

Výukový modul pro předmět programování mikropočítačů: informační LED displej

Educational module for microcontrollers programming course:
information LED display

Jiří Pobořil



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří POBOŘIL**
Osobní číslo: **A09677**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Výukový modul pro předmět Programování
mikropočítačů: informační LED displej**

Zásady pro vypracování:

1. Prostudujte hardwarové vlastnosti vývojového kitu M68EVB908GB60 a navrhnete způsob připojení modulu informačního LED displeje.
2. Provedte hardwarový návrh modulu informačního LED displeje na bázi maticových displejů s organizací 5x7 bodů.
3. Realizujte prototyp modulu a ověřte jeho funkci na vývojovém kitu.
4. Vytvořte podpůrné programové vybavení pro obsluhu modulu ve formě knihovny podprogramů v jazyce symbolických adres a v C jazyce.
5. Vytvořte ukázkovou aplikaci s využitím realizované knihovny podprogramů, která bude demonstrovat funkci modulu.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. FREESCALE SEMICONDUCTOR. CPU08 Central Processor Unit Reference Manual [online]. 2001. Dostupné z: www.freescale.com
2. FREESCALE SEMICONDUCTOR. HCS08 Family Reference Manual, Rev.1. [online]. 2003. Dostupné z: www.freescale.com
3. FREESCALE SEMICONDUCTOR. MC9S08GB/GT Data Sheet, Rev.2.3. [online]. 2004. Dostupné z: www.freescale.com
4. JURÁNEK, Antonín a Miroslav HRABOVSKÝ. EAGLE pro začátečníky /: uživatelská a referenční příručka :. 2. vydání. Praha: BEN – technická literatura, 2007, 191 s. ISBN 80-730-0213-2.
5. MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy – linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Vyd. 1. Praha: BEN, 2003, 279 s. ISBN 80-730-0077-6.

Vedoucí bakalářské práce:

Ing. Petr Dostálek, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

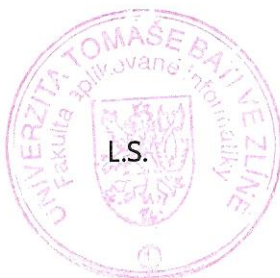
24. února 2012

Termín odevzdání bakalářské práce:

8. června 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Cílem této bakalářské práce je vytvořit přídatný modul k vývojovému kitu M68EVB908GB60. Tímto modulem je informační LED displej, který bude používán pro podporu cvičení z předmětu „Programování mikropočítačů“.

Práce zahrnuje návrh a vytvoření zmenšeného modelu, tvořeného pomocí matice LED diod. Tento panel umožňuje plynulé rolování textu. Mimo jiné bylo nutné vymyslet vhodné propojení modulu s vývojovým kitem. Knihovny byly vytvořeny pomocí programovacího jazyka C a jazyka symbolických adres. Ovladač obsahuje funkce, které je možno používat pro vykreslení jednoho bodu, jednoho znaku, statického textu nebo rolovacího textu na displeji.

Klíčová slova: Freescale, vývojový kit, HCS08, mikropočítač, maticový displej.

ABSTRACT

The main goal of this bachelor thesis is to create additional module for development kit M68EVB908GB60. This module is informational LED display that will be used to support seminar called „Microcomputers programming“.

The thesis contains design and creation of the miniaturized display module built by LED diodes matrix. The display enables smooth text scrolling. Among other things, it was necessary to invent appropriate interconnection between the module and the development kit. Libraries were created using the programming language C and assembly language. Driver contains functions that can be used for drawing of one point, one symbol or text string which can be static or it can scroll in selected direction.

Keywords: Freescale, development kit, HCS08, microcomputer, matrix display

Rád bych poděkoval vedoucímu mé práce, panu Ing. Petru Dostálkovi za odbornou pomoc, cenné rady a připomínky, které byly velice prospěšné při tvorbě této práce. Dále bych chtěl poděkovat kamarádům, jež mi svými zkušenostmi pomohli zorientovat se v dané problematice.

Motto:

„Ničeho se v životě nemusíme bát – jen to pochopit!“

Marie Curie-Sklodovská

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.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST	10
1 JEDNOČIPOVÉ MIKROPOČÍTAČE	11
1.1 HISTORIE MIKROPROCESORŮ	11
1.2 ZÁKLADNÍ POJMY	11
1.3 ARCHITEKTURA	12
1.3.1 Von Neumannova architektura.....	13
1.3.2 Harvardská architektura	14
2 VÝVOJOVÝ KIT FREESCALE M68EVB908GB60	15
2.1 ZÁKLADNÍ VLASTNOSTI VÝVOJOVÉHO KITU	15
2.2 ZÁKLADNÍ VLASTNOSTI MIKROPOČÍTAČE MC9S08GB60.....	16
3 SÉRIOVÉ PERIFERNÍ ROZHRANÍ SPI.....	18
3.1 KLADY A ZÁPORY SPI	18
3.2 PRINCIP KOMUNIKACE PO SBĚRNICI SPI.....	19
3.3 IMPLEMENTACE SPI NA MIKROPOČÍTAČI MC9S08GB60.....	19
3.4 PINY A REGISTRY SPI	21
3.4.1 Ovládací piny SPI.....	21
3.4.2 Ovládací registry SPI.....	21
Registr přenosové rychlosti (SPI1BR):	22
SPI stavový register (SPI1S):	23
SPI datový register (SPI1D):	24
4 CODEWARRIOR	25
4.1 VYTVOŘENÍ NOVÉHO PROJEKTU	25
4.2 HLAVNÍ OKNO PROGRAMU	27
4.3 PŘEKLAD A SPUŠTĚNÍ PROGRAMU	28
5 EAGLE	29
5.1 VLASTNOSTI PROGRAMU	29
5.2 EDITOR SPOJŮ	29
5.3 EDITOR SCHÉMAT	29
5.4 AUTOROUTER	30
6 ZOBRAZOVACÍ JEDNOTKY	31
6.1 PARAMETRY ZOBRAZOVACÍCH JEDNOTEK	31
6.2 ROZDĚLENÍ PODLE MOŽNOSTI ZOBRAZOVÁNÍ	31
6.2.1 Numerické displeje.....	32
6.2.2 Alfnumerické displeje	32
6.2.3 Grafické displeje	33

6.3	ROZDĚLENÍ PODLE TECHNOLOGIE ZOBRAZOVÁNÍ	33
6.3.1	Zobrazovací jednotky na bázi žárovek	33
6.3.2	Zobrazovací jednotky na bázi LED	33
6.3.3	Zobrazovací jednotky na bázi LCD	34
6.3.4	Obrazovkové zobrazovací jednotky	35
6.3.5	Plazmové zobrazovací jednotky	35
II	PRAKTICKÁ ČÁST	36
7	NÁVRH INFORMAČNÍHO LED DISPLEJE	37
7.1	POUŽITÉ SOUČÁSTKY NA MODULU	37
	Maticový displej:	37
	Sériový registr 74HC595 ve funkci budiče sloupců:	38
	Dekodér 74HC138 ve funkci budiče řádků a tranzistor IRF7104:	38
7.2	BOKOVÉ SCHÉMA	39
7.3	NÁVRH SCHÉMATU A DESKY PLOŠNÝCH SPOJŮ	40
7.4	ZOBRAZOVACÍ MODUL	40
7.4.1	Návrh schématu zapojení	40
7.4.2	Návrh desky plošných spojů	42
7.5	NAPÁJECÍ MODUL	43
7.5.1	Návrh schématu zapojení	43
7.5.2	Návrh desky plošných spojů	45
8	PROGRAMOVÁ ČÁST	47
8.1	KNIHOVNA FUNKCÍ	47
8.1.1	LEDInit	48
8.1.2	Funkce Fill	48
8.1.3	Funkce Clear	48
8.1.4	Timer_int	48
8.1.5	Funkce SetBit	49
8.1.6	Funkce ZobrazZnak	49
8.1.7	Funkce ZobrazText	50
8.1.8	Funkce RotujText	50
8.2	UKÁZKOVÝ PROGRAM DEMONSTRUJÍCÍ FUNKCI MODULU	51
	ZÁVĚR	53
	ZÁVĚR V ANGLIČTINĚ	54
	SEZNAM POUŽITÉ LITERATURY	54
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	56
	SEZNAM OBRÁZKŮ	58
	SEZNAM TABULEK	60
	SEZNAM PŘÍLOH	61

ÚVOD

V dnešní době se s různými displeji setkáváme na každém kroku. Bez této vymoženosti si většina z nás nedovede dnešní svět představit. Různé typy zobrazovacích jednotek jsou použity ve spotřebičích, jako jsou mobilní telefony, televize, počítačové monitory a v mnoha dalších zařízeních.

Proto byla vytvořena práce na téma informační LED displej. Tato práce má ukázat studentům, jak se modul propojuje s mikropočítačem, a jak se pomocí vytvořených knihoven ovládá. Vše je realizováno na vývojovém kitu M68EVB908GB60 od firmy Freescale, na němž je umístěn mikropočítač MC9S08GB60. Modul je možno ovládat ve dvou programovacích jazycích, které se v předmětu „Programování mikropočítačů“ vyučují. Prvním je jazyk symbolických adres. Tento jazyk je blízký zápisu strojového kódu. Druhým programovacím jazykem je C. Jeho výhodou je jednoduchá přenositelnost zdrojových kódů programu neboli multiplatformnost. Pro uživatele je jednodušší pochopit ho a dále s ním pracovat.

V této práci byly v první části nejprve popsány jednočipové mikropočítače obecně. Dále byly vylíčeny základní vlastnosti vývojového kitu od firmy Freescale. Jelikož je používáno pro komunikaci mezi mikroprocesorem a vytvořeným modulem sériového periferního rozhraní, jsou popsány jeho piny a registry. Následuje popis vývojového prostředí CodeWarrior, v němž byly vytvořeny knihovny. Tvorba schématu a DPS byla provedena programem EAGLE, pomocí kterého je možno vytvořit i 3D vizualizaci. Závěrečným bodem je popis a rozdělení zobrazovacích jednotek.

V praktické části můžeme nalézt funkci použitých součástek na výukovém informačním modulu. Dále bylo vytvořeno blokové schéma, z něhož se vycházelo při tvorbě schématu. Následuje rozdělení a popis schémat programem EAGLE. Z těchto schémat byly vytvořeny DPS, jejichž rozložení součástek je také pečlivě popsáno. Posledním bodem je programová část, ve které je popsána knihovna funkcí. Jednotlivé funkce jsou podrobně rozebrány a ke každé funkci je přiřazena i vzorová ukázka použití.

I. TEORETICKÁ ČÁST

1 JEDNOČIPOVÉ MIKROPOČÍTAČE

1.1 Historie mikroprocesorů

V počátku vývoje mikroprocesorů byly dva technologické směry. Technologie bipolární a technologie unipolární. V dnešní době se mikroprocesory vyrábí technologií unipolární. Rychlý rozvoj unipolárních technologií od PMOS po CMOS umožnil zvýšit rychlost a hustotu prvků na čipu. Mezi další důležité vlastnosti CMOS patří odolnost proti šumu a nízká spotřeba ve statickém stavu.

Vývoj prvního mikroprocesoru se datuje od roku 1971, kdy byl vynalezen firmou Intel mikroprocesor 4004. Tento mikroprocesor byl součástí kalkulaček Busicom. Šířka datové sběrnice tohoto mikroprocesoru byla 4 bity. I4004 byl dodán v pouzdře CERDIP se 16 vývody. Obsahoval 2300 tranzistorů. Zatímco v dnešní době mají mikroprocesory stovky miliónů tranzistorů.

Roku 1972 firma Intel přišla s typem 8008, který byl osmibitový a vyžadoval mnoho podpůrných obvodů. Jeho hlavním nedostatkem byla omezená hloubka zásobníku a neschopnost uložit všechny registry při příchodu přerušení. Proto byl roku 1974 představen typ 8080, byl osmibitový, s adresovatelným paměťovým prostorem 64 kB. Tento mikroprocesor se stal součástí prvního osobního počítače Altair. Dále následují osmibitové mikroprocesory Motorola 6800, Intel 8085, Zilog Z80.

V roce 1978 byl uveden na trh první 16bitový mikroprocesor, typ 8086 vyráběný firmou Intel NMOS technologií. Vývoj byl velice rychlý, a proto jsou uvedeny jen ty nejzákladnější mikroprocesory. Následovaly 286, 386, 486, různé verze Pentium až po dnešní AMD Phenomy či Intel Core i7 [1] [8].

1.2 Základní pojmy

Mikroprocesor – je to centrální jednotka (CPU), která je základní částí mikropočítače. Mikroprocesor sám o sobě nedokáže pracovat. Obsahuje aritmeticko – logickou jednotku (ALU), pracovní registry vyrovnávací paměti a řídicí obvody - řadič.

Mikropočítač – vznikne doplněním mikroprocesoru o další podpůrné obvody. Především o paměť a obvody zajišťující komunikaci s okolím – vstupní/výstupní obvody. Činnost mikropočítače je řízena programem a je schopen pracovat samostatně.

Řadič – obvod řídící chod mikroprocesoru.

Port – fyzické zařízení mikropočítače umožňující komunikaci s vnějšími jednotkami.

Zásobník – vymezený úsek paměti RAM pro dočasné ukládání dat.

Instrukce – jednoduchá akce, jež umí provést procesor.

Program – posloupnost instrukcí příkazujících procesoru, co má dělat.

Podprogram – část programu, kterou je možno opakovaně volat z různých míst kódu.

Programování – zápis programu v jakémkoliv programovacím jazyce.

Algoritmus – je daný postup, pomocí něhož lze vyřešit danou úlohu.

Vývojový diagram – grafické znázornění algoritmu.

Programovací jazyk – prostředek pro zápis algoritmů.

Strojový kód – posloupnost instrukcí procesoru adres a dat vyjádřená čísly.

Assembler – jazyk symbolických adres, blízký strojovému kódu [1].

1.3 Architektura

Existuje a existovalo mnoho představ o tom, jak by počítače měly vypadat a fungovat. Jednu z nich utvořil všestranný vědec a matematik John von Neumann.

Existují dvě základní architektury mikroprocesorů:

- Von Neumannova architektura
- Harvardská architektura

Každá z těchto architektur má svoje výhody i nevýhody.

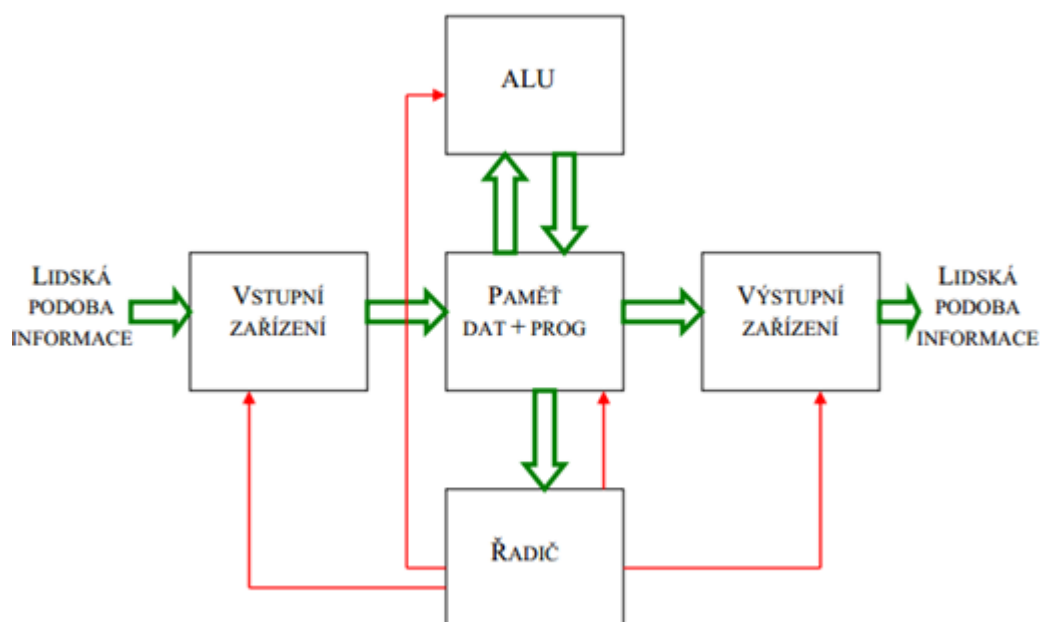
Architekturu mikropočítačů lze rozdělit na následující bloky:

- Operační paměť
- Řadič
- Aritmeticko-logická jednotka
- Vstupní a výstupní jednotka
- Blok registrů

1.3.1 Von Neumannova architektura

Charakteristickým rysem této architektury je popis počítače se společnou pamětí pro instrukce i data. To můžeme vidět na obrázku (obr. 1). Omezení této architektury spočívá v rychlosti zpracování instrukcí dnešními procesory, která je výrazně vyšší než rychlost komunikace s pamětí. Komunikace s pamětí je proto velkou nevýhodou Von Neumannovy architektury. V dnešní době se částečně nevýhoda komunikace s pamětí eliminujeme pamětí cache. Jsou to rychlé mezipaměti. Potřebná data z pomalejší hlavní paměti jsou načítána dříve, než jsou při zpracování potřeba.[9]

Vstupní zařízení se stará o změnu lidské informace na podobu binární informace. Tu ukládá do paměti. Data přesouvána z paměti jsou posléze zpracována pomocí ALU. Výsledky jsou ukládány zpátky do paměti. Podobu počítačové informace mění na podobu srozumitelnou pro člověka výstupní zařízení. Paměť uchovává data a program. Podle instrukcí řadiče je posílá k dalšímu zpracování. Řadič čte instrukce programu z paměti, ukládá je do svých registrů a generuje posloupnost řídících elektrických impulsů, které zajistí vykonání činnosti. Ta je předepsaná instrukcí programu [2].

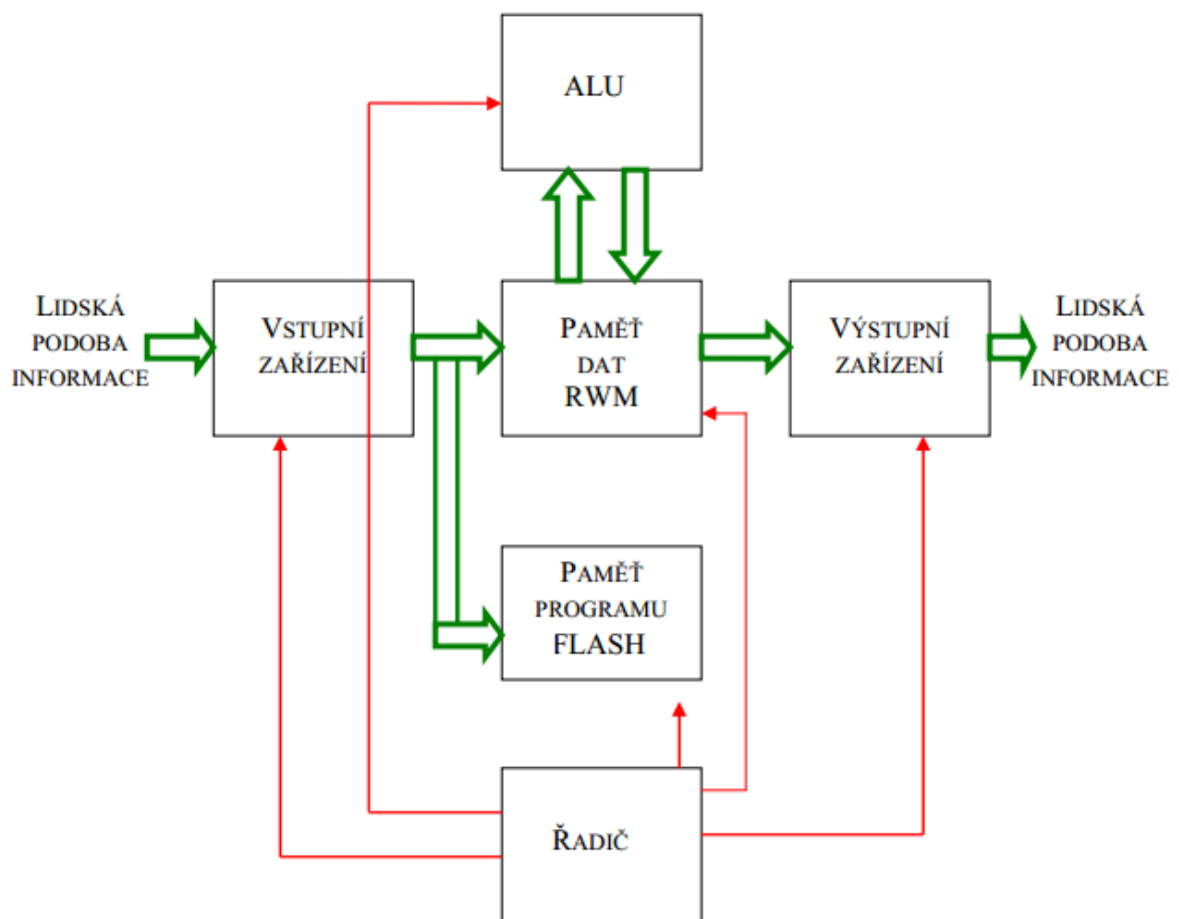


Obr. 1 John Von Neumannova architektura

1.3.2 Harvardská architektura

V této architektuře je oddělena paměť dat od paměti programu (obr. 2). V paměti programu je často umístěn statický program, který může být vyměněn pouze přeprogramováním. Zatímco data jsou uchovávány v pamětech typu RAM. Paměti nemusí mít shodnou délku slova ani způsob adresace.

Harvardská koncepce se uplatňuje tam, kde uživatel dlouhodobě pracuje ve svém zařízení s jedním programem. Tento program je trvale uložen v paměti programu Flash [2].



Obr. 2 Harvardská architektura

2 VÝVOJOVÝ KIT FREESCALE M68EVB908GB60

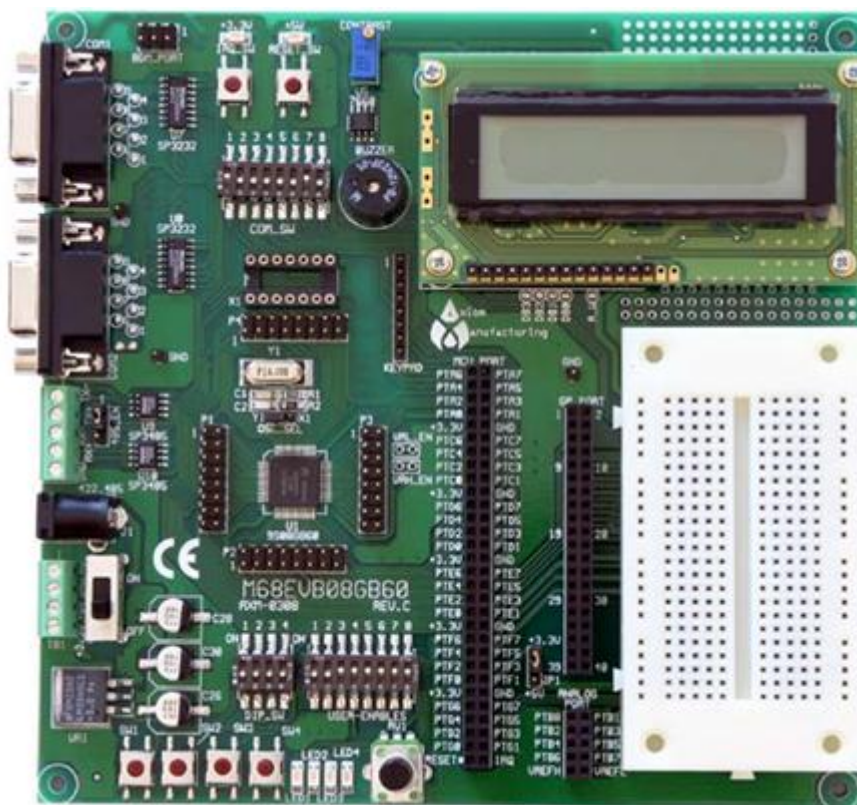
Vývojový kit M68EVB908GB60 slouží pro vývoj aplikací, určených pro mikroprocesory Freescale. Vývoj aplikací se provádí pomocí vývojového prostředí CodeWarrior.

2.1 Základní vlastnosti vývojového kitu

Rozložení součástek vývojového kitu je možné vidět na obrázku (obr. 3). Rozměry kitu jsou 15,24 x 15,75 cm. Na vývojovém kitu, používaném ve škole, můžeme nalézt mikroprocesor řady HCS08 MC9S08GB60. Propojení s počítačem se provádí přes sériový port RS232. Komunikace mezi jednotlivými periferiemi vývojového kitu probíhá přes vyvedené porty mikropočítače. Interní oscilátor je 32 kHz a externí 4 MHz. Dále se na kitu nachází RS422/485 konektor, konektor pro napájení 9 V, vypínač a signalizace napájení, tlačítko IRQ, tlačítko reset, kontaktní nepájivé pole o rozměrech 5,08 x 12,70 cm, MCU port konektor s digitálními vstupy a výstupy. Jednotlivé periferie používají stabilizované napětí 3,3V [3].

Jednotlivé uživatelské periferie:

- 4x vratné tlačítko (PTA4-PTA7)
- 4x LED dioda (PTF0 – PTF3)
- 4x DIP přepínač (PTB4-7)
- Potenciometr (PTB0/AD0)
- Bzučák
- LCD displej 2x16 znaků



Obr. 3 Vývojový kit M68EVB908GB60

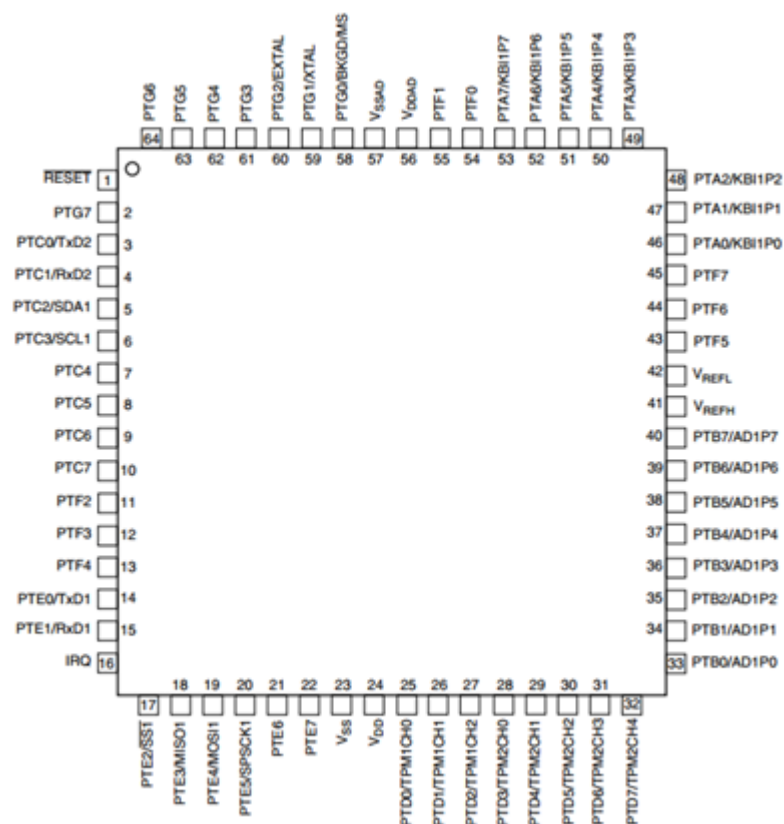
2.2 Základní vlastnosti mikropočítače MC9S08GB60

Je to osmibitový počítač vyroben firmou Freescale. Patří do řady HCS08, která vychází ze starší řady HC05. Struktura je založena na Von Neumannovy koncepci, protože data i program jsou umístěny ve stejném paměťovém prostoru. Z historického hlediska je struktura podobná jako u mikroprocesoru Motorola 6800. Tvar a rozložení jednotlivých pinů můžeme vidět na obrázku (obr. 4).

Základní vlastnosti:

- Osmibitová centrální procesní jednotka HCS08, taktovací frekvence max. 40MHz
- 60 KB Flash paměti
- 4 KB RAM paměti
- 56 I/O linek (64 pinů) na 7 portech (Port A-G)
- 5 kanálový TPM2 časovač

- Kanálový TPM1 časovač
- 8 kanálový, 10 bitový A/D převodník
- Synchronní sériové periferní rozhraní SPI a I2C rozhraní
- 2x Asynchronní sériové periferní rozhraní SCI
- BDM rozhraní pro pokročilé ladění
- Generátor hodin w/PLL [3] [5]



Obr. 4 Význam pinů mikropočítače MC9S08GB60

3 SÉRIOVÉ PERIFERNÍ ROZHRANÍ SPI

Sériové periferní rozhraní se používá pro komunikaci mezi řídicími mikroprocesory a ostatními integrovanými obvody, jako jsou např. displeje nebo A/D převodníky. Jejich implementace je snadná, jak po hardwarové, tak i po softwarové stránce. Sériové periferní rozhraní vzájemně propojuje dva nebo více komunikujících uzlů. Jeden z těchto uzlů obvykle bývá nastavován do režimu master. Ostatní uzly pracují v režimu slave. Režim master obsahuje generátor hodinového signálu, který je rozveden do všech ostatních uzlů. Z tohoto důvodu je umožněn synchronní přenos dat. Uzly jsou propojeny nejen hodinovým signálem, ale také dvojicí vodičů označených většinou symboly MISO (Master In, Slave Out) a MOSI (Master Out, Slave In), pomocí nichž jsou data přenášeny obousměrně. Poslední signál, používaný u SPI, je signál SSEL (Slave Select). Tento signál slouží k výběru některého z uzlů pracujících v režimu slave. Všechny čtyři výše popsané signály vyžadují pouze jednosměrné porty. Proto je implementace této sběrnice levná [4].

3.1 Klady a zápory SPI

Klady sběrnice SPI:

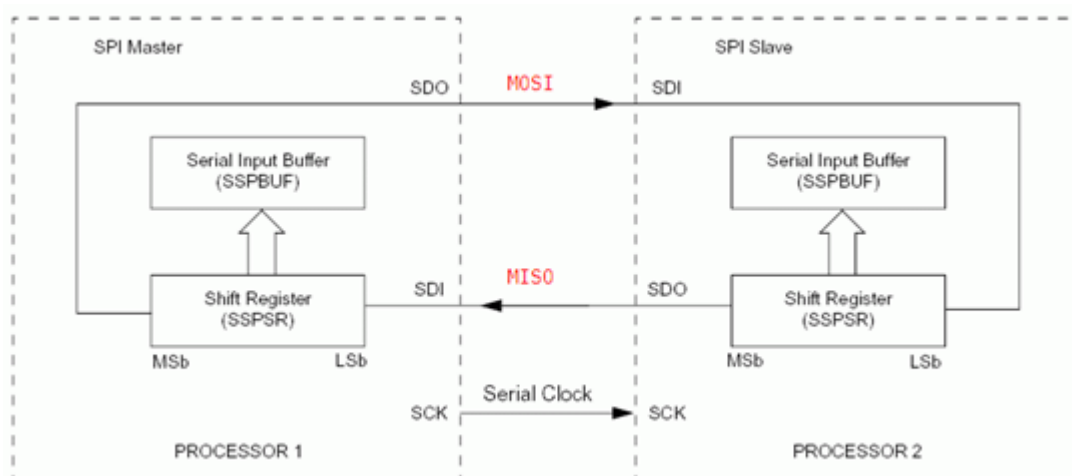
- Jednoduchost, snadná implementace
- Obousměrné přenášení dat po samostatných vodičích
- Dostatečně vysoká přenosová rychlost
- Hodinová frekvence 1MHz až 10 MHz, ale i 70MHz

Zápory sběrnice SPI:

- Pouze jedno zařízení v režimu master
- Přenášení dat na kratší vzdálenosti
- Lze použít několik způsobů synchronizace dat
- Nutnost použít minimálně čtyři vodiče pro obousměrnou komunikaci

3.2 Princip komunikace po sběrnici SPI

Propojení dvou zařízení pomocí SPI je popsáno na obrázku (obr. 5). Můžeme na něm vidět dva uzly, každý z těchto uzlů obsahuje dva registry. Prvním je datový registr, tím druhým posuvný registr. Do posuvného registru SSPSR je zapsán jeden bajt. Tento bajt byl přijat, ale nebyl ještě zpracován. Registr SSPSR hlídá, aby nedošlo ke ztrátě dat. Dále slouží posuvný registr jako vysílač a přijímač jednoho bitu. Posun obsahu registru doprava způsobí vysunutí bitu na pin SDO. Logická hodnota přečtená pinem SDI je zapsána do nejnižšího bitu posuvného registru. Každý z uzlů pracuje v jiném režimu. Jeden tedy jako master, který generuje hodinové impulsy a posílá je po vodiči SCK. Druhý pracuje v režimu slave. Polaritu hodin lze u většiny zařízení nakonfigurovat dvěma způsoby. Prvním způsobem je posun proveden při sestupné hraně. Při druhém se posun provede při vzestupné hraně hodinového signálu. Polaritu lze nastavit bitem CKP. K vysílání a příjmu dat dochází vždy po ustálení MISO i MOSI datových vodičů [4].



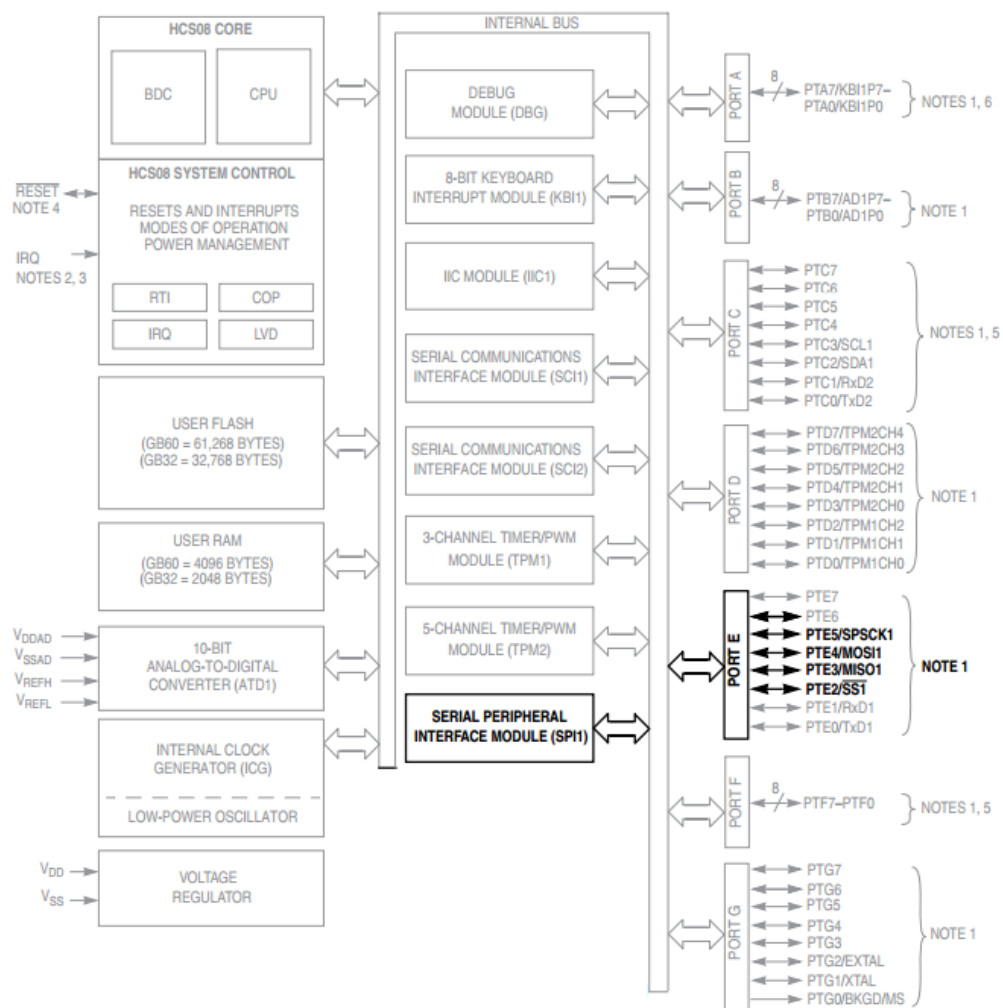
Obr. 5 Princip propojení dvou zařízení pomocí SPI

3.3 Implementace SPI na mikropočítači MC9S08GB60

Mikropočítač nabízí jeden modul SPI. Nachází se na portu E. Na pinech dva až pět, což můžeme vidět na obrázku (obr. 6).

SPI modul obsahuje:

- Master, Slave režim provozu
- Plně duplexní nebo jednovodičový obousměrný režim
- Programovatelná přenosová rychlost
- 2xbuffer pro vysílání a příjem
- Nastavení polarity a fáze u časovací frekvence
- Výběr slave výstupu
- Volitelný MSB - nejvýznamnější bit nebo LSB - nejméně významný bit [5]



Obr. 6 Blokové schéma mikropočítače MC9S08GB60 – zvýrazněné SPI

3.4 Piny a registry SPI

3.4.1 Ovládací piny SPI

Funkce těchto čtyř pinů závisí na nastavení SPI. Pokud je SPI vypnuto ($SPE = 0$), vrátí se všechny čtyři piny do režimu univerzální I/O. Ty pak nejsou řízeny SPI.

Popis řídicích pinů:

- **SPSCK1** – SPI časovací hodinový signál – pokud je povoleno SPI jako slave, tak hodinovou frekvenci přijímá, pokud je SPI nastaveno jako master, hodinovou frekvenci vysílá
- **MOSI1** – Master Data Out, Slave Data In – master – vysílání dat, slave – příjem dat
- **MISO1** – Master Data In, Slave Data Out – master – příjem dat, slave – vysílání dat
- **SS1** – Slave Select – přepíná mezi slave jednotkami [5]

3.4.2 Ovládací registry SPI

Řídicí registr 1 (SPI1C1):

Je to 8-bitový registr, který slouží pro nastavení komunikace s SPI.

	Bit 7	6	5	4	3	2	1	Bit 0
Čtení:	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Zápis:								
Reset:	0	0	0	0	0	1	0	0

Obr. 7 Řídicí registr SPI1C1


Popis jednotlivých bitů:

- **SPIE** — povolí přerušení (pro SPRF a MODF)
- **SPE** — povolí hardwarovou sběrnici SPI
- **SPTIE** — přerušení při vysílání
- **MSTR** — výběr módu Master/Slave

- CPOL — polarita hodinového signálu
- CPHA — fáze hodinového signálu
- SSOE — povolení výběr slave zařízení
- LSBFE — výběr pořadí vysílání bitů - nejméně významný bit (LSB) se vysílá jako první[5]

Řídící registr 2 (SPI1C2):

	Bit 7	6	5	4	3	2	1	Bit 0
Čtení:	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
Zápis:								
Reset:	0	0	0	0	0	0	0	0

 = Neimplementovaný nebo Rezervovaný


Obr. 8 Řídící registr SPI1C2

Popis jednotlivých bitů:

- MODFEN — detekce chyby v režimu Master
- BIDIROE — řízení směru komunikace v obousměrném režimu
- SPISWAI — zastavení hodinové frekvence ve WAIT režimu
- SPC0 — nastavení half-duplex nebo full-duplex módu [5]

Registr přenosové rychlosti (SPI1BR):

	Bit 7	6	5	4	3	2	1	Bit 0
Čtení:	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
Zápis:								
Reset:	0	0	0	0	0	0	0	0

 = Neimplementovaný nebo Rezervovaný

Obr. 9 Registr přenosové rychlosti SPI1BR

- Dělička frekvence pro SPI, přičemž jako referenční frekvence je použita frekvence hlavní sběrnice
- Dělička se nastavuje pomocí bitů SPPR2:SPPR1:SPPR0.

Tab. 1 Nastavení dělitele frekvence SPI

SPPR2:SPPR1:SPPR0	DĚLIČKA
0:0:0	1
0:0:1	2
0:1:0	3
0:1:1	4
1:0:0	5
1:0:1	6
1:1:0	7
1:1:1	8


- Dělička frekvence hlavní frekvence se nastaví pomocí bitů -SPR2:SPR1:SPR0. [5]

Tab. 2 Dělička frekvence

SPR2:SPR1:SPR0	DĚLIČKA
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

SPI stavový register (SPI1S):

	Bit 7	6	5	4	3	2	1	Bit 0
Čtení:	SPRF	0	SPTEF	MODF	0	0	0	0
Zápis:								
Reset:	0	0	1	0	0	0	0	0

 = Neimplementovaný nebo Rezervovaný

Obr. 10 Stavový registr SPI1S

Popis jednotlivých bitů:

- SPRF — příznak naplnění vstupního bufferu
- SPTEF — příznak vyprázdnění výstupního bufferu
- MODF — detekce kolize [5]

SPI datový register (SPI1D):

Je to paměť typu sériový registr. Přečtením tohoto registru získáme příchozí data z příchozího bufferu. Zápisem do toho registru jsou zapsána data pro přenos na sběrnici SPI. Pokud je zařízení v režimu master, tak přenos dat začne zapsáním do tohoto registru. Data nemůžou být zapisována, dokud nebudou přečtená příchozí data (nastaven SPTEF).

	Bit 7	6	5	4	3	2	1	Bit 0
Čtení:	Bit 7	6	5	4	3	2	1	Bit 0
Zápis:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Obr. 11 Datový registr SPI1D

4 CODEWARRIOR

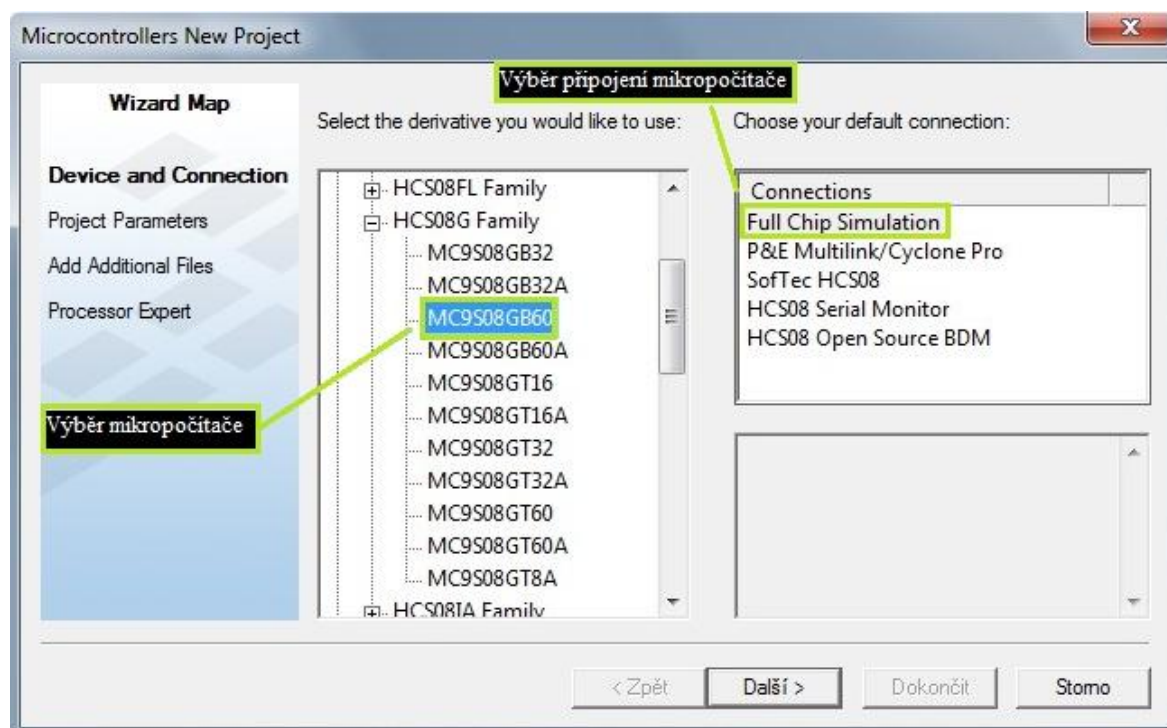
Vývojové prostředí CodeWarrior od firmy Freescale je používáno ve škole a to ve verzi 6.3. Vývojové prostředí komunikuje s použitým mikropočítačem MC9S08GB60. Programování tohoto mikropočítače je možné realizovat v jazyce symbolických adres neboli assembleru a vyšším programovacím jazyce C. Uživatel se musí rozhodnout, zda bude použito obou programovacích jazyků v jednom projektu. Pokud se nerozhodne, je možné použít mixovaného projektu, kde lze programovat oběma způsoby. Mimo jiné vývojové prostředí CodeWarrior obsahuje i debugger, pomocí něhož se po jednotlivých krocích zkoumá funkčnost programu.

4.1 Vytvoření nového projektu

Po spuštění programu vyskočí vstupní obrazovka, na které se nabízí tyto možnosti:

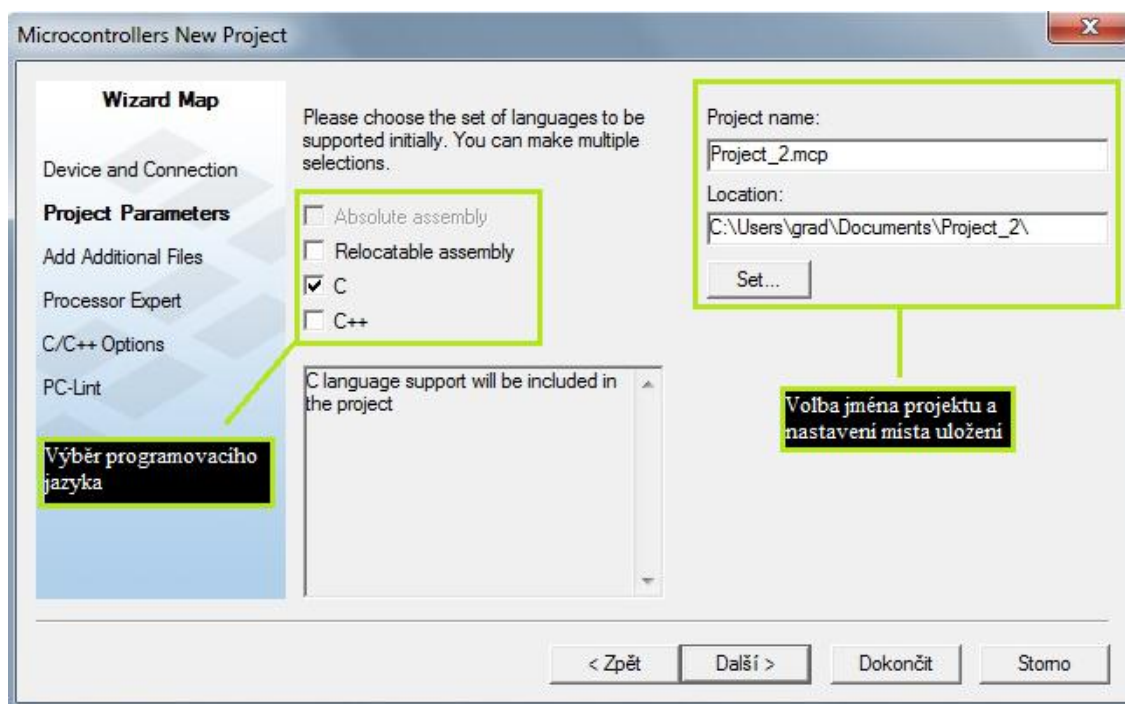
- Vytvořit nový projekt
- Načíst vytvořený projekt
- Načíst předchozí projekt
- Spustit tutoriál nebo začít používat
- Začít pracovat s programem CodeWarrior.

Po volbě vytvořit nový projekt se uživateli objeví obrazovka (obr. 12), kde lze vybrat používaný mikropočítač. Dále je možné zvolit režim simulace nebo komunikovat přímo s mikropočítačem.



Obr. 12 Vytváření projektu programem CodeWarrior – volba mikropočítače

Po stisknutí tlačítka „další“ dojde k zobrazení dalšího okna (obr. 13). Zde je nutno vybrat umístění, název projektu a výběr jazyka, se kterým se bude pracovat.

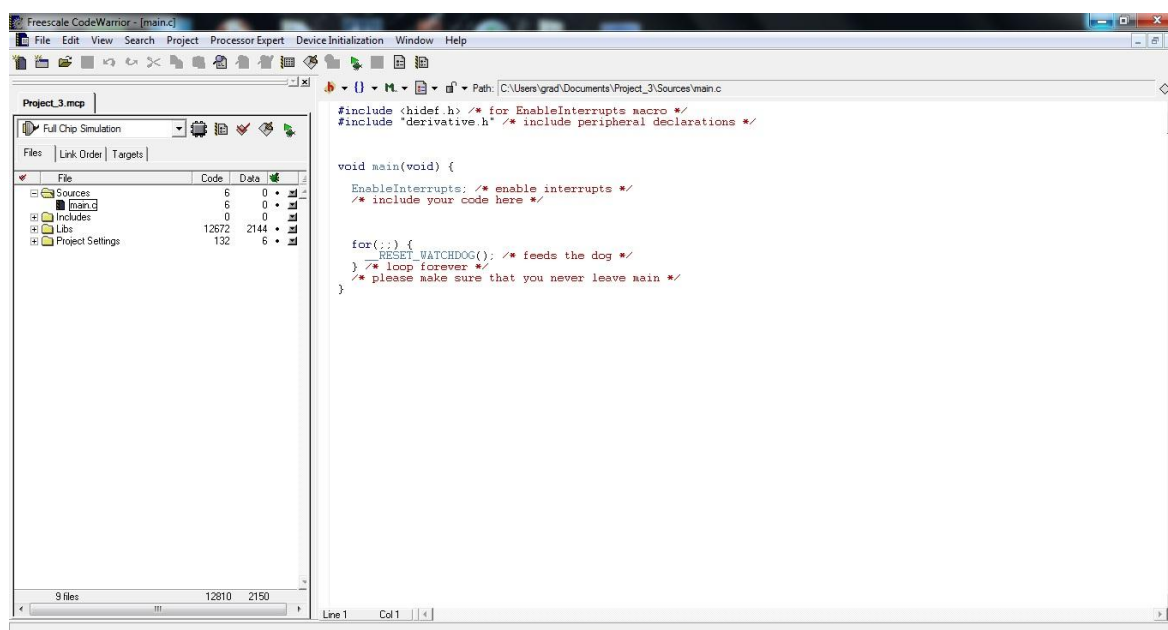


Obr. 13 Volba programovacího jazyka a umístění programu programem CodeWarrior

Po stisknutí tlačítka dokončit bude vytvořen zadaný projekt. Pokud ovšem uživatel klikne na ikonu „další“, bude mít více možností projektu, jako je například přidání podpůrné knihovny.

4.2 Hlavní okno programu

Po úspěšném dokončení vytváření projektu se v levé části programu zobrazí adresářová struktura (obr. 14). Otevřením souboru main.c se zobrazí předdefinovaná kostra programu, odlišná pro jazyk C a assembler. Do předem vytvořené kostry může uživatel libovolně doplňovat text v daném programovacím jazyce.



Obr. 14 Kostra vytvořená programem CodeWarrior

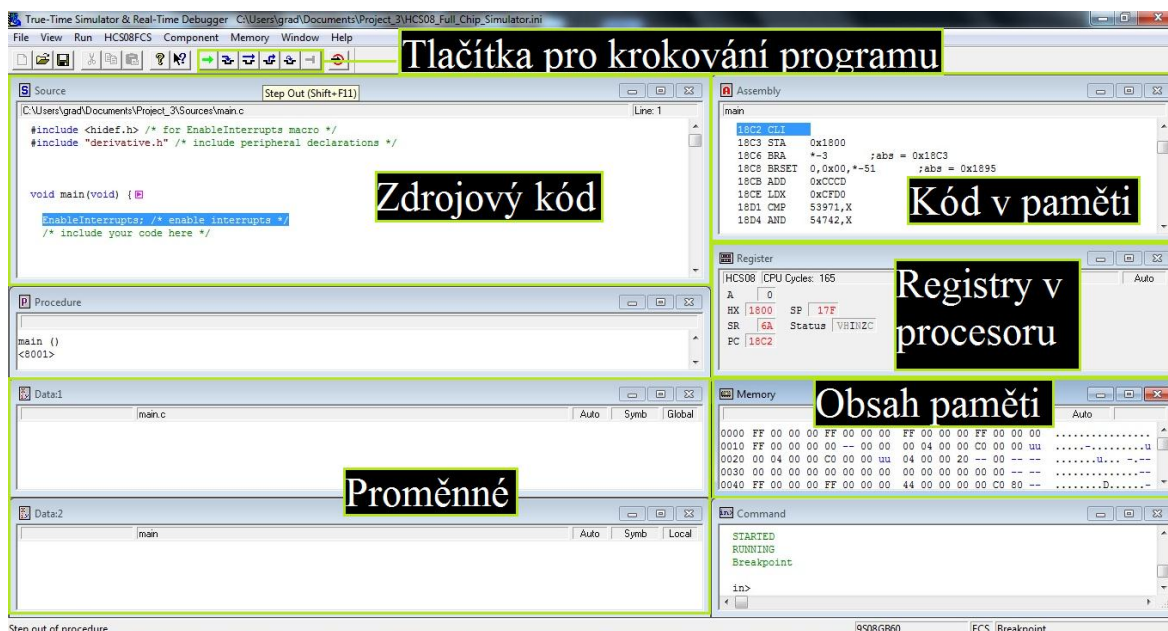
4.3 Překlad a spuštění programu

Po úspěšném dokončení vytvořeného programu je nejprve potřeba program zkompileovat (obr. 15). To znamená převést algoritmus zapsaný v programovacím jazyce do strojového kódu. Poté je program zaveden do mikropočítače nebo simulátoru a na závěr je spuštěn.



Obr. 15 Tlačítka pro sestavení a ladění programem CodeWarrior

Debugger slouží k ladění programu. Je rozdělen do více oken, které lze vidět na obrázku (obr. 16). Ve vrchní části programu CodeWarrior jsou umístěny tlačítka pro krokování. První tlačítko je zelená šipka, ta program spustí. Dalšími tlačítky jsou Single step – přeskočení o jednu instrukci, Step Over – přeskočení podprogramu, Step Out – vyskočení z podprogramu, Assembly Step a Halt – zastavení. Další obrazovky jsou na zobrazení zdrojového kódu, proměnných programu, umístění kódu v paměti, obsahu registrů v procesoru a na závěr okno obsahu paměti.



Obr. 16 Rozložení oken v debuggeru programu CodeWarrior

5 EAGLE

Zkratka EAGLE pochází z anglického názvu Easily Applicable Graphical Layout Editor neboli editor snadně použitelného grafického rozhraní. Je to program pro tvorbu plošných spojů. Program se skládá ze tří hlavních modulů: Editor spojů, Editor schémat, Autorouter. Tyto tři moduly jsou ovládány jedním uživatelským prostředím. Dále budou všechny informace směřovány k verzi Professional.

5.1 Vlastnosti programu

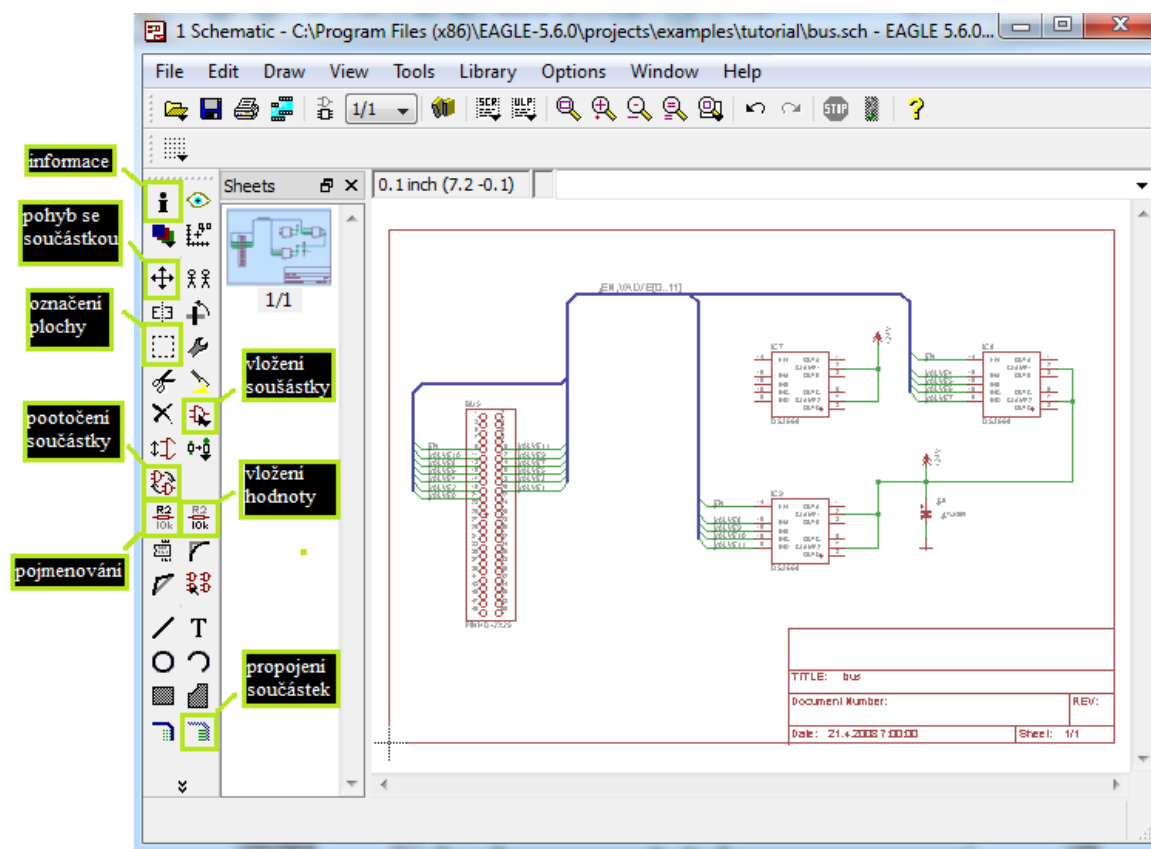
- Dopředná a zpětná anotace v reálném čase
- Nápověda orientovaná podle obsahu
- Žádná hardwarová ochrana programu
- Vícenásobná okna pro desku, schéma a knihovnu
- Výkonný uživatelský jazyk
- Integrovaný textový editor [6]

5.2 Editor spojů

Největší rozměr výkresu může být 1,6 x 1,6 m při rozlišení 1/10 000 mm. Využívá se 16-ti signálových vrstev. Může být použito klasických i SMD součástek, které jsou dodávány v knihovnách. Pokud součástka není dostupná, je snadné vytvořit vlastní součástku v plně integrovaném editoru knihoven. Uživatel má možnost použít funkci vpřed/vzad pro libovolný editační příkaz, libovolné hloubky. Po vytvoření schématu se provede kontrola pravidel návrhu [6].

5.3 Editor schémat

Při tvorbě se používá až 99 listů jednoho schématu. Dále program umí zkontrolovat elektrické pravidla zapojení, popřípadě prohodit hradla a piny. Ze schématu se zrealizuje DPS jediným příkazem. Při vytváření schématu lze použít základních vlastností, jako je kopírování, vyjmutí, vložení a řada dalších tlačítek. Ty nejdůležitější jsou popsány na obrázku (obr. 17) [6].



Obr. 17 Ukázka nejdůležitějších funkcí programu EAGLE

5.4 Autorouter

Autorouter byl vytvořen pro usnadnění a urychlení práce uživatele. Při větších projektech by nebylo možné spojovat jednotlivé cesty součástek ručně. Autorouter pracuje metodou ripup&retry router. Po umístění součástek na desku se spustí autorouter. Ten má mnoho nastavení, ovšem nenáročný uživatel nastavovat nic nemusí a program mu vše spojí automaticky. Nejzákladnějším nastavitelným parametrem je tloušťka cesty. Po provedení autorouteru má možnost uživatel automatické spojení ručně poopravit podle vlastního uvážení [6].

6 ZOBRAZOVACÍ JEDNOTKY

Zobrazovací jednotka je klasický výstup počítače. Informuje uživatele o všech programech a procesech probíhajících na počítači. Tyto jednotky nahrazují velké množství indikačních zařízení, jež jsou soustředěny na jednu plochu. Při výběru zobrazovací jednotky musí uživatel vždy dbát na dvě hlavní kritéria. Prvním kritériem je kvalita grafického výstupu a tím druhým je cena. Proto se uživatel musí rozhodnout dopředu, zda v daný moment potřebuje kvalitní zobrazovací jednotku, kterou využije například při náročných grafických aplikacích nebo mu postačí pouze obyčejná [7].

6.1 Parametry zobrazovacích jednotek

Mezi klíčové parametry zobrazovacích jednotek patří:

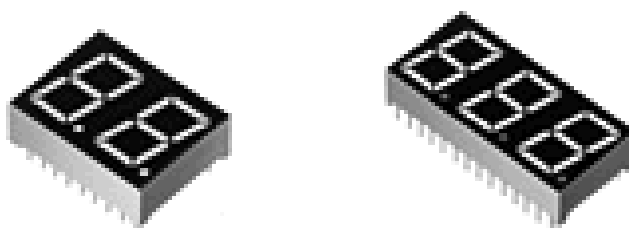
- Rozměry obrazovky – velikost nejčastěji udávána v palcích
- Rozlišovací schopnost – počet bodů, které zobrazí zobrazovací jednotka
- Obnovovací frekvence – používáno u CRT obrazovek
- Doba odezvy – u LCD, doba během níž se bod rozsvítí a zhasne (ms)
- Rozteč zobrazovaných bodů
- Počet zobrazitelných barev nebo stupňů šedi
- Jas, Kontrast
- Elektrická spotřeba
- Pozorovací úhly
- Hmotnost

6.2 Rozdělení podle možnosti zobrazování

- Numerické displeje
- Alfnumerické displeje
- Grafické displeje

6.2.1 Numerické displeje

Nejstarší zobrazovací jednotka. Umí zobrazit pouze číslice 0 až 9 a znaky plus, mínus a tečka. Díky uspořádání displeje je možné zobrazovat i písmena A až F. Proto lze zobrazit celou hexadecimální soustavu. [7]



Obr. 18 Dvou a třímístný sedmisegmentový displej

6.2.2 Alfanaumerické displeje

Na rozdíl od numerických displejů umožňují zobrazovat jak číslice, tak i písmena. Nejčastěji jsou používány matice 8x8 nebo 5x7, které jsou použity i v této bakalářské práci. Alfanaumerické displeje bývají vyráběny na bázi LED nebo LCD. Mezi alfanumerické displeje patří čtrnácti a šestnácti segmentové displeje. Protože u segmentových displejů text nevypadá tak přesvědčivě a neumí používat českou diakritiku, jsou používány jen zřídka. [7]



Obr. 19 Maticový LED displej 8x8

6.2.3 Grafické displeje

V dnešní době jsou grafické displeje nejvíce rozšířeny. Použití si najdou ve všech různých odvětvích od mobilních telefonů až po ploché monitory počítačů. Na těchto displejích je možno zobrazovat libovolné grafické obrazce. Matice, z nichž bývají složeny, jsou $M \times N$ bodů. Grafické displeje jsou vyráběny nejrozličnějšími technologiemi, například: LED, LCD, obrazovkové displeje a displeje na bázi žárovek.

6.3 Rozdělení podle technologie zobrazování

- Zobrazovací jednotky na bázi žárovek
- Zobrazovací jednotky na bázi LED
- Zobrazovací jednotky na bázi LCD
- Obrazovkové zobrazovací jednotky
- Plazmové zobrazovací jednotky

6.3.1 Zobrazovací jednotky na bázi žárovek

Zobrazovací jednotky na bázi žárovek patří mezi nejjednodušší zobrazovací jednotky. Díky jejich velké svítivosti jsou vhodné pro panely velkých rozměrů. Skládají se ze žárovek uspořádaných do matice $M \times N$ bodů. Je to už poněkud zastaralá technologie. V minulosti jsme na obrazovky na bázi žárovek mohli narazit na reklamních panelech nebo například na výsledkových tabulích na fotbalových stadionech. Jejich velkou nevýhodou je velká spotřeba a malá životnost. Tuto technologii nahradily zobrazovací jednotky na bázi LED [7].

6.3.2 Zobrazovací jednotky na bázi LED

Zobrazovací jednotky na bázi LED nahradily technologii na bázi žárovek. Jak lze z názvu poznat, jedná se o technologii, jejíž součástí jsou LED diody. Dnes se LED diody vyrábí ve čtyřech barvách: červená, zelená, modrá a žlutá. K dispozici jsou diody o různé velikosti a svítivosti. Vyrábějí se také vícebarevné LED v jednom pouzdře. Vícebarevná LED vznikne složením více diod na jednom čipu, zalitých v jednom pouzdře. Elektrický proud

prochází LED diodou pouze v propustném směru. Pokud se zapojí v závěrném směru, proud skrz diodu neprochází, a proto nevyzařuje žádné světlo [7].

6.3.3 Zobrazovací jednotky na bázi LCD

Na rozdíl od jednotek na bázi LED, které se používají na větších plochách, jsou zobrazovací jednotky na bázi LCD zpočátku používány v menších displejích. Zobrazovací jednotky na bázi LCD nemají žádný negativní vliv na zrak člověka. Jsou to displeje složené z kapalných krystalů. Klasická technologie LCD vychází ze dvou skleněných destiček, mezi nimiž je prostor s náplní kapalných krystalů. Na vnitřní straně skleněných destiček je napařena průhledná kovová vrstva, tvořící na jedné destičce reliéf požadovaných zobrazených segmentů, na druhé destičce společnou elektrodu. Vývody jednotlivých segmentů a společné elektrody jsou provedeny technikou tlusté kovové vrstvy na skle. Z obou stran jsou nalepeny polarizační fólie. Bez těchto fólií není zobrazení patrné. Za klidového stavu prochází světlo tekutými krystaly a buňka svítí. Zapnutím zdroje střídavého napětí se změní vnitřní struktura tekutého krystalu a světlo je zablokováno, buňka nesvítí. Mají velmi nízkou spotřebu proudu. LCD displeje jsou vyráběny ve třech skupinách. První skupinou jsou numerické zobrazovače, které jsou složené z několika sedmisegmentových displejů. Do druhé skupiny patří alfanumerické jednotky a do třetí grafické jednotky. V dnešní době se LCD displeje používají od notebooků, plochých monitorů až po velkoplošné obrazovky [7][10].



Obr. 20 Alfa-numerický LCD displej 16x4

6.3.4 Obrazkové zobrazovací jednotky

Obrazkové zobrazovací jednotky neboli CRT byly v minulosti nejvyžívanější technologií pro výrobu obrazovek. Mají větší rozměry i hmotnost než LCD. Pro televizory byla používána obrazovka s elektromagneticky vychylovaným paprskem a pro osciloskopy s elektrostaticky vychylovaným paprskem. U černobílé obrazovky bylo používáno jediného paprsku, na rozdíl od barevné obrazovky, kde jsou použity 3 elektronové paprsky. Barevné body vznikají dopadem elektronového paprsku na daný fosforový bod neboli luminofor. Největší nevýhodou CRT obrazovek je velikost zobrazovacího zařízení.[11]

6.3.5 Plazmové zobrazovací jednotky

Plazmová obrazovka je používána pro ploché a zároveň velké televizory. Tato technologie využívá iontů a elementárních částic. Proto plazma není plynem, kapalinou ani pevnou látkou. V klidovém stavu se v displejích nachází směs vzácných plynů. Plazma se vytvoří zavedením elektrického proudu do plynu, čímž se objeví mnoho volných elektronů. Srážkami mezi těmito elektrony vznikají kladně nabitě ionty, a tím získáváme plazmu. Jelikož plazmové displeje samy o sobě emitují světlo, mají vynikající pozorovací úhly. Spotřeba těchto obrazovek je větší, jak u LCD i CRT technologie.[11]

II. PRAKTICKÁ ČÁST

7 NÁVRH INFORMAČNÍHO LED DISPLEJE

Při návrhu informačního LED displeje se bral ohled zejména na to, aby byl vytvořený modul plně kompatibilní s vývojovým kitem M68EVB908GB60. Dále se vytvořilo propojení modulu s vývojovým kitem. Jelikož je napájení modulu 3,3 V, byl návrh přizpůsoben tomuto napětí.

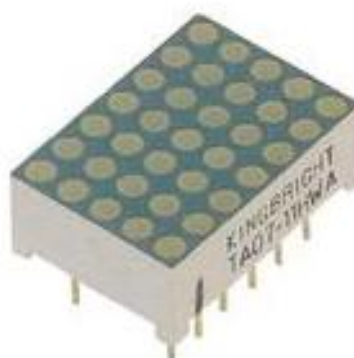
7.1 Použité součástky na modulu

Maticový displej:

Byl využit LED maticový displej od firmy Kingbright řady TA07-11HWA (obr. 21). Jeden displej se skládá ze sedmi řádků a pěti sloupců. Aby byla výsledná matice vytvořeného modulu 7 řádků a 50 sloupců, bylo použito deseti těchto součástek.

Technické parametry displeje:

- Rozměr – 12,7 x 17,8 x 6,3 mm (šířka x výška x hloubka)
- Barva – červená
- Výška číslic – 18 mm
- Svítivost – 900 μ cd
- Vlnová délka – 700 nm
- Společný kontakt – katoda

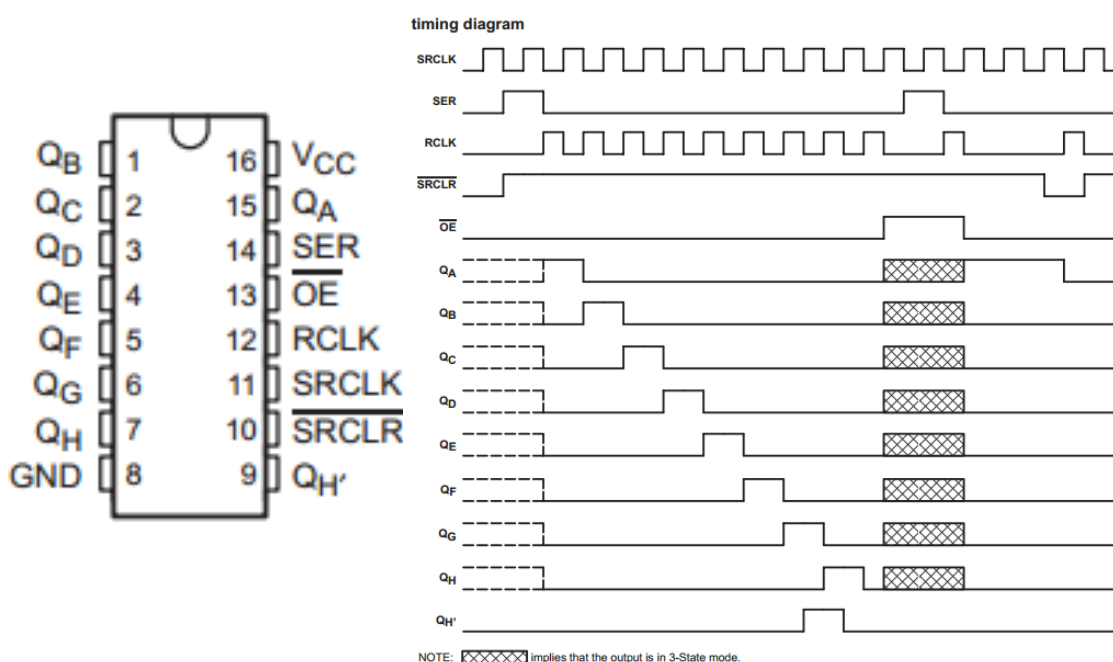


Obr. 21 Maticový LED displej 5x7

Sériový registr 74HC595 ve funkci budiče sloupců:

Bylo použito osmibitového posuvného registru 74HC595. Jelikož má výsledná matice modulu 50 sloupců, bylo potřeba 7 těchto součástek. Šest registrů bylo plně využito, poslední sedmý registr využije jen 2 vývody. Komunikace probíhá přes sériové rozhraní SPI na frekvenci 5 MHz. Do sériového registru se jen zapisuje. Řídící vodiče posuvného registru jsou:

- enable, který se stará o zobrazení informace na displeji
- latch, jenž přepisuje informace z datových registrů na výstup



Obr. 22 Rozložení pinů posuvného registru 74HC595 s časovým diagramem

Dekodér 74HC138 ve funkci budiče řádků a tranzistor IRF7104:

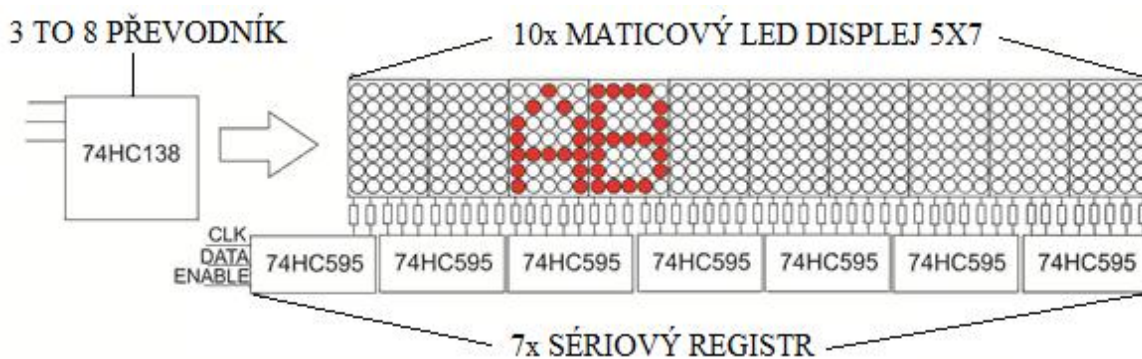
Dekodér 74HC138 převádí binární kombinace na dekadickou. Tři binární vstupy (2^3 kombinací), reprezentující číslo v binární podobě, zobrazí jako osm negovaných dekadických kombinací na výstupu. Podle datasheetu je proud na jednom pinu 25 mA, proto jsou na modulu umístěny také unipolární tranzistory, které posílí výstupní proud. Jedná se o MOSFET tranzistory s P-kanálem. Tyto tranzistory jsou použity z toho důvodu, že mají malý odpor (R_{ds}), který je $0,3 \Omega$. Při průchodu proudem nevzniká odpadní teplo.

		VSTUPY						VÝSTUPY							
		\bar{E}_1	\bar{E}_2	E_3	A_0	A_1	A_2	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7
A_0	1	H	X	X	X	X	X	H	H	H	H	H	H	H	H
A_1	2	X	H	X	X	X	X	H	H	H	H	H	H	H	H
A_2	3	X	X	L	X	X	X	H	H	H	H	H	H	H	H
\bar{E}_1	4	L	L	H	L	L	L	L	H	H	H	H	H	H	H
\bar{E}_2	5	L	L	H	H	L	L	H	L	H	H	H	H	H	H
E_3	6	L	L	H	H	H	L	H	H	H	L	H	H	H	H
\bar{Y}_7	7	L	L	H	L	L	H	H	H	H	H	L	H	H	H
\bar{Y}_6	8	L	L	H	H	L	H	H	H	H	H	H	L	H	H
\bar{Y}_5	9	L	L	H	H	H	H	H	H	H	H	H	H	L	H
\bar{Y}_4	10	L	L	H	H	H	H	H	H	H	H	H	H	H	L
\bar{Y}_3	11	L	L	H	H	H	L	H	H	H	L	H	H	H	H
\bar{Y}_2	12	L	L	H	H	L	L	H	L	H	H	H	H	H	H
\bar{Y}_1	13	L	L	H	L	L	L	L	H	H	H	H	H	H	H
\bar{Y}_0	14	X	X	X	X	X	X	H	H	H	H	H	H	H	H
V_{CC}	16														
\bar{Y}_0	15														
\bar{Y}_1	14														
\bar{Y}_2	13														
\bar{Y}_3	12														
\bar{Y}_4	11														
\bar{Y}_5	10														
\bar{Y}_6	9														
\bar{Y}_7	8														
GND	8														

Obr. 23 Rozložení pinů převodníků 74HC138 s funkční tabulkou

7.2 Blokové schéma

Z tohoto blokového schématu bylo vycházeno při návrhu informačního LED displeje. Blokové schéma se skládá z deseti maticových LED displejů Kingbright. Každý displej je složen z 5 sloupců a 7 řádků. Ve výsledku se získá matice o velikosti 7x50, což bylo pro výukové účely dostačující. Dále jsou použity pro buzení jednotlivých sloupců sériové registry 74HC595. Ty musí být použity sedmkrát, jelikož je potřeba 50 sloupců. Jeden z registrů dovede ovládat pouze osm sloupců. Pro buzení řádků bylo použito dekodéru z 3 na 8 a to konkrétně 74HC138.



Obr. 24 Blokové schéma informačního LED displeje

7.3 Návrh schématu a desky plošných spojů

Při návrhu schématu byly respektovány všechny požadavky, které byly dány vlastnostmi vývojové kitu. Těmito požadavky je například napájecí napětí 3,3 V. Proto je použito DC/DC měniče. Zpočátku bylo vytvořeno pouze jedno velké schéma. Jelikož by byla DPS příliš velká a její výroba složitá, bylo rozhodnuto, že bude modul rozdělen na dvě části. První schéma bylo pojmenováno jako zobrazovací deska, protože obsahuje displeje se sériovými registry. Ve druhém schématu je umístěno napájení, proto se jmenuje napájecí. Propojení vývojového kitu s modulem zajišťuje konektor 30x2 piny.

7.4 Zobrazovací modul

7.4.1 Návrh schématu zapojení

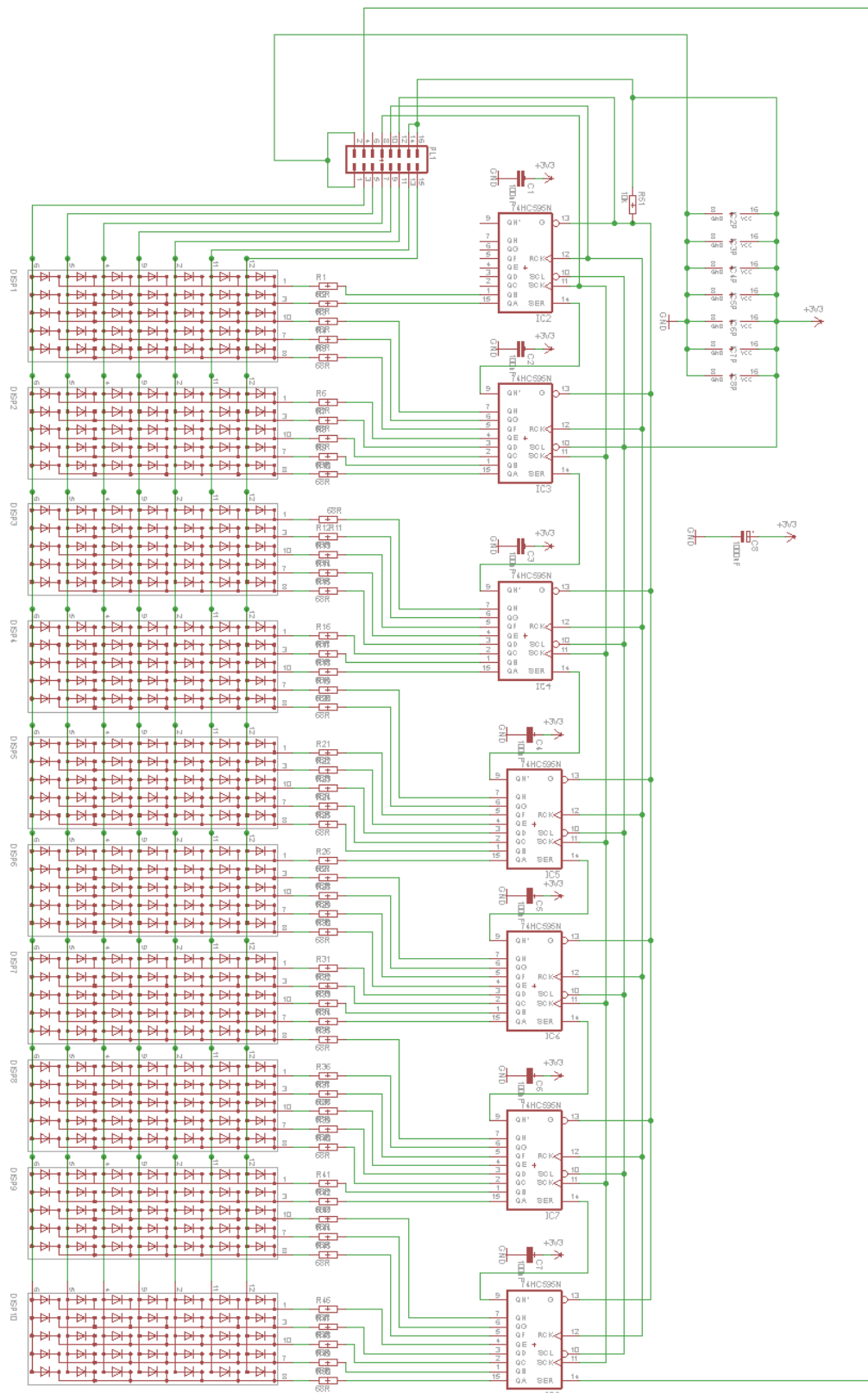
Nejdůležitější součástí tohoto schématu je LED maticový displej, ten lze vidět v levé části schématu (obr. 25). Je to displej od značky Kingbright TA07-11HWA. Výsledný obraz je složen z 10 těchto součástek. Jelikož je to displej se společnou katodou, funguje následovně. Po přivedení napájecího napětí na příslušný řádek a uzemněním příslušného sloupce bude rozsvícen daný bod. Tento displej svítí červenou barvou. Ke každému sloupci je přiveden rezistor omezující proud do LED. Velikost rezistoru byla vypočtena pomocí Ohmova zákona.

$$R = \frac{U_N - U_D}{I_D} \quad (1)$$

Do výrazu (1) bylo doplněno za napájecí napětí (U_N) 3,3 V. Napětí na diodě (U_D) 2 V a proud na diodě (I_D) 20 mA. Výsledný odpor má hodnotu 65 Ω . Podle odporové řady byl zvolen rezistor 68 Ω .

Nad rezistory jsou posuvné registry 74HC595. Do těchto posuvných registrů se zapíše data. Pokud je nastavená log. 0, rozsvítí se LED ovládaná posuvným registrem. Jestliže je nastavená log. 1, zůstane LED zhaslá. Vedle posuvných registrů jsou umístěny blokovací kondenzátory omezující rušení napájení. Ve schématu se ještě nachází rezistor R51 o velikosti 10 k Ω . Funkce tohoto rezistoru je následující. Jakmile není nakonfigurován port jako výstup a je ve stavu vysoké impedance, tento rezistor zakáže zobrazení na displeji. Vpravo uprostřed se nachází filtrační kondenzátor pro vyrovnávání poklesů napětí.

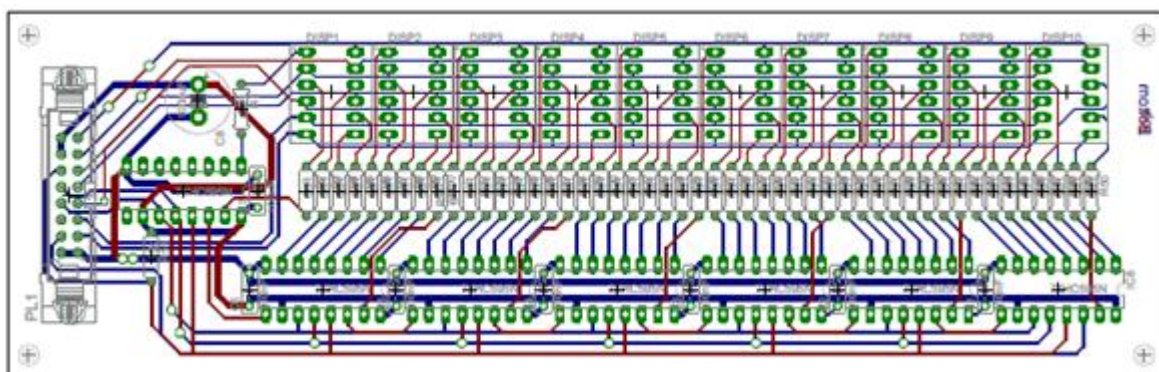
Poslední součástíou, kterou lze na schématu nalézt, je PSL 16 konektor. Tímto konektorem se zajišťuje propojení zobrazovacího modulu s napájecím.



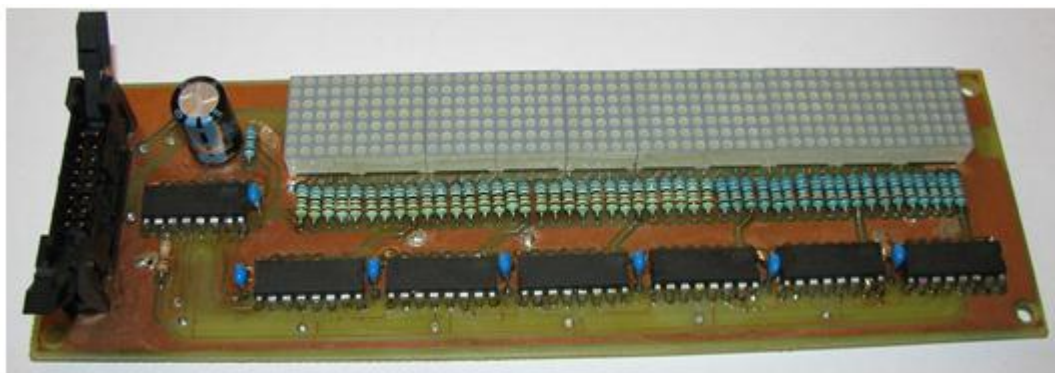
Obr. 25 Schéma zobrazovacího modulu

7.4.2 Návrh desky plošných spojů

Jak lze na obrázku (obr. 26) vidět, realizace DPS je provedena programem EAGLE. Je jasné, že navrhnutá DPS je oboustranná. Jelikož se i přes rozdělení schématu na dvě desky nepodařilo vyhnout oboustrannému DPS, bylo velice náročné ji vyrobit. Všechny součástky jsou umístěny z vrchní strany desky (top vrstva). Ze spodní strany desky (bottom vrstva) vedou pouze spojovací cesty. Rozvržení součástek je velice jednoduché. Na levém okraji je umístěn PLS konektor. Umístění na kraji je zvoleno proto, aby byl konektor snadno dostupný. Na vrchní straně DPS jsou displeje. Pod displeji se nachází rezistory a v dolní části DPS sériové registry s blokovacími kondenzátory. Na obrázku (obr. 27) je možné si prohlédnout výsledné zhotovení zobrazovací desky informačního LED displeje.



Obr. 26 DPS zobrazovacího modulu

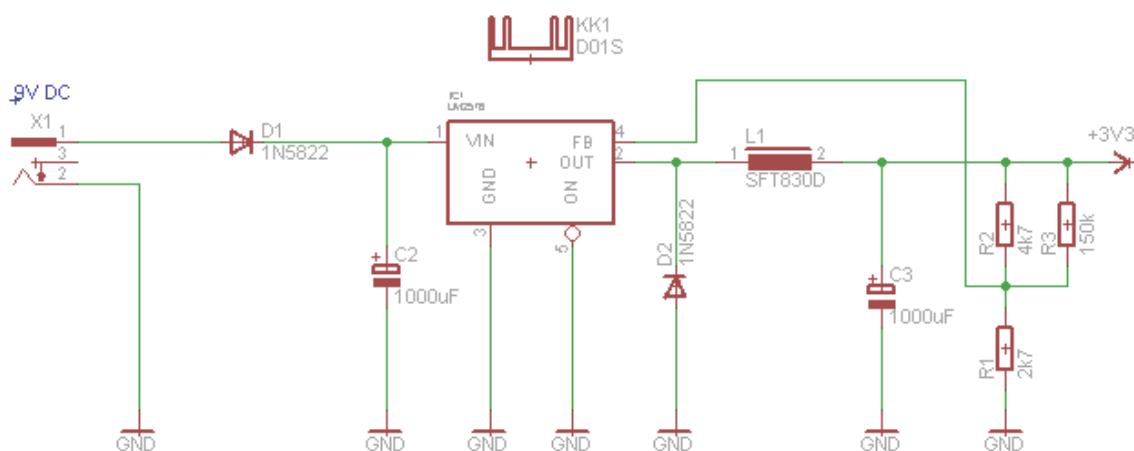


Obr. 27 Zobrazovací modul osazený součástkami

7.5 Napájecí modul

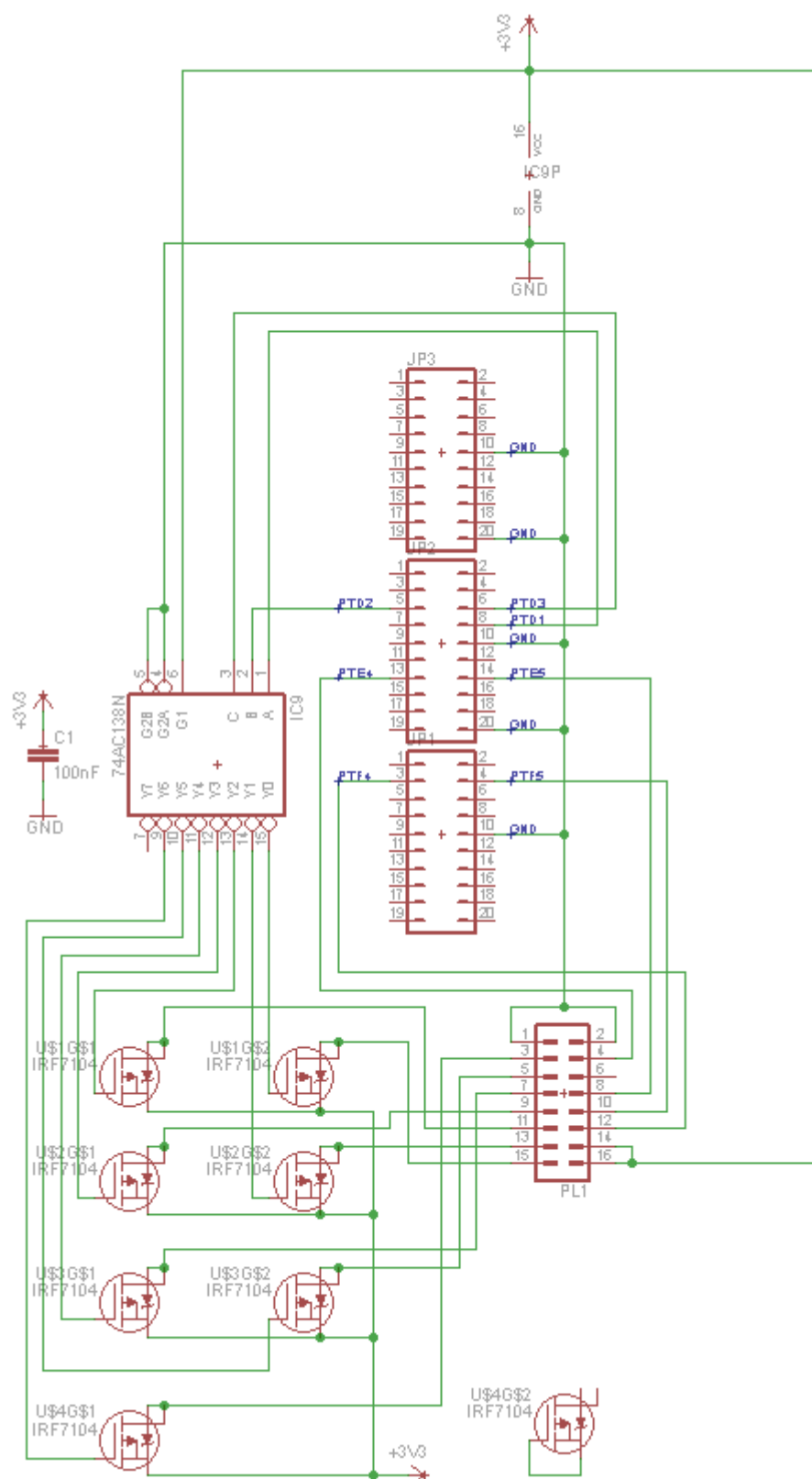
7.5.1 Návrh schématu zapojení

Napájecí schéma je rozděleno do dvou částí. Tou první je DC/DC měnič z 9 V na 3,3 V (obr. 28). Zleva lze vidět napájecí konektor PC, který dodává stejnosměrné napětí o hodnotě 9 V. Ochrana proti přepólování je provedena diodou D1. Jako integrovaný DC/DC konvertor je použit LM2576. Jeho výstup je blokován diodou a filtrován LC propustí.

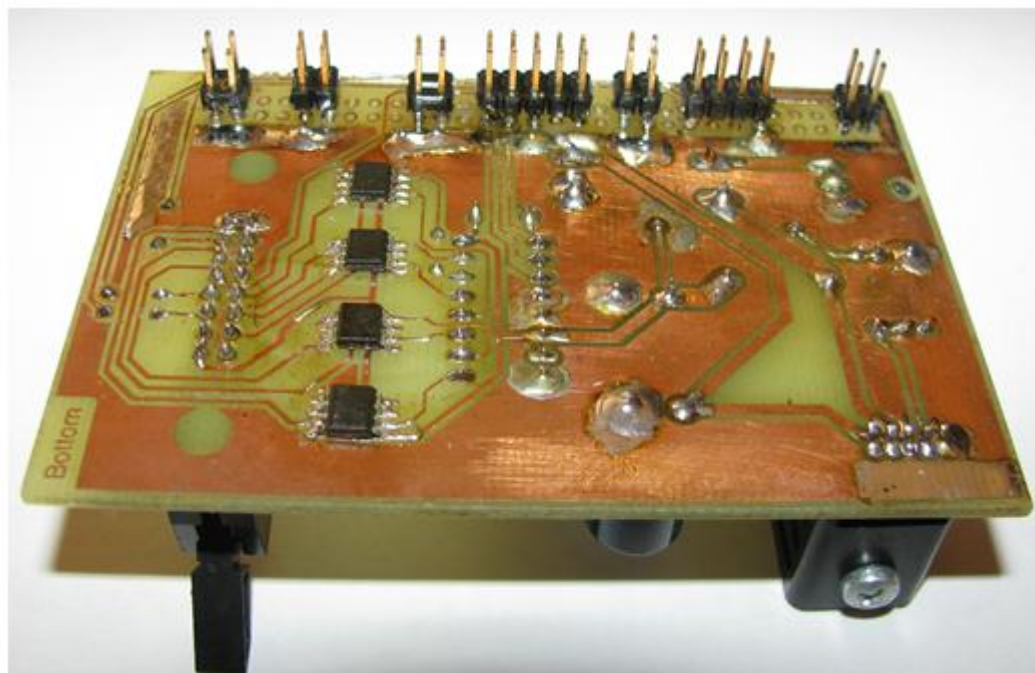


Obr. 28 Schéma napájecího modulu – část 1. DC/DC měnič

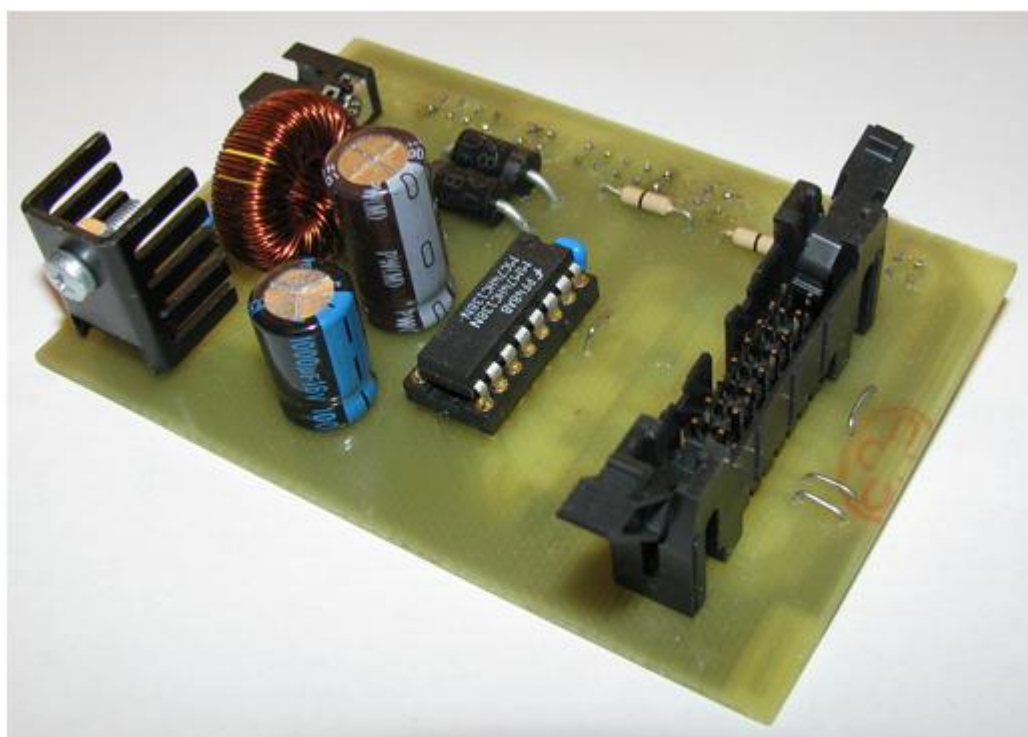
Ve druhé části schématu se nachází převodník 74HC138, neboli převodník 3 na 8 (obr. 29). 74HC138 slouží k převodu binární kombinace na dekadickou. Vedle něj je blokovací kondenzátor C1. V levé spodní části jsou zapojené výkonové tranzistory MOSFET s P-kanálem. Byly použity unipolární tranzistory IRF7104. Tyto tranzistory jsou použity proto, že při průchodu proudu vzniká díky malému vnitřnímu odporu jen nepatrný úbytek napětí a tím i malá tepelná ztráta. Napravo schématu je použit PSL 16 konektor, propojující napájecí a zobrazovací modul. Poslední součástí ve schématu je pin header konektor 30x2. Tento konektor slouží pro propojení s vývojovým kitem M68EVB908GB60.



Obr. 29 Schéma napájecího modulu – část 2.



Obr. 31 Napájecí modul osazený součástkami - Bottom



Obr. 32 Napájecí modul osazený součástkami - Top

8 PROGRAMOVÁ ČÁST

Nejprve byla vytvořena podpůrná knihovna funkcí v jazyce C. Posléze je přepsána i do jazyka symbolických adres. Tato knihovna například umožňuje zobrazit bit, znak, statický text a rolovací text. Knihovna byla vytvořena ve vývojovém prostředí CodeWarrior od firmy Freescale.

8.1 Knihovna funkcí

Nejprve bylo potřeba vytvořit vlastní znakovou sadu, kterou program ukládá do RAM paměti pomocí proměnné Abeceda. Vytvořená sada obsahuje 110 znaků. Rozsah znaků podle ASCII je 32 až 126. Jako nadstavba byly vytvořeny i znaky s českou diakritikou, proto vznikne celkový počet 110 znaků. Jelikož má displej pět sloupců, musí každý znak zabírat 5 bajtů.

Programová knihovna má implementovány následující funkce:

- LedInit
- Fill
- Clear
- Timer_int
- SetBit
- ZobrazZnak
- ZobrazText
- RotujText

8.1.1 LEDInit

Tato funkce se stará o veškeré nastavení. Pomocí vztahu (2) je vypočtena perioda přerušení od časovače. Vztah je následující:

$$MODULO = \frac{t_{TOF} - f_{SOURCE}}{Pr\,escaler} \quad (2)$$

Dosazením všech potřebných údajů do rovnice (2), vyšla hodnota přerušení od časovače (t_{TOF}) 1,43 ms. Jelikož je řádků na displeji 7, výsledná obnovovací frekvence je rovna 100 Hz.

Dále se funkce stará o aktivaci hardwarové SPI sběrnice, nastavení rychlosti komunikace, režimu master, polarity a fáze hodinového signálu SPI sběrnice. Jsou nastaveny i piny pro řízení sériových registrů a dekodéru.

8.1.2 Funkce Fill

Funkce Fill, jak už z názvu vyplývá, modul rozsvěcuje. Nemá žádné vstupy. Všechny bity pole se nastaví do log. 0, a tím se rozsvítí celý modul.

8.1.3 Funkce Clear

Funkce Clear je opakem funkce Fill. Nemá žádné vstupy. Clear nastaví všechny bity pole na log. 1, a tím celý modul zhasne. Funkce se používá pro vymazání (zhasnutí) celého displeje.

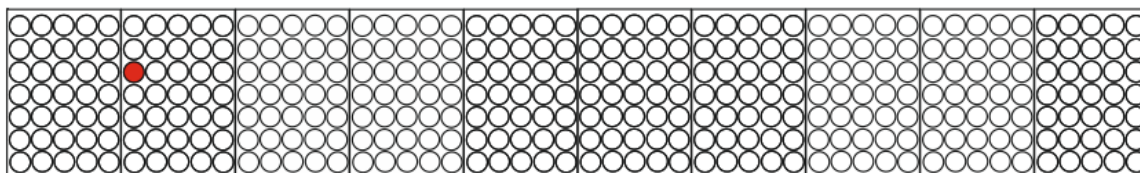
8.1.4 Timer_int

Funkce Timer_int nemá žádné vstupy a výstupy. Stará se o časový multiplex displeje, který je 100Hz. Volá se každých 1,43ms. Přes SPI se do sériových registrů odešle vždy jeden celý řádek.

8.1.5 Funkce SetBit

- Funkce: `int SetBit(int radek, int sloupec, int zobraz)`
- Příklad použití: `SetBit(2,5,1)`

Vstupy pro tuhle funkci jsou „radek“, „sloupec“ a „zobrazení“. Parametrem „radek“ se nastaví číslo řádků, na kterém je potřeba bit rozsvítit. Parametrem „sloupec“ se určí číslo sloupce, který se rozsvítí. Poslední parametr je „zobraz“. Lze buď nastavit do log. 0 – to znamená, že se znak nerozsvítí nebo do log. 1 – rozsvítí nastavený bit. Pro uvedený příklad použití se nastavený bit rozsvítí na třetím řádku a šestém sloupci, protože je číslování počítáno od nuly (obr. 33).



Obr. 33 Příklad použití funkce `SetBit(2,5,1)`

8.1.6 Funkce ZobrazZnak

- Funkce: `void ZobrazZnak(unsigned char znak, int sloupec, int radek, int invert)`
- Příklad použití: `ZobrazZnak ('A', 0, 0, 1)`

Pro tuhle funkci jsou připravené čtyři parametry. Prvním parametrem je „znak“. Jako parametr se zadá do apostrofů znak nebo je možno zadat číslo ASCII kódu. Výsledný znak se vykreslí na displeji. Dalšími parametry jsou již dříve zmiňované nastavení sloupce a řádku. Čtvrtým parametrem je „invert“. Nastavením tohoto parametru do log. 1 se zobrazí znak invertovaný. Pro uvedený příklad použití je zobrazen znak ‘A’, invertovaný, který se rozsvítí v levém horním rohu.

8.1.7 Funkce ZobrazText

- Funkce: void ZobrazText(unsigned char znak[], int sloupec, int radek, int invert)
- Příklad použití: ZobrazText („Zašováá!“,0,0,0)

Funkce ZobrazText má stejné parametry jako funkce ZobrazZnak. Jediným rozdílem je, že do parametru „znak“ se zadá v uvozovkách celý text. Pro uvedený příklad použití je zobrazen text „Zašováá!“ neinvertovaný.



Obr. 34 Příklad použití funkce Zobraz text na reálném modulu - neinvertovaný



Obr. 35 Příklad použití funkce Zobraz text na reálném modulu - invertovaný

8.1.8 Funkce RotujText

- Funkce: void RotujText(unsigned char znak[], int sloupec, int radek, int smer, int invert)
- Příklad použití: RotujText („Jiří Pobořil“,0,0,3,0)

Funkce RotujText vychází z funkce ZobrazZnak. Do této funkce je přidán nový parametr „smer“. Nastavením tohoto parametru se zvolí směr, kterým bude text rotovat. Jak nastavit směr rotace je patrné z přiložené tabulky (tab. 3). Pro uvedený příklad použití se zobrazí text „Jiří Pobořil“, nebude invertovaný a bude rotovat zdola nahoru.

Tab. 3 Volba směru rotace

Směr rotace	Nastavení parametru „směr“
zprava doleva	0
zleva doprava	1
shora dolů	2
zdola nahoru	3

8.2 Ukázkový program demonstrující funkci modulu

Program demonstrující funkci modulu využívá nejen vytvořeného LED displeje, ale také LCD displeje, potenciometru a vratných tlačítek na vývojovém kitu. Ukázkový program byl nazván Autobus. Skládá se ze dvou částí. První část, pojmenovanou jako ovládací, má k dispozici řidič autobusu. Druhou vidí cestující v autobuse, proto je nazvána zobrazovací.

Po spuštění programu se na LCD displeji vypíše uvítací obrazovka. Na této obrazovce se ukáže jméno řidiče a verze firmwaru po dobu pěti sekund. Po uplynutí této doby se oba displeje smažou. Na LCD displeji, ovládaném řidičem, naskočí nápis „výběr trasy“. Ten po dvou sekundách zmizí. Na prvním řádku se zobrazí nápověda, na druhém trasa, zvolená řidičem. Rozložení výpisu těchto dvou řádků platí v celém programu. Volba trasy se provádí potenciometrem. Po nalezení požadované trasy zmáčkne řidič tlačítko SW1 a tím potvrdí volbu. Proveďte se skok na druhou obrazovku - volba směru jízdy. Tato volba je důležitá pro správné označení zastávek. Není jedno, jestli se pojedje trasa Vsetín – Zlín nebo Zlín – Vsetín. Potvrzení se opět provede tlačítkem SW1. Pokud se bude chtít řidič vrátit do menu volba trasy, musí zmáčknout tlačítko SW3. Návrat zpět tlačítkem SW3 platí v celém programu. Poslední volbou řidiče je zastávka, z které bude vyjíždět. Jakmile je potenciometrem vybrána aktuální zastávka, na vytvořeném modulu LED displeje se zobrazí číslo zastávky ve tvaru 01,02,03... Číslo zastávky se zobrazí invertovaně. Za číslem je umístěno jméno zastávky. Po dvou sekundách začne jméno zastávky rotovat displejem zprava doleva. Co bude následovat po projetí celého textu, záleží na tom, jestli to byla poslední nebo bude následovat další zastávka. Pokud řidič dojedie na konečnou zastávku, přejede LED displejem nápis „Konečná“ zdola nahoru. Když nebude zastávka konečná, projede displejem nápis „Příští“ shora dolů. Po tomto nápisu následuje název příští zastávky zprava doleva. Na LCD displeji vidí řidič aktuální zastávku na prvním

řádku, na druhém řádku zastávku příští. Přepnutí na další zastávku se provádí tlačítkem SW1. Může se stát, že řidič udělá chybu a zmáčkne tlačítko SW1 dvakrát. Proto je ošetřeno druhým tlačítkem SW2 přepnutí o zastávku zpět.

ZÁVĚR

Cílem této práce bylo zhotovení výukového modulu pro předmět programování mikropočítačů. Modul informační LED displej bude sloužit jako výuková pomůcka tohoto předmětu.

Byl vyroben zmenšený model jednořádkového informačního LED displeje. Modul je složen z maticových LED displejů 5x7. Na modulu je možné zobrazit statický nebo pohyblivý text, který je může rolovat různými směry. Postupně byly přidány funkce jako možnost invertování textu nebo nastavení bodu, v němž se daný text zobrazí. Knihovna funkcí je tvořena dvěma programovacími jazyky. Těmito jazyky jsou vyšší programovací jazyk C a jazyk symbolických adres. Pro tvorbu knihovny bylo použito programu CodeWarrior od firmy Freescale.

V teoretické části byl popsán vývojový kit M68EVB908GB60, na kterém je umístěn mikropočítač MC9S08GB60. K tomu vývojovému kitu je připojen vytvořený modul. Bylo nutné navrhnout vhodné propojení kitu s modulem. Pro propojení je použito pin header konektoru 30x2 pinů. Komunikace mezi řídicím mikroprocesorem a informačním LED displejem probíhá pomocí sériového periferního rozhraní. SPI je také důkladně rozebráno v teoretické části. Dále se v teoretické části nachází popis programů EAGLE a CodeWarrior. Poslední částí je přiblížení zobrazovacích jednotek, jež jsou důležitou součástí modulu.

V praktické části se nachází dvě důležité kapitoly. V kapitole o návrhu informačního LED displeje je popis funkcí součástek, jejichž znalost byla potřeba pro programování funkcí. Další částí je návrh schématu, vygenerování DPS a na závěr výsledné zhotovení informačního LED displeje. Návrh schématu a DPS byl proveden programem EAGLE. Ve druhé polovině praktické části je umístěn popis vytvořené knihovny a jejích funkcí.

Při tvorbě informačního LED modulu vznikla spousta problémů, s nimiž bylo potřeba se vypořádat. Po výsledném zapájení součástek vše nefungovalo, jak mělo. Byla potřeba spousta hodin doladování a hledání chyb, které vznikly při výrobě modulu. Nakonec se podařilo všechny chyby odladit a modul se chová tak, jak je potřeba.

ZÁVĚR V ANGLIČTINĚ

The main goal of this thesis is construction of learning module for seminar called Microcomputers Programming. Information LED display module will be used as a teaching tool of this seminar.

There was made a miniature of single-line information LED display. Module is composed of matrix LED displays 5x7. Module can display static or scrollable text, text can scroll in different directions. Functions such as option of text inverting or setting the point at which text appear, were gradually added. Function's library consists of two programming languages. These languages are high level programming language C and assembler language. Freescale's programme CodeWarrior was used to create the library .

Development kit M68EVB908GB60 where is located a microcomputer MC9S08GB60 was described in the theoretical part of thesis. Created module is connected to this development kit. It was necessary to design a suitable connection between kit and module. For the connection is used pin header connector 30x2 pins. Communication between the directing microprocessor and information LED display is done via a serial peripheral interface. SPI is also thoroughly discussed in the theoretical part. Furthermore, in the theoretical part is located a description of Eagle and CodeWarrior programmes. In the last part, there is an approximation of display units that are an important components of the module.

Two important chapters are situated in the practical part of thesis. In the chapter about the information LED display design, there is a description of the functions of components whose knowledge was needed for programming functions. Another part is the scheme design, generation by DPS and information LED display construction in conclusion. Scheme design and DPS were performed by program EAGLE. In the second half of the practical part, you can find description of the created libraries and their functions.

During an information LED module construction, there was a lot of problems to deal with. After the final soldering of components, some parts did not work as expected. A lot of hours were needed to finalize module and to search and fix errors that arise in the process of construction of module. In the end, all errors were successfully detected and fixed and the module works correctly.

SEZNAM POUŽITÉ LITERATURY

- [1] CHOCHOLÁČ, Jaromír. ELEKTRONICKÉ POČÍTAČE: Mikroprocesorová technika. SPŠE Rožnov p. Radhoštěm., 1996
- [2] EPO. [online]. S. 1-93 [cit. 2012-05-23]. Dostupné z: <http://corella-web.yc.cz/epo.pdf>
- [3] M68EVB908GB60: Development Board for Freescale MC9S08GB60. Dostupné z: http://cache.freescale.com/files/soft_dev_tools/doc/user_guide/M68EVB908GB60UM.pdf?fp=1
- [4] TIŠNOVSKÝ, Pavel. Externí sériové sběrnice SPI a I²C. 20008. Dostupné z: <http://www.root.cz/clanky/externi-seriove-sbornice-spi-a-i2c/>
- [5] FREESCALE SEMICONDUCTOR. MC9S08GB60/D: HCS08 Microcontrollers - Data Sheet. [online]. 2004 [cit. 2012-05-23].
- [6] Eagle Online: Informace o programu EAGLE. 2001 CADSOFT COMPUTER GMBH. [online]. [cit. 2012-05-23]. Dostupné z: <http://www.eagle.cz/info.htm>
- [7] LATTENBERG, Ivo. Moderní zobrazovací jednotky. [online]. 1999 [cit. 2012-05-23]. Dostupné z: <http://www.elektrorevue.cz/clanky/99002/index.html>
- [8] KLABAZŇA, Petr. Procesor slaví třicátiny: Procesory. 2001. Dostupné z: http://www.svethardware.cz/art_doc-5747EB35528EF0BBC1256B0B004E49C3.html
- [9] Von Neumannova architektura. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-24]. Dostupné z: http://cs.wikipedia.org/wiki/Von_Neumannova_architektura
- [10] Displej z tekutých krystalů. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-05-24]. Dostupné z: http://cs.wikipedia.org/wiki/Displej_z_tekut%C3%BDch_krystal%C5%AF
- [11] OLIVKA, Petr a Tomáš KOTULA. Zobrazovací jednotky počítačů - monitory: Studijní materiál pro předmět Architektury počítačů. [online]. 2010 [cit. 2012-05-24]. Dostupné z: <http://poli.cs.vsb.cz/edu/arp/down/monitory.pdf>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

A/D	Analog/Digital
ALU	Arithmetic logic unit
AMD	Advanced Micro Devices
ASCII	American Standard Code for Information Interchange
BDM	Background Debug Mode
CMOS	Complementary Metal-Oxide-Semiconductor
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DC	Direct current
DIP	Dual in package
DPS	Deska plošných spojů
EEPROM	Electrically Erasable Programmable Read-Only Memory
I ² C	Inter-Integrated Circuit
I/O	Input/Output
IRQ	Interrupt Request
LCD	Liquid crystal display
LED	Light-Emitting Diode
LSB	Least significant bit
MCU	Micro Controller Unit
MISO	Master In, Slave Out
MOSI	Master Out, Slave In
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MSB	Most significant bit
NMOS	N-type Metal-Oxide-Semiconductor

PMOS	P-type Metal-Oxide-Semiconductor
RAM	Random Acces Memory
RWM	Reaa Write Memory
SCI	Serial Communications Interface
SCK	Serial Clock
SDI	Serial Data Input
SDO	Serial Data Output
SMD	Surface mount device
SPE	SPI System Enable
SPI	Seriál Peripheral Interface
SPSCK1	SPI Serial Clock
SS1	Slave Select
SSEL	Slave Select
SSPSR	Shift Register
TPM	Timer/Pulse Module

SEZNAM OBRÁZKŮ

Obr. 1 John Von Neumannova architektura	13
Obr. 2 Harvardská architektura	14
Obr. 3 Vývojový kit M68EVB908GB60	16
Obr. 4 Význam pinů mikropočítače MC9S08GB60	17
Obr. 5 Princip propojení dvou zařízení pomocí SPI	19
Obr. 6 Blokové schéma mikropočítače MC9S08GB60 – zvýrazněné SPI	20
Obr. 7 Řídící registr SPI1C1	21
Obr. 8 Řídící registr SPI1C2	22
Obr. 9 Registr přenosové rychlosti SPI1BR	22
Obr. 10 Stavový registr SPI1S	23
Obr. 11 Datový registr SPI1D	24
Obr. 12 Vytváření projektu programem CodeWarrior – volba mikropočítače	26
Obr. 13 Volba programovacího jazyka a umístění programu programem CodeWarrior.....	26
Obr. 14 Kostra vytvořená programem CodeWarrior	27
Obr. 15 Tlačítka pro sestavení a ladění programem CodeWarrior	28
Obr. 16 Rozložení oken v debuggeru programu CodeWarrior	28
Obr. 17 Ukázka nejdůležitějších funkcí programu EAGLE	30
Obr. 18 Dvou a třímístný sedmisegmentový displej	32
Obr. 19 Maticový LED displej 8x8.....	32
Obr. 20 Alfnumerický LCD displej 16x4.....	34
Obr. 21 Maticový LED displej 5x7.....	37
Obr. 22 Rozložení pinů posuvného registru 74HC595 s časovým diagramem	38
Obr. 23 Rozložení pinů převodníků 74HC138 s funkční tabulkou	39
Obr. 24 Blokové schéma informačního LED displeje	39
Obr. 25 Schéma zobrazovacího modulu	41
Obr. 26 DPS zobrazovacího modulu	42
Obr. 27 Zobrazovací modul osazený součástkami.....	42
Obr. 28 Schéma napájecího modulu – část 1. DC/DC měnič	43
Obr. 29 Schéma napájecího modulu – část 2.	44
Obr. 30 Napájecí DPS informačního LED displeje	45
Obr. 31 Napájecí modul osazený součástkami - Bottom	46

Obr. 32 Napájecí modul osazený součástkami - Top.....	46
Obr. 33 Příklad použití funkce SetBit(2,5,1)	49
Obr. 34 Příklad použití funkce Zobraz text na reálném modulu - neinvertovaný	50
Obr. 35 Příklad použití funkce Zobraz text na reálném modulu - invertovaný	50

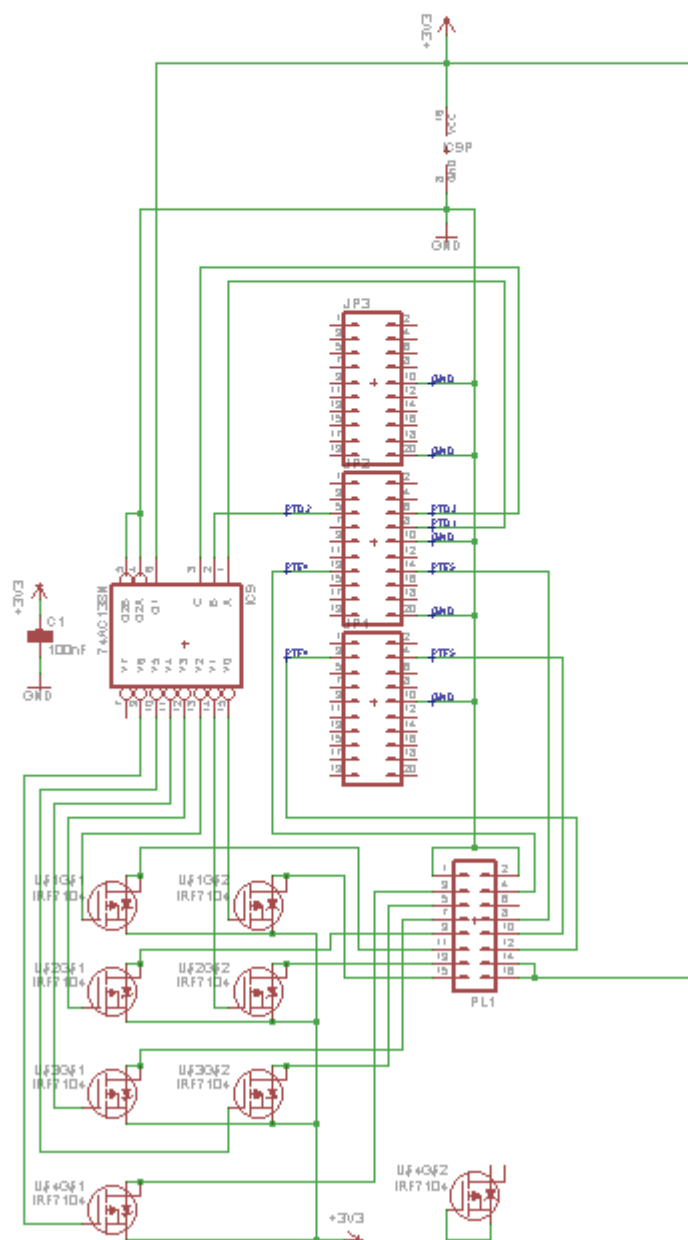
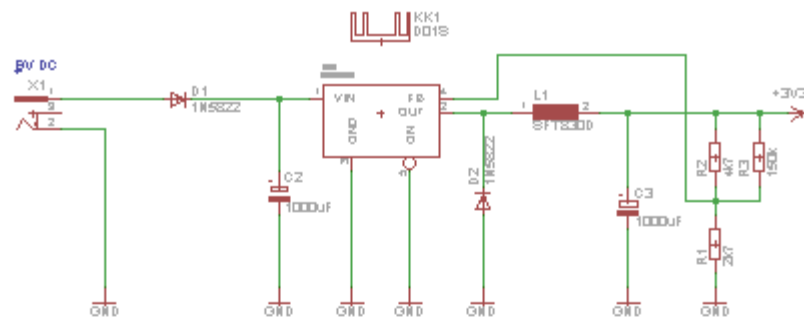
SEZNAM TABULEK

Tab. 1 Nastavení dělitele frekvence SPI	23
Tab. 2 Dělička frekvence	23
Tab. 3 Volba směru rotace	51

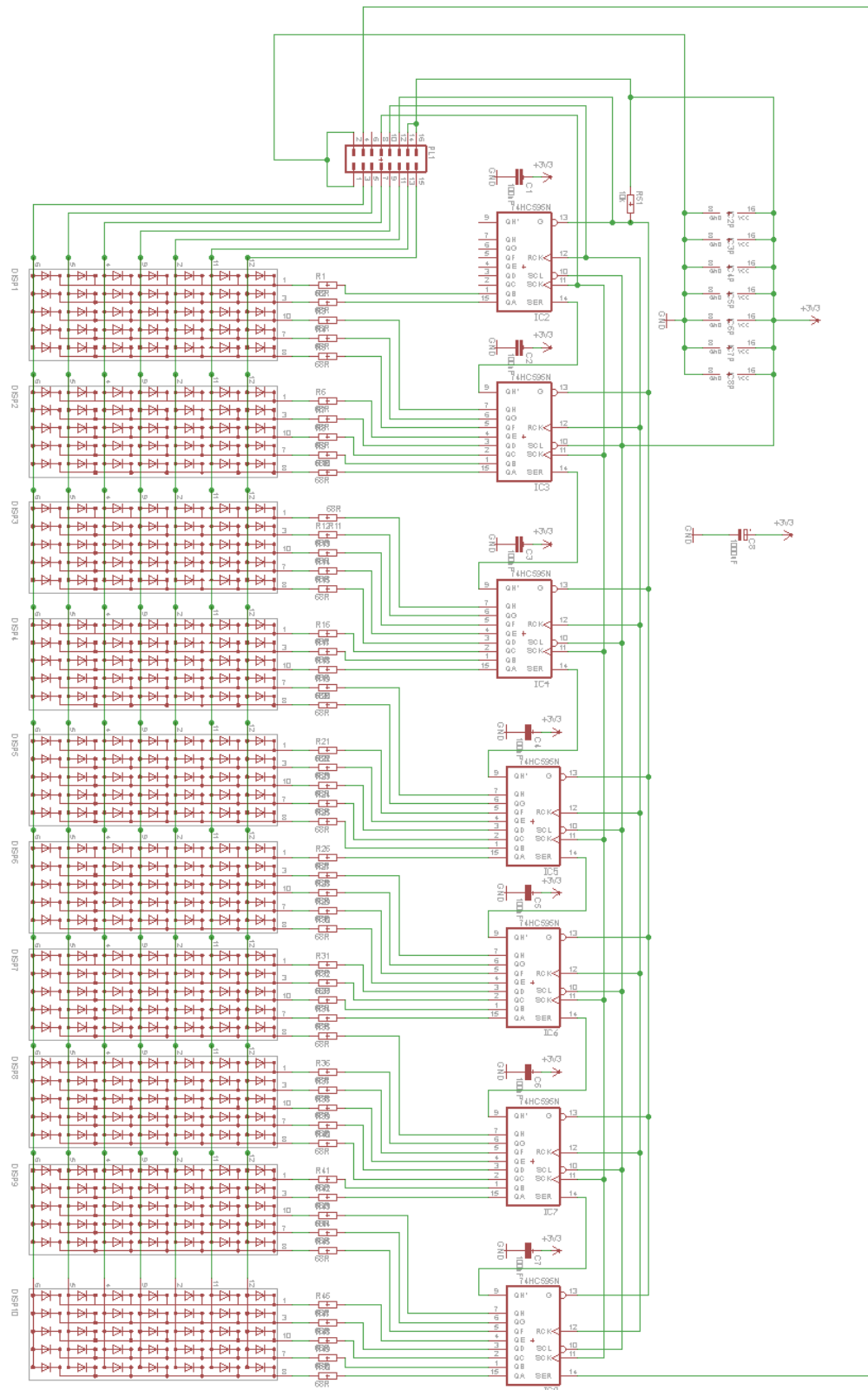
SEZNAM PŘÍLOH

- P I Schéma zapojení napájecí části modulu
- P II Schéma zapojení zobrazovací části modulu
- P III Motivy plošných spojů
- P IV Osazovací plány
- P V Rozpis použitých součástek
- P VI Verze bakalářské práce a elektronické přílohy na CD

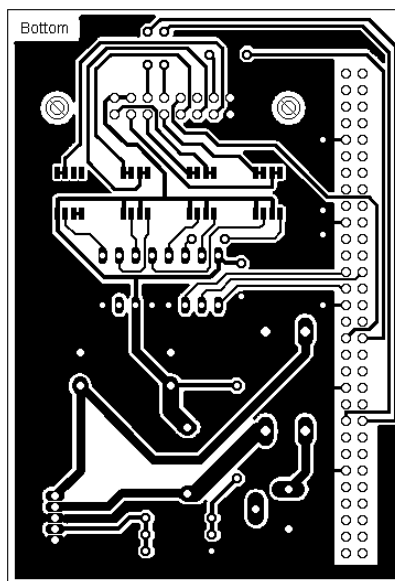
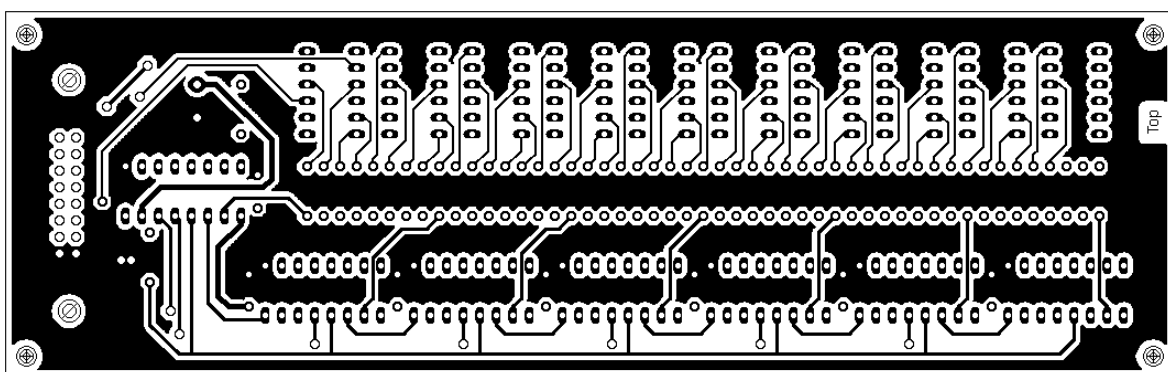
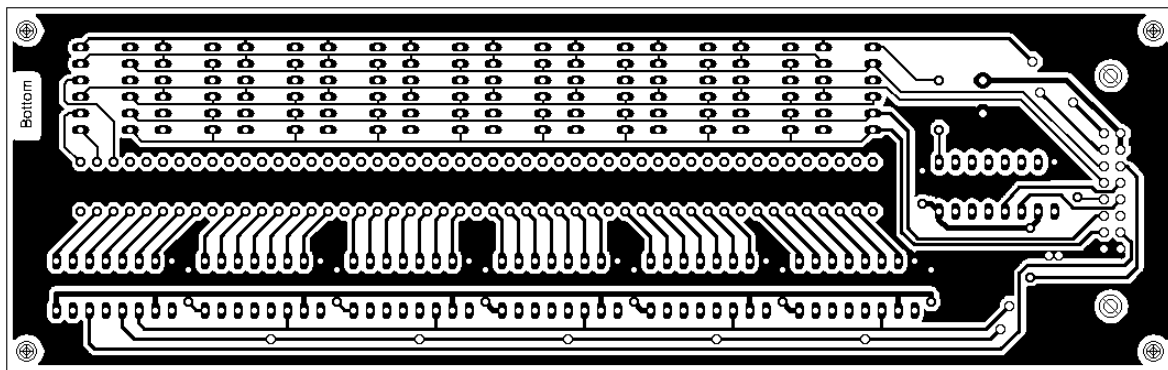
PŘÍLOHA P I: SCHÉMA ZAPOJENÍ NAPÁJECÍ ČÁSTI MODULU



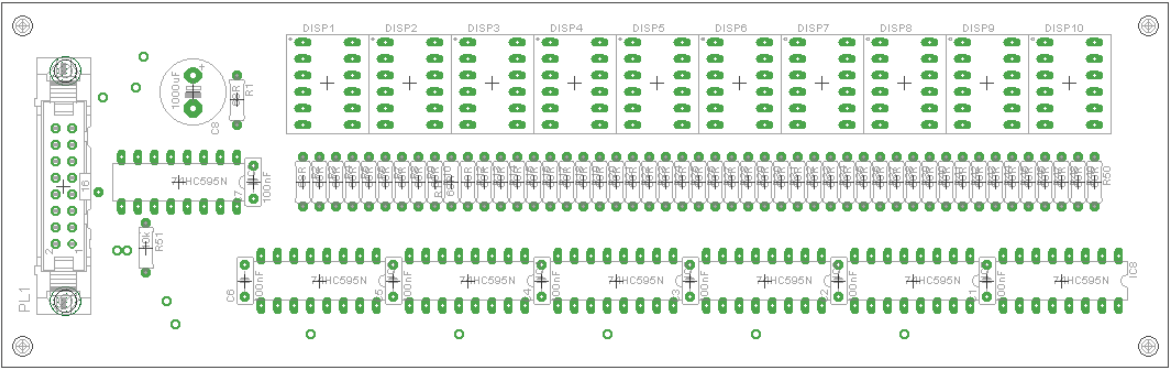
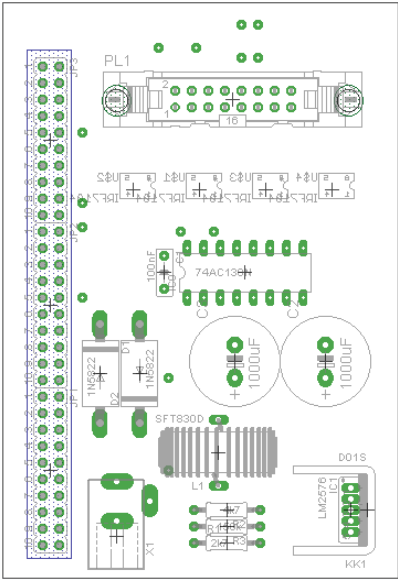
PŘÍLOHA P II: SCHÉMA ZAPOJENÍ ZOBRAZOVACÍ ČÁSTI MODULU



PŘÍLOHA P III: MOTIVY PLOŠNÝCH SPOJŮ



PŘÍLOHA P IV: OSAZOVACÍ PLÁNY



PŘÍLOHA P V: ROZPIS POUŽITÝCH SOUČÁSTEK

SEZNAM SOUČÁSTEK NAPÁJECÍ DPS			
	KS	TYP	REFERENCE
1	1	PC konektor	X1
2	2	1N5822	D1,D2
3	1	100nF	C1,C2
4	2	1000uF/16V	C2,C3
5	1	SFT830D	L1
6	1	2k7	R1
7	1	4k7	R2
8	1	150kΩ	R3
9	1	LM2576	IC1
10	1	chladič DO01S	KK1
11	1	74AC138N	IC9
12	4	IRF7104	T1-T4
13	1	PSL 16 konektor	PL1
14	1	2x30 pin header konektor	JP1-JP3

SEZNAM SOUČÁSTEK ZOBRAZOVACÍ DPS			
	KS	TYP	REFERENCE
1	2	1000uF/16V	C8
2	7	100nF	C1-C7
3	7	74HC595	IC2-IC8
4	1	10kΩ	C2,C3
5	50	68R	R1-R50
6	10	5x7 maticový displej	DISP1-DISP10
7	1	PSL 16 konektor	PL1