu, který bude ilustrovat ovládání servomotoru pomocí mikropočítače a bude připojitelný na tento vývojový kit.
3. Realizujte funkční prototyp navrženého modulu a ověřte jeho funkci na vývojovém kitu.
4. Vytvořte podpůrné programové vybavení pro obsluhu modulu v jazyce symbolických adres.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

- MC9S08GB/GT Data Sheet, Rev.2.3. Freescale Semiconductor, 2004
- HCS08 Family Reference Manual, Rev.1. Freescale Semiconductor, 2003
- CPU08 Central Processor Unit Reference manual. Motorola, 2001
- VAŠEK V., VAŠEK L.: Programování mikropočítačů. FT VUT v Brně, Zlín 1989
- VÁŇA V.: Začínáme s mikrokontroléry Motorola HC08 Nitron. BEN — technická literatura, Praha 2003
- LADMAN J.: Elektronické konstrukce pro začátečníky. BEN – technická literatura 2002.

Vedoucí bakalářské práce:

**Ing. Jan Dolinay**

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

**13. února 2007**

Termín odevzdání bakalářské práce:

**24. května 2007**

Ve Zlíně dne 13. února 2007

prof. Ing. Vladimír Vašek, CSc.  
*děkan*



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

## **ABSTRAKT**

Cílem bakalářské práce je vytvořit modul pro připojení k vývojovému kitu M68EVB908GB60, který bude sloužit při výuce programování mikropočítače Motorola HCS08. Modul bude umožňovat řízení polohy modelářského servomotoru pomocí výstupních portů s pojistkou proti překročení povoleného rozsahu pohybu a indikací případné chyby. Součástí je návrh a realizace hardwaru modulu i základního softwarového vybavení pro jeho ovládní v jazyce symbolických adres.

Klíčová slova:

Mikropočítač, Motorola, M68EVB908GB60, Servomotor

## **ABSTRACT**

Abstrakt ve světovém jazyce

Aim of this bachelor thesis is to create module for interface with developing kit M68EVB908GB60, which will support education of programming microcontroller Motorola HCS08. Module will enable controlling position of modeler servomotor by the aid of output ports with safety fuse against over-range allowed range movement and indication of eventual mistakes. In this work the design and realization of the hardware of the module is performed and also essential software equipment for its control in assembly language is created.

Keywords:

Microcomputer, Motorola, M68EVB908GB60, Servomotor

Nejprve bych chtěl velmi poděkovat vedoucímu mé práce panu ing. Janu Dolinayovi za projevenou vstřícnost a toleranci při společných konzultačních hodinách na téma mé bakalářské práce. Za jeho podporu ze stránky přínosu informací, odborné vedení a veškeré rady při realizování myšlenek na přídavný hardware pro vývojový kit. Dále pak panu ing. Aloisovi Mynaříkovi za pomoc při výrobě desky plošných spojů na profesionální úrovni. V neposlední řadě patří můj dík rodině a všem, kteří mi poskytli vybavení, prostor a morální podporu pro mou práci.

*Motto*

”

Konec světa bude, až přestanou lidé zpívat.

”

Albert Einstein (\*1879 - †1955)

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně

.....  
Podpis diplomanta

**OBSAH**

<b>ÚVOD .....</b>	<b>7</b>
<b>I TEORETICKÁ ČÁST .....</b>	<b>8</b>
<b>1 MIKROPROCESOROVÉ SYSTÉMY .....</b>	<b>9</b>
1.1 ČINNOST MIKROPROCESOROVÝCH SYSTÉMŮ .....	9
1.1.1 Počítač, procesor, mikroprocesor .....	9
1.1.2 Mikroprocesory v průmyslové automatizaci .....	10
1.1.3 Architektura mikroprocesorů .....	11
1.1.4 Cykly, způsoby adresování .....	12
1.2 MOTOROLA HCS08.....	13
1.2.1 Historie.....	13
1.2.2 Vývojový kit M68EVB908GB60 .....	14
1.2.3 Vývojový kit - základní vlastnosti.....	16
<b>2 SERVOMOTOR .....</b>	<b>17</b>
2.1 VLASTNOSTI SERVOMOTORŮ .....	17
2.2 PRINCIP SERVA.....	19
2.3 VNITŘNÍ STAVBA SERVA .....	20
2.4 ŘÍZENÍ SERVA .....	22
2.4.1 Řídící impulsy .....	22
2.4.2 Rušivé vlivy .....	23
<b>3 EAGLE .....</b>	<b>24</b>
3.1 HLAVNÍ ČÁSTI.....	24
3.2 OMEZENÍ .....	25
<b>II PRAKTICKÁ ČÁST.....</b>	<b>26</b>
<b>4 POPIS HARDWARU PŘÍDAVNÉHO MODULU .....</b>	<b>27</b>
4.1 NAVRŽENÍ ELEKTRICKÉHO ZAPOJENÍ.....	27
4.2 DESKA PLOŠNÝCH SPOJŮ .....	29
<b>5 PROGRAMOVÁ PODPORA PŘÍDAVNÉHO MODULU .....</b>	<b>31</b>
5.1 KNIHOVNA PRO MODUL ŘÍZENÍ SERVA .....	31
5.2 UKÁZKOVÝ PROGRAM ŘÍZENÍ SERVA .....	35
5.2.1 Hlavička programu.....	35
5.2.2 Tělo programu .....	36
<b>ZÁVĚR .....</b>	<b>38</b>
<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>39</b>
<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>40</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>41</b>

## ÚVOD

S vyvíjejícími se trendy velmi narůstá využití mikroelektroniky nejen v průmyslu, ale i v domácnostech. Aniž si to mnohdy uvědomujeme, setkáváme se denně v běžném životě s mnoha rozmanitými prvky obsahujícími integrovanou elektroniku, mezi které patří i mikropočítače. Vyznačují se velkou spolehlivostí a kompaktností, proto jsou určeny především pro jednoúčelové aplikace jako je řízení nebo regulace. Úkolem této bakalářské práce je pokusit se vysvětlit základní pojmy z oblasti mikropočítačů, představit jejich základní vlastnosti a uvést ukázkový příklad jejich aplikace jako řídicí jednotky servomotoru. Podrobný popis tvorby výukového modulu a jeho programové podpory má za úkol pomoci překonávat hranice našich vědomostí v oblasti mikropočítačů, jejich vnitřní stavby, programování a možností jejich reálného využívání v praxi pro zjednodušení a zvyšování efektivity naší práce.

Servomotor stejně jako mikropočítač proniká stále častěji do naší blízkosti, a proto se s přibývajícimi novinkami zvyšují požadavky na naše znalosti z těchto technických okruhů. Dnes se používají serva elektrická, avšak existují také hydraulická, pneumatická či dokonce parní. Často se s nimi setkáváme mnohdy několikrát za den, a proto jedním z cílů této práce je seznámení se základním principem jejich funkčnosti, přiblížit jejich vnitřní stavbu a seznámit s několika základními parametry ovlivňujícími jejich využití. Vhodným příkladem je elektrické servo malých rozměrů, které bude součástí vyrobeného modulu.

Praktická část práce obsahuje základy konstrukce elektrotechnických zařízení pomocí počítačového softwaru od navržení elektronického schéma a jeho četných úprav až po vyhotovení desky plošného spoje, kde je spousta funkcí ulehčujících náročné kroky zjednodušování. Na konci jsou ukázky tvorby programové podpory modulu pro řízení serva v programu Code Warrior programované v jazyku symbolických adres. Vysvětlivky jednotlivých částí kódu doprovázené několika obrázky se snaží pomoci začátečníkům při studii programování nejen v předmětu Mikropočítače.

## **I. TEORETICKÁ ČÁST**



# 1 MIKROPROCESOROVÉ SYSTÉMY

Hlavním cílem této kapitoly je poskytnout základní informace o činnosti mikroprocesorů a mikroprocesorových systémů. Přiblížit jak závisí jednotlivé typy procesorů na zaměření pro jejich aplikaci.

## 1.1 Činnost mikroprocesorových systémů

### 1.1.1 Počítač, procesor, mikroprocesor

Počítač samostatně pracuje podle programu, který je předem vložen do jeho paměti. Je to stroj na číslicové a logické zpracování dat, grafických údajů nebo údajů z výrobního procesu či vědeckého experimentu. Mezi jeho základní části patří procesor, operační paměť a periferní systém.

Nejdůležitější částí počítače je procesor, programovatelný sekvenční logický obvod, který obsahuje **aritmeticko-logickou jednotku** (ALU), která provádí aritmetické operace (binární sčítání, odčítání, porovnávání, případně násobení a dělení) a logické operace (AND,OR,XOR,negaci). Dále pak **řadič** (Control Unit), který řídí činnost procesoru i celého počítače. Neméně důležitou funkci mají **pracovní a pomocné registry**, do kterých jsou ukládány operandy a výsledky aritmetických nebo logických operací. Právě v nich se nachází akumulátor (střadač), který slouží pro uchovávání stavových informací procesoru, různých nastavení a podobně.

Řadič je blok logických obvodů, a obsahuje **registr instrukcí**, který uchovává operační kód instrukce po dobu jejího vykonávání. Dále pak **dekodér instrukcí**, jenž dekoduje operační kód a generuje řídicí signály pro ostatní obvody procesoru i pro vnější okolí (paměti, periférie,...). A **čítač instrukcí** ukazující na adresu v paměti právě prováděné instrukce nebo její části, jehož hodnota je nastavována signály z dekodéru instrukcí. V případě instrukcí větvení programu nebo volání či návratu z přerušení je programový čítač naplněn adresou cíle skoku, adresou začátku podprogramu, nebo návratovou adresou. Při obsluze žádosti o přerušení je do programového čítače uložena adresa začátku přerušovací rutiny obvykle dodaná řadičem přerušení.

Pro uložení kódu, proměnných a konstant slouží **operační paměť**.

Samozřejmostí je styk s vnějším okolím, který zajišťuje **periferní systém**, a může se lišit podle typu mikropočítače. Typické jsou klávesnice, několika řádkový displej, jednotky binárních a analogových vstupů/výstupů, řadiče průmyslových sběrnic, čítače a časovače, nebo třeba jednotky pro zpracování signálů z inkrementálních čidel.

Technologie výroby integrovaných obvodů se na konci 60. let velmi rozvíjela a přinášela výsledky jako integrace všech obvodů procesoru na jednom čipu. Tak vznikl **mikroprocesor**.

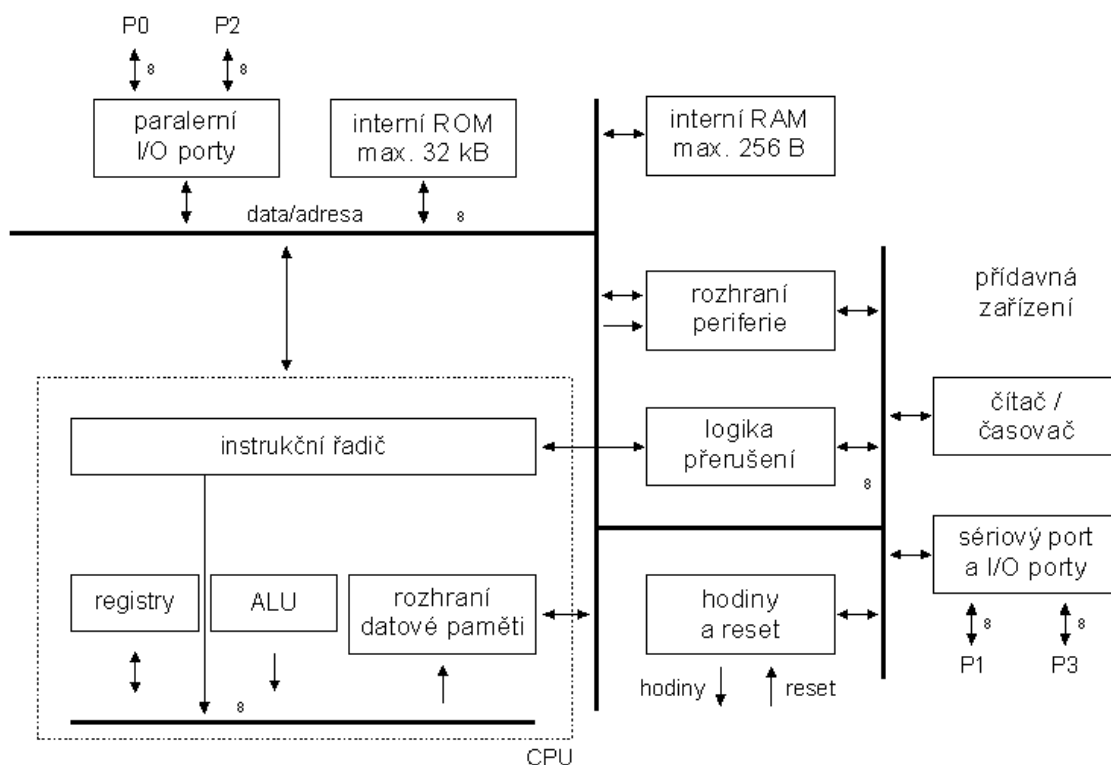


*Obrázek 1. Mikroprocesor průmyslového počítače*

### 1.1.2 Mikroprocesory v průmyslové automatizaci

Hned jedny z prvních mikroprocesorů byly používány kromě personálních počítačů také v zařízeních měřicí techniky, průmyslové automatizace i v oblasti zpracování a analýzy signálů. Bylo však potřeba rozšířit desku s mikroprocesorem o spoustu dalších obvodů jako jsou převodníky, komunikační kanály, čítače, časovače atd. To znamenalo další obvody a součástky. Dalším krokem byla tedy miniaturizace těchto částí a integrování do jednoho čipu. Vzniká integrovaný obvod nazývaný **mikropočítač**, který je svou architekturou přizpůsoben speciálně pro monitorování a řízení různých mechanismů a procesů. Zpravidla obsahují oddělenou paměť programu (ROM, PROM, EPROM) a paměť dat (RAM). Časovač umožňuje synchronizaci s vnějším okolím. Aplikační rozsah těchto systémů je prakticky neomezený. Od běžných elektronických přístrojů používaných v domácnosti (videorekordér, pračka, šicí stroj, kuchyňský sporák) až po nejnáročnější aplikace

v automobilovém průmyslu, leteckém průmyslu a ve vojenství.



Obrázek 2. Obecné blokové schéma mikropočítače

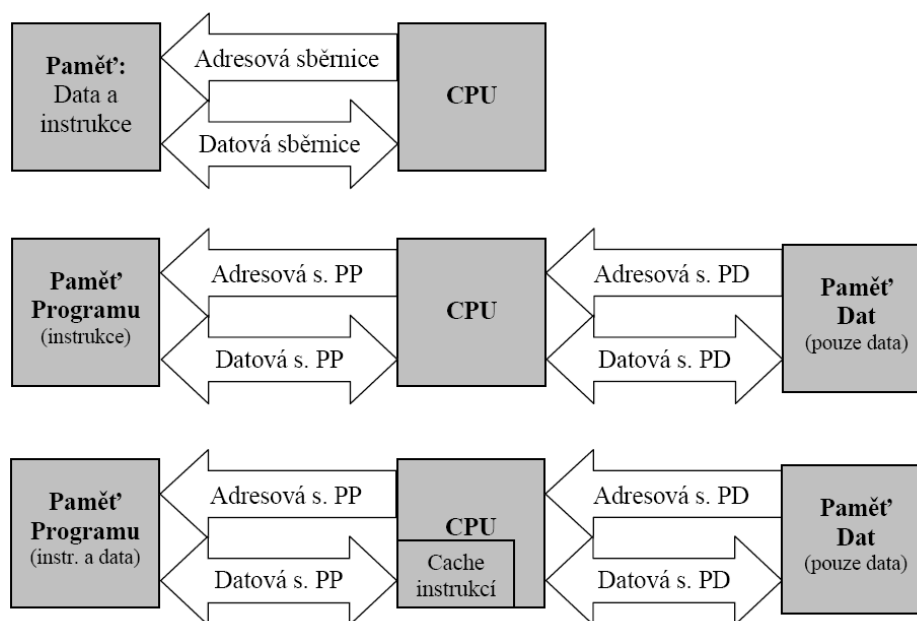
### 1.1.3 Architektura mikroprocesorů

**Von Neumannova architektura** se vyznačuje tím, že procesory mají společnou paměť pro kód i data, tudíž existuje pouze jedna datová a adresová sběrnice. To znamená že zpracovávání je sekvenční. Omezením je rychlost zpracování instrukcí dnešními procesory, která je výrazně vyšší než rychlost komunikace s pamětí. Komunikace s pamětí se tak stává nejslabším článkem řetězu ve von Neumannově architektuře.

**Harvardská architektura** je typická charakteristicky oddělenou pamětí pro kód a pamětí pro data, což procesoru umožňuje číst instrukce i proměnné současně a je tak typickým příkladem paralelního zpracování. Přesto, že se musí zdvojnásobit počet sběrnic (tudíž i vývodů), jsou cennější možnosti na rychlost, které touto úpravou vzniknou. Je vhodná u procesorů s menším adresovým prostorem, ale s požadavky na již zmíněnou rychlost provádění programu.

**Modifikovaná harvardská architektura** vychází z harvardské architektury, neboť ani ta později nestačila požadavkům na rychlost stále novějších procesorů. Má oproti ní

upravenou programovou paměť tak, že do ní lze ukládat nejen instrukce, ale i data. To umožňuje, například při provádění instrukce pro násobení dvou operandů, přístup k oběma operandům umístěným v oddělených paměťových prostorech najednou. Jedna operace násobení a sčítání je provedena v jednom instrukčním cyklu, tak se opět přibližuje vysokým nárokům na rychlost. [10]



Obrázek 3. Architektury mikroprocesorů

#### 1.1.4 Cykly, způsoby adresování

Cykly rozumíme časy, za jaké vykoná mikroprocesor určité úkony nebo jejich skupiny. **Instrukční cyklus** je doba, která je potřebná k výběru instrukce z paměti a její následné vykonání. **Strojový cyklus** je každá komunikace procesoru s okolím (např. přístup do paměti). Je základní ucelenou aktivitou procesoru a z několika strojových cyklů se pak skládá cyklus instrukční. **Hodinový cyklus** je definován jako interval mezi dvěma hranami hodinových impulsů. Strojový cyklus se skládá z jednoho nebo více hodinových cyklů.

Proměnná s níž chceme provést například aritmetickou operaci se nazývá **operand**. Operand je třeba během programu ukládat a k tomu slouží adresování. Existuje několik způsobů jak jej adresovat. Zde naprosto postačí je pouze zmínit, například adresování indexové, nepřímé, nultého řádu, prvního a druhého řádu atd.

## 1.2 Motorola HCS08

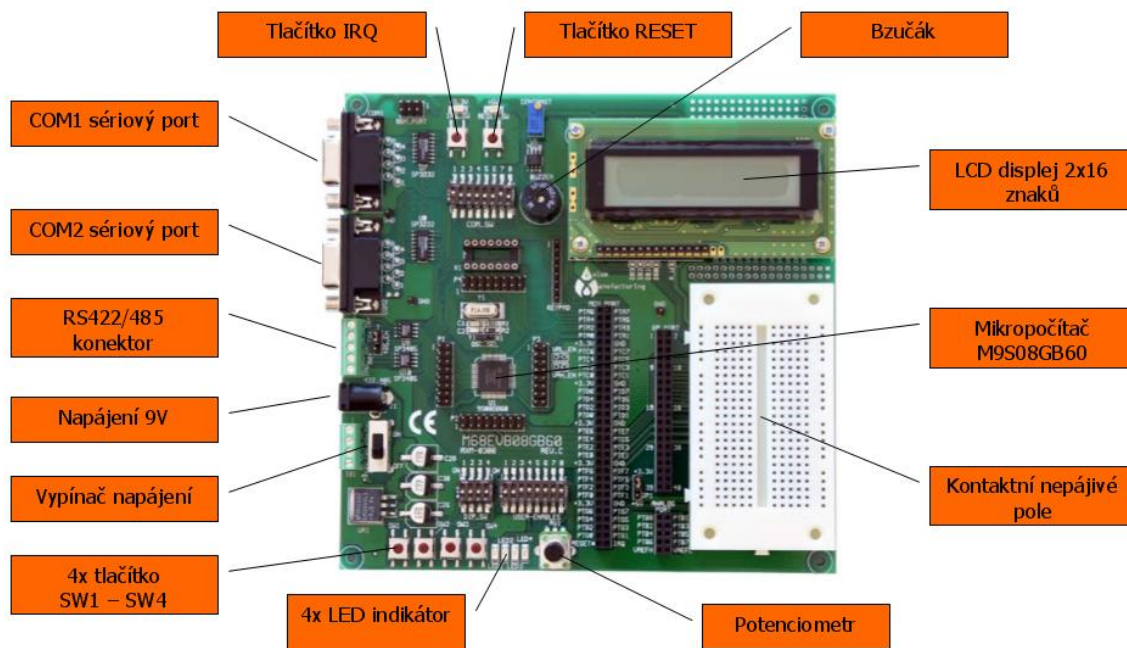
Největšího rozšíření z hlediska aplikací doznaly mikropočítače dvou významných výrobců mikroprocesorů – firem Motorola a Intel. Obecně platí, že pro aplikaci se použije ten nejjednodušší mikropočítač, který pro splnění účelu vyhovuje. Tato motivace je také důvodem k tomu, že Motorola nabízí velmi široké spektrum konfigurací svých výrobků určených pro průmyslové řízení lišících se drobnými změnami v kapacitě paměti, počtu vstupů/výstupů, použitelných periférií apod.

### 1.2.1 Historie

Historie firmy Motorola sahá do roku 1928, kdy bratři Paul a Joseph Galvinové založili v Chicagu, USA malou ruční dílnu na výrobu elektronických součástek radiopřijímačů. Určili tím budoucí primární orientaci firmy na bezdrátovou mobilní komunikaci. V roce 1949 vyčleňuje firma pobočku pro výzkum, vývoj a využití enormního potenciálu tranzistoru a brzy vlastní jednu z předních světových továren na výrobu polovodičových součástek. Prvním mikroprocesorem, uvedeným na trh v roce 1974, byl osmibitový 6800. Obsahoval 4000 tranzistorů a nejpodstatnějším rozdílem, oproti konkurenčnímu Intelu je mapování vstupně/výstupních periférií do paměti, což umožňuje využít pro práci s nimi standardní instrukce MOV. Mikroprocesor 6800 se stal rodičem dnešních moderních mikropočítačů i evolučním stavebním kamenem PC firmy Apple Macintosh.

Mikropočítače rodiny HC08 představují typické zástupce své kategorie, navazující na řadu HC05, se kterou jsou zpětně kompatibilní na úrovni zdrojového kódu. Konstrukčně odpovídají osmibitové architektuře s uspořádáním modelu CISC (komplexní instrukční soubor), von Neumannova typu, kdy data i program jsou umístěny ve stejném paměťovém prostoru. Struktura jádra CPU08, vyráběného technologií HCMOS, vychází z historického mikroprocesoru 6800 a staršího jednočipového mikropočítače 6801. Charakteristickou vlastností je minimum pracovních registrů. Odstranění druhého střadače bylo vyváženo doplněním o instrukce pro rychlou a flexibilní práci s pamětí RAM a registry periférií. Jádro je také zdokonaleno lepším řízením smyček, upravenými instrukcemi BCD kódu a předvídáním instrukcí, s úsporou 20-30%. [6]

### 1.2.2 Vývojový kit M68EVB908GB60



Obrázek 4. Hlavní části M68EVB908GB60

Na výše uvedeném obrázku je uveden popis vývojového kitu s mikroprocesorem motorola, který je vynikající pomůckou při výuce programování a řízení moderní elektroniky. Jeho cena je přímo úměrná možnostem poskytujícím rozmanité prvky pro vizuální i manuální cvičení. Rozměry celé desky jsou přibližně 15x15 cm a její hlavní částí je výše zmíněný mikroprocesor. Součástí kitu je i napájecí adaptér poskytující stejnosměrné napájecí napětí 9V a sériový kabel sloužící pro naprogramování a komunikaci s počítačem. Pro stejný účel jsou na okrajích kitu dva konektory COM1 a COM2 kam se kabel sériového rozhraní připojí. Bez Pc se dá samotný kit ovládat (reset, přerušení, nastavení) pomocí několika tlačítek a potenciometru. Při výuce oceníme možnost výstupu akustického signálu z integrovaného bzučáku či signalizace několika led diod. Značnou výhodou je také dvouřádkový displej a kontaktní nepájivé pole poskytující místo pro malé doplňky. Většina těchto periférií úzce souvisí s výstupním portem a jejich vzájemné ovlivňování se dá jednotlivě vypínat pomocí tzv. dipů umístěných hned pod mikroprocesorem.

Díky stabilizátoru lze z desky odebírat v rozumné proudové míře napětí 5 a 3,3 V, vhodné jako zdroje vstupně-výstupních signálů na portu nebo napájení malých zařízení s malým odběrem připojených k mikropočítači.

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. Výstupní port

Na obrázku je vidět rozložení výstupů na portu. Každý pin portu má přesný název a mezi každých 8 je vložen zdroj napětí 3,3V a zem pro práci se signály. Na portu PTF0-3 jsou připojeny 4 optické indikátory (LED 1-4) umístěné ve spodní části kitu vedle tlačítek SW1-4, které jsou na portu PTA4-7. Na PTD0 je umístěn bzučák (buzzer) a LCD modul na portech PTG3-7, PTE6-7. PTB0 je potenciometr (RV1). Celkově máme možnost využívat 56 vstupně/výstupních linek na 7 portech (A-G). Veškeré zbylé přesné informace jako je instrukční sada aj. se dají nalézt v manuálech volně stažitelných na internetu. [1]

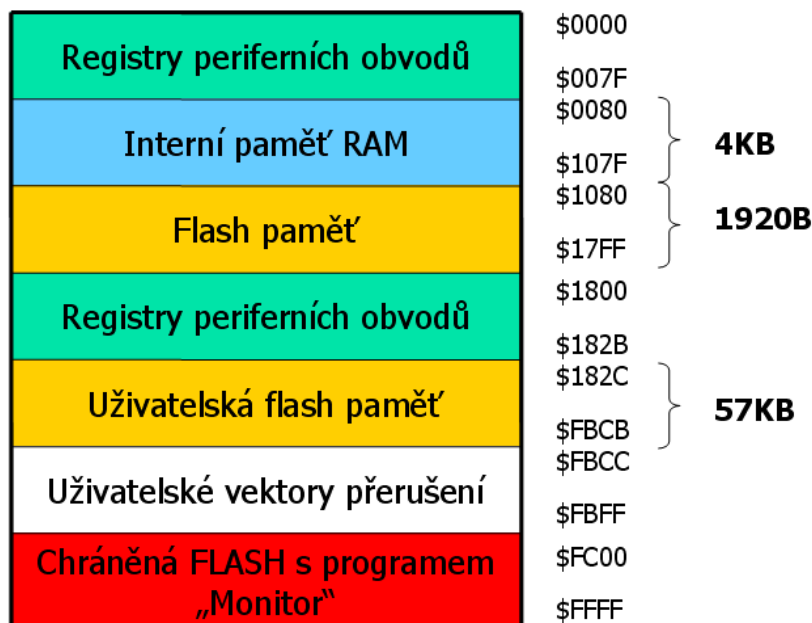
### LCD Character Codes

\$20	Space	\$2D	-	\$3A	:	\$47	G	\$54	T	\$61	A	\$6E	n	\$7B	{
\$21	!	\$2E	.	\$3B	;	\$48	H	\$55	U	\$62	B	\$6F	o	\$7C	
\$22	"	\$2F	/	\$3C	{	\$49	I	\$56	V	\$63	C	\$70	p	\$7D	}
\$23	#	\$30	0	\$3D	=	\$4A	J	\$57	W	\$64	D	\$71	q	\$7E	>
\$24	\$	\$31	1	\$3E	}	\$4B	K	\$58	X	\$65	E	\$72	r	\$7F	<
\$25	%	\$32	2	\$3F	?	\$4C	L	\$59	Y	\$66	F	\$73	s		
\$26	&	\$33	3	\$40	Time	\$4D	M	\$5A	Z	\$67	G	\$74	t		
\$27	'	\$34	4	\$41	A	\$4E	N	\$5B	[	\$68	H	\$75	u		
\$28	(	\$35	5	\$42	B	\$4F	O	\$5C	Yen	\$69	I	\$76	v		
\$29	)	\$36	6	\$43	C	\$50	P	\$5D	]	\$6A	J	\$77	w		
\$2A	*	\$37	7	\$44	D	\$51	Q	\$5E	^	\$6B	K	\$78	x		
\$2B	+	\$38	8	\$45	E	\$52	R	\$5F	~	\$6C	L	\$79	y		
\$2C	,	\$39	9	\$46	F	\$53	S	\$60	`	\$6D	M	\$7A	z		

Obrázek 6. Znakové možnosti displeje

### 1.2.3 Vývojový kit - základní vlastnosti

Základem je tedy 8 bitová centrální procesní jednotka HCS08 a její taktovací frekvence je max. 40MHz. Součástí je 60KB FLASH paměti a 4KB RAM paměti (podrobněji viz obr).



Obrázek 7. Paměťová mapa

Dále pak 5 kanálový TPM2 časovač, 3 kanálový TPM1 časovač, interní generátor hodinového kmitočtu s FLL obvodem (32KHz – 20MHz frekvence sběrnice) a 8 kanálový, 10 bitový A/D převodník.

Byl přidán COP watchdog systém s nastavitelnou časovou prodlevou na 218 nebo 213 cyklů sběrnice, který je nepřehlédnutelnou oporou při programování a postupném odlaďování. Pro větší bezpečnost především hardwaru slouží systém kontroly napájecího napětí umožňující detekovat pokles napětí pod stanovenou mez. Především při plně reálném užívání mimo zdroje síťového napětí je vítaným doplňkem podpora režimů se sníženou spotřebou umožňující nasazení v bateriově napájených aplikacích.

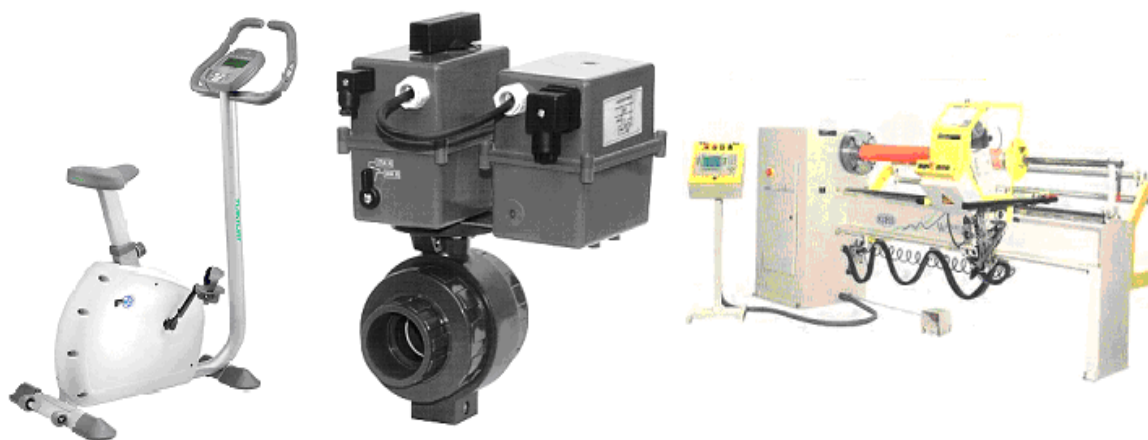


## 2 SERVOMOTOR

Nemalou roli v této práci zaujímá dnes již hojně rozšířená mechanicko-elektronická součástka nazvaná servomotor (servo). Je určena pro pohony, u kterých lze na rozdíl od běžného motoru nastavit přesnou polohu natočení osy.

### 2.1 Vlastnosti servomotorů

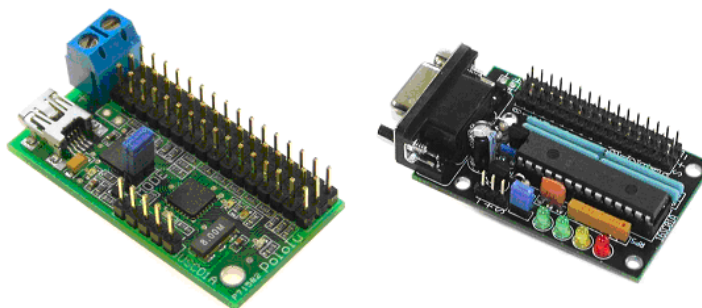
Jak bylo výše zmíněno, rozšíření servomotorů v dnešní době stále navyšuje hranice svého využití. Většinou se používají tak jako v tomto případě serva elektrická, avšak existují také hydraulická, pneumatická či dokonce parní. Ta nejmenší najdeme jako zařízení pro nastavení čtecí hlavičky u pevného disku, naopak v provedení větších rozměrů a s vyšším výkonem se jimi dají například přesně určovat posuvy u CNC strojů. Denně se s nimi setkáváme mnohdy několikrát. Najdeme je jako řízení pro přesnou a plynulou magnetickou brzdu na moderních rotopedech nebo také jako kulový kohout na potrubích, kterými se rozvádí voda.



*Obrázek 8. Příklady užití serva*

Elektrické servomotory jsou řízeny prakticky výhradně tranzistorovými měniči s pulzní šířkovou modulací. Námí používané servo nachází častého a oblíbeného využití v modelářství u složitějších především rádiově řízených modelů. V úloze zaměřené na používání čítače a časovače integrovaného v mikroprocesoru Motorola nám poskytuje vynikající výstup pro vizualizaci výsledku práce s impulsy. Způsob řízení se liší od obyčejných a krokových elektromotorků, jelikož vstupním signálem je měnicí se tvar průběhu nízkého střídavého napětí, který v praxi většinou zajišťuje mikrokontrolér

s přesnou definicí užití (viz obr). Samotné řízení není výkonově příliš náročné a aplikace už je pouhou otázkou naší fantazie.



Obrázek 9. Mikrokontroléry pro řízení serva

Aby bylo možné si ze serv vybírat, je třeba si ujasnit, k čemu bude využíváno. Na obrázku je přehled parametrů, které je třeba uvážit při výběrání pro předem známý účel.

parametr	význam	Poznámky
torque	kroucí moment	Často bývají udávány v kg/cm (uncích na palec).
speed	rychlost	
precision	přesnost	Přesnost nastavení polohy.
bearing	ložiska	
	převody	
size	rozměry	Často bývají udávány v palcích. Můžete si je převést na jednotky metrické.
weight	hmotnost	Často bývají udávány v uncích. Můžete si je převést na jednotky metrické.
angle	úhel otáčení	Běžně 45° na každou stranu. Existují specializovaná serva s větším úhlem otáčení: 180° , 1260° (3 a 1/2 celé otočky), aj.
	životnost	
applications	doporučené použití	
	rozsah napájení	zpravidla 4.8 - 6 V
	odběr v klidu	
	odběr v chodu	
	šířka impulsu	Udává se šířka řídicího impulsu pro polohu serva ve středu. Standard je 1.5 ms. Multiplex používá 1.6 ms.
		digitální servo

Obrázek 10. Parametry servomotoru

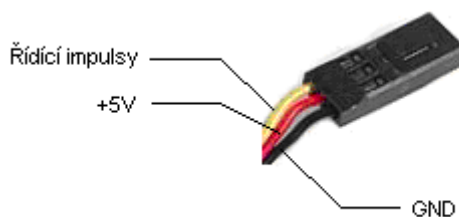
## 2.2 Princip serva

V tomto případě naprosto postačí zabývat se tedy servomotorem z řad těch nejmenších a to modelářským. Pro názorné vysvětlení principu řízení a funkce jsem si vybral levné servo Hitec (viz obr.), které se dá pořídit za cenu někde kolem 300,- Kč.



Obrázek 11. Modelářský servomotorek

Jak poměry součástek na obrázku napovídají, je jeho velikost jen 40x20x36mm a hmotnost pouhých 45g. Přesto dokáže vyvinout rychlost pohybu až 0,16s na 60 stupních maximální dráhy pohybu a kroučící moment 3kg na páce 1cm. Přívodní kabel pro napájení takového serva bývá přibližně 25cm dlouhý a na konci mívá konektor, jenž se dá nastrčit na špičky určené pro jumpery (typ S1G20, S1G36, S1G40). Kabel bývá převážně 3 žilový, přičemž jednotlivé žíly mají různé barvy, kterými jsou odlišovány podle svého významu.



Obrázek 12. Zapojení napájecího konektoru

Černý vodič – GND (0V) a červený vodič +5V ( $\pm 0,2V$ ) jsou určeny pro přívod napájení serva, zbývající žlutý vodič slouží pro vstup řídicích impulsů. Toto připojení bývá zpravidla totožné u servomotorků takovýchto parametrů i u jiných výrobců. Těm silnějším, s většími rozměry a hmotností se říká „jumbo“, naopak ty menší, miniaturní, se nazývají „mikroserva“. Jejich velikost se srovnatelná s kovovou mincí a hmotnost někde kolem 10g. Čím více se však liší od běžných typů, jsou cenově několikanásobně dražší.

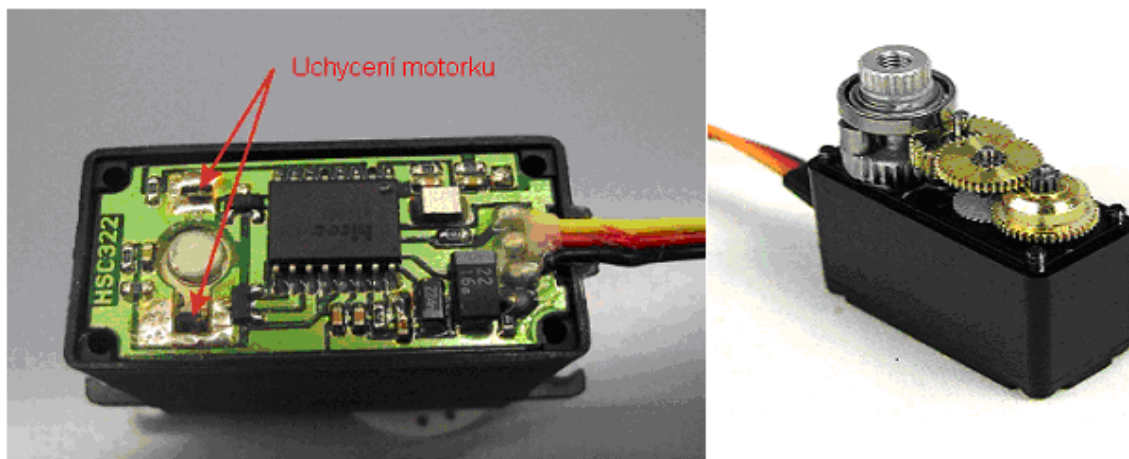
### 2.3 Vnitřní stavba serva

Servo nepatří mezi příliš složité součástky, a na následujícím obrázku je znázorněna jeho vnitřní stavba.



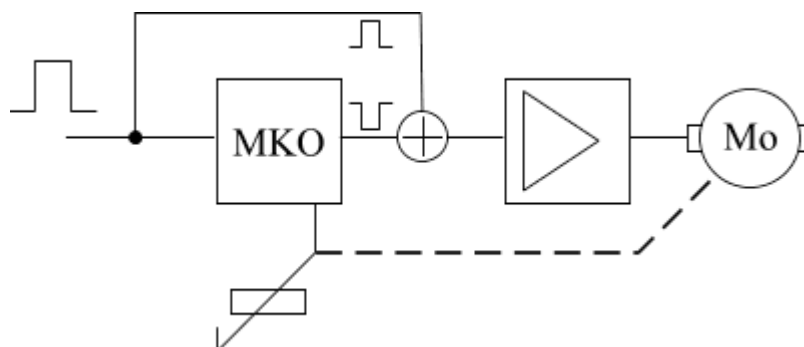
Obrázek 13. Vnitřní stavba serva

Hlavní úlohu má motorek a převodovka, která bývá plastová, nebo v lepším případě kovová. Ty nejvyšší kategorie jsou vyrobeny z titanové slitiny. Přebodovka ovlivňuje důležitý parametr, kterým je výstupní krouticí moment. Její kvalita musí být úměrná výkonu motorku vybranému podle požadovaných výstupních hodnot zatížení. Motorek serva je připojován k plnému napájecímu napětí na krátký časový okamžik s opakovací periodou. Délka tohoto sepnutí je proměnná a je určena požadovanými otáčkami motorku.



Obrázek 14. Řídící elektronika a převodovka servomotoru

Další důležitou částí je řídicí elektronika. Na vstup přichází řídicí impuls například v mém případě vyslaný z mikropočítače. Tento impuls spustí monostabilní klopný obvod, který vygeneruje impuls nový, odpovídající aktuální poloze serva, a navíc opačné polarity proti impulsu vstupnímu.



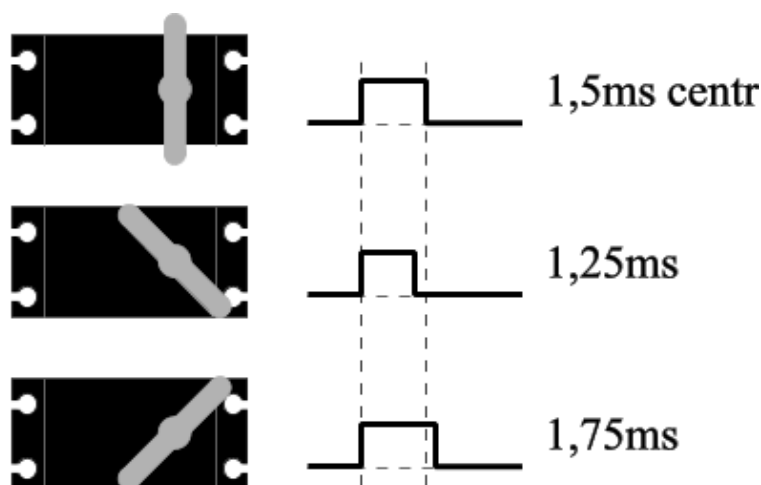
Obrázek 15. Vnitřní stavba serva blokově

Následovně jsou navzájem porovnány a výsledný rozdílový impuls po zesílení přes můstkový spínač způsobí roztočení elektromotorku. Pohyb je přes převodovku přenášen na výstupní hřídel a současně potenciometr, který posílá do monostabilního klopného obvodu zpětnou vazbu. Směr otáčení vyplývá ze snahy přiblížit délku impulsu generovaného monostabilním klopným obvodem délce impulsu vstupního (řídicího). V době, kdy jsou si oba rovny, se elektromotor zastaví, neboť je dosaženo požadované polohy.

## 2.4 Řízení serva

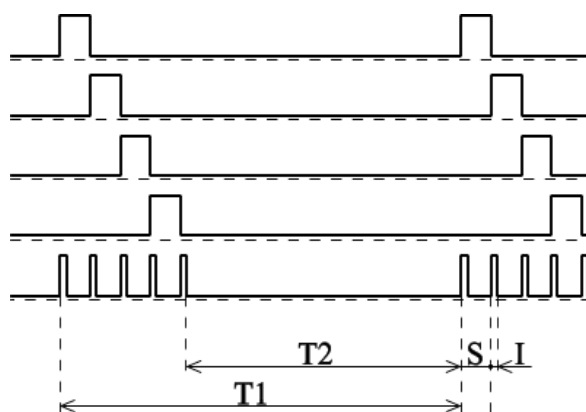
### 2.4.1 Řídící impulsy

Dnes používaná serva pracují s kladnými řídicími impulsy o délce 1-2ms. Délce impulsu 1,5ms odpovídá střední poloha serva, 1ms je jedna a 2ms druhá krajní poloha. Mechanické provedení může být takové, že servo je schopno pohybu v rozsahu o něco větším než 180 stupňů, ale není to pravidlem. Většina jich má na koncích rozsahu pohybu mechanické blokování, na což je potřeba dávat pozor, protože při dojetí na doraz se výrazně zvýší odebíraný proud a může to skončit tím, že shoří jeho řídicí elektronika.



Obrázek 16. Závislost polohy na řídicím impulsu

Řídící impulsy se opakují většinou s frekvencí kolem 50Hz, ale pokud budou řídicí impulsy přicházet méně často, pouze se zpomalí rychlost pohybu serva. Velikost impulsů je rovná velikosti napájecího napětí serva, což je 4,8-5,2V. První čtyři průběhy na níže uvedeném obrázku ukazují standardní řídicí impulsy na výstupu časovače. [12]



Obrázek 17. Charakteristika řídicích impulsů pro více motorků

Pátý průběh ukazuje, jak vypadá výstupní signál kodéru modelářského vysílače pro 4 serva. Podle typu kodéru zůstává konstantní buď T1 a skupiny impulsů se opakují v pravidelném rytmu (kolem 50Hz), nebo je konstantní mezera mezi skupinami impulsů T2 (většinou 10-15ms). S je délka řídicího impulsu pro jedno servo (1-2ms), I je oddělovací impuls (0,25-0,3ms)

#### 2.4.2 Rušivé vlivy

Nevýhodou u serva je princip regulace jeho elektromotoru, který nikdy neběží ve vysokých otáčkách, ale je vlastně nárazově roztáčen v krátkých časových intervalech. Proto protinapětí, které se v něm indukuje je nízké a proud procházející motorkem je tedy omezen hlavně jeho vnitřním odporem. Dosahuje tak stále rozběhových hodnot, neboť je napájen proudovými impulsy přibližně o velikosti proudu stojícího motorku. Změní-li se však smysl otáčení hlavní hřídele za chodu, je na motorek skokově přivedeno napětí opačné polarity, které se pak sečte s protinapětím indukovaným motorkem a výsledkem je proudový skok  $I=(U_{nap}+U_{in})/R$ . Tato proudová špička bývá u rychle se pohybujícího serva až o polovinu větší, než proud stojícím motorkem. Běžnými ampérmetry proud protékajících servem nelze změřit, respektive naměřená hodnota neodpovídá skutečnosti. Lze jej změřit jako úbytek napětí na odporu 0,5 ohmů v záporné větvi napájecího napětí. Vnitřní odpor motorků bývá od 2 do 10 ohmů v závislosti na velikosti a požadované rychlosti. U mikroserv bývá vnitřní odpor 5-8 ohmů, což znamená popěrně vysoké impulsní proudy, pro představu je běžné když odebírá okolo 1A. Problém se dá vyřešit pomocí kondenzátoru s kapacitou 100  $\mu\text{F}$ , který je však zatím rozměrově příliš velký a do malé krabičky serva se nevejde. Pracujeme-li s například s více servy najednou, stává se, že v časové posloupnosti najednou odebírají proud dvě, tři najednou a výsledkem je přetížení stabilizátoru a výpadek napájení. Obrana proti tomuto rušení je možná dvěma způsoby:

- Do série s motorkem serva se vřadí malý odpor 5-8ohmů. Tím snížíme rozběhový proud motorku na polovinu.
- Do napájení serva vestavíme kondenzátor alespoň 50  $\mu\text{F}$  s malým ekvivalentním sériovým odporem

### 3 EAGLE

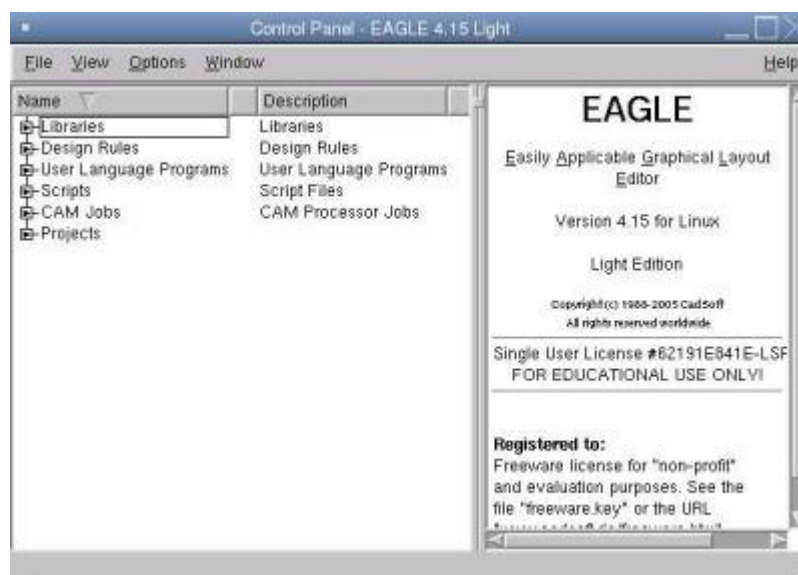
Při výrobě nepřiliš složitých elektrických modulů bývá častým pomocníkem program Eagle (Easily Applicable Graphical Layout Editor) pro návrh a editaci elektronických schémat a plošných spojů od Německé firmy CadSoft, která má pobočku také v USA. Skládá se ze tří základních částí.

#### 3.1 Hlavní části

**Editor spojů** - umožňuje navrhnout desku plošného spoje o velikosti až 1,6 x 1,6 m při rozlišení 1/10.000 mm (0,1 mikronu). Můžete použít klasické i SMD součástky, které jsou uloženy v knihovně součástek, která se standardně dodává s programem. Pokud se součástka nenachází v knihovně, lze ji vytvořit v editoru za použití až 16-ti signálových vrstev. Vynikajícím doplňkem je kontrola pravidel návrhu a vytvoření desky ze schématu jediným příkazem.

**Editor schémat** - schéma je možné navrhnout až na 99 listů. Desku ze schématu vytvoříte jediným jednoduchým příkazem.

**Autorouter** - modul pro automatické rozmístění součástek na desce a jejich propojení vodiči. Uživatel samozřejmě může nastavit pravidla propojování pomocí váhových vektorů.

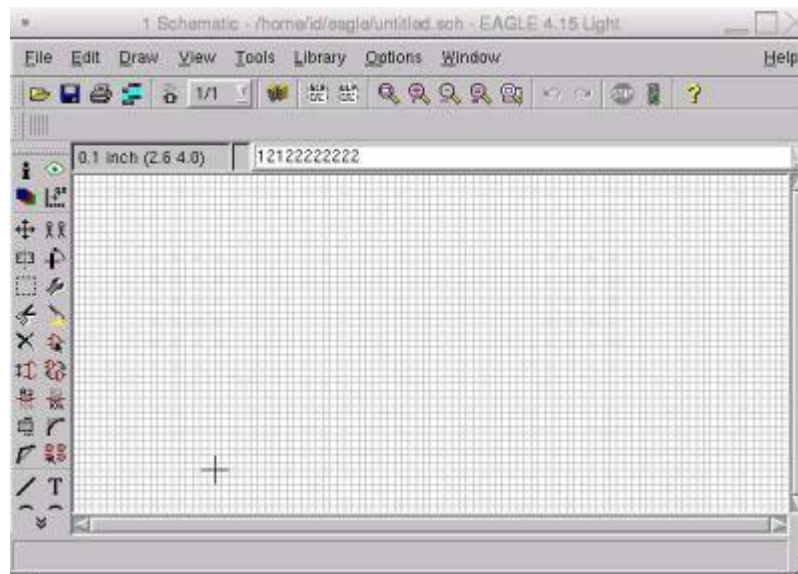


Obrázek 18. Kontrolní panel Eaglu

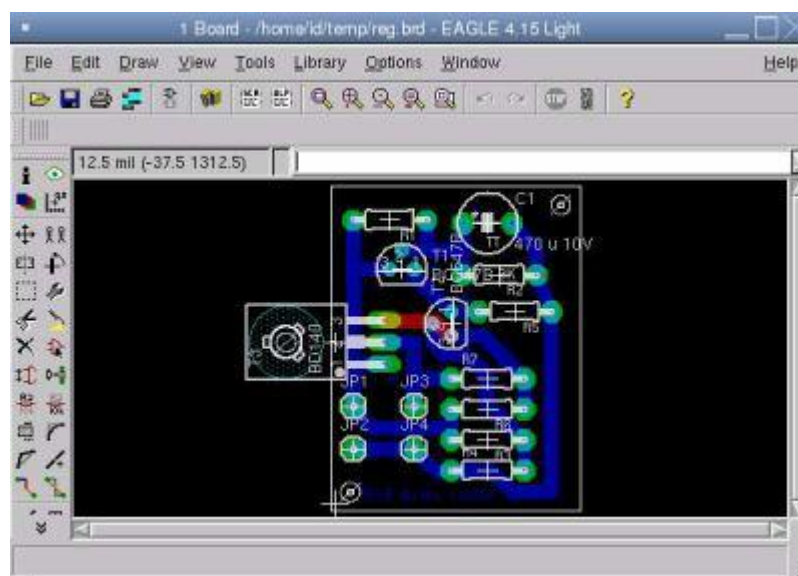


### 3.2 Omezení

Program se dá volně stáhnout pouze v Lite (omezené) verzi, která je určena pro nevýdělečné aplikace a studijní účely. Omezení spočívá v základních vlastnostech. Použitelná plocha desky je omezena na 100 x 80mm, mohou být použity jen dvě signálové vrstvy spojů (vrchní a spodní vrstva) a editor schémat může vytvořit schéma pouze na jednom listu. Zbytek lze používat v plném rozsahu, jako v plné verzi.



Obrázek 19. Editor schémat



Obrázek 20. Návrh desky plošných spojů

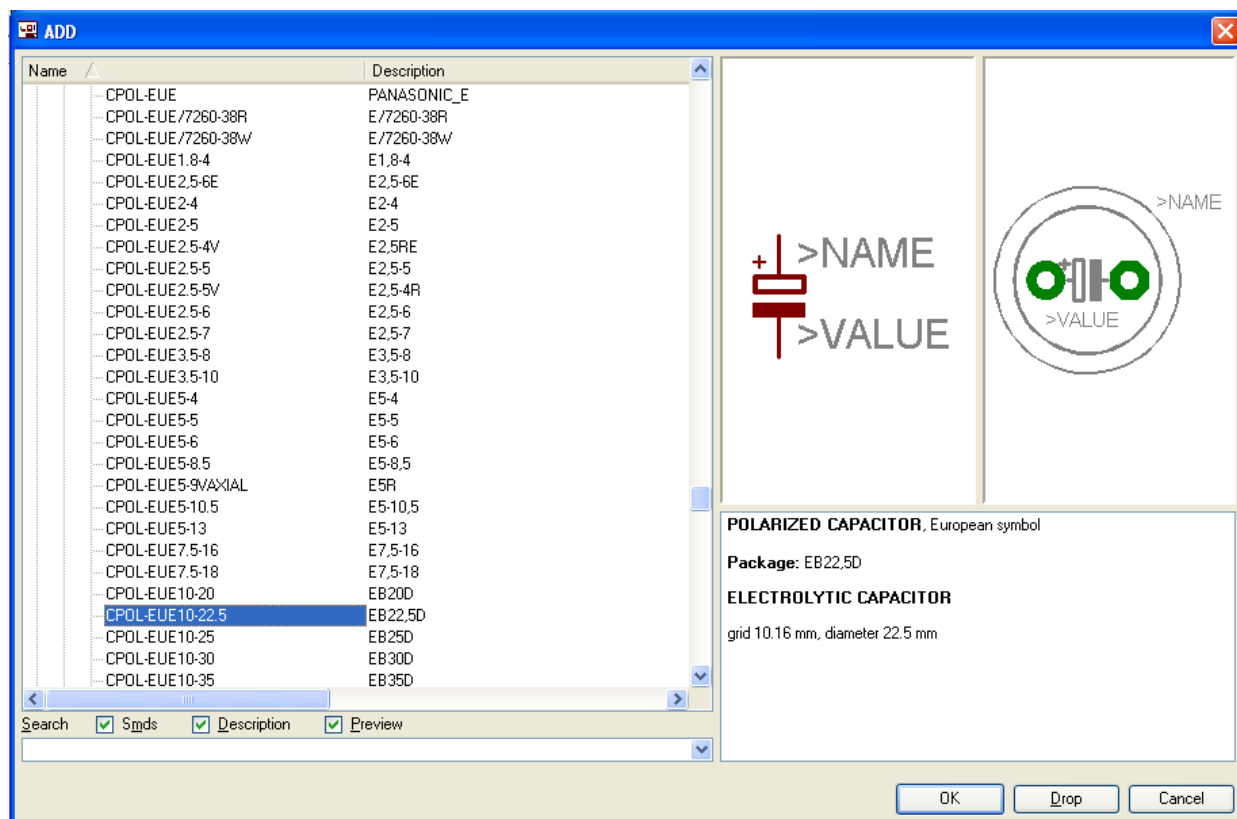
## **II. PRAKTICKÁ ČÁST**

## 4 POPIS HARDWARU PŘÍDAVNÉHO MODULU

Před výrobou přídatného modulu pro řízení serva mikro počítačem HCS08 je nejprve potřeba získat několik informací týkajících se jak vývojového kitu M68EVB908GB60 tak serva samotného. Je třeba znát základy programování v jazyku symbolických adres, kterým se programuje veškerá činnost mikro počítače, a mít alespoň malé zkušenosti s konstrukcí elektronických zařízení. Po splnění uvedených podmínek je ještě potřeba důkladně nastudovat vše o řídicím signálu a funkčnosti serva. Poté přichází na řadu návrhnutí elektrického zapojení.

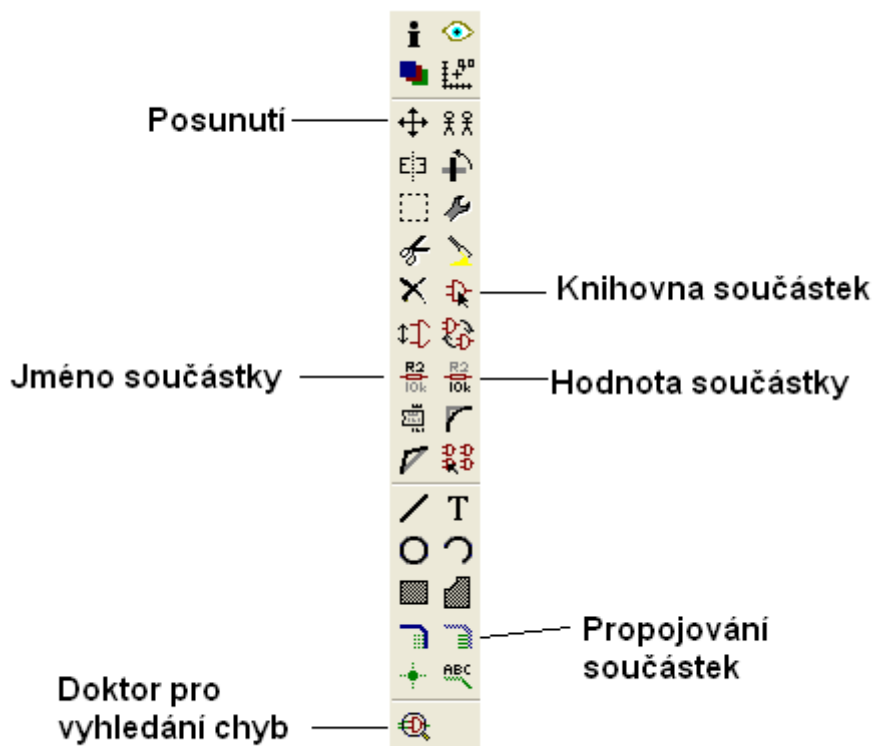
### 4.1 Navržením elektrického zapojení

V dnešní době máme velkou výhodu, že při navrhování elektrických schémat je k dispozici celá řada programů vytvořených přímo pro tuto činnost. Schéma bylo v tomto případě vytvářeno v programu Eagle. Veškeré součástky je potřeba vyhledat v knihovně která je součástí programu. Při výběru, pokud nenajdeme přesný tip, se řídíme podle vzdálenosti vývodů součástky, aby ji později bylo možno osadit do plošného spoje.



Obrázek 21. Knihovna součástek v Eaglu

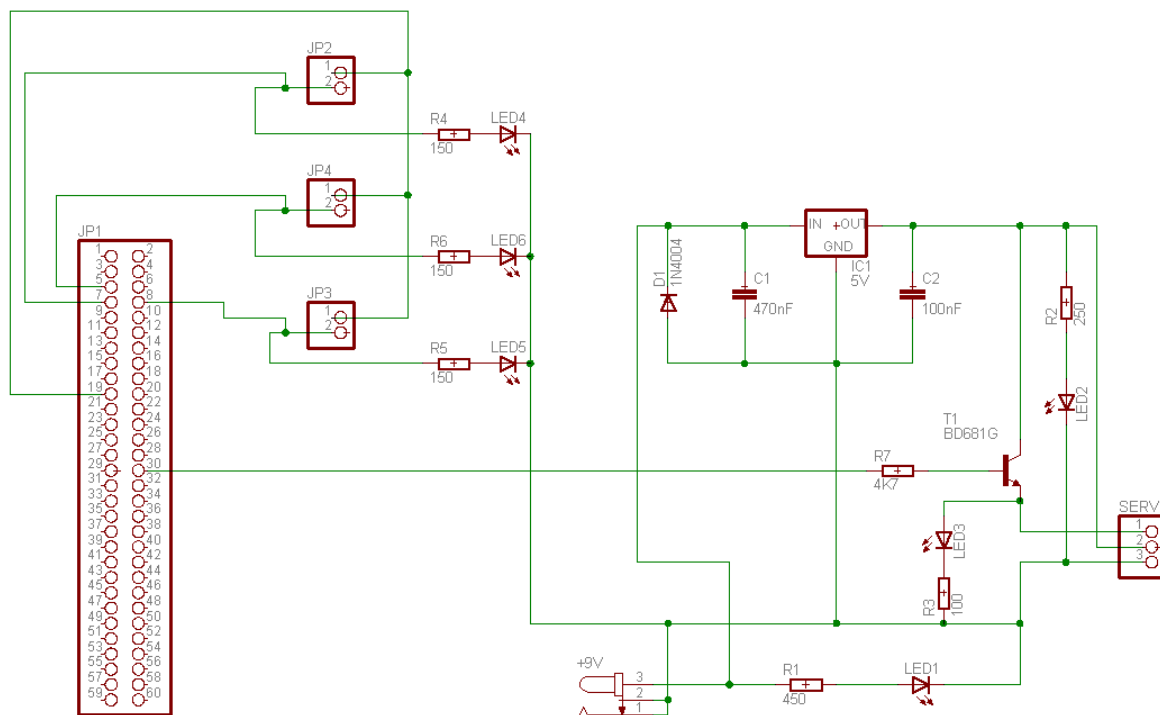
Po vložení je potřeba přidat název a hodnotu. Tímto způsobem si přidáme všechny potřebné součástky, které následovně propojíme.



Obrázek 22. Panel s nástroji pro tvorbu schémat

Víme, že servomotor je napájen napětím 5V a řízen střídavým signálem s amplitudou rovněž 5V. Abychom proudově nenamáhali mikropočítač, použijeme stabilizátor napětí L7805CV s výstupem 5V připojený přímo na napájecí napětí 9V. Na vstup přidáme diodu D1 chránící stabilizátor před opačnou nežádoucí polaritou vstupního napětí a kondenzátor C1-470nF jako odrušení. Na výstupu bude kondenzátor C2 o kapacitě 100nF pro vyladění. Dále do obvodu přidáme tranzistor BD675-D na jehož bázi přivedeme řídicí signál z mikropočítače přes odpor R7-1k7 a na kolektor přivedeme výstupní napětí ze stabilizátoru. Z emitoru pak vedeme zesílený signál přímo na vstup serva. LED1 diodu s odporem R1-450Ω připojíme přímo na vstupní napětí 9V pro kontrolu vstupního napětí, LED2-250Ω připojíme za stabilizátor a čímž signalizujeme výstupní napětí z něj. Třetí dioda LED3 s R-100Ω je umístěna paralelně na řídicí vstup serva a změna jasu signalizuje měnící se šířku impulsu. Pro připojení k mikropočítači je potřeba patice JP1 z jumperů o rozměrech 2x30 vývodů. Pro kontrolu povoleného rozsahu polohy jsou přidány tři tlačítka napájené napětím přímo z portu 3,3V, které při sepnutí přivedou napětí na PTA0-2, program bude tyto vstupní

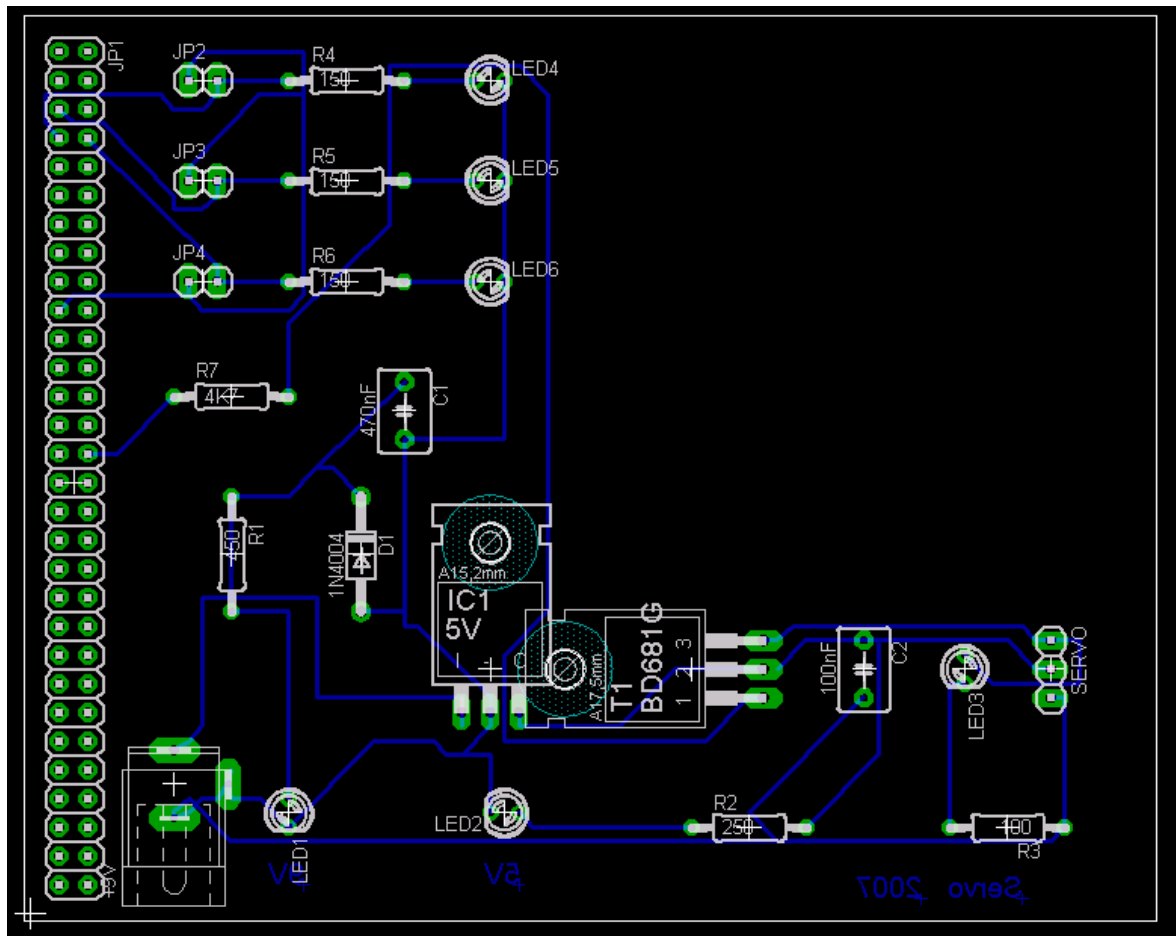
hodnoty vyhodnocovat bude tak mít pod kontrolou rozsah polohy a polohu ve středu dráhy pohybu. LED4-6 s odpory R4-6  $150\Omega$  signalizují krajní polohy. Zde na obrázku je vidět celé schéma.



Obrázek 23. Schéma zapojení modulu pro řízení serva

## 4.2 Deska plošných spojů

Vytváření desky plošných spojů je jednou z výhod Eaglu. Stačí pouhý jeden příkaz z nabídky Soubor a to **Switch to board** a ze schématických značek se automaticky vytvoří prostředí vývoje spojů s obrysy skutečných velikostí součástí spolu s naznačenými vývody. Vývody jsou však prozatím propojeny pouze naznačenými pohyblivými cestami. Přichází na řadu funkce Autorouter, která hledá varianty možných propojení na pevně. Ale je i nadále možné výsledné schéma upravovat ručně. Je vhodné nastavit ještě pouze jednu vrstvu plošného spoje, neboť implicitně vytváří desku ve dvou. Dají se nastavit ještě další parametry jako minimální vzdálenost dvou měděných cest.



Obrázek 24. Deska plošného spoje modulu pro řízení serva

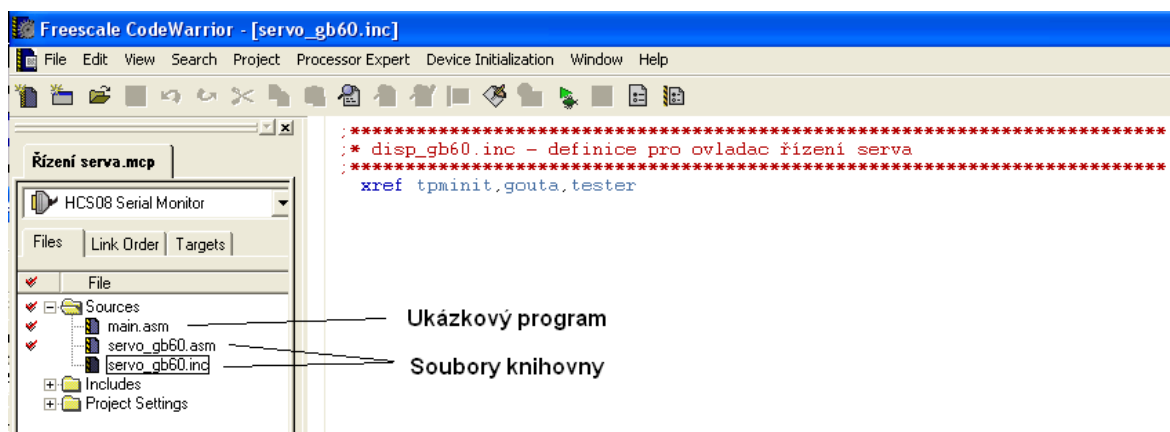
Na desce je vidět hlavní připojovací port, konektory pro napájení, konektory pro kontrolní spínače rozsahů polohy a další rozložení součástek. Modré cesty naznačují, kudy povedou měděné spoje. Zelené body, kde bude díra pro vývod a měď pro jeho následné připájení.

## 5 PROGRAMOVÁ PODPORA PŘÍDAVNÉHO MODULU

Softwarová část poskytuje programovou podporu pro řízení servomotoru mikropočítačem. Nejeftivnější je vytvoření knihovny, která uživateli nastaví a inicializuje časovač vhodným způsobem právě pro řízení serva a poskytne několik základních podprogramů pro práci s ním.

### 5.1 Knihovna pro modul řízení serva

Knihovna se skládá ze dvou základních souborů **servo\_gb60.asm** a **servo\_gb60.inc**. V souboru \*.inc jsou definované názvy podprogramů, které mají být přístupné pro program využívající knihovnu a naopak. V souboru \*.asm je pak kompletní kód knihovny s jednotlivými podprogramy. Na obrázku je náhled souboru \*.inc.



Obrázek 25. Ukázka souboru knihovny v prostředí CodeWarrior

Impulsy řídicí servomotor generuje časovač. Frekvenci je třeba nastavit předem propočítanými parametry. Zdrojový signál pro časovač mikropočítače má frekvenci 20Mhz. Servomotor je řízen impulsy o frekvenci 50Hz z toho vyplývá doba periody  $T=0,02s$ . Vynásobením zdrojové frekvence a doby periody se získá modulo, které v tomto případě by činilo číslo větší, než s kterým je mikropočítač schopen pracovat, je proto potřeba nosnou frekvenci snížit pomocí děliče (prescaler)  $20MHz / 16 = 1,25MHz$ , následně se vypočítá modulo  $1,25000000 * 0,02 = 25000$ . Na 25000 hodnot v registru modulo je tedy rozdělena jedna perioda časovače o délce 20ms, motorek je však řízen pouze v čase 1-2ms proto se do registru časovače budou vkládat pouze hodnoty od 1200 do 2500, což je 1300 hodnot. Na obrázku je hlavička souboru z knihovny s příponou \*.asm.

```

;*****
;* Ovladac pro řizeni serva pro vyvojovy kit M68EVB08GB60
;* UTB ve Zline, 2007
;* Implementovane podprogramy:
;*
;*      tpminit   - inicializace TPM1
;*      gouta     - nastavi servo na procentualni pozici drahy pohybu zadanou v reg.A (0-100)
;*      tester    - 10x projede a tim otestuje posuv od min do max vystupni polohy serva
;*
;*****
;
; PWM
; frekvence cca 50 Hz, duty od 0 do 100% nastavitelny potenciometrem na kitu
; Pro vypocet modulo podle vzorce:
; tof = 0,02 sec
; fsorce = 20 MHz
; Prescaler = 16;      (1,25 MHz)
; modulo = 25000
;
; prescaler=dělič => 20 Mhz / 16 = 1.25 Mhz
; 1250000 * 0.02 = 25000
;
; celkova T = 20 ms rizeni serva 1-2 ms (nastavujem modulo 1250-2500 = 1250 hondnot)
;
; Vystup PTD1
;
; Pro 0% duty zapsat do TMP1C1 VH a VL hodnotu 0, pro 100% hodnotu větší než modulo
;
; NA PTD1 pripojit ridici vstup serva popripade Led aby bylo vidět vysledek modulace = menici se poloha.
;
; XDEF      tpminit,gouta,tester          ; zprístupneni funkci ovladace okoli
;
; INCLUDE 'derivative.inc'
;
; variable/data section
MY_ZEROPAGE: SECTION SHORT          ; Insert here your data definition
duty DS.W 1
cyklu DS.W 1
pocetst DS.W 1

```

Obrázek 26. Hlavička souboru knihovny

Časovač je potřeba inicializovat. Bit 1 portu D jako nastavíme jako výstup. Při inicializaci se nastaví předem vypočítaný dělič a zdroj nosné frekvence. Uloží se také předem spočítaný obsah registru modulo pro frekvenci 50 Hz a přednastaví duty cycle na hodnotu 0% pomocí předem definované proměnné. Nakonec inicializace kanálu CH0 časovače TPM1.

```

;*****
; podprogram tpminit - inicializace TPM1
;*****
minit
MOV    #$02, PTDDD          ; Bit 1 portu D jako vystup. Pozn. bzucak je na bitu 0
MOV    #%00001100,TPM1SC    ; inicializace TPM1, fsorce=busclk, prescaler=16
LDHX   #25000               ; obsah modulo registru pro frekvenci 50 Hz
STHX   TPM1MOD              ; ulozime do modulo registru

LDHX   #0
STHX   duty                 ; prednastavime duty cycle na 0%

LDHX   duty
STHX   TPM1C1V              ; nastav pocatecni duty cycle
MOV    #%00101000, TPM1C1SC ; inicializace kanalu CH0 casovace TPM1,
; rezim PWM high true(pin 28 = PTD1)

RTS

```

Obrázek 27. Inicializace časovače



Podprogram gouta nastaví servo na procentuální pozici dráhy pohybu zadanou v reg.A (0-100). Princip spočívá v tom, že po spuštění podprogramu se nejprve porovná hodnota v akumulátoru A, nepřesahuje-li stanovené hranice ( $0 > x < 100$ ). Je-li hodnota mimo stanovený rozsah, nastaví se automaticky přesně doprostřed dráhy pohybu. Tím je ošetřeno, aby se servo nedostalo do mezní krajní polohy. Odpovídá-li číslo stanovenému rozsahu, přesune se pomocí zásobníků z akumulátoru A do akumulátoru X. Do A se nahraje nová hodnota 13, kterou číslo v X vynásobíme. Tím rozšíříme rozsah čísel nastavitelných potenciometrem z 0-100 na 0-1300. To jsou hodnoty modulo přenesené na periodu v oblasti 0-1ms, pro řízení serva je však potřeba hodnot v oblasti 1-2ms. Proto se po násobení pomocí zásobníků přesune  $(H:X) \leftarrow (X:H)$  následuje smyčka **pricti**, která snadnou inkrementací v akumulátoru (H:X) přičítá po stovce hodnoty v 12 cyklech, až je dosaženo čísla 1200. Tím se posouvá spodní hranice rozsahu potenciometru násobená 13 a zvětšená 1200 na možný rozsah 1200-2500, cílený výsledek. Nyní se potenciometrem nastavuje hodnota v čase periody 1-2ms. Hodnoty z akumulátoru (X:A) se ukládají do registru časovače. Celý kód podprogramu je uveden nyní na obrázku.

```

;*****
;*
;* podprogram gouta - nastavi servo na procentualni pozici drahy pohybu zadanou v reg.A (0-100)
;*
;*****

gouta      PSHH
           PSHX

           CMP #0
           BLO chyba
           CMP #100
           BHI chyba
           BRA pokračuj

chyba      lda #50
           BRA pokračuj

pokracuj   PSHA
           PULX

           LDA #13
           MUL

           PSHX
           PSHA
           PULX
           PULH

           PSHA
           MOV #12,cyklad

pricti     AIX #100

           LDA cyklad
           DECA
           STA cyklad
           CMP #0
           BNE pricti

           PULA

           PSHH
           PSHX
           PULA
           PULX

           STX TPM1C1VH ; presun z XA do registru timeru = tim se zmeni
           STA TPM1C1VL ; aktualni duty cycle (dalka pulzu)

           JSR cekmal

           PULX
           PULH

           RTS

```

Obrázek 28. Podprogram pro nastavení polohy serva

Posledním podprogramem který knihovna obsahuje je program testovací. Nese název tester a automaticky pošle na výstup 10x změnu výstupní polohy z 0 na 100%. Servo se po té s použitím podprogramu gouta 10x pootočí z jedné krajní polohy do druhé a zpět. Funkce slouží pro testování funkčnosti obvodu a serva. Jeho kód je opět na následujícím obrázku.

Zbylé dva podprogramy knihovny nejsou přístupné. Slouží pouze jako zpoždovací smyčky při provádění výše uvedených podprogramů gouta a tester.

```

;*****
;*
;* podprogram tester - 10x projede a tím otestuje posuv od min do max vystupni polohy serva
;*
;*****
tester      PSHA
            PSHX
            PSHH

            MOV     #10,poctst

testuj      LDA     #0
            JSR     gouta
            JSR     cekmoc

            LDA     #100
            JSR     gouta
            JSR     cekmoc

            LDA     poctst
            DECA
            STA     poctst
            CMP     #0
            BHI     testuj

            PULH
            PULX
            PULA

            RTS

```

Obrázek 29. Podprogram pro testování

## 5.2 Ukázkový program řízení serva

### 5.2.1 Hlavička programu

Ukázkový program začíná tak jako každý standardní program, navíc je zde pouze nově vložen INCLUDE nově vytvořené knihovny 'servo\_gb60.inc', který zavede soubory knihovny do programu. Definuje se pin PTB0 jako vstup převodníku a provede se jeho inicializace. Na konci obrázku je vidět volání prvního podprogramu z nově vytvořené knihovny **JSR tpmnit**, který inicializuje časovač TPM1.

```

;*****
;* This stationery serves as the framework for a user application. *
;* For a more comprehensive program that demonstrates the more *
;* advanced functionality of this processor, please see the *
;* demonstration applications, located in the examples *
;* subdirectory of the "Freescale CodeWarrior for HC08" program *
;* directory. *
;*****
;~~~~~
;Ukazkovy program uziti knihovny 'servo_gb60.inc'
;~~~~~
; export symbols
XDEF _Startup, main
; we export both '_Startup' and 'main' as symbols. Either can
; be referenced in the linker .prm file or from C/C++ later on

; Device Initialization function declaration
XREF MCU_init

XREF __SEG_END_SSTACK ; symbol defined by the linker for the end of the stack

; Include derivative-specific definitions
INCLUDE 'derivative.inc'
INCLUDE 'servo_gb60.inc'

; variable/data section
MY_ZEROPAGE: SECTION SHORT ; Insert here your data definition

; code section
MyCode: SECTION
main:
_Startup:
LDHX #__SEG_END_SSTACK ; initialize the stack pointer
TXS

; Uncomment this function call after using Device Initialization
; to use the generated code
;JSR MCU_init

MOV #00000001,ATD1PE ; pin PTB0 prepnut do rezimu vstup AD převodniku
MOV #11100100,ATD1C ; inicializace převodniku
; (AD zap, 8bit, zarovnaní vpravo, ATD clock 2MHz)

JSR tpmnit

```

Obrázek 30. Hlavička ukázkového programu

## 5.2.2 Tělo programu

Jako první zavoláme funkci **tester**, která sama automaticky pošle na výstup 10x změnu výstupní polohy z 0 na 100%. Servo se 10x nastaví z jedné krajní polohy a zpět. Funkce slouží pro testování funkčnosti obvodu. Poté je spuštěn převod na 8 bitovém převodníku a smyčka čekající na jeho dokončení. Výsledek převodu je násoben 20ti a ihned vydělen 51, čímž získáme z rozsahu převodníku 0-255 hodnoty 0-100, které jsou podmínkou vstupního parametru níže volaného podprogramu **gouta**, jenž nastavuje výstupní polohu serva podle parametru v akumulátoru A 0-100%. Otáčením potenciometru je ovládán pohyb po plném rozsahu serva.

```
                jsr     tester

opakuj         mov     #00000000,ATD1SC      ; spusteni prevodu - preruseni zakazano,
nav1          lda     ATD1SC                ; cekej na dokonceni prevodu
                and     #010000000
                beq     nav1

                ldx    ATD1RH              ; nacteme do reg. H:X vysledek prevodu v rozsahu 0-255
                lda    #20
                mul    ; X:A ← (X) * (A)
                PSHX  ; H:A ← X:A
                PULH
                ldx    #51                ; A ← (H:A) / (X)
                div

dale          jsr     gouta

                feed_watchdog
                BRA    opakuj
```

Obrázek 31. Tělo ukázkového programu pro řízení serva

## ZÁVĚR

Cílem bakalářské práce bylo vytvořit modul pro připojení k vývojovému kitu M68EVB908GB60, který bude sloužit při výuce programování mikropočítače Motorola HCS08. Modul umožňuje řízení polohy modelářského servomotoru pomocí výstupních portů s pojistkou proti překročení povoleného rozsahu pohybu a indikací případné chyby. Součástí práce je návrh a realizace hardwaru modulu i základního softwarového vybavení pro jeho ovládání v jazyce symbolických adres.

Bakalářská práce nejprve vysvětluje obecně základní pojmy mikropočítačů, přibližuje jejich vlastnosti a vnitřní stavbu. V první řadě přibližuje jejich architekturu a způsoby provádění instrukcí, následně uvádí příklady jednotlivých aplikací.

V praktické části byla vytvořena knihovna obsahující základní inicializační a nastavovací podprogramy pro řízení servomotoru navrženým a postupně sestrojeným modulem.

Snahou při vytváření bylo dosáhnout takové kvality, aby vynaložené úsilí a všechny strávený čas zúročily přínosy, které získá čtenář ze všech v této práci obsažených textů nebo alespoň jejich částí.

## ZÁVĚR V ANGLIČTINĚ

Aim of this bachelor thesis was to create module for interface with developing kit M68EVB908GB60, which will support education of programming microcontroller Motorola HCS08. Module allows controlling position of modeler servomotor by the aid of output ports with safety fuse against over-range allowed range movement and indication of eventual mistakes. In this work the design and realization of the hardware of the module was performed and also essential software equipment for its control in assembly language was created.

The thesis first explains basic terms of microcontrollers, introduces their properties and internal operation. In the first part the architecture of microcontroller is explained and also the principle of its work. Then examples of applications are given.

In the practical part a program library was created which contains initialization and position-setting routines for controlling the servomotor. The purpose of this work was to offer students easy way of learning microcontroller programming.

## SEZNAM POUŽITÉ LITERATURY

### Monografie

- [1] Freescale Semiconductor. *MC9S08GB/GT Data Sheet*. [s.l.] : [s.n.], 2004.
- [2] Freescale Semiconductor. *HCS08 Family Reference Manual*. [s.l.] : [s.n.], 2003.
- [3] LADMAN, J. *Elektronické konstrukce pro začátečníky*. [s.l.] : [s.n.], 2002.
- [4] Motorola. *CPU08 Central Processor Unit Reference manual*. [s.l.] : [s.n.], 2001.
- [5] MACHO, Tomáš. *Mikroprocesory*. [s.l.] : [s.n.], 2004. 38 s.
- [6] REICH, Jakub. *Archivační teploměr s využitím mikropočítače řady Motorola HC08*. [s.l.] : [s.n.], 2006. 68 s.
- [7] ŠEBESTA, Jiří. *Mikroprocesorová technika*. [s.l.] : [s.n.], 2002. 34 s.
- [8] VAŠEK, V., VAŠEK, L. *Programování mikropočítačů*. [s.l.] : [s.n.], 1989.
- [9] VÁŇA, V. *Začínáme s mikrokontroléry Motorola HC08 Nitron*. [s.l.] : [s.n.], 2003.

### Internetové zdroje

- [10] MATEMATICKO-FYZIKÁLNÍ FAKULTA. *Elektronika* [online]. 2002 [cit. 2007-05-23]. Dostupný z WWW: <<http://lucy.troja.mff.cuni.cz/~tichy/>>.
- [11] SWEB. *Úprava a využití serva HS-322 jako motoru* [online]. 2007 [cit. 2007-05-19]. Dostupný z WWW: <[http://www.sweb.cz/e78/clanky/uprava\\_serva\\_cla.htm](http://www.sweb.cz/e78/clanky/uprava_serva_cla.htm)>.
- [12] Vlastíkd. *Jak fungují modelářská serva* [online]. 2006 [cit. 2007-05-18]. Dostupný z WWW: <<http://vlastikd.webz.cz/bastl/serva.htm>>.
- [13] Wikipedia. *Servo motor* [online]. 2007 [cit. 2007-05-20]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Servo>>.



**SEZNAM OBRÁZKŮ**

<i>Obrázek 1. Mikroprocesor průmyslového počítače .....</i>	10
<i>Obrázek 2. Obecné blokové schéma mikropočítače .....</i>	11
<i>Obrázek 3. Architektury mikroprocesorů .....</i>	12
<i>Obrázek 4. Hlavní části M68EVB908GB60 .....</i>	14
<i>Obrázek 5. Výstupní port .....</i>	15
<i>Obrázek 6. Znakové možnosti displeje .....</i>	15
<i>Obrázek 7. Paměťová mapa .....</i>	16
<i>Obrázek 8. Příklady užití serva .....</i>	17
<i>Obrázek 9. Mikrokontroléry pro řízení serva .....</i>	18
<i>Obrázek 10. Parametry servomotoru .....</i>	18
<i>Obrázek 11. Modelářský servomotorek .....</i>	19
<i>Obrázek 12. Zapojení napájecího konektoru .....</i>	19
<i>Obrázek 13. Vnitřní stavba serva .....</i>	20
<i>Obrázek 14. Řídící elektronika a převodovka servomotoru .....</i>	21
<i>Obrázek 15. Vnitřní stavba serva blokově .....</i>	21
<i>Obrázek 16. Závislost polohy na řídicím impulsu .....</i>	22
<i>Obrázek 17. Charakteristika řídicích impulsů pro více motorků .....</i>	22
<i>Obrázek 18. Kontrolní panel Eaglu .....</i>	24
<i>Obrázek 19. Editor schémat .....</i>	25
<i>Obrázek 20. Návrh desky plošných spojů .....</i>	25
<i>Obrázek 21. Knihovna součástek v Eaglu .....</i>	27
<i>Obrázek 22. Panel s nástroji pro tvorbu schémat .....</i>	28
<i>Obrázek 23. Schéma zapojení modulu pro řízení serva .....</i>	29
<i>Obrázek 24. Deska plošného spoje modulu pro řízení serva .....</i>	30
<i>Obrázek 25. Ukázka souboru knihovny v prostředí CodeWarrior .....</i>	31
<i>Obrázek 26. Hlavička souboru knihovny .....</i>	32
<i>Obrázek 27. Inicializace časovače .....</i>	32
<i>Obrázek 28. Podprogram pro nastavení polohy serva .....</i>	34
<i>Obrázek 29. Podprogram pro testování .....</i>	35
<i>Obrázek 30. Hlavička ukázkového programu .....</i>	36
<i>Obrázek 31. Tělo ukázkového programu pro řízení serva .....</i>	37

