Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Študijný program: Počítačové systémy a siete

Tímový projekt

NÁVRH A REALIZÁCIA EXPERIMENTÁLNYCH MIKROPOČÍTAČOV

Ľuboš Rabčan
Miroslav Figura
Peter Knotka
Rado Oršula
Ján Tomko
Peter Zubčák

Tím číslo 10

12.6.2008

Obsah

1 Úvod

1.1 Účel a rozsah dokumentu

Predkladaný dokument obsahuje špecifikáciu projektu Návrh a realizácia experimentálnych mikropočítačov v rámci predmetu Tímový projekt v zimnom semestri 2007.

Cieľom projektu je (viď zadanie kapitola 2) navrhnúť, vytvoriť a oživiť experimentálny mikropočítač.

Pre rozsiahlosť projektu boli úlohy rozdelené pre jednotlivých členov tímu, podľa ich znalostí a doterajších skúseností. Úlohy členov tímu v rámci prvej fázy projektu boli špecifikované v zápise zo stretnutia dňa 22.10.2007.

Tím pracuje pod pedagogickým vedením Ing. Adriana Bagalu.

1.2 Odkazy a zdroje

[1] M. ONDROVIČ a kol.: Experimetálne mikropočítače
Bratislava, FIIT STU, 2007, Tímový projekt

[2] S. ANGELOVIČ a kol.: Experimetálne mikropočítače
Bratislava, FIIT STU, 2007, Tímový projekt

[3] Atmel Corporation - Industry Leader in the Design and Manufacture of Advanced Semiconductors
http://www.atmel.com/

[4] Welcome to Intel
http://www.intel.com/

[5] Motorola
http://www.motorola.com/

[6] http://www.research.ibm.com/journal/rd/414/huott.html

[7] FT232BL
http://www.ftdichip.com/Products/FT232BM.htm http://hw.cz/teorie-praxe/art1945-teplomer-pro-usb.html

[8] MAX232
http://www.dhservis.cz/dalsi/prevodnik.htm

http://programujte.com/index.php?akce=clanek&cl=2006110301-Komunikace-po-RS232-komunikace-po-rs232

[9] USB

http://www.mcu.cz/modules/news/article.php?storyid=281

http://cs.wikipedia.org/wiki/Universal_Serial_Bus http://hw.cz/Teorie-a-praxe/Dokumentace/ART327-USB---Universal-Serial-Bus---Popis-rozhrani.html

[10] RS rozhrania, IR rozhranie, prevodníky

http://rs232.hw.cz/

http://programujte.com/index.php?akce=clanek&cl=2006111804-Hardware-5:-Komunikacne-rozhrania-hardware-5-komunikacne-rozhrania

[11] RS485 a RS422

http://hw.cz/docs/rs485/poucha.html

[12] Implementácia USB rozhrania do mikrokontroléra

http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20(AVR).htm

2 Zadanie projektu

Navrhnite a zrealizujte dva experimentálne mikropočítače a program pre hostiteľský počítač.

Riešenie každého mikropočítača musí spĺňať nasledovné požiadavky:

- dve sériové linky
- ďalšie definované vstupné a výstupné zariadenia a indikačné prvky
- základné programové vybavenie (monitor), umožňujúce demonštrovať funkčnosť mikropočítača a ladenie aplikačných programov

Požiadavky na monitor:

- znakovo orientovaná komunikácia s hostiteľským počítačom
- otestovanie funkčnosti mikropočítača
- práca s registrami
- práca s pamäťou
- načítanie vykonateľného programu v definovanom formáte
- nastavenie / zrušenie bodov prerušenia
- spustenie / zastavenie vykonávania programu
- ďalšie špecifické funkcie pre daný typ mikroprocesora

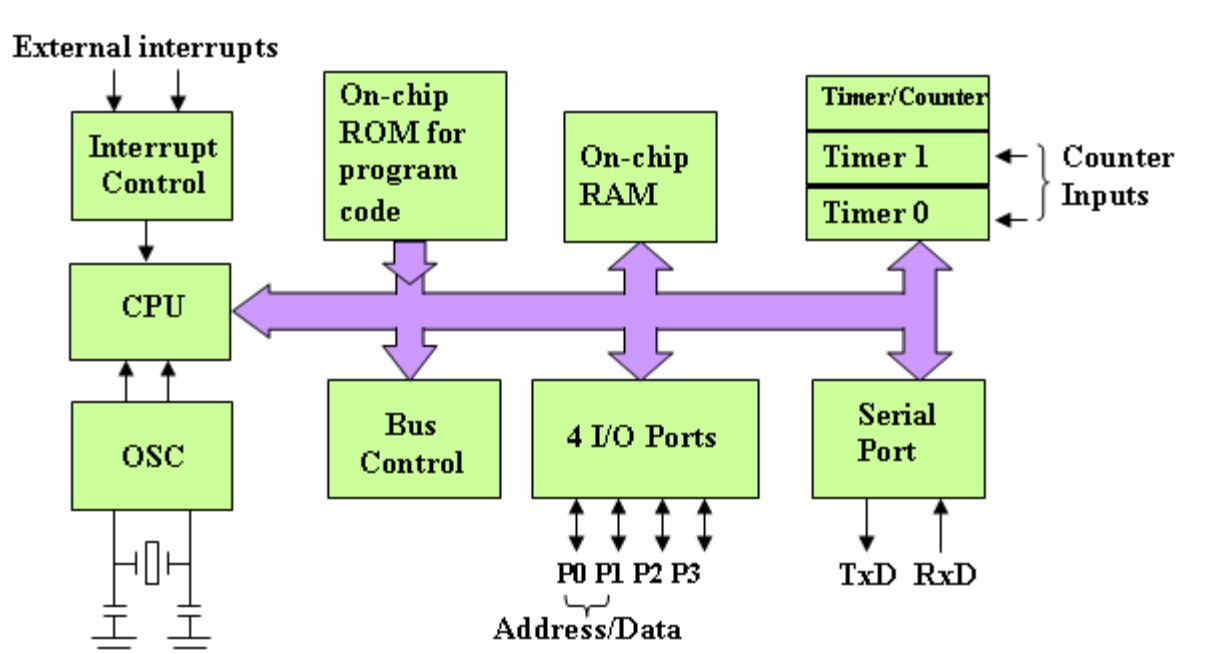
Požiadavky na program pre hostiteľský počítač:

- grafické používateľské rozhranie pre OS Windows alebo Linux
- univerzálny program pre prácu s obidvomi mikropočítačmi
- výber rozhrania, cez ktoré bude komunikácia realizovaná
- terminál na znakovo orientovanú komunikáciu s pripojeným mikropočítačom
- kompilácia zdrojového programu

3 Analýza a špecifikácia riešenia

3.1 Mikropočítač

Cieľom tejto kapitoly je preštudovanie jestvujúcich riešení a štandardných postupov, preštudovanie mikroprocesorov a z nich výber 2 najvhodnejších, ktoré použijeme na realizáciu mikropočítačov, daných špecifikáciou.



Obrázok 1. Všeobecná bloková schéma mikropočítača.

3.1.1 Mikropočítač na báze mikroprocesora 8086

Tím ZHE v rámci tímového projektu 2006/07 [1] navrhol a zrealizoval experimentálny mikropočítač na báze procesora 8086. Tento 16 bitový mikroprocesor s multiplexovanou zbernicou má 20 bitovú adresnú zbernicu a 16 bitovú dátovú zbernicu. K tomuto mikroprocesoru pripojili dve EPROM pamäte, každá o veľkosti 64 kB (obvody 27512). Ako pamäť programu použili dve zapisovateľné pamäti RWM, každá o veľkosti 32 kB (obvody 62256). Mikropočítač ďalej disponuje dvoma sériovými linkami a síce obvodom RS 232 a infračerveným portom, vstupným portom so spínačmi, LED diódami a šiestimi 7 segmentovými displejmi. Mikropočítač sa im bohužiaľ nepodarilo oživiť, keďže vznikli nemalé problémy pri realizácií, no napriek tomu je projekt veľmi pekne spravený, je na vysokej technickej úrovni a ak by sa na ňom znova trochu popracovalo, mikropočítač by sa podarilo oživiť.

3.1.2 Mikropočítač na báze mikroprocesora 8086 – MPS86

S týmto mikropočítačom sme sa stretli a pracovali s ním na cvičeniach z predmetu Mikropočítače. Jadro mikropočítača tvorí mikroprocesor Intel 8086, ktorý má pamäť programu o veľkosti 64 kB, pamäť údajov o veľkosti 4 kB, 3 externé čítače / časovače, šesť 8 bitových vstupno-výstupných portov a prerušovací podsystém. Tým, že počítač je pekne a prehľadne zrealizovaný a navyše má možnosť krokovať po strojových cykloch, je výbornou pomôckou pre študentov.

3.1.3 Experimentálny mikropočítač EMP386EX

Tento mikropočítač v rámci záverečného projektu navrhol a zrealizoval Viktor Tlacháč. Mikroprocesor Intel80386EX obsahuje veľa vnorených prvkov a tým pádom je mikroprocesor použitý ako mikropočítač. Pamäť programu typu ROM má 512 kB, pamäť údajov typu RWM má 64 kB. Mikropočítač ďalej obsahuje šesť miestny 7 segmentový displej, maticový LCD displej, 2 sériové rozhrania RS 232 a konektor PC/104, ktorý dáva možnosť rozšíriť mikropočítač o ďalšie periféria (napríklad myš, klávesnica). Mikropočítač je taktiež možné krokovať.

3.1.4 Mikropočítač na báze 8051

Už vyššie spomínaný tím ZHE [1] taktiež navrhol a zrealizoval aj mikropočítač na báze mikroprocesora 8051. 8051 je 8-bitový CISC procesor s harwardskou architektúrou. V základnej verzií, vnútorná pamäť údajov má kapacitu 128 B. Šírka údajovej zbernice je 8 bitov, šírka adresnej zbernice je 15 bitov. Spodných 8 bitov je multiplexovaných na adresu a údaje. V základnej verzii ďalej obsahuje dve počítadlá, možnosť obslúžiť dve externé prerušenia a má jeden sériový port. Tím použil mikroprocesor AT89S8253, ktorý obsahuje navyše WatchDog Timer a komunikačné rozhranie SPI. Pamäť programu typu EEPROM má veľkosť 8 kB, pamäť údajov typu RWM má taktiež 8 kB.

3.1.5 Mikropočítač na báze AVR

Tím Stanislava Angeloviča [2] použil vo svojom tímovom projekte mikroprocesor rodiny AVR, ktoré vyrába firma ATMEL. Jedná sa o RISC procesory s Harwardskou architektúrou. AVR ponúka tri typy mikroprocesorov: AVRtiny (menší počet pinov a menej prvkov než ostatné typy), AVR (klasické AVR) a AVRmega (viac pridaných prvkov a väčšia

zložitosť procesora). Tím sa rozhodol pre návrh a realizáciu mikropočítača s použitím mikroprocesora typu AVRmega a síce ATmega8515. ATmega8515 je 8-bitový procesor obsahujúci 8 kB Flash pamäte, 512 B pamäte EEPROM a 512 B pamäte SRAM. Má vstavaný oscilátor, teda nie je potrebné pripojiť k nemu aj kryštál a disponuje aj obvodom na resetovanie. Mikropočítač má pamäť údajov o kapacite 8kB RWM a je realizovaná obvodom 6264, sériové rozhrania typu RS232, 16C450 a USB rozhranie, šesť 7 segmentových LED displejov, maticový displej. Projekt je na vysokej technickej úrovni, mikropočítač sa im podarilo oživiť.

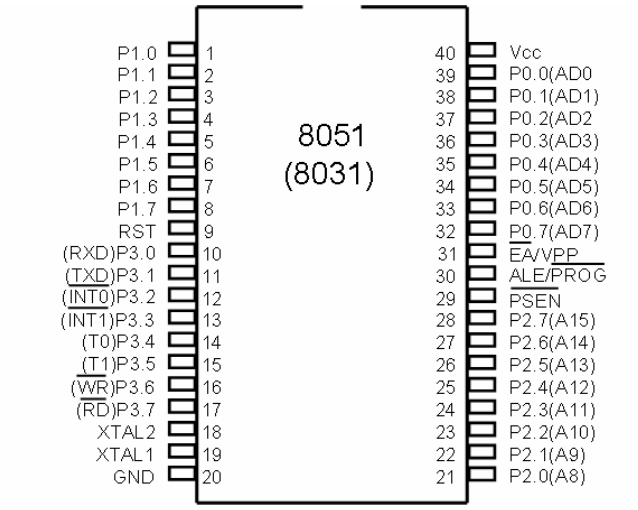
3.2 Mikroprocesor

Zanalyzovali sme nasledovné rodiny mikroprocesorov:

3.2.1 Procesor rodiny 8051 – AT89S52

Obsahuje: [3]

- 8 kB Flash pamäť
- 256 B RAM pamäť
- 32 vstupno-výstupných bitov
- Watchdog Timer
- 2 dátové ukazovatele
- Full duplex sériový port
- 3 16-bitové čítače/časovače
- 8 zdrojov prerušenia
- Sériový port
- Oscilátor

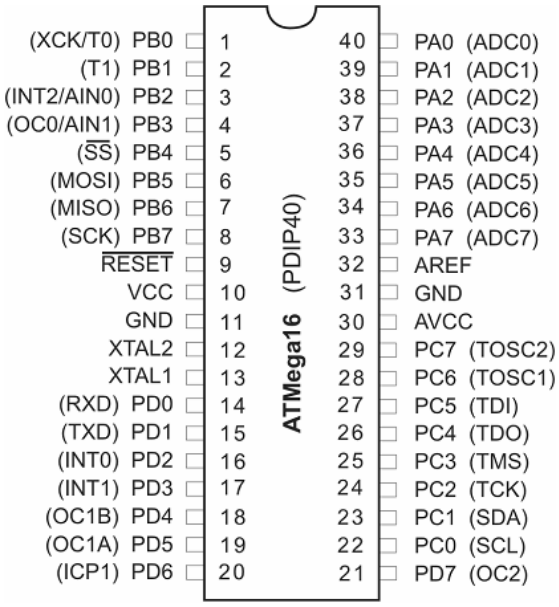


Obrázok 2. Rozloženie pinov 8051.

3.2.2 Procesor rodiny AVR – ATMega16

Obsahuje:

- 16 kB Flash
- 512 B EEPROM
- 1 kB SRAM
- 32 vstupno výstupných bitov
- JTAG debugger
- 3 čítače / časovače
- Programovateľný sériový USART
- Vonkajšie a vnútorné prerušenia
- Dve sériové rozhrania
- WatchDog Timer s interným oscilátorom
- SPI sériový port



Obrázok 3. Rozloženie pinov ATMega16.

3.2.3 Procesor rodiny 8086

Obsahuje: [4]

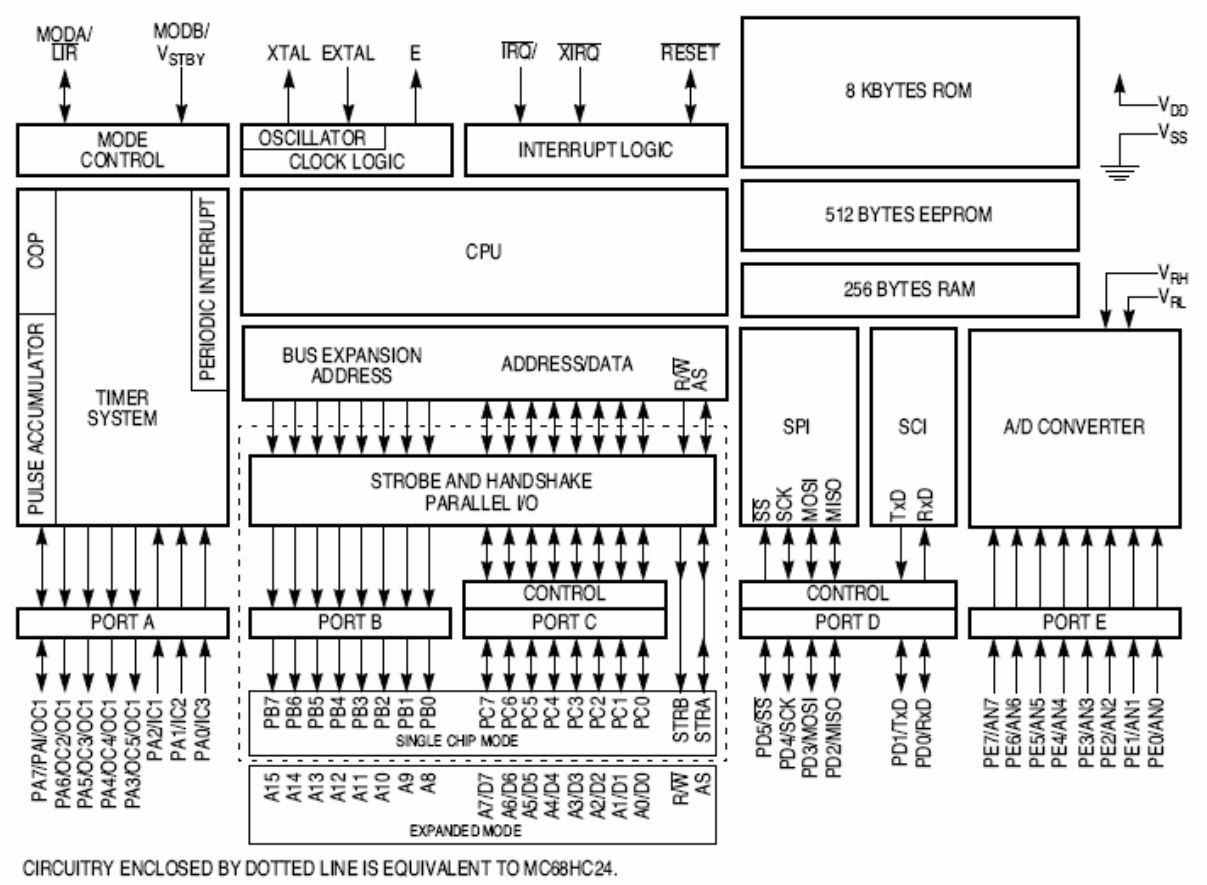
- Jadro tvorí 32 bitový procesor 80386 statická verzia
- Bus Interface Unit (jednotka spolupráce so zbernicou) – rozhranie na 16 bitovú údajovú zbernicu, adresná zbernica redukovaná z 32 na 26 bitov
- Programovateľný dekóder s 8 výstupmi, každý sa dá individuálne naprogramovať pre pamäť alebo V/V zaradenia
- Dvojkanálový obvod DMA s rozšírenou adresáciou na 26 bitov
- 3 čítače / časovače (ekvivalent 8254)
- 2 riadiace obvody prerušenia (8259 A)
- 2 asynchrónne sériové interfejsy a 1 synchrónny sériový interfejs
- Watchdog Timer
- 24 vstupno-výstupných liniek
- JTAG boundary scan

3.2.4 Procesor rodiny Motorola – M68HC11

Obsahuje: [5]

- 8 bitový procesor, 2 MHz
- 8 kB ROM pamäť

- 512 B EEPROM pamäť
- 256 B RAM pamäť
- 8 kanálový A/D prevodník
- 1 asynchrónny sériový interfejs (SCI) a 1 synchrónny sériový interfejs (SPI)
- 16 bitový časovač



Obrázok 4. Bloková schéma Motorola M68HC11

3.3 Prerušovací podsystém

Prerušenie (Interrupt - INT) je udalosť vyvolaná vonkajším impulzom alebo zodpovedajúcou programovou inštrukciou, na základe ktorej procesor preruší práve vykonávaný program a zavolá rutinu určenú príslušným vektorom prerušenia.

Signál prerušenia z externého zdroja sa môže k procesoru dostať cez jeden z dvoch vývodov na puzdre procesora : INTR a NMI.

- **NMI** - (Non Mascable Interrupt) **nemaskovateľné prerušenie** sa používa na signalizovanie “katastrofických “ udalostí ako je napríklad náhly pokles

napájacieho napätia alebo chyba parity operačnej pamäti či zbernice. Tento vývod procesora je teda určený pre prerušenia, ktoré nie je možné zakázať.

- **INTR** - (Mascable Interrupt) **maskovateľné prerušenie**, resp. vývod INTR tohto zakázateľného prerušenie je väčšinou budený ďalším obvodom - programovateľným kontrolérom prerušenia - (PIC - Priority Interrupt Controller) typu 8259A. Obsluhu tohto prerušenia možno programovo zakázať vynulovaním bitu IF (Interrupt Flag) v príznakovom registri procesora. Pokiaľ je IF = 0 procesor prerušenie ignoruje. Povinnosťou kontroléra PIC je udržať informáciu o aktívnej požiadavke na prerušenie do doby keď sa uvoľní bit IF = 1 a procesor môže vo vhodnom okamihu akceptovať požiadavku na prerušenie tak, že vyšle 2 zbernicové cykly v doprovode so signálom INTA (Interrrupt Asknowledge). Počas druhého z cyklov INTA uloží PIC na dátovú zbernicu číslo vektora prerušenia, na základe ktorého sa vykoná príslušná rutina z adresy podľa tabuľky vektorov prerušení. Po začatí obsluhy prerušenia je automaticky zakázaný príjem ďalšieho prerušenia. V prípade ak sa požaduje príjem ďalších vnorených prerušení možno ich prístup odblokovať programovo pomocou príznakového bitu IF.

Podľa toho, čím je prerušenie generované, rozlišujú sa na prerušenia:

- **Interné INT**, tiež nazývané softwarové **SW_INT**, ktoré sú generované programovo, synchrónne s okamihom výskytu inštrukcie SW_INT v postupnosti inštrukcii vykonávaného programu. Sú to vlastne špecializované obslužné rutiny, ktorých začiatočná adresa je prístupná pomocou tabuľky vektorov prerušení. Vyznačujú sa tým, že zo všetkých volacích inštrukcii sú najrýchlejšie. Tabuľka vektorových prerušení len zjednodušuje volanie príslušných obslužných procedúr a tiež umožňuje rôznym výrobcom konštruovať vzájomne kompatibilné počítače. Preto sú nepoužité prerušenia pre obsluhu hardware využívané ako rozhranie pre volanie služieb DOS a BIOS. K interným INT patrí podľa definície firmy Intel aj prvých 32 typov (INT 0 až INT 31), ktoré sú určené pre potrebu samotného procesora v neštandardnej situácii, napríklad INT 0 pri delení nulou, INT 1 pri

krokovaní, INT 2 ako NMI, INT3 ako jednobajtová inštrukcia pre breakpoint a INT 4 pri pretečení.

- **Externé INT**, tiež nazývané hardwarové **HW_INT**, ktoré sú generované vonkajšími udalosťami (na základe žiadosti o prerušenie IRQ), asynnchrónne voči bežiacemu programu.

Procesor 80x86 rozlišuje 256 možných prerušení. Každému prerušeniu prislúcha 32-bitová logická adresa (segment:offset). Táto adresa sa nazýva **vektor prerušenia** a ukazuje na miesto v pamäti, kde sa začína podprogram, ktorý sa vykoná pri vyvolaní daného prerušenia. Pre každý typ prerušenia je v operačnej pamäti od fyzickej adresy 0 uložená úplná logická adresa obslužného programu pre príslušné prerušenie (v poradí najprv offset pre IP potom segment pre CS), teda celkove je vyhradená oblasť 256 * 4 byte = 1000D = 400H. Táto oblasť pamäti sa označuje ako tzv. **tabuľka vektorov prerušení**.

3.3.1 Dôležité služby BIOS

- INT 00H - delenie nulou
- INT 01H - krokovanie programu
- INT 02H - NMI
- INT 03H - zastavenie programu - Break point
- INT 04H - aritmetické pretečenie
- INT 05H - kláves PrintScreen
- INT 08H - časovač
- INT 09H - klávesnica
- INT 08H - časovač
- INT 10H - VideoBIOS
- INT 13H - Diskové operácie
- INT 14H - Sériové rozhranie
- INT 15H - Rozšírenie AT BIOSu
- INT 16H - Obsluha buffera klávesnice
- INT 17H - Obsluha tlačiarne
- INT 1AH - Systémové hodiny

- INT 1BH - Kláves Ctrl - Break
- INT 1CH - Užívateľský časovač
- INT 1DH - Tabuľka videoparametrov

3.3.2 Prehľad HW zdrojov žiadostí IRQ o prerušenie INT

Typ IRQ	Adresa INT vektora	Typ INT	Význam v PC AT
IRQ0	20H	INT_8	HW časovač (18,2 Hz)
IRQ1	24H	INT_9	Klávesnica
IRQ2	28H	INT_10 (0AH)	kaskádne zapojenie 2 PIC
IRQ3	2CH	INT_11 (0BH)	COM2
IRQ4	30H	INT_12 (0CH)	COM1
IRQ5	34H	INT_13 (0DH)	LPT2
IRQ6	38H	INT_14 (0EH)	disketa
IRQ7	3CH	INT_15 ()FH)	LPT1
IRQ8	1C0H	INT_112 (70H)	kalendár
IRQ9	1C4H	INT_113 (71H)	
IRQ10	1C8H	INT_114 (72H)	
IRQ11	1CCH	INT_115 (73H)	
IRQ12	1D0H	INT_116 (74H)	
IRQ13	1D4H	INT_117 (75H)	kooprocesor
IRQ14	1D8H	INT_118 (76H)	pevný disk
IRQ15	1DCH	INT_119 (77H)	

3.3.3 Príklad výpisu obsahu začiatku pamäte, v oblasti tabuľky vektorových prerušení

(tabuľka získaná pomocou príkazu debuggera d 0:0)

Pre iný PC sa môže výpis odlišovať.

Adresa	Obsah pamäte
0000:0000	8A 10 16 01 F4 06 70 00-16 00 A0 05 F4 06 70 00
0000:0010	F4 06 70 00 54 FF 00 F0-4C E1 00 F0 6F EF 00 F0
0000:0020	5F 02 07 0C DE 01 5F D3-6F EF 00 F0 6F EF 00 F0
0000:0030	6F EF 00 F0 6F EF 00 F0-B7 00 A0 05 F4 06 70 00
0000:0040	78 16 85 CC 4D F8 00 F0-41 F8 00 F0 8A 16 85 CC
0000:0050	39 E7 00 F0 65 17 85 CC-2E E8 00 F0 FD 01 07 0C
0000:0060	CC E3 00 F0 55 17 85 CC-6E FE 00 F0 EE 06 70 00
0000:0070	53 FF 00 F0 A4 F0 00 F0-22 05 00 00 CA 53 00 C0

Z tabuľky vidno, že prerušeniu INT 8H , zodpovedá obsah pamäte 20H - 24H (32D - 36D), ktorý tvoria bajty 5F02070C => adresa : 0C07:025F (najprv offset LSB, MSB potom segment LSB, MSB adresy obslužného programu).

Poprípade tiež vidno, že prerušenia INT 0BH a INT 0CH (pre obsluhu sériových portov) začínajú na spoločnej adrese F000:EF6F H.

3.3.4 Kontrolér prerušenia

Programovateľný **obvod PIC** (Priority Interrupt Controller) tzv. kontrolér prerušenia, obyčajne realizovaný obvodom 8259A, umožňuje (v PC XT) spracovať až 8 žiadostí IRQi o prerušenie, prideľovať a meniť im prioritu, zakazovať a povoľovať jednotlivé prerušenia. Výstup z PIC ako už bolo spomenuté je pripojený k vývodu pre maskovateľné prerušenie procesora INTR. Pre programátora predstavuje ďalšiu vrstvu prerušovacieho systému, ktorá mu umožňuje selektívne zakázať niektoré externé prerušenia, pokiaľ by nepostačovali inštrukcie STI a CLI procesora. V PC XT sa používa jeden obvod 8259A, v PC AT sú zapojené dva v kaskáde, takže umožňuje až 15 rôznych žiadostí o prerušenie.

Pripojenie obvodu PIC na zbernicu počítača je podobné (obr. 80) ako pripojenie podobných programovateľných obvodov typu PPI-8255, PIT-8253 a pod. S obvodom prvého Master PIC možno komunikovať cez porty 20H, 21H a s obvodom druhého Slave PIC cez porty A0H, A1H.

Žiadosti o prerušenie IRQi (obr. 81) nevyvolávajú priamo prerušenie na vývode INTR procesora, ale môžu byť v PIC-8259A rôznym spôsobom blokované (maskované) a vyhodnocované podľa naprogramovanej priority. V PC sa používa spôsob priradenia, keď IRQ_0H (HW časovač 18,2Hz) má najvyššiu prioritu a volá vektorovú adresu INT_8H, klávesnicový IRQ_1H vyvoláva prerušenie INT_9H, asynchrónna komunikácia cez COM2 (resp.COM1) sa uskutočňuje s pomocou IRQ_3 (resp.IRQ_4) a vyvoláva INT_0BH (resp. INT_0CH) a pod. Niektoré požiadavky IRQ sú voľné a poskytujú tak možnosť pre obsluhu ďalších vonkajších zariadení. Požiadavka IRQ_02H slúži na kaskádové pripojenie druhého obvodu PIC.

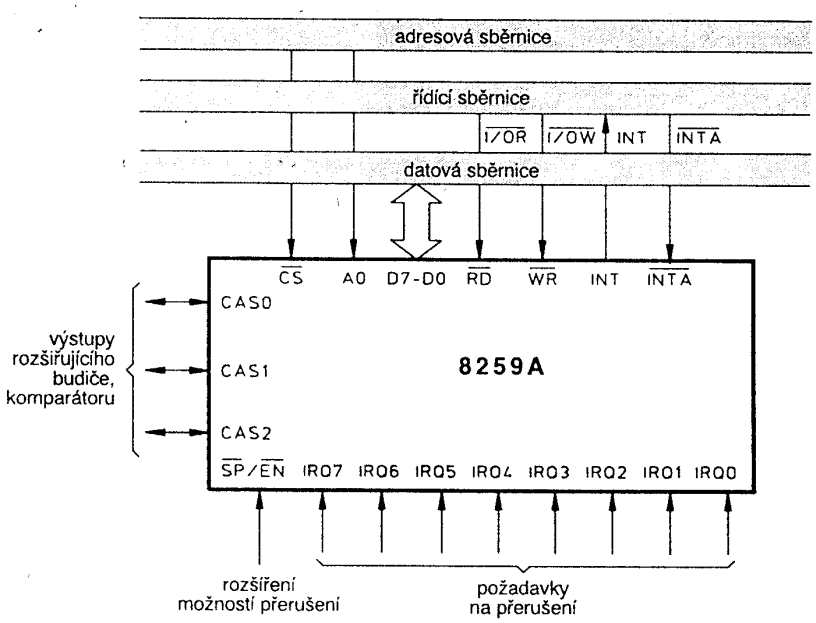
Činnosť obvodu PIC sa ovplyvňuje programovaním jeho registra masiek prerušení IMR, registra požiadaviek na prerušenie IRR a stavového registra obsluhy prerušení ISR.

- **IMR - register masiek požiadaviek** IRQ (Interrupt mask register) sa používa na maskovanie, t.j. povolenie alebo zákaz jednotlivých požiadaviek na prerušenie

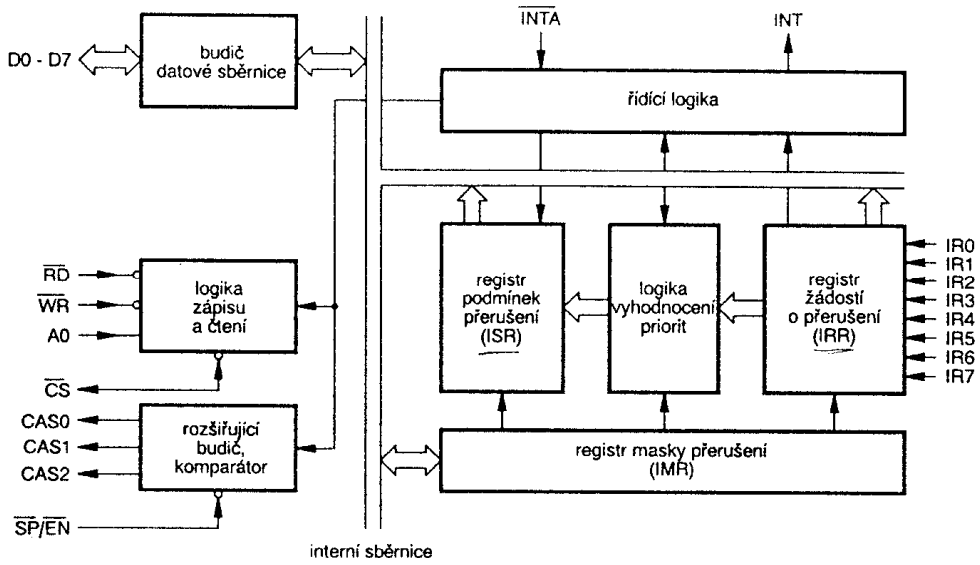
IRQi. Hodnota 1 na príslušnej pozícii slova OCW1 blokuje príslušný vstup IRQi a hodnota 0 ho povoľuje.

- **IRR - register požiadaviek** IRQ(Interrupt request register). Požiadavky na prerušenie IRQi sa privádzajú cez vstupy IR0 až IR7 do registra IRR (IR0 má najvyššiu prioritu, IR7 najnižšiu). Až po vhodnom odmaskovani a pri vhodnej priorite požiadavky môže sa stať táto požiadavka podnetom na vyvolanie prerušenia - INT.
- Priority coder (resolver) - **logika vyhodnocovania priorít** porovnáva obsah registrov IMR a IRR a vyberá žiadosť IRQi s najvyššou prioritou a zavedie ju do stavového registra požiadaviek na obsluhu prerušení ISR.
- **ISR - stavový register požiadaviek** na obsluhu prerušení (In service register). Až obsah tohto registra vyhodnocuje mikroprocesor ako žiadosť o INT. Jeho príslušný bit indikuje, či je zodpovedajúce prerušenie obsluhované alebo nie. (Hodnota 0 oznamuje, že nie je práve obsluhované).

Inicializácia registrov IRR, IMR a ISR obvodu 8259A po zapnutí počítača sa vykonáva zápisom inicializačných slov ICW. V PC je štandardná inicializácia záležitosťou BIOSu.



Obrázok 5. Kontrolér prerušenia PIC 8259A v systéme s PC.



Obrázok 6. Vnútorná štruktúra kontroléra prerušenia PIC 8237A.

3.3.5 Obsluha prerušenia

Po vzniku prerušovacej udalosti prebehne určitá postupnosť akcii, cieľom ktorých je poskytnúť prvú inštrukciu obslužného programu, adekvátneho prerušovacej udalosti. Aby táto postupnosť mohla prebehnúť je potrebné prerušovací podsystém naprogramovať.

3.3.5.1 Inicializačná časť a nainštalovanie rezidentného obslužného programu

Program k obsluhe prerušenia - ISR (Interrupt service routine) od vonkajšieho zariadenia treba trvale t.j. rezidentne uložiť do pamäti tak, aby ho bolo možné vyvolať aj počas behu iných programov. Užívateľské programy sú totiž tranzientné - zavedú sa do operačnej pamäti, vykonajú sa a uvoľnia pamäť.

- Program k obsluhe prerušenia ISR pozostáva z inštalačnej a reinštalačnej časti a z vlastného tela rezidentne uloženého programu. Základom inštalačnej časti je uloženie kódu obslužnej procedúry ISR (vlastného tela rezidentne uloženého programu) so špeciálnym **zakončením tejto procedúry - TSR** (terminate and stay resident - ukončenie s rezidentným zostatkom), vďaka ktorej zostane kód procedúry uložený v rezervovanej rezidentnej pamäti a bude tak k dispozícii vždy v prípade potreby obsluhy tohto prerušenia.
- Pred ukončením inštalačnej časti programu je potrebné ešte **uložiť do tabuľky vektorov INT adresu** začiatku nového obslužného programu, ktorý zostane rezidentne uložený v pamäti.

3.3.5.2 Realizácia tela programovej obsluhy prerušenia.

Vlastný obslužný program okrem úschovy obsahov požívaných registrov (PUSH - POP + uloženie starej adresy pôvodného obslužného programu ISR - napríklad pre potreby deinštalácie), ukončenia procedúry obsluhy pomocou IRET (IRET= RET + POPF) može byť:

- Naviazaný na činnosť určitej obslužnej rutiny, spravidla BIOS alebo DOS (**modifikuje sa pôvodný obslužný program** a nová obsluha je zaradená na začiatok alebo na koniec "reťazca" pôvodných obslužných rutín).
- Tvorí samostatný obslužný program s originálnym vektorom prerušenia, napríklad pre určité vonkajšie zariadenie, ktoré má svoju individuálnu požiadavku IRQ, realizovanú cez spoj k obvodu PIC-8259A , napríklad ako tlačítko cez príslušný kolík na konektory zbernice, umožňujúce po stlačení vyvolať žiadosť o prerušenie IRQi. Predpokladom úspešnej funkcie (okrem povolenia prerušení od vonkajších zariadení - IF pomocou inštrukcie STI) je tiež správne nainicializovanie kontroléra PIC 8259A (inicializácia cez BIOS, vhodná maska pomocou OCW1 do registra IMR). Po skončení obslužnej procedúry , vyvolanej vonkajším prerušením treba oznámiť kontroléru PIC 8259A koniec aktivity obsluhy pomocou slova OCW2, buď ako nešpecifikovaný koniec EOI (rovnaký pre všetky IRQ a zakončujúci naraz všetky žiadosti IRQ) alebo ako koniec len vybranej žiadosti IRQ_i, napríklad v prípade vnoreného prerušenia.

3.3.6 Postup činností pri obsluhe prerušenia

Prvou udalosťou je **vznik požiadavky IRQ** (Interrupt Request) periférneho zariadenia na určitú činnosť, ktorá sa má realizovať pomocou prerušenia INT aktuálne bežiaceho programu.

- Signál tejto požiadavky (zapamätaný v preklápacom obvode adaptéra periférie) je pripojený na jeden zo vstupov IRQi kontroléra prerušenia PIC. Ak je náhodou na vstupe PIC súčasný výskyt požiadaviek na INT od viacerých periférii, tak PIC vyhodnotí, **ktorá z požiadaviek IRQI má najvyššiu prioritu** a pre túto vybranú požiadavku aktivuje do mikroprocesora **signál INTR**.

- Ďalší postup záleží na tom, či je prerušenie mikroprocesora programovo povolené. V kladnom prípade sa dokončí rozpracovaná inštrukcia a mikroprocesor začne vysielať **potvrdzovací signál INTA** (Interrupt Acknowledge). V zápornom prípade pokračuje mikroprocesor nerušene vo svojej predtým zahájenej činnosti, ako keby signál INTR neprišiel. Situáciu môže zmeniť len inštrukcia STI (povolenie prerušenia v registri flagov mikroprocesora).
- Potvrdzovacie impulzy INTA vysiela mikroprocesor vždy dva. Na základe prijatia prvého rozhodne kontrolér PIC o najdôležitejšej požiadavke na prerušenie IRQ a po prijatí druhého mikroprocesor vyšle späť k PIC po dátovej zbernici osembitovú hodnotu tzv. **typu prerušenia (INT_vector)**. Teoreticky sú možné všetky kombinácie z intervalu <0, 255>, ale v IBM PC sú BIOSom pre obslužné programy najbežnejších periférii naprogramované typy prerušení INT 8 - INT 15 a INT 112 - INT119.
- Typ prerušenia INT_vector prepočíta mikroprocesor na **ukazovateľ do tabuľky prerušení** - 4*INT_Vector a po adresovej zbernici **vyšle adresu** začiatku obslužného programu prerušenia ISR (4 bajty - v poradí najprv offset pre IP potom segment pre CS).
- Po **uložení adresy nového ISR do registrov CS a IP sa začne vykonávať** postupnosť inštrukcii z programu ISR. ISR by mal uschovať do zásobníka obsahy všetkých registrov, s ktorými bude pracovať, aby nedošlo po návrate z ISR k stratám pôvodných hodnôt. Pred návratom (inštrukcia IRET) ich procedúra ISR zase obnoví.
- Na záver procedúry ISR by sa mala **vynulovať pôvodná žiadosť o prerušenie** v technických prostriedkoch adaptéra periférie. Procedúra ISR musí tiež oznámiť kontroléru PIC **koniec aktivity - EOI** (End of Interrupt). Zaslanie príkazu EOI sa vykoná pomocou zaslania príkazu OCW2 pre register ISR kontroléra PIC typ 8259A .
- Prechodom na vykonávanie obslužnej procedúry ISR sa automaticky zakázali ďalšie prerušenia. Zmeniť tento stav možno dvojako:

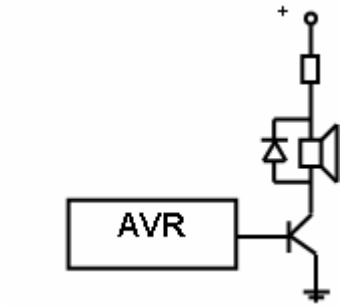
- (pre prípad viacúrovňového prerušenia v priebehu procedúry ISR) použitím inštrukcie STI v procedúre ISR;
- automaticky po vykonaní návratovej inštrukcie IRET sa tiež obnoví možnosť vyvolania prerušenia v následnom priebehu užívateľského programu.

3.4 Periféria

3.4.1 Reproduktor

Reproduktor je elektromechanické zariadenie, ktoré mení elektrický signál na mechanický pohyb, ktorý spôsobuje akustické vlnenie. Nazýva sa teda elektromechanický a mechanickoakustický menič, avšak zaužívané je elektroakustický menič. Reproduktor využíva spätný princíp elektromagnetu.

V širšom zmysle sa ako reproduktor označuje sústava (ozvučnica), ktorá pozostáva z elektroakustického meniča spojeného s akustickým zariadením, na zosilnenie a / alebo usmernenie zvuku.



Obrázok 7. Schéma zapojenia reproduktora.

3.4.2 Sériový port 8051

SCON – riadiaci register sériového portu

SBUF – buffer sériového prenosu

3.4.2.1 Štruktúra registra SCON

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

- SM0 - Bit výberu režimu
- SM1 - Bit výberu režimu
- SM2 - Povoľuje multiprocesorovú komunikáciu v móde 2 a 3.SM2=1 =>v móde 2 a 3 sa neaktivuje bit RI, keď má 9 prijatý bit hodnotu 0, SM2=1 => v móde 1 sa RI neaktivuje, ak nebol prijatý platný stop bit. V režime 0 má biť SM2=0.

- REN - Povolenie sériového prijímu, REN=1 => povolený sériový príjem, REN=0 => sériový príjem je zakázaný
- TB8 - 9. bit pri vysielaní v režime 2 a 3.Nastavuje sa softwarovo.
- RB8 - 9. bit pri príjme. Prijíma sa v režime 2 a 3. V režime 1 obsahuje prijatý stop bit.
- TI - Príznak prerušenia pri vysielaní. V režime 0 sa nastavuje hardwarovo na konci doby 8. bitu. V ostatných režimoch sa nastavuje na začiatku stop bitu. Musí sa nulovať softwarovo.
- RI - Príznak prerušenia pri príjme. V režime 0 sa nastavuje hardwarovo na konci doby 8. bitu. V ostatných režimoch sa nastavuje v strede stop bitu. Musí sa nulovať softwarovo.

3.4.2.2 Režimy sériového portu UART

SM0	SM1	Režim	Popis režimu	Prenosová rýchlosť
0	0	0	Posuvný register Vývod TxD - hodinový signál Vývod RxD – dáta	Je pevná BaudRate = Fosc/12
0	1	1	8 bitový UART Vývod TxD - vysielanie Vývod RxD - príjem	Premenná BaudRate = (K*Fosc) / (384*(256-TH1)) => TH1 = 256-((K*Fosc) / (384*BaudRate)) **!! TH1 musí byť celé číslo !!**. Ak pre danú rýchlosť nevychádza číslo celé je nutné použiť iný kryštál TH1 je horný bajt časovača 1. Časovač 1 sa používa na synchronizáciu prenosu. Musí byť v režime 2 K=1, ak SMOD=0 K=2, ak SMOD=1
1	0	2	9 bitový UART Vývod TxD - vysielanie Vývod RxD - príjem	Je pevná BaudRate = Fosc/64, ak SMOD=0 BaudRate = Fosc/32, ak SMOD=1
1	1	3	9 bitový UART Vývod TxD - vysielanie Vývod RxD - príjem	Premenná To isté ako v režime 1

3.4.2.3 Štruktúra registra PCON

Nie je bitovo adresovateľný. Adresa je 87H.

SMOD	-	-	-	GF1	GF0	PD	IDL

- SMOD - Dvojitá prenosová rýchlosť sériového portu.
- SMOD=1 => prenosová rýchlosť je dvojnásobná
- SMOD=0 => prenosová rýchlosť je ako nastavená

3.4.3 Displej

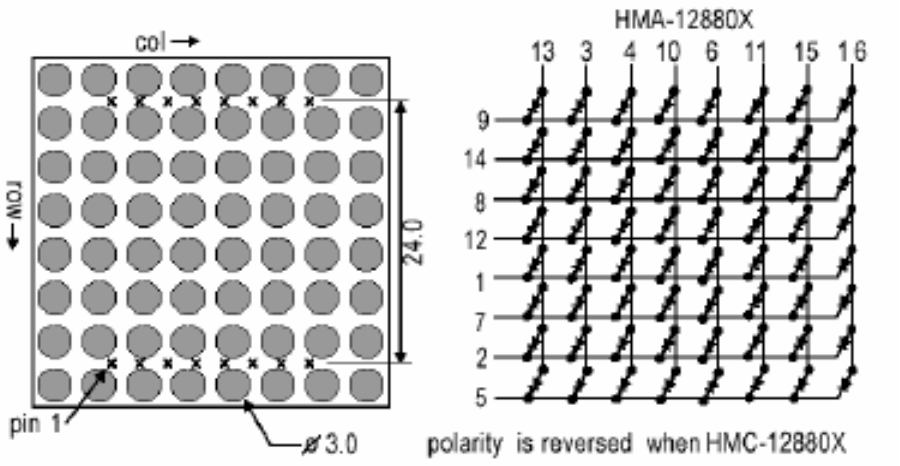
3.4.3.1 LED displeje

- 7 segmentové
- alfanumerické
- maticové

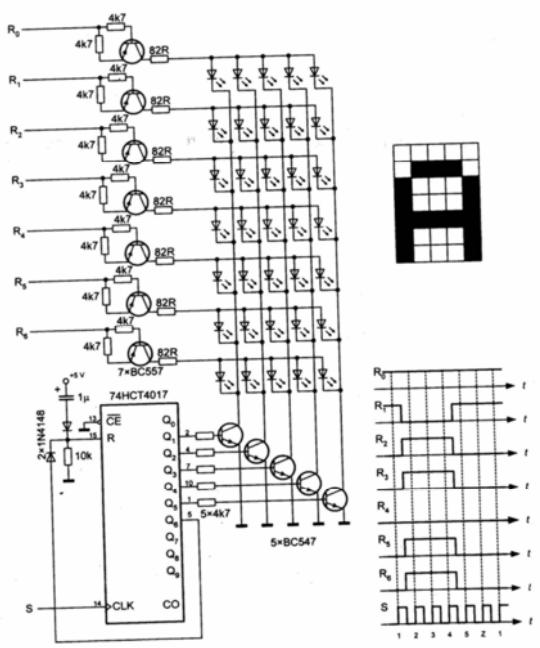
3.4.3.2 LCD displeje

- 7 segmentové
- znakové/textové
- grafické

Počet registrov je závislí od veľkosti maticového displeja, na naše potreby bude postačovať matica veľkosti 5x7, 5x8 alebo 8x8.



Obrázok 8. Maticový LED displej a jeho vnútorné zapojenie.



Obrázok 9. Schéma zapojenia maticového displeja.

3.5 Rozhrania

3.5.1 RS232

RS232 je sériové rozhranie pre prenos informácií vytvorené pôvodne pre komunikáciu dvoch zariadení do vzdialenosti 20 m. Pre väčšiu odolnosť proti rušeniu je informácia po prepojovacích vodičoch prenášaná väčším napätím, ako je štandardných 5 V. Prenos informácií prebieha asynchrónne, pomocou pevne nastavenej prenosovej rýchlosti a synchronizácii zostupnou hranou štartovacieho impulzu. Sériový port je známy aj ako **UART** (Universal Asynchronus Reciever/Transmiter), ktoré obsluhujú sériový port a zabezpečujú konverziu z paralelných údajov na sériové. Štandardná dosahovaná rýchlosť pri komunikácii pomocou čipov UART je **115 kbps**.

Zapojenie pinov pre 9 pinový konektor:

PIN	NÁZOV	SMER	POPIS
1	CD	<--	Carrier Detect
2	RXD	<--	Receive Data▫
3	TXD	-->	Transmit Data▫
4	DTR	-->	Data Terminal Ready▫
5	GND	---	System Ground▫
6	DSR	<--	Data Set Ready▫
7	RTS	-->	Request to Send▫
8	CTS	<--	Clear to Send▫
9	RI	<--	Ring Indicator

Popis signálov:

DCD - Data Carrier Detect	Detekcia nosnej. Modem oznamuje terminálu, že na telefónnej linke detekoval nosný kmitočet.
RXD - Receive Data	Tok dát z modemu (DCE) do terminálu (DTE).
TXD - Transmit Data	Tok dát z terminálu (DTE) do modemu (DCE).
DTR - Data Terminal Ready	Terminál týmto signálom oznamuje modemu, že je pripravený komunikovať.
SGND - Signal Ground	Signálová zem
DSR - Data Set Ready	Modem týmto signálom oznamuje terminálu, že je pripravený komunikovať.
RTS - Request to Send	Terminál týmto signálom oznamuje modemu, že komunikačná cesta je voľná.
CTS - Clear to Send	Modem týmto signálom oznamuje terminálu, že komunikačná cesta je voľná.
RI - Ring Indicator	Indikátor zvonenia. Modem oznamuje terminálu, že na telefónnej linke detekoval signál zvonenia.

RS 232 používa dve napäťové úrovne, logickú 1 a logickú 0. Log. 1 je indikovaná zápornou úrovňou, logická 0 je prenášaná kladnou úrovňou výstupných vodičov.

Povolené napäťové úrovne sú uvedené v tabuľke.

Dátové signály		
Úroveň	Vysielač	Prijímač
Log. 0	+5 V to +15 V	+3 V to +25 V
Log. 1	-5 V to -15 V	-3 V to -25 V
Nedefinovaný	-3 V to +3 V	

Riadiace signály		
Signál	Driver	Terminátor
"Off"	-5 V to -15 V	-3 V to -25 V
"On"	5 V to 15 V	3 V to 25 V

3.5.2 USB

Rozhranie USB predstavuje najnovší štandard pre sériový prenos dát, ktorý behom posledných pár rokov pomaly vytláča klasické sériové porty počítača založené na RS-232. Oproti RS-232 má celú radu výhod:

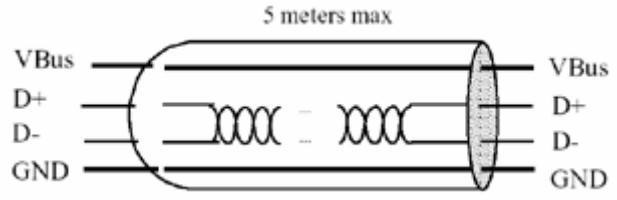
- vyššia rýchlosť (1,5 Mbit/s do 480 Mbit/s podľa špecifikácie),
- možnosť pripojenia až 127 zariadení na jediný koreňový rozbočovač
- 5V napájanie pre pripojenie zariadení umožňujúce odber až 0,5 A.
- USB zaisťuje správne pridelenie prostriedkov (IRQ, DMA).

V súčasnej dobe sa používajú dve verzie USB rozhraní líšiacich sa maximálnou prenosovou rýchlosťou:

- USB1.0, resp. 1.1 - 12 Mbit/s (Full Speed) a 1,5 Mbit/s (Low Speed)
- USB2.0 - 480 Mbit/s (High Speed)

Jednotlivé štandardy sú vzájomne kompatibilné. Prenosová rýchlosť pochopiteľne odpovedá pomalšiemu radiču. Uvedené štandardy sa od seba líšia prevedením kábla, elektrickými parametrami rozhraním pripojeného zariadenia.

Napäťové úrovne sú približne TTL - logická nula je 0,3 V so zaťažením do 1,5 kΩ proti napájaniu, logická jednotka 2,8 V so zaťažením do 15 kΩ proti zemi. Pre využitie maximálnej prenosovej rýchlosti (12 Mbps) môže byť kábel dlhý max. 5 metrov, pričom musí byť tienený a krútený, pre nízko-rýchlostné prenosy (do 1,5 Mbps) môže byť použitý netienený a nekrútený kábel s max. dĺžkou 3 metre.



Obrázok 10. Zapojenie USB

Pin	Meno	Farba	Popis
1	VBus	Red	+5 VDC
2	D-	White	Data -
3	D+	Green	Data +
4	GND	Black	Ground

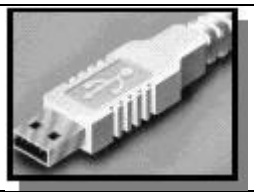
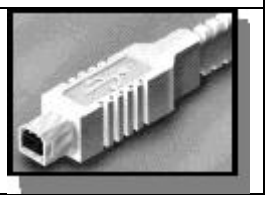
3.5.2.1 Vybrané softwarové a systémové vlastnosti USB

- Auto - identifikácia periférií, automatické mapovanie funkcie a konfigurácie ovládača
- Dynamicky prepojiteľné a prekonfigurovateľné periférie
- Pripojenie periférií požadujúcich prenosové pásmo od niekoľko kb/s do niekoľko Mb/s
- Podpora synchrónnych i asynchrónnych prenosov
- Súčasná funkcia viac zariadení pripojených na spoločnú USB zbernicu

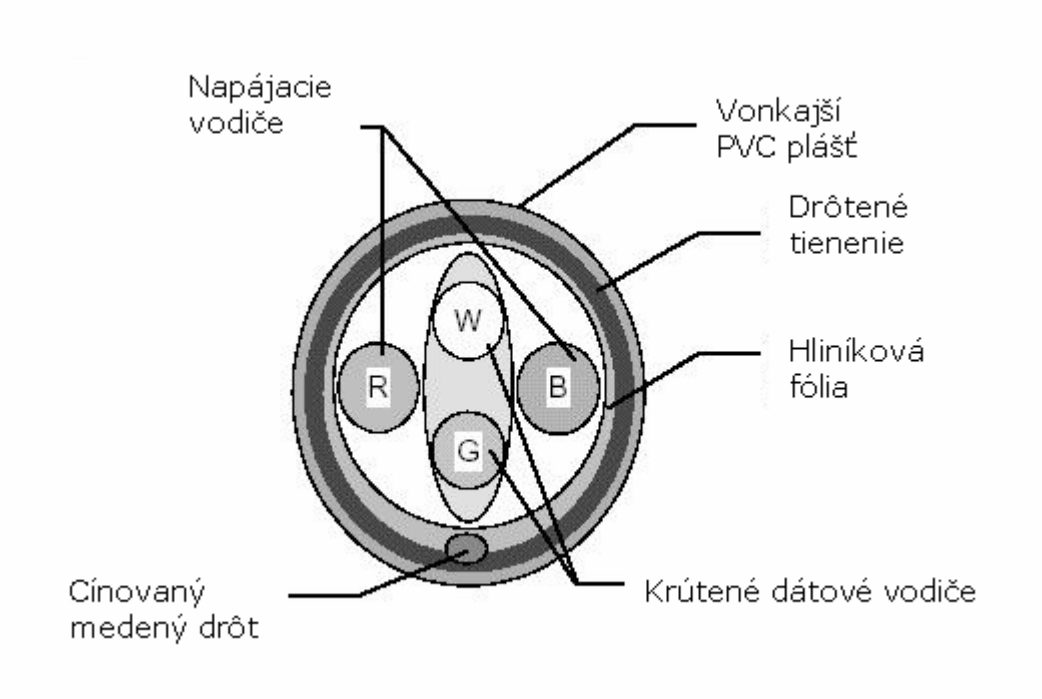
- Podpora združených zariadení
- Jednoduchý protokol
- Malý overhead protokolu (protokol má vysokou efektivitu)
- Garantované šírky pásma pre zariadenia, ktoré ju vyžadujú (telefón, audio atd.)
- Možnosť využitia celej šírky pásma jedným zariadením
- Robustnosť - spracovanie chýb je súčasťou protokolu
- Dynamické pridávanie a odoberanie zariadení pri ich pripojení/odpojení
- Konzistencia s PC Plug and Play architektúrou

3.5.2.2 Konektory pre USB

Táto kapitola popisuje konektory pre USB rozhranie, ktoré sa používajú pre pripojenie zariadení s hostiteľským systémom. Štandardne sa používajú dva typy konektorov a to konektory typu A a konektory typu B. Vzhľad a miesto umiestnenia konektorov popisuje nasledujúca tabuľka.

Konektory typu A	**Konektory typu B**
Používa sa na strane hostiteľského systému	Štandardne sa používa na strane zariadenia
 Zásuvka typu A - Výstup z USB hostiteľa	 Zásuvka typu B - Vstup do USB zariadenia
Zástrčka typu A - do USB hostiteľa 	Zástrčka typu B - do USB zariadenia 

Prenosové médium (Obrázok 11) používané na pripojenie USB zariadení z hostiteľským systémom sa skladá zo štyroch vodičov. Dva vodiče slúžia na napájanie a dva vodiče sú používané na prenos dát. V prenosovom režime s vyššou rýchlosťou (High Speed, Full Speed), je potrebné aby dátové vodiče boli krútené. Takéto médium je možné použiť aj v režime s nižšou prenosovou rýchlosťou (Low speed).



Obrázok 11. Prenosové médium USB

3.5.3 IRDA

IrDA je štandard vytvorený IrDA konzorciom (Infrared Data Association), ktorý definuje, ako bezdrôtovo prenášať digitálne dáta pomocou infračerveného žiarenia. IrDA vo svojich špecifikáciách definuje štandardy, ako fyzických koncových zariadení, tak protokolu, ktorým komunikujú IrDA zariadenia. IrDA štandard vznikol z potreby mobilne prepojiť rôzne zaradenia medzi sebou.

3.5.3.1 Fyzická vrstva

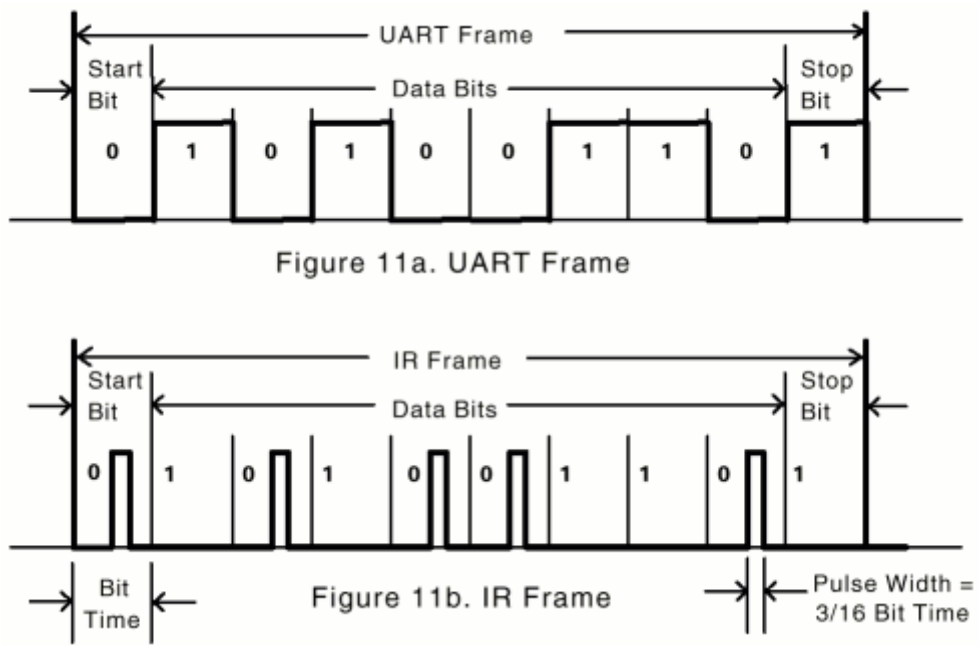
IrDA zariadenia komunikujú pomocou infračervených LED diód s vlnovými dĺžkami vyžarovaného svetla 875 nm +/- tolerancie výroby (asi 30nm). Na túto vlnovú dĺžku sú citlivé i mnohé CCD kamery. Prijímačom sú PIN fotodiódy, ktoré pracujú v generačnom režime (pri dopade svetla na prijímač "vyrazí" svetlo elektróny, ktoré sa odvedú do filtra (elektrického), ktorý prepustí len tie frekvencie, ktoré sú povolené pre daný typ IrDA modulácie). Existuje priama úmera medzi energiou dopadnutého žiarenia a nábojom, ktorý optická časť prijímača vygeneruje.

3.5.3.2 Dosah a používané rýchlosti IrDA

IrDA zariadenia podľa normy IrDA 1.0 a 1.1 pracujú do vzdialenosti 1.0 m pri bitovej chybovosti BER (bit error ratio, pomer chybne prenesených bitov ku správne preneseným)

10-9 a maximálnou úrovňou okolitého osvetlenia 10klux (denný svit slnka). Tieto hodnoty sú definované pre nesúostrosť vysielača a prijímača 15 stupňov, pre jednotlivé optické prvky sa meria výkon do 30 stupňov. Existujú smerové vysielače (IR LEDky) pre väčšie vzdialenosti, ktoré nedodržujú predpísaný uhol 30 stupňov od osy, pre ktorú ma vysielač útlm 3 dB.

Rýchlosti sú pre IrDA v. 1.0 od 2400 do 115200 kbps, používa sa pulzná modulácia 3/16 dĺžky pôvodnej doby trvania bitu. Formát dát je rovnaký ako na sériovom porte, teda asynchrónne vysielané slovo uvedené štartbitom.



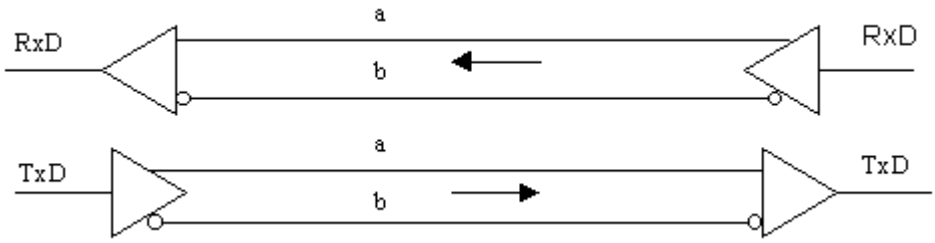
Obrázok 12. Formát dátového prenosu.

3.5.4 RS485 a RS422

Komunikácia po linke RS232 patrí k najrozšírenejším, pretože je ňou vybavený každý bežný počítač. Používa pre komunikáciu dvoch zariadení tak do vzdialenosti 20m. Ako už bolo spomenuté v časti o rozhraní RS232, okrem vodičov na prenos dát (RxD, TxD), používa aj ďalšie vodiče pre riadenie toku dát. Tieto podporné signály nie sú obsiahnuté v linkách RS485 a RS422 a musia byť nahradené komunikačným protokolom. Nevýhodou RS232 je teda malá vzdialenosť prenosu a nemožnosť jeho vetvenia. Pre prenos údajov sériovou komunikáciou je vhodné použiť rozhrania RS485 a RS422.

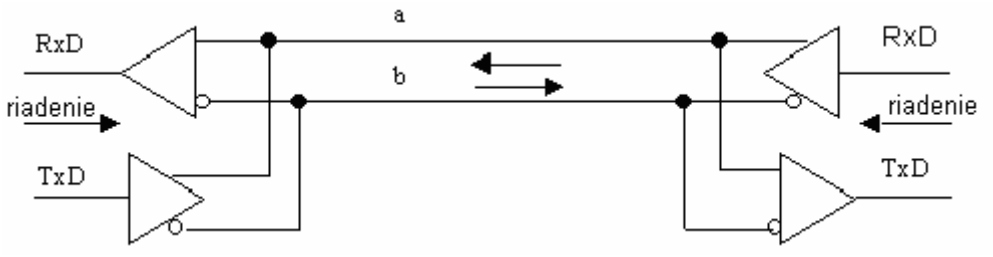
3.5.5 RS422

Linka RS422 používa jeden pár vodičov pre signál RxD a jeden pár pre signál RxT.



Obrázok 13. Prevedenie nevetvenej linky RS422.

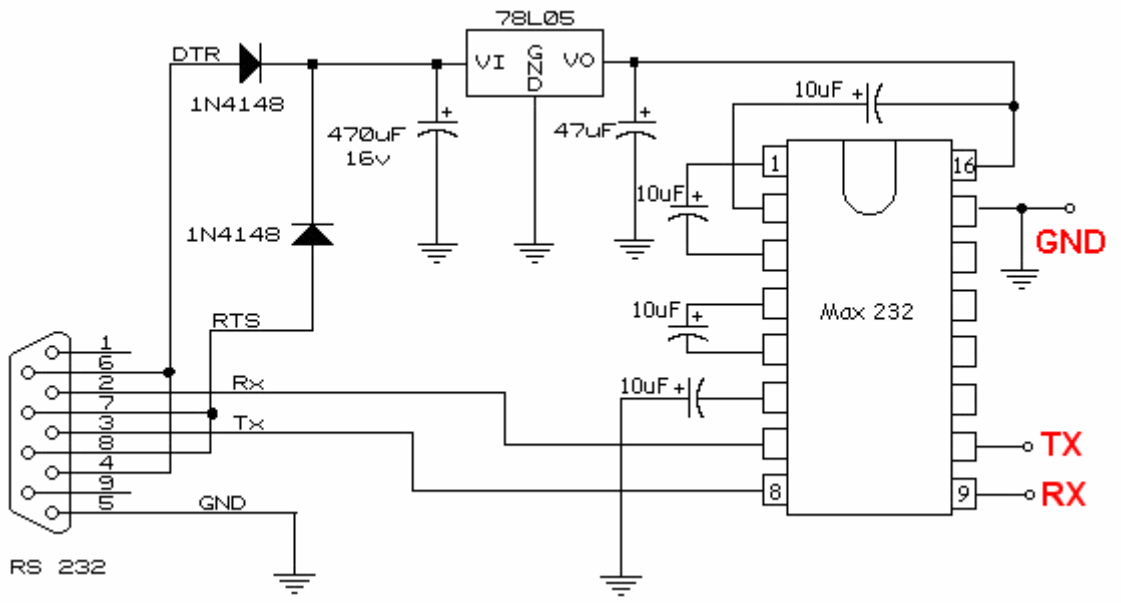
Linka RS485 používa len jeden pár vodičov pre obidva smery toku dát, je preto potrebné smer komunikácie prepínať.



Obrázok 14. Prevedenie nevetvenej linky RS485.

3.5.6 MAX232

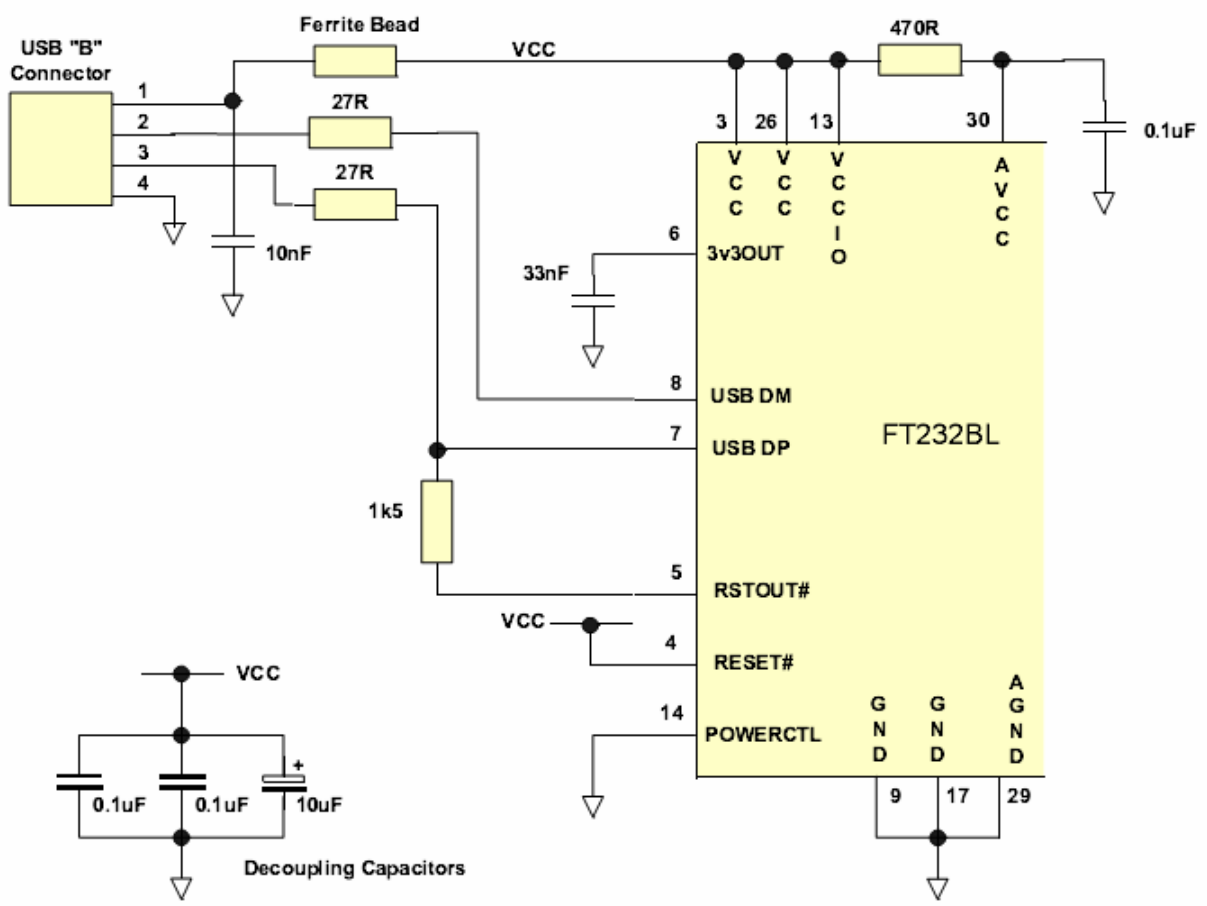
U rozhrania RS232 je logická 1 reprezentovaná záporným napätím. Logická 0 je reprezentovaná kladným napätím. V prípade vysielača môže napätie dosahovať minimálne −15 V a maximálne +15 V, a to by obvody pracujúce s TTL úrovňami poškodilo. Z toho dôvodu je potrebné použiť prevodník, ktorý nám prevedie úrovne z RS232 na úrovne TTL. K najpoužívanejším prevodníkom patrí obvod MAX232.



Obrázok 15. Schéma zapojenia RS232 s MAX232.

3.5.7 FT232BL

Pre imlpementáciu USB rozhrania k jednotlivým procesorom použijeme prevodník FT232BL.



Obrázok 16. Schéma zapojenia USB s FT232BL.

3.6 Analýza mikroprogramovej časti

Požiadavky zadania projektu vyžadujú dve softvérové časti. Monitor ako určitý firmware, zjednodušený operačný systém pre daný mikropočítač, poskytujúci taktiež prepojenie s hostiteľským počítačom a obslužným softvérom pre rozšírenie funkcionality požadovaného experimentálneho mikropočítača.

3.6.1 Monitor

Spôsobov, ako monitorovací softvér, alebo aj firmware implementovať je viacero. Pri výbere je potrebné brať ohľad na požadované vlastnosti, ale taktiež na schopnosti navrhovaného mikropočítača spolu s možnosťami pri tvorbe softvéru. Medzi faktory, ktoré môžu zasiahnuť do výberu konkrétneho návrhu riešenia spadajú:

- hardvérový návrh mikropočítača a s tým súvisiace testovanie, funkcionalita
- veľkosť pamäte pre uchovanie vytvoreného monitora
- voľba implementačného prostredia alebo programovacieho jazyka

- schopnosti a skúsenosti realizačného tímu, ale taktiež
- možnosti a zdroje

Keďže program má byť schopný plniť úlohy monitora a testera jednotlivých častí mikropočítača, samotný proces testovania je možné správne navrhnúť a vytvoriť až po kompletnom návrhu mikropočítača.

Na Internete je možné nájsť niekoľko už vytvorených monitorovacích programov pre podobné experimentálne mikropočítače, avšak vo väčšine prípadov je hardvérová časť odlišná, alebo nespĺňa nami požadované vlastnosti. Taktiež je možné nájsť vývojové prostredia pre rôzne procesory a mikropočítače, avšak väčšinou sa jedná a platené produkty, preto s nimi nebudeme počítať. Pri výbere implementačného prostredia je taktiež dôležité myslieť na programovací jazyk, kde je možné program vytvoriť:

- priamo v jazyku strojových inštrukcií – výhoda vo väčšej kontrole nad mikropočítačom, efektivita, veľkosť. Taktiež ľahká možnosť overiť a otestovať v emulátore.
- vyšší programovací jazyk (C, C++, C#, ...) – výhoda knižníc, pohodlnejší vývoj. Ľahšia manipulácia a úprava kódu. Avšak menej efektívny, problém s knižnicami volajúcimi systémové volania (ktoré nebudú obslúžené). Potrebná dobrá znalosť týchto jazykov pre ovládanie periférií a pod. Od tohto sa potom vyvíja aj hostiteľská platforma. Taktiež výhoda kvalitných vývojových prostredí. Nevýhodou môže byť chýbajúci, resp. neprístupný kompilátor pre zvolený typ mikroprocesora.

Testovacia sekvencia bude prvou vecou, čo sa po aktivácií programu spustí. Testom by mala vyhovieť každá logická jednotka mikropočítača, inak monitor nebude poskytovať všetky, respektíve služby. Jednotlivé testy sa budú líšiť vzhľadom na rôznu povahu rôznych prvkov mikropočítača. Tie však budú určite obsahovať:

- mikroprocesor
- externá pamäť

- sériové rozhranie a iné
- externé zariadenia ako napr. LED diódy, displej, ...
- test spojenia s hostiteľským počítačom

Podstatnou časťou mikropočítača je mikroprocesor vybranej triedy, ktorý je riadiacim prvkom a zároveň najkomplexnejším čipom v zariadení. Jeho správna činnosť je esenciálna, preto sekvencia príkazov pre otestovanie by mala byť vykonaná skôr než ostatné.

3.6.2 Testovanie mikroprocesora

Možností, ako otestovať mikroprocesor je veľa. Líšia sa hlavne komplexnosťou a odvíjajú sa tiež od procesora, ktorý sa práve testuje. Pre účely projektu bude využitý jednoduchší typ a preto aj testovanie bude menej náročné. Vzhľadom na predpokladaný vybraný typ je možné vyhodnotiť správnosť týchto logických jednotiek:

- vstavaná pamäť typu ROM a RAM v prípade režimu mikropočítača
- podporované porty
- sériové rozhranie
- časovače
- registre, príznaky
- zásobníky
- akumulátor
- inštrukčná sada
- aritmeticko-logická jednotka

Otestostovanie všetkých súčastí je však pomerne náročné a taktiež by výslednej veľkosti programu nemusela stačiť vnútorná pamäť procesora v režime mikropočítača, preto sa test obmedzí len na určité časti a po ich splnení sa prehlási test za úspešný. Metodiku pokročilejšieho testovania je možné nájsť aj na [1]. Ak je test mikroprocesora úspešný, je možné prejsť na potrebné testovanie pamätí, ktoré rozoberieme v ďalšej kapitole.

3.6.3 Testovanie pamätí

Test pamätí navrhovaného mikropočítača a ich správna činnosť patrí medzi nevyhnutné súčasti. Pri vzniku chýb by funkcionalita nebola korektná a mohla by viesť až k úplnému zlyhaniu zariadenia. Testovanie je potrebné rozdeliť na:

- vonkajšiu pamäť – v režime mikroprocesor, keď nevyužíva svoju
- vnútornú pamäť

ale taktiež na:

- prepisovateľnú pamäť (RAM)
- neprepisovateľnú pamäť (ROM)

Medzi testovaním vonkajších alebo vnútorných pamätí nie je zásadný rozdiel.

Pod podmienky správnej funkčnosti patria:

- schopnosť pamäťovej bunky zapísať (v prípade RAM), uchovať a prečítať hodnotu zapísanú programom
- schopnosť adresovať každú pamäťovú bunku
- schopnosť vykonávať príkazy zvolenou rýchlosťou

Pamäťové bunky môžu ovplyvniť prierazy, presluchy, skraty, fyzické poškodenie, ale aj nesprávne časovanie a adresovanie. Nesprávne adresovanie môže spôsobiť zlý návrh adresnej zbernice, či poškodený mikroprocesor. Pri softvérovom testovaní je možné využiť citlivosť porúch na vzorky. Analýza z pohľadu týchto vzoriek je však menej podstatná, nakoľko nevieme presné fyzické rozmiestnenie pamäťových buniek a preto vybrať tú správnu nie je jednoznačné. Otestovať všetky pamäťové časti, prípadne aj viac krát je často krát časovo náročné.

Medzi možné testovacie vzorky pre test RAM pamätí spomeniem:

- samé 0 alebo 1 - najjednoduchší spôsob testovania, avšak netestuje presluchy medzi bunkami ani správne adresovanie
- šachovnica - lepší spôsob otestovania, ktorý skontroluje aj presluchy medzi susednými bunkami

- samé riadky alebo stĺpce – podobné šachovnici
- diagonála – v matici jednotlivých bitov sa diagonála vyplní 1, ostatné 0. Po kontrole sa bity posunú a testuje znova
- parita - nastavené párne, resp. nepárne počty 1 alebo 0
- putujúca 1/0 - postupné zapisovanie 1, resp. 0, do pamäťových buniek. Dobrá testovacia schopnosť
- pochodujúca 1/0 - postupné zapisovanie 1, resp. 0, do pamäťových buniek, pričom v predchádzajúcich bunkách ostáva zapísaná hodnota. Taktiež dobrá testovacia schopnosť
- galoping – časovo náročný. Testovanie postupným zápisom 1 a 0 s návratom.

Iná metóda spočíva v náhodnom zápise na vybrané miesta v pamäti, napríklad na začiatok, strede a na koniec adresného priestoru alebo iné zvolené miesta. Takýto test je menej presný, avšak menej časovo náročný. Vzhľadom na fakt, že presné rozloženie pamäťových buniek nepoznáme a jeho schopnosť pomerne správne určiť funkčnosť sa javí ako vhodná alternatíva.

Pre testovanie pamäte ROM ale aj Flash je potrebné nájsť iný spôsob kvôli neschopnosti týchto pamätí dáta prepisovať. Medzi možné spôsoby detekcie patria:

- prečítanie vopred vybraných miest a porovnať ich s hodnotami, ktoré monitor pozná a má uložené
- využitie kontrolnej sumy pamäte.

Obidve metódy sa opierajú o fakt, že monitorovací program je už vopred daný a nemenný, vpísaný do pamäte flash. Prvá sa snaží o prečítanie definovaných adries a porovnaním s uloženou „databázou“ hodnôt, ktoré sa na daných adresách nachádzajú. Druhá metóda využíva rôznu metódu výpočtu kontrolnej sumy pamäte vytvoreného monitora a porovnáva otestovanú s vopred známou kontrolnou sumou. Jednotlivé funkcie pre výpočet je možné kombinovať a tak zaručiť lepšiu kontrolu. Kontrolnú suma sa vypočítava po napísaní celého programu. Ohľad treba dávať tiež na výpočtovú náročnosť zvoleného spôsobu testovania.

Kontrolné sumy je možné počítať napríklad XORom (detekcia 1 chyby), paritou, kontrolou rôznych bitov a bajtov po blokoch, atď.

3.6.4 Testovanie ostatných súčastí mikropočítača

Pre správnu funkčnosť navrhovaného mikropočítača je potrebné taktiež otestovať ostatné prídavné zariadenia. Medzi ne patrí sériové rozhranie, systém externých prerušení a iné externé zariadenia ako LED diódy, displej, reproduktor.

Otestovanie činnosti sériového rozhrania môže spočívať v zapísaní a následnom prečítaní hodnôt konfiguračných registrov, prípadne skúšky nadviazania spojenia s hostiteľským počítačom. Ak je spojenie vytvorené, je pravdepodobné, že sériové rozhranie funguje správne.

Funkčnosť diód či iných podobných zariadení je správne overiť zložitejšie, nakoľko spätné overenie správnosti činnosti je podmienené buď prídavnými senzormi, alebo vstupmi od obsluhovateľa zariadenia. Možnosťou na zariadenia poslať aktivačné príkazy bez možnosti spätnej väzby. Ako príklad je možné uviesť pípnutie v prípade reproduktora, či rozsvietenie diód.

Systém externých prerušení taktiež vzhľadom na potrebu ďalšieho potvrdzujúceho mechanizmu nebude predmetom testovania.

Medzi požiadavky na monitor patria aj znakovo orientovaná komunikácia s hostiteľským počítačom, načítanie vykonateľného programu a možnosť ovplyvňovať beh programu.

3.6.5 Komunikácia s hostiteľským počítačom

Pre splnenie podmienok, aby bolo možné nahrať, spúšťať a zastavovať vykonávanie programu či už počas jeho behu, alebo na vopred definovaných bodoch prerušenia je potrebné vytvoriť a určiť komunikačný protokol, ktorý zabezpečí prenos údajov a príkazov cez sériový port. Tento protokol bude poskytovať riadenie monitora a mikropočítača ako takého. Jeho návrh ovplyvní hlavne počet potrebných príkazov (funkcií) a ako aj ich charakter. Taktiež je potrebné myslieť na vlastnosti sériového portu a potrebu synchronizácie, či začiatok a koniec vysielania blokov údajov.

Samotné programy, ktoré chce užívateľ na mikropočítači spúšťať je možné prenášať buď v binárnej forme, alebo zakódovane. Jedným často používaným kódovaním je aj Intel HEX Code, ktorého výhodou je, že obsahuje kontrolné súčty pre čiastočnú kontrolu nad prenosom a pozná ho veľa kompilátorov. Kódovanie z a do HEX formátu je možné uskutočniť buď na strane monitora, alebo programu na hostiteľskom počítači. Druhá možnosť je však výhodnejšia z hľadiska jednoduchosti a kompaktnosti monitora a ľahšej realizácie.

3.6.6 Analýza softvérovej časti

Vytváraný softvér má slúžiť ako nadstavba vytvorených mikropočítačov a bude poskytovať rozšírenie monitora o grafické používateľské rozhranie s podporou ďalších funkcií.

Ovládací program spúšťaný na hostiteľskom počítači by mal spĺňať minimálne tieto požiadavky:

- grafické používateľské rozhranie pre OS Windows alebo Linux
- univerzálny program pre prácu s obidvomi mikropočítačmi
- výber rozhrania, cez ktoré bude komunikácia realizovaná
- terminál na znakovo orientovanú komunikáciu s pripojeným mikropočítačom
- kompilácia zdrojového programu

Medzi ďalšie možno uviesť:

- zobrazenie obsahu pamätí a registrov
- spustenie, zastavenie programu
- podpora navrhovaných rozhraní pre prenos údajov – RS232 a USB

Pre splnenie prvej požadovanej vlastnosti grafického rozhrania existujú rôzne kvalitné nástroje a vývojové prostredia pre oba operačné systémy. Vzhľadom na fakt, že stále najpoužívanejším operačným systémom je Microsoft Windows bude vhodnejšie využiť písanie programu v pre nás dostupnom Microsoft Visual Studiu a v niektorom jazyku C. Medzi možnosti, ako zabezpečiť multiplatformovosť patrí Java, nevýhodou je však pre nás menej skúseností s tvorbou programov v tomto jazyku a väčšou hardvérovou náročnosťou.

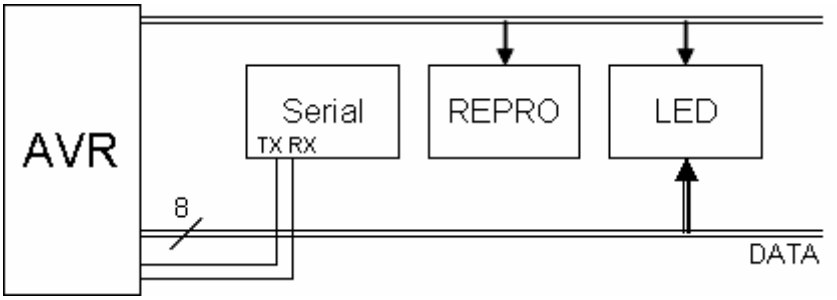
MS Visual Studio poskytuje tiež dobré možnosti pre splnenie ďalších požiadaviek pre prispôsobenie sa používateľským požiadavkám, aby bol program prispôsobiteľný a čo najodolnejší voči nechceným zásahom používateľa.

4 Hrubý návrh riešenia

Pre naše riešenie navrhujeme mikropočítač na báze ATMega16 a mikroprocesora 8051.

4.1 Návrh riešenia mikropočítača na báze ATMega16

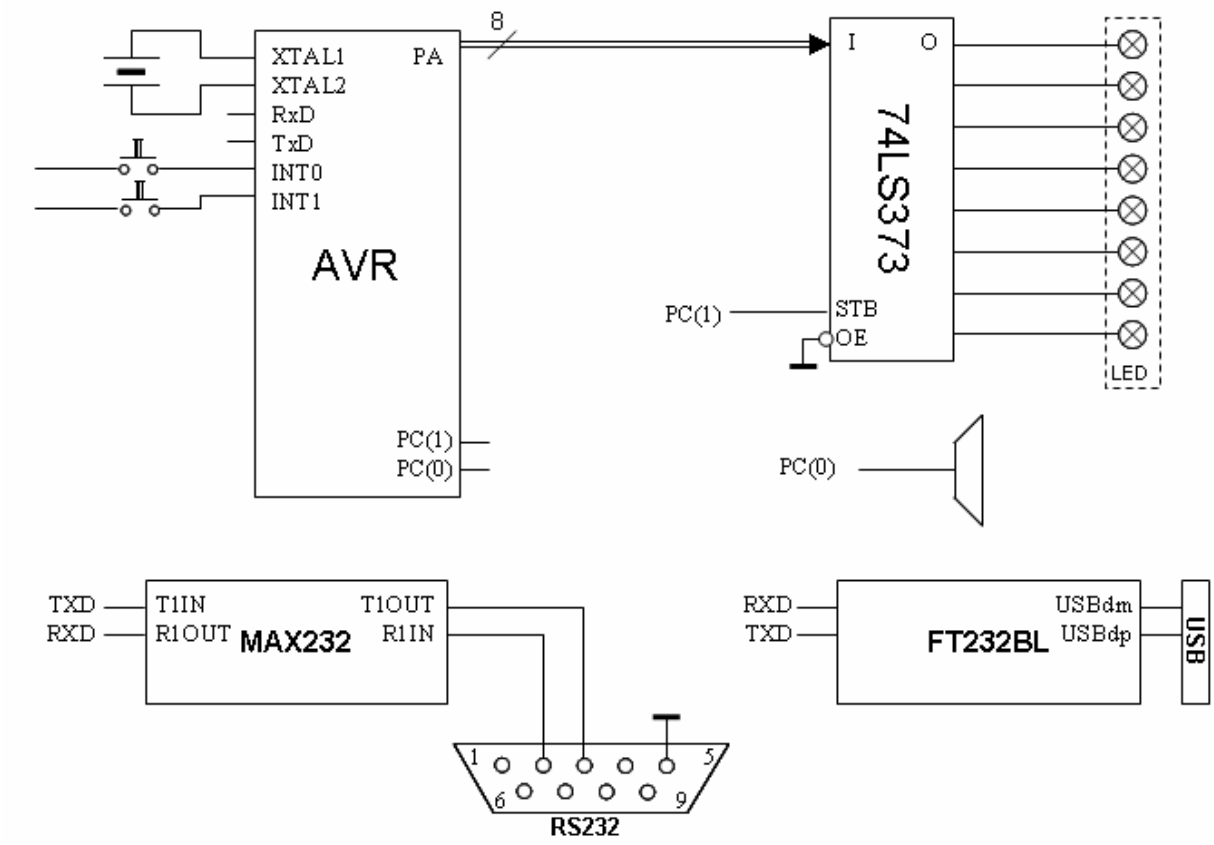
ATMega16 bude zapojený ako mikropočítač (bude využívať vnútornú pamäť programu a údajov). Ako periférne zariadenia budú pripojené LED diódy a reproduktor. S okolím bude komunikovať pomocou dvoch sériových liniek – RS232 a USB.



Obrázok 17. Bloková schéma zapojenia ATMega16.

Na pripojenie USB portu použijeme prevodníkový obvod FT232BL, RS232 použijeme obvod MAX232.

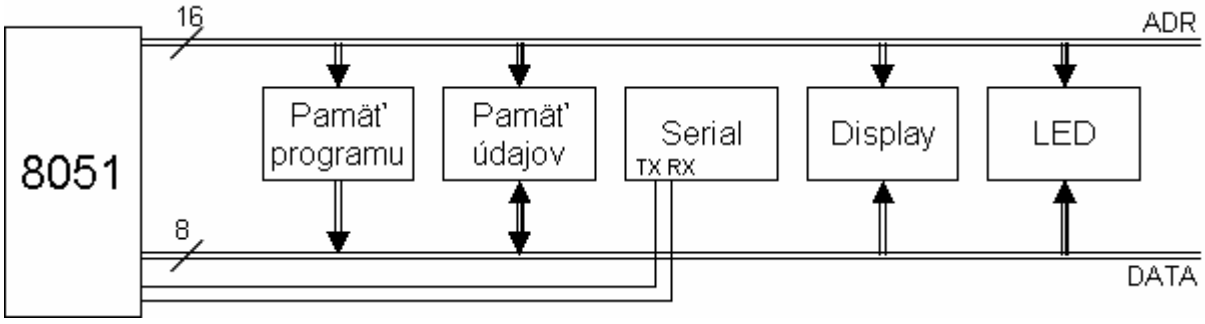
LED diódy budú pripojene k dátovej zbernici pomocou záchytného registra, aby sa udržal stav.



Obrázok 18. Logická schéma zapojenia ATMega16.

4.2 Návrh riešenia mikropočítača na báze 8051

8051 bude zapojený ako mikroprocesor (bude využívať externú pamäť programu a údajov). Ako periférne zariadenia budú pripojené LED diódy a maticový displej. S okolím bude komunikovať pomocou dvoch sériových liniek – RS232 a USB.



Obrázok 19. Bloková schéma zapojenia 8051.

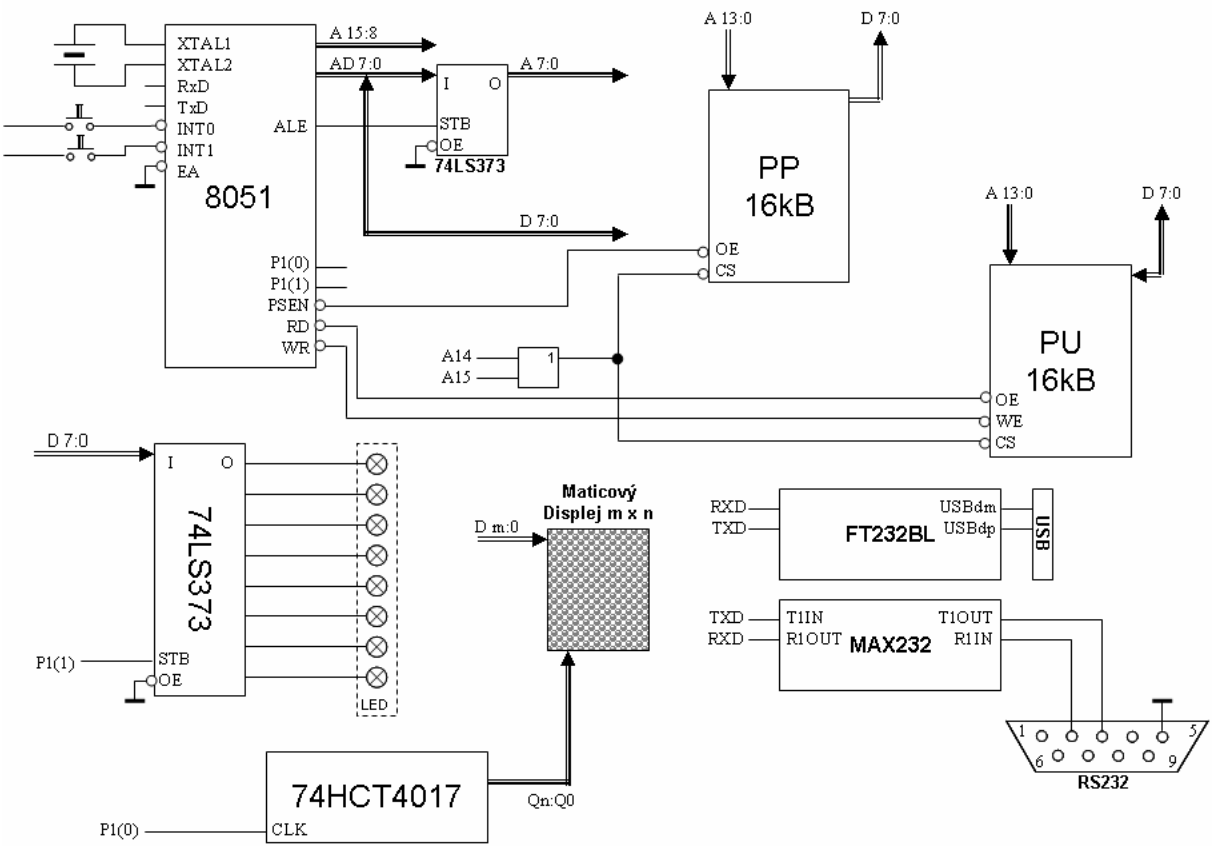
Pamäť programu aj pamäť údajov budú o veľkosti 16 kB (program: EPROM, údaje: RWM), z toho vyplýva, že na adresáciu pamäte bude použitých spodných 14 bitov adresy.

Mapovanie adries:

pamäť	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0	adresy
PP/PU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH

Na pripojenie USB portu použijeme prevodníkový obvod FT232BL, RS232 použijeme obvod MAX232.

Na zobrazovanie informácií pomocou maticového displeja použijeme obvod 74HCT4017. LED diódy budú pripojene k dátovej zbernici pomocou záchytného registra, aby sa udržal stav.



Obrázok 20. Logická schéma zapojenia 8051.

5 Návrh mikroprogramovej a softvérovej časti

Návrh a neskôr realizácia softvérovej časti projektu sa opiera o potrebné funkcie, ktoré sú uvedené v požiadavkách a špecifikácii. Pri zostavovaní finálneho riešenia je potrebné dbať hlavne na splnenie funkcionálnych a nefunkcionálnych požiadaviek, ale aj ľahkú prenositeľnosť a modularitu. Medzi rozhodujúce faktory patrí taktiež výber hostiteľského operačného systému a implementačného vývojového prostredia. Napriek tomu, že požadované funkcie pre oba mikropočítače sú rovnaké, implementácia prevažne mikroprogramovej časti je rôzna vzhľadom na odlišnú architektúru použitých mikroprocesorov.

Pre lepšie pochopenie a podchytenie funkcionálnych požiadaviek je dobré si problém rozdeliť na menšie podproblémy. Identifikovali sme troch hráčov, pričom prvým je používateľ ako zadávateľ príkazov a je v kontakte s druhým hráčom - programom na hostiteľskom počítači. Tento program potom na základe komunikácie a výmene príkazov riadi tretieho hráča, mikropočítač s pomocou mikroprogramu.

Vhodným vyjadrením dekompozície systému je aj diagram prípadov použitia. Jednotlivé funkcie sú ďalej popísané bez pohľadu na technickú podstatu.

5.1 Mikroprogramovacia časť

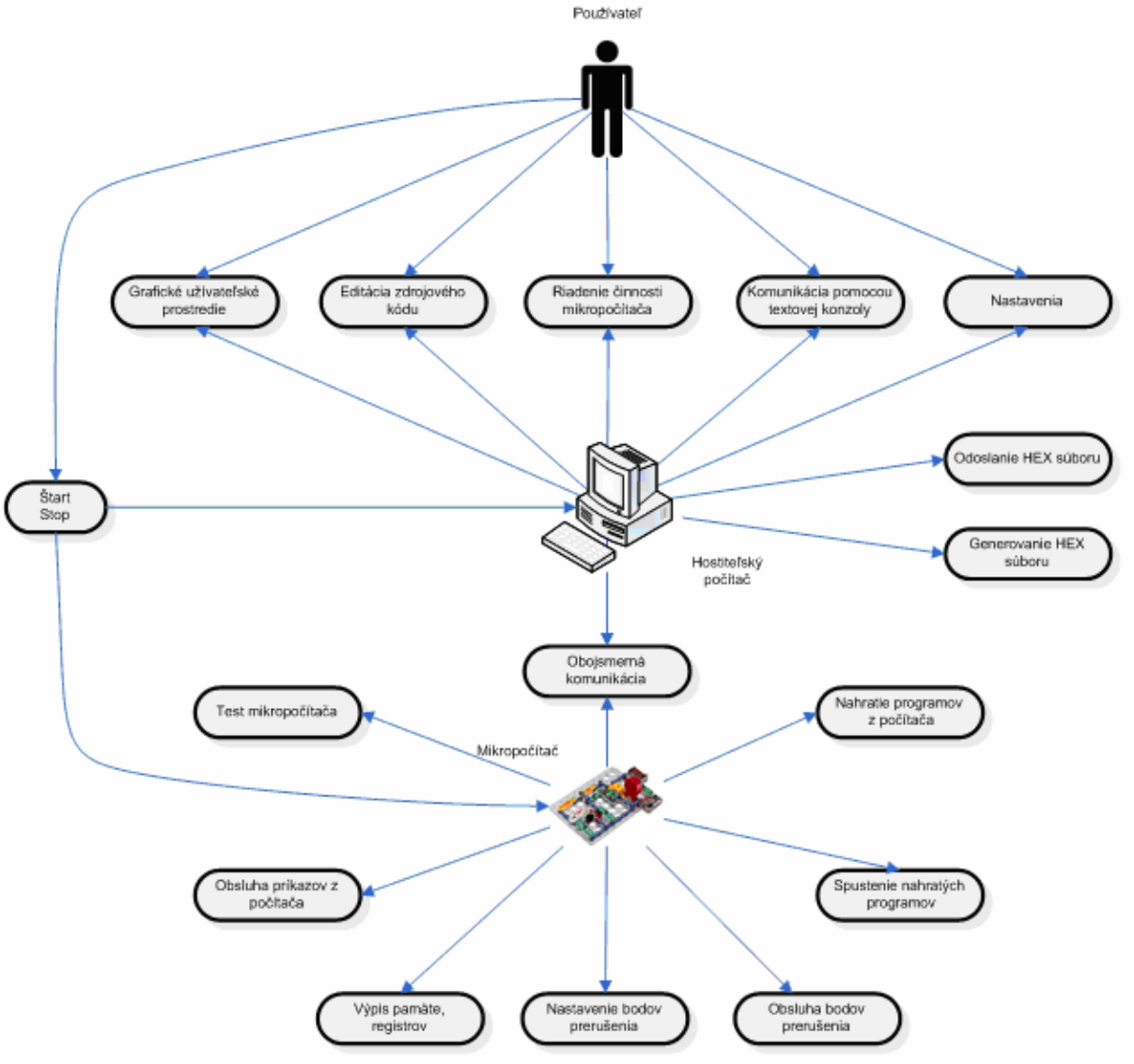
Nasledujúca časť obsahuje stručne popísané úlohy spĺňajúce potrebné testovacie, programovacie a monitorovacie funkcie firmvéru.

5.1.1 Štart a stop mikropočítača

Užívateľ fyzicky spustí zariadenie. Po spustení je okamžite vykonávaná sekvencia príkazov inicializačnej časti, ktorá má za úlohu naštartovať ostatné časti monitora. Tieto sú spolu s architektúrou uvedené neskôr.

5.1.2 Test mikropočítača

Medzi prvými úkonmi po štarte sa nachádza testovanie súčastí mikropočítača a po kontrole základnej funkcionality prenecháva riadenie monitoru.



Obrázok 21. Konceptuálny diagram prípadov použitia softvérovej časti projektu

5.1.3 Obsluha príkazov z počítača

Aby mikropočítač mohol vykonávať používateľove príkazy, musí vedieť komunikovať so softvérom na hostiteľskom počítači a spracovávať jeho požiadavky. Medzi takéto požiadavky patrí aj

5.1.4 Nahratie programov z PC

Aby mohol byť potrebný program vykonávaný na mikropočítači, musí byť najskôr premiestnený vo vhodnom formáte a uložený v prepisovateľnej pamäti.

5.1.5 Spustenie nahratých programov

Užívateľovi má byť umožnené požadovaný program spustiť. Z oblasti, pre ktorú je mikropočítač vyvíjaný plynú aj ďalšie požiadavky na sledovanie správania programu a s tým súvisiace:

5.1.5.1 Nastavenie a obsluha bodov prerušenia

Slúži pre lepšiu analýzu bežiaceho programu a ladeniu chýb. Používateľovi je tak umožnené nastaviť si v programe body prerušenia a firmvér má byť schopný takúto požiadavku vedieť obslúžiť.

5.1.5.2 Výpis registrov a pamäte

Pomáha pri bodoch prerušenia sledovať obsah pamätí a registrov mikropočítača a tak sledovať či hľadať chyby v programoch.

5.1.6 Proces štartovania

Po zapnutí mikropočítača prebieha počiatočný test, na základe ktorého úspešnosti je rozhodnuté, či inicializačný modul alebo subsystém dokončí proces štartovania a nahrá/spustí kompletnú inštanciu firmvéru so všetkou potrebnou funkcionalitou. Počas štartu sa samozrejme inicializujú aj zapojené časti mikropočítača. Na pripojenom displeji bude zobrazený príslušný kód pre výsledok testu vhodným nastavením záchytných registrov, registre pre reproduktor sa vynulujú, sériový port nastaví zmenou hodnôt SCON registra, či nastaví zásobník stavu monitora v pamäti.

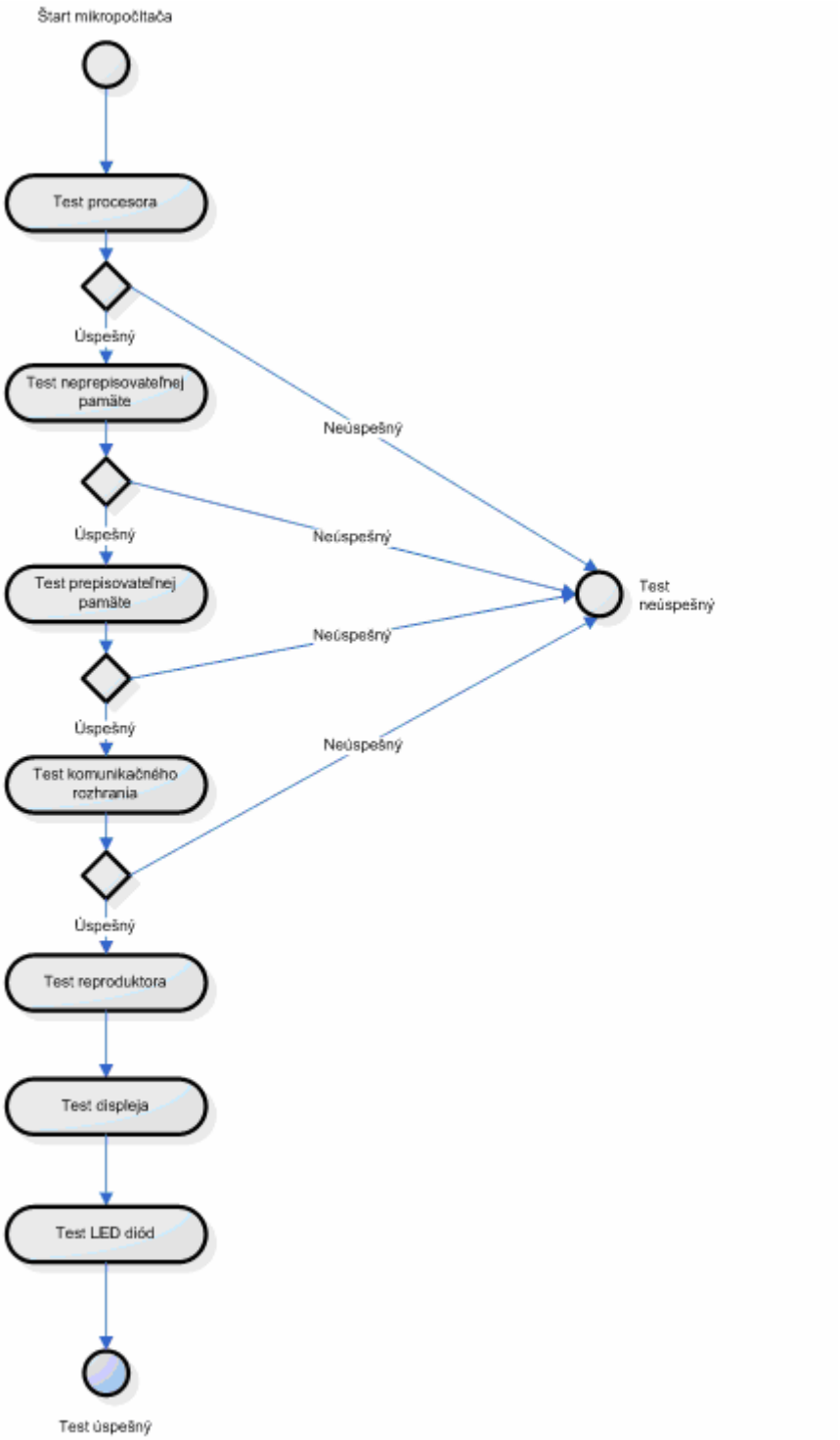
Obrázok 22. Proces štartu monitora

5.1.7 Test mikropočítača

Ako bolo spomenuté, všetky funkcie mikropočítača budú prístupné až po úspešne vyhodnotenom teste jeho súčastí. Testovací modul sa skladá z niekoľkých podtestov, ktoré krok za krokom preverujú správne zapojenie a funkčnosť mikropočítača. Medzi prvými sa nachádzajú súčasti nevyhnutne potrebné pre jeho chod, neskoršie už kvôli chýbajúcej spätnej väzbe neovplyvňujú výsledok testu. Nasledujúci obrázok zobrazuje postupnosť subtestov.

V prípade vyhodnotenia jedného z testov procesora, prepisovateľnej a neprepisovateľnej pamäte a komunikačného sériového rozhrania modul sa pokúsi externé zariadenia vyslať kódy chyby a tak dať používateľovi informáciu o probléme, pričom následný chod mikropočítača je pozastavený. Test externých zariadení neovplyvní výsledok celého testu a aj v prípade ich nefunkčnosti umožňovať pristupovať k ostatným funkciám mikroprocesora.

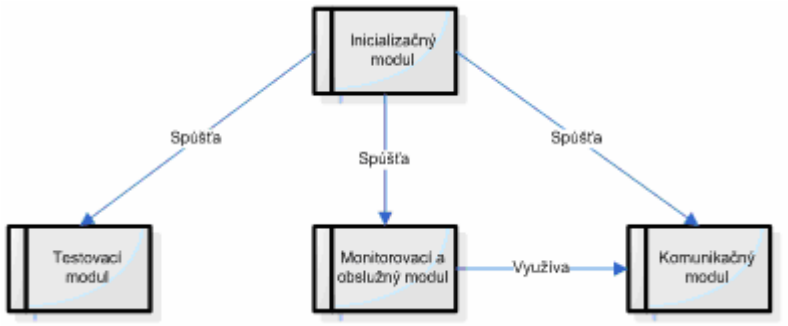
Spôsob testovania jednotlivých jednotiek je možné nájsť v analýze softvérovej časti (podkapitolách číslo 3.6.1).



Obrázok 23. Proces testovania mikropočítača

5.1.8 Návrh architektúry monitora

Ako už bolo taktiež spomínané, funkčnosť celého problému a zároveň návrhu si je dobré rozčleniť na menšie časti alebo celky. V našom prípade sa jedná o rozdelenie na základe potrieb počiatočnej inicializácie, testovania a komunikácie s PC aplikáciou. Monitor vystupuje ako server, pričom obsluhuje požiadavky klienta – PC aplikácie.



Obrázok 24. Členenie architektúry monitora

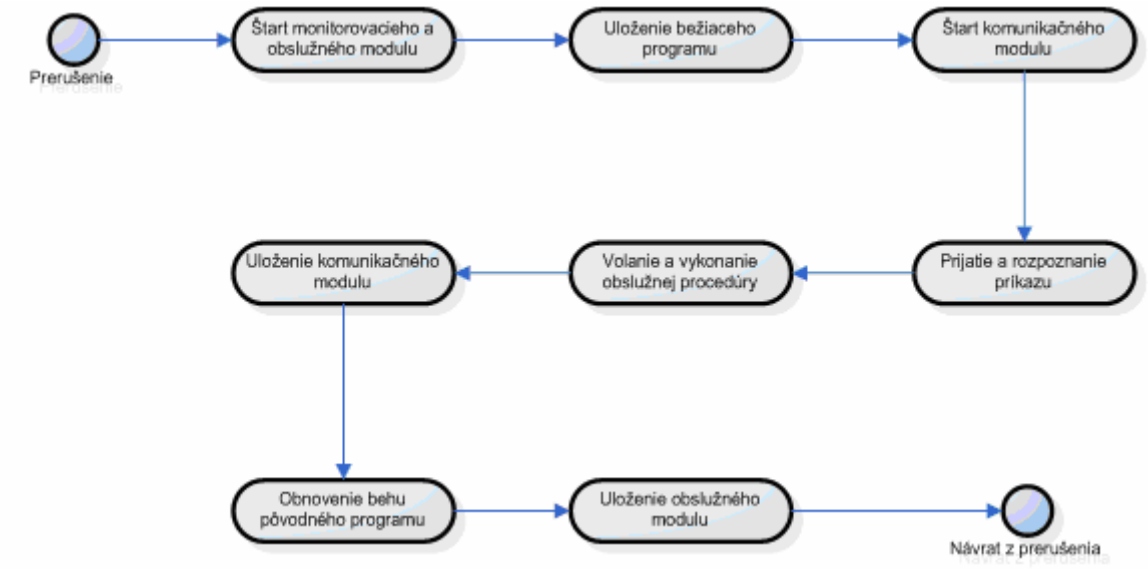
5.1.9 Medzi moduly tvoriace monitor preto patria:

- inicializačný modul
- testovací modul
- monitorovací a obslužný modul
- komunikačný modul

Inicializačná časť sa stará o správne spustenie činnosti zvyšných modulov, pričom po testovacej fáze chod riadi prevažne monitorovací a obslužný modul s využitím komunikačného rozhrania pre interpretáciu príkazov z počítača a užívateľa.

5.1.10 Komunikačný modul

Zabezpečuje komunikáciu medzi PC aplikáciou a mikropočítačom na strane mikropočítača. Jeho činnosť je aktivovaná na základe prerušení zo sériového rozhrania. Vďaka komunikačnému protokolu rozozná posielané príkazy. Obsluhu prerušenia na sériovom porte lepšie vyjadruje nasledujúci diagram (Obrázok 25).



Obrázok 25. Proces obslúženia prerušení mikropočítačom

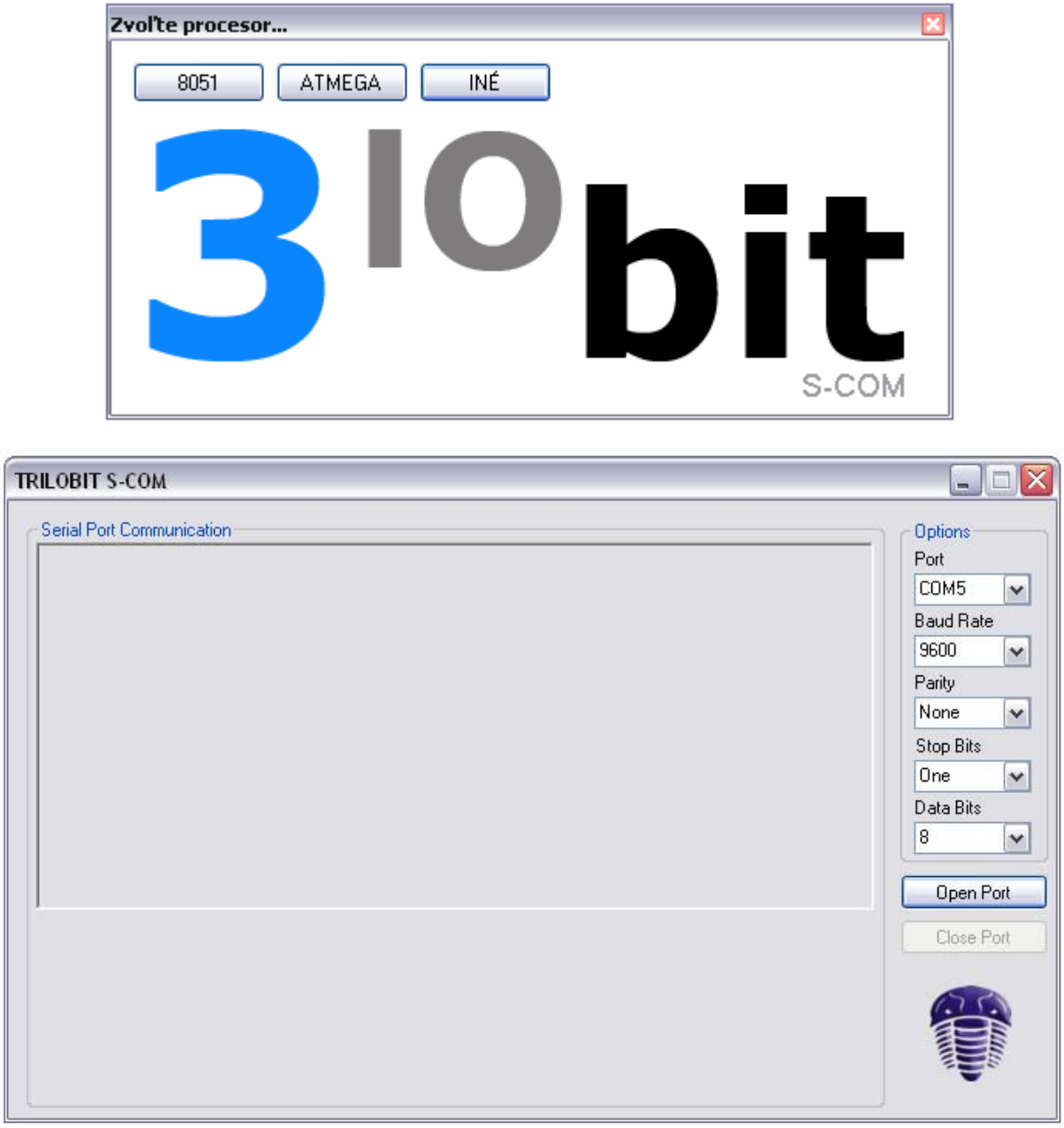
5.1.11 Monitorovací a obslužný modul

Slúži na poskytovanie požadovaných služieb mikropočítača naprogramovanými rutinami a poskytuje tak prácu s pamäťou, jej výpis, zmenu a nahrávanie používateľských programov. Umožňuje taktiež štart programu a obsluhu v bodoch prerušenia.

Implementácia všetkých modulov bude prebiehať v jazyku strojových inštrukcií pre daný mikroprocesor. Dôvody sú uvedené v kapitole 3.6.1.

5.2 Softvérová časť

Aplikácia je implementovaná programovacím jazykom C#. Na obrázku nižšie je znázornené používateľské rozhranie softvéru na komunikáciu s plošnými spojmi.



Obrázok 26. Softvérová aplikácia.

Na komunikáciu po sériovej linke sa využíva knižnica PCCom verzia 1.0.0.0.

6 Ďalšie požiadavky a ohraničenia

Pri výbere komponentov sme vychádzali hlavne z dostupnosti jednotlivých obvodov na slovenskom trhu (vid kapitoly 8.3 a 8.6).

7 Implementácia prototypu

Z počiatočného návrhu (vid prílohy 8.1 a 8.4) sme v programovom prostredí EAGLE vytvorili rozloženie jednotlivých komponentov a vodičov na doske plošného spoja (vid prílohy 8.2 a 8.5) Počas oživovania sme prišli na mnohé nedostatky návrhu, ktoré sú bližšie popísané v prílohe 8.9. Po oživení oboch mikropočítačov sme aplikovali riadiaci program (vid prílohy 8.7 a 8.8). Návod na používanie programu pre obsluhu mikropočítačov pomocou PC je uvedený v kapitole 8.10.

8 Záverečné zhodnotenie

Primárnym cieľom nášho projektu bolo navrhnúť, vytvoriť a oživiť dva experimentálne mikropočítače, čo sa nám v oboch prípadoch úspešne podarilo v rámci daných časových vymedzení.

Naše vymedzenie špecifikácie a primárne zameranie vychádzalo najmä z týchto bodov, a preto nespĺňa úplne všetky body pôvodnej špecifikácie projektu. Z riešenia bolo vynechané nasledovné:

- načítanie vykonateľného programu v definovanom formáte
- nastavenie / zrušenie bodov prerušenia

a to hlavne z dôvodu komplikovanosti takéhoto riešenia a časových obmedzení.

Všetky ostatné body zadania boli úspešne splnené podľa špecifikácie.

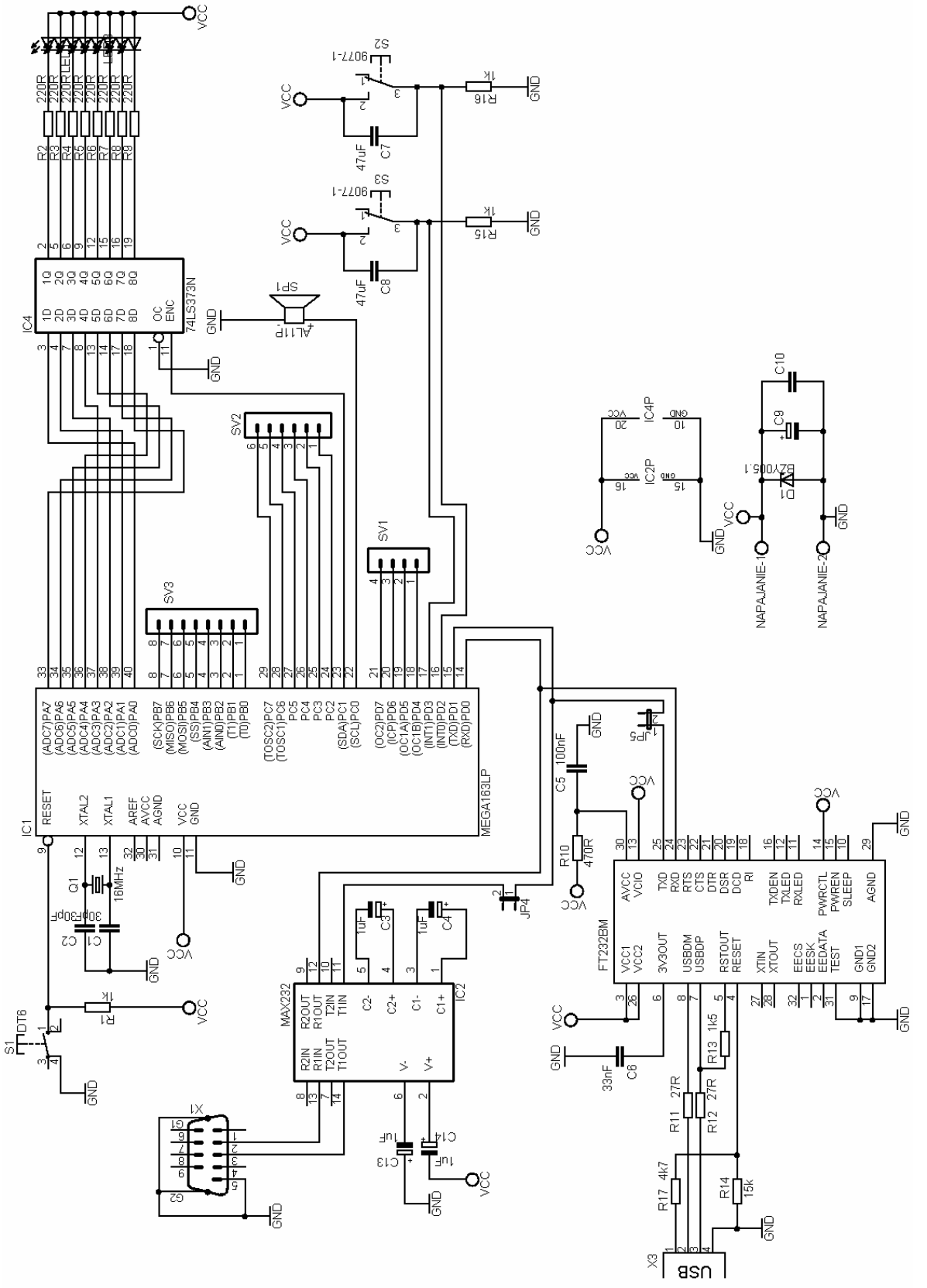
Postup tímu bol neustále spomaľovaný nepriaznivými externými udalosťami, a to nie len písomnými skúškami a inými školskými aktivitami členov tímu, ale aj dlhými dodacími lehotami pre obe dosky plošných spojov a jednotlivých integrovaných obvodov.

Tvorba firmvéru sa taktiež nezaobišla bez zistených chýb v návrhu alebo zapojení, no po odstránení všetkých prekážok vývoj pokračoval a po implementovaní programových rutín sa skompletizovanie riadiacich programov významne zjednodušilo.

Jednotlivý členovia tímu sa podľa svojich znalostí a skúseností podieľali na jednotlivých častiach projektu a prispeli tak k tímovému riešeniu úlohy diverzifikáciou na jednotlivé podúlohy pod manažérskou aktovkou vedúceho tímu a pedagogického vedúceho.

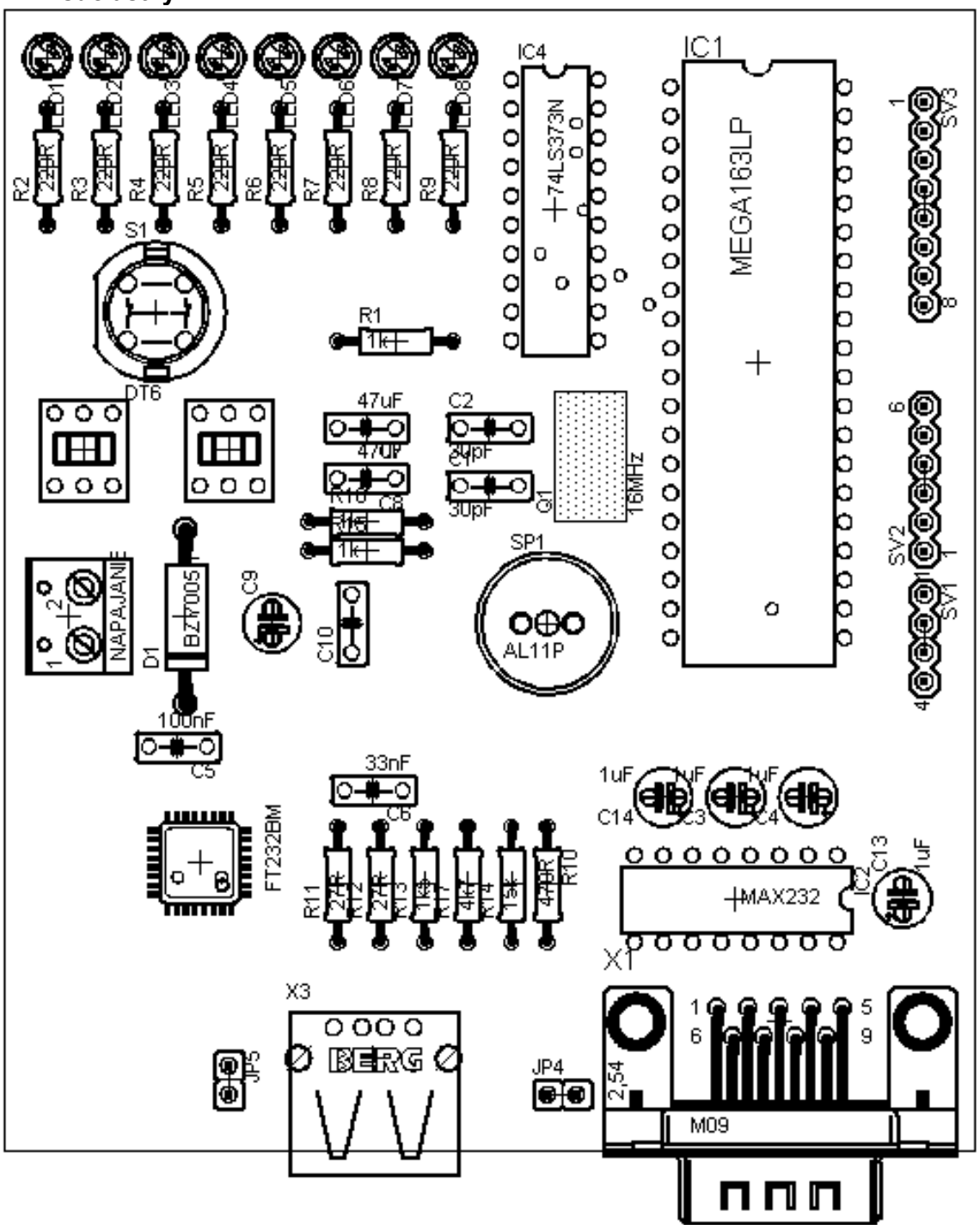
Prílohy

8.1 Schéma zapojenia ATMega16

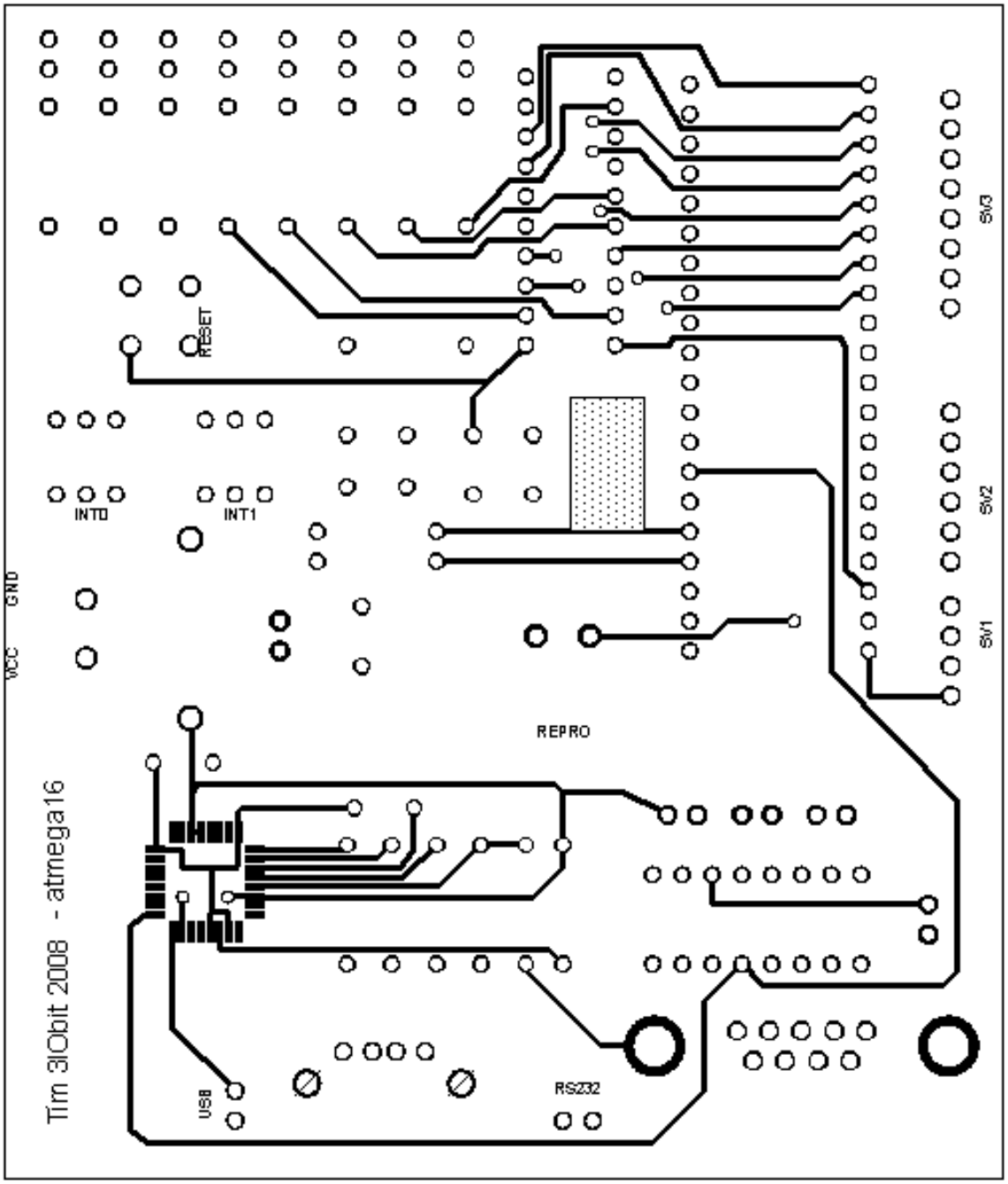


8.2 Schéma plošného spoja ATMega16

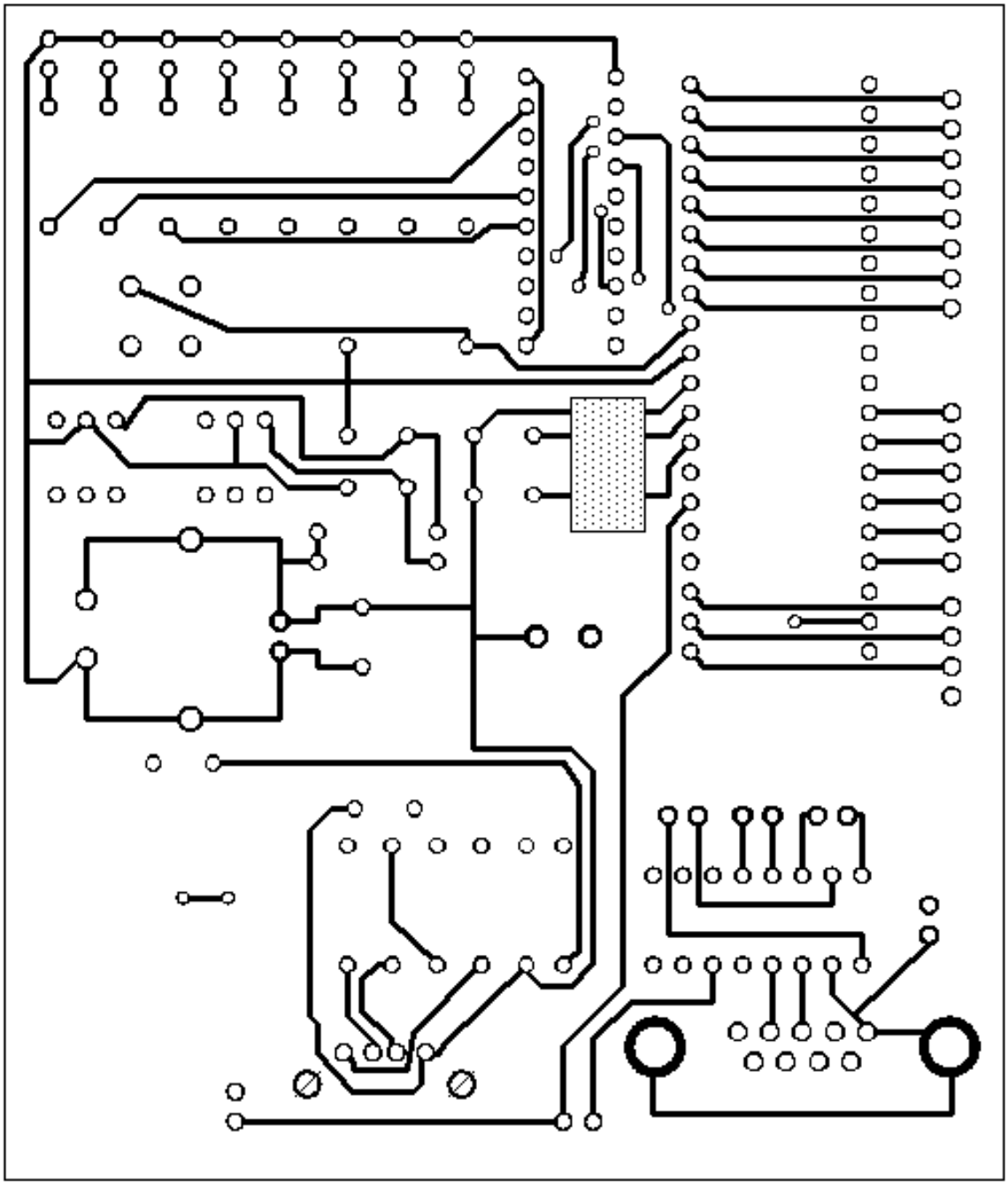
8.2.1 súčiastky



8.2.2 horná strana



8.2.3 dolná strana



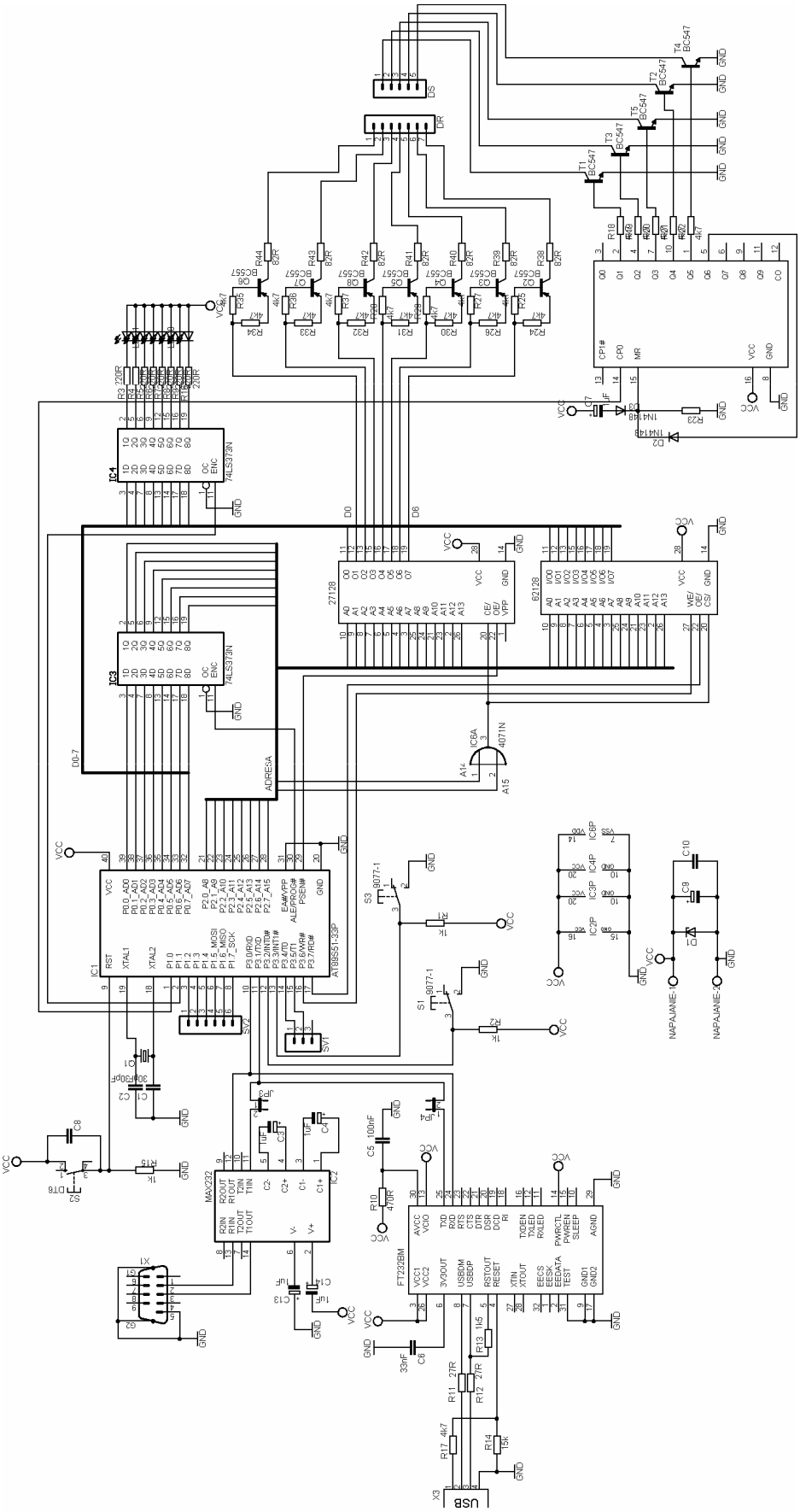
8.3 Zoznam súčiastok pre ATMega16

```
Partlist

Exported from atmega16.brd at 06.03.2008 08:57:41

EAGLE Version 4.16 Copyright (c) 1988-2005 CadSoft
```

```
Part   Value     Package   Library   Position (mm)     Orientation

C1    30pF      C050-025X075 rcl     (119.38 57.785)    R0
C2    30pF      C050-025X075 rcl     (119.38 62.865)    R0
C3    1uF      E2,5-5    rcl     (140.97 30.48)    R180
C4    1uF      E2,5-5    rcl     (147.32 30.48)    R180
C5    100nF     C050-025X075 rcl     (92.075 34.925)    R180
C6    33nF      C050-025X075 rcl     (109.22 31.115)    R180
C7    47uF      C050-025X075 rcl     (108.585 62.865)   R180
C8    47uF      C050-025X075 rcl     (108.585 58.42)    R180
C9           E2,5-5    rcl     (100.33 45.72)    R90
C10          C050-025X075 rcl     (107.315 45.72)    R90
C13    1uF      E2,5-5    rcl     (155.575 21.59)    R90
C14    1uF      E2,5-5    rcl     (134.62 30.48)    R180
D1    BZY005.1    C1702-15   diode    (92.71 46.355)    R90
FT232BM         QFP-32    ftdi2    (93.98 24.765)    R90
IC1    MEGA163LP   DIL40    atmel    (142.875 68.58)    R270
IC2    MAX232     DIL16    maxim    (140.97 21.59)    R180
IC4    74LS373N    DIL20    74xx-us   (125.095 81.915)   R270
JP4          JP1     jumper    (125.73 4.445)    R270
JP5          JP1     jumper    (96.52 5.715)     R180
LED1          LED3MM    led     (80.645 95.25)    R270
LED2          LED3MM    led     (85.725 95.25)    R270
LED3          LED3MM    led     (90.805 95.25)    R270
LED4          LED3MM    led     (95.885 95.25)    R270
LED5          LED3MM    led     (100.965 95.25)    R270
LED6          LED3MM    led     (106.045 95.25)    R270
LED7          LED3MM    led     (111.125 95.25)    R270
LED8          LED3MM    led     (116.205 95.25)    R270
NAPAJANIE ARK120/2    W237-102   con-wago-500 (82.55 46.355)    R90
Q1    16MHz     HC18U-V   crystal   (128.27 60.325)    R90
R1    1k       0207/10   rcl     (111.125 70.485)   R0
R2    220R      0207/10   rcl     (80.645 85.725)    R90
R3    220R      0207/10   rcl     (85.725 85.725)    R90
R4    220R      0207/10   rcl     (90.805 85.725)    R90
R5    220R      0207/10   rcl     (95.885 85.725)    R90
R6    220R      0207/10   rcl     (100.965 85.725)   R90
```

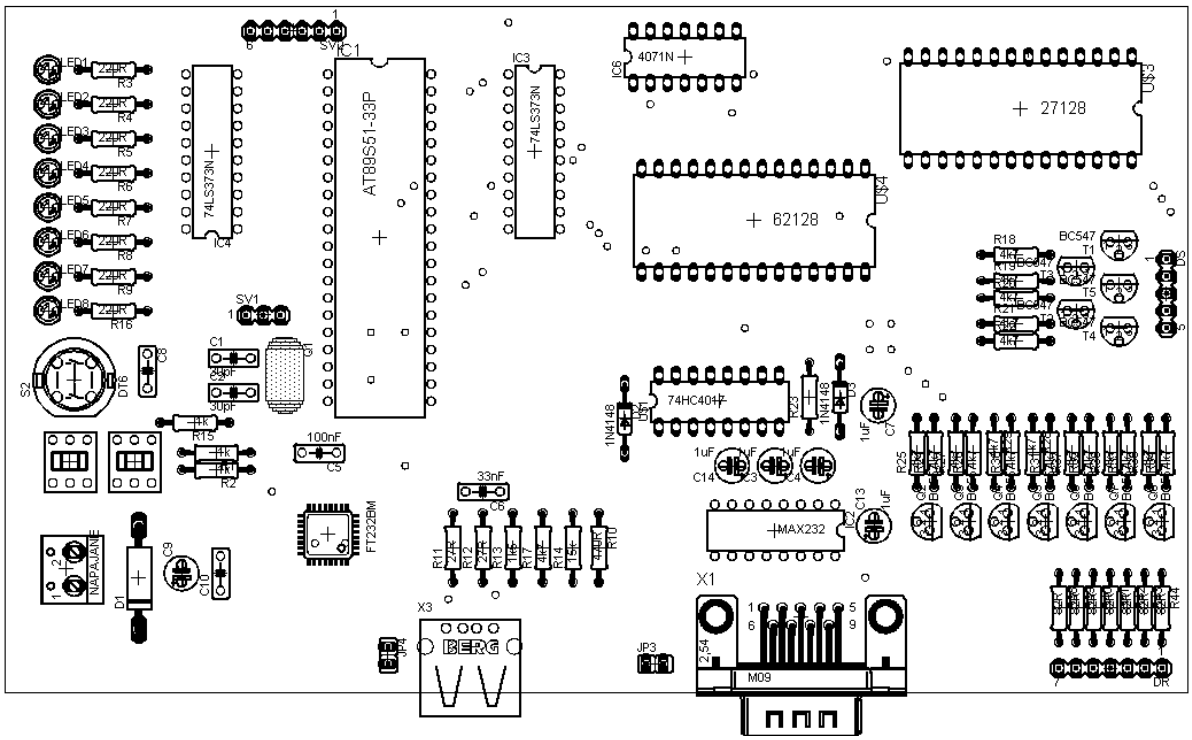
```
R7    220R      0207/10   rcl     (106.045 85.725)   R90
R8    220R      0207/10   rcl     (111.125 85.725)   R90
R9    220R      0207/10   rcl     (116.205 85.725)   R90
R10    470R      0207/10   rcl     (124.46 22.86)    R270
R11    27R      0207/10   rcl     (106.045 22.86)    R90
R12    27R      0207/10   rcl     (109.855 22.86)    R90
R13    1k5      0207/10   rcl     (113.665 22.86)    R90
R14    15k      0207/10   rcl     (121.285 22.86)    R90
R15    1k       0207/10   rcl     (108.585 52.07)    R0
R16    1k       0207/10   rcl     (108.585 54.61)    R0
R17    4k7      0207/10   rcl     (117.475 22.86)    R90
S1    P-DT6RT    DT6     switch-misc (97.79 71.12)     R90
S2    P-B170H    9077-1    switch-misc (83.82 60.96)     R0
S3    P-B170H    9077-1    switch-misc (96.52 60.96)     R0
SP1             AL11P    buzzer    (124.46 45.72)    R0
SV1    S1G20           MA04-1    con-lstb   (157.48 44.45)    R270
SV2    S1G20     MA06-1    con-lstb   (157.48 58.42)    R90
SV3    S1G20     MA08-1    con-lstb   (157.48 82.55)    R270
X1    CAN 9 Z 90   M09HP    con-subd   (144.78 10.795)    R0
X3    USB1X90    PN87520   con-berg   (109.22 7.62)     R0



+ pätice pre všetky io
```

8.4 Schéma zapojenia 8051

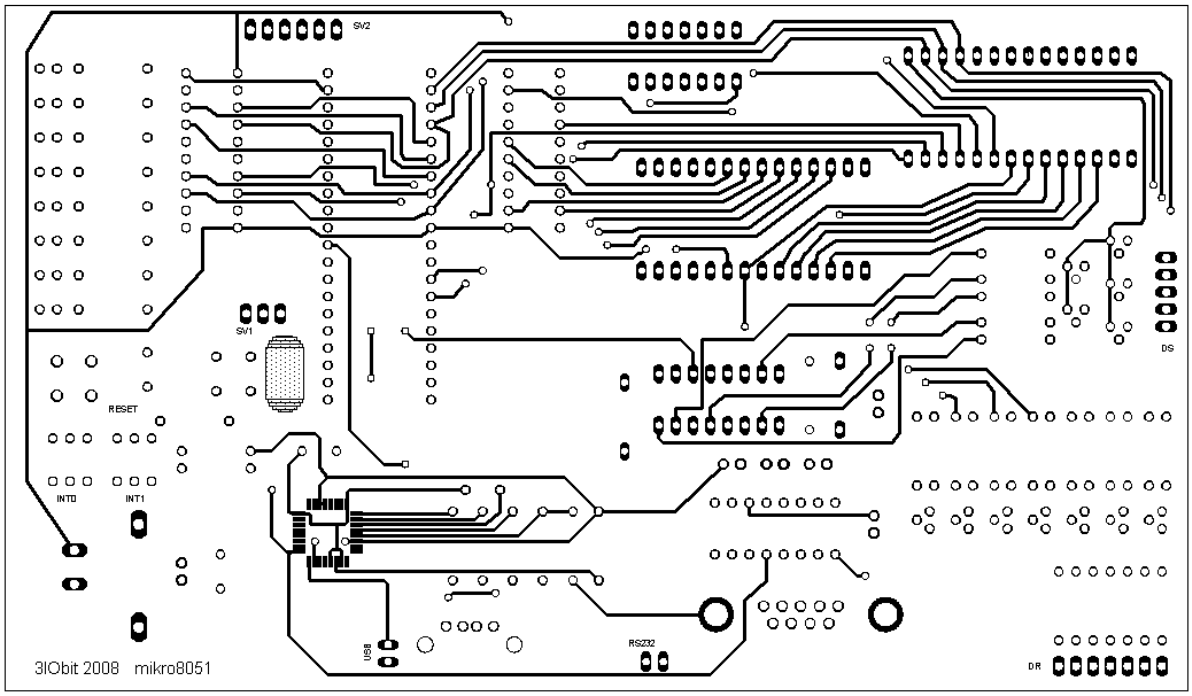


8.5 Schéma plošného spoja 8051

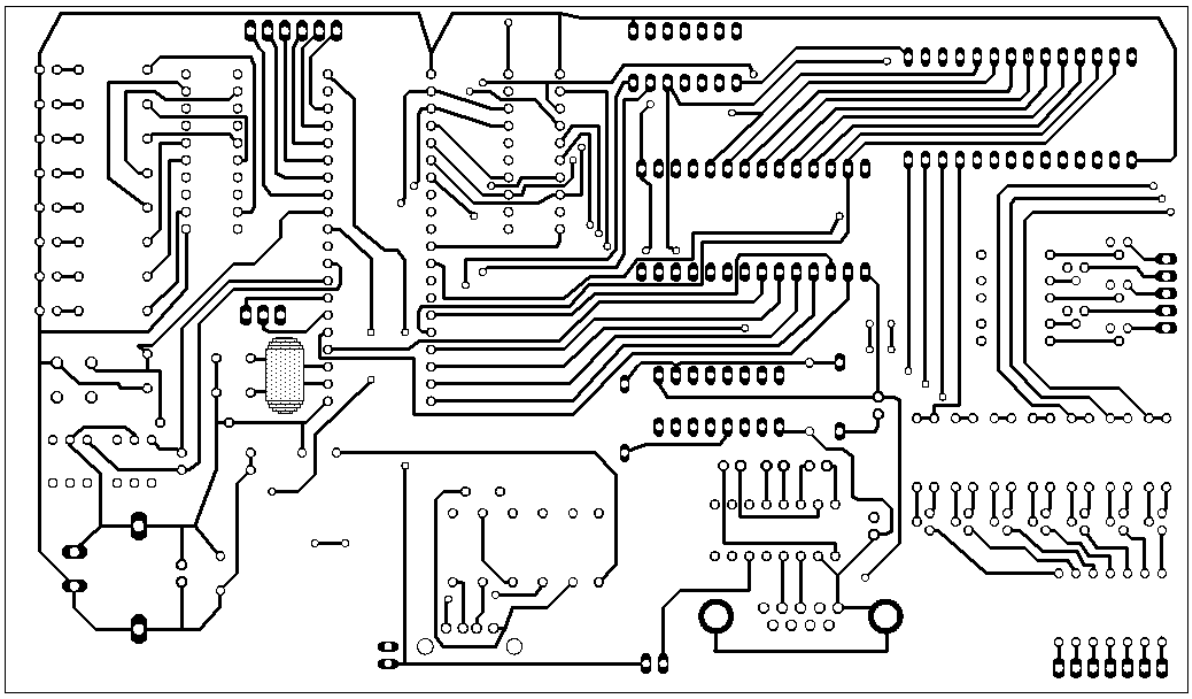
8.5.1 súčiastky



8.5.2 horná strana



8.5.3 dolná strana



8.6 Zoznam súčiastok pre 8051

```
Partlist

Exported from mikro8051_2.brd at 06.03.2008 10:25:20

EAGLE Version 4.16 Copyright (c) 1988-2005 CadSoft

Part   Value     Package   Library    Position (mm)     Orientation

C1    30pF      C050-025X075 rcl      (33.02 46.99)     R0
C2    30pF      C050-025X075 rcl      (33.02 41.91)     R0
C3    1uF      E2,5-5    rcl      (113.03 31.115)    R180
C4    1uF      E2,5-5    rcl      (119.38 31.115)    R180
C5    100nF     C050-025X075 rcl      (45.72 33.02)     R180
C6    33nF      C050-025X075 rcl      (69.85 27.305)    R180
C7    1uF      E2,5-5    rcl      (128.27 40.005)    R270
C8            C050-025X075 rcl      (17.78 45.085)    R270
C9            E2,5-5    rcl      (25.4 20.32)     R90
C10           C050-025X075 rcl      (31.115 20.32)    R90
C13    1uF      E2,5-5    rcl      (127.635 22.225)   R90
```

```
C14    1uF      E2,5-5    rcl      (106.68 31.115)    R180
D1    BZY005.1    C1702-15   diode     (19.05 19.685)    R90
D2    1N4148     DO35-10   diode     (90.805 38.1)     R270
D3    1N4148     DO35-10   diode     (122.555 41.275)   R270
DR    S1G20     MA07-1    con-lstb    (162.56 1.27)     R180
DS            MA05-1    con-lstb    (170.815 56.515)   R270
FT232BM         QFP-32    ftdi2     (46.99 20.955)    R90
IC1    AT89S51-33P  DIL40    atmel     (54.61 64.77)     R270
IC2    MAX232     DIL16    maxim     (113.03 21.59)    R180
IC3    74LS373N    DIL20    74xx-eu    (77.47 77.47)     R270
IC4    74LS373N    DIL20    74xx-eu    (29.845 77.47)    R90
IC6    4071N     DIL14    40xx      (99.695 91.44)    R0
JP3           JP1     jumper     (95.25 1.905)     R270
JP4           JP1     jumper     (55.88 3.175)     R180
LED1           LED3MM    led      (5.715 89.535)    R0
LED2           LED3MM    led      (5.715 84.455)    R0
LED3           LED3MM    led      (5.715 79.375)    R0
LED4           LED3MM    led      (5.715 74.295)    R0
LED5           LED3MM    led      (5.715 69.215)    R0
LED6           LED3MM    led      (5.715 64.135)    R0
LED7           LED3MM    led      (5.715 59.055)    R0
LED8           LED3MM    led      (5.715 53.975)    R0
NAPAJANIE ARK120/2    W237-102   con-wago-500  (8.255 20.955)    R90
Q1            HC49U-V   crystal    (40.64 44.45)     R270
Q2    BC557     TO92     transistor-pnp (135.89 22.86)    R90
Q3    BC557     TO92     transistor-pnp (141.605 22.86)    R90
Q4    BC557     TO92     transistor-pnp (147.32 22.86)    R90
Q5    BC557     TO92     transistor-pnp (153.035 22.86)    R90
Q6    BC557     TO92     transistor-pnp (170.18 22.86)    R90
Q7    BC557     TO92     transistor-pnp (164.465 22.86)    R90
Q8    BC557     TO92     transistor-pnp (158.75 22.86)    R90
R1    1k       0207/10   rcl      (30.48 33.02)     R180
R2    1k       0207/10   rcl      (30.48 30.48)     R180
R3    220R      0207/10   rcl      (15.24 89.535)    R180
R4    220R      0207/10   rcl      (15.24 84.455)    R180
R5    220R      0207/10   rcl      (15.24 79.375)    R180
R6    220R      0207/10   rcl      (15.24 74.295)    R180
```

```
R7    220R      0207/10   rcl      (15.24 69.215)    R180
R8    220R      0207/10   rcl      (15.24 64.135)    R180
R9    220R      0207/10   rcl      (15.24 59.055)    R180
R10    470R      0207/10   rcl      (86.995 19.05)    R270
R11    27R      0207/10   rcl      (65.405 19.05)    R90
R12    27R      0207/10   rcl      (69.85 19.05)     R90
R13    1k5      0207/10   rcl      (74.295 19.05)    R90
R14    15k      0207/10   rcl      (83.185 19.05)    R90
R15    1k       0207/10   rcl      (27.305 37.465)    R180
R16    220R      0207/10   rcl      (15.24 53.975)    R180
R17    4k7      0207/10   rcl      (78.74 19.05)     R90
R18    4k7      0207/10   rcl      (148.59 62.23)    R0
R19    4k7      0207/10   rcl      (148.59 58.42)    R0
R20    4k7      0207/10   rcl      (148.59 55.88)    R0
R21    4k7      0207/10   rcl      (148.59 52.07)    R0
R22    4k7      0207/10   rcl      (148.59 49.53)    R0
R23    10k      0207/10   rcl      (118.11 41.275)    R90
R24    4k7      0207/10   rcl      (136.525 33.02)    R90
R25    4k7      0207/10   rcl      (133.985 33.02)    R90
R26    4k7      0207/10   rcl      (142.24 33.02)    R90
R27    4k7      0207/10   rcl      (139.7 33.02)     R90
R28    4k7      0207/10   rcl      (151.13 33.02)    R270
R29    4k7      0207/10   rcl      (145.415 33.02)    R270
R30    4k7      0207/10   rcl      (147.955 33.02)    R90
R31    4k7      0207/10   rcl      (153.67 33.02)    R90
R32    4k7      0207/10   rcl      (159.385 33.02)    R90
R33    4k7      0207/10   rcl      (165.1 33.02)     R90
R34    4k7      0207/10   rcl      (170.815 33.02)    R90
R35    4k7      0207/10   rcl      (168.275 33.02)    R90
R36    4k7      0207/10   rcl      (162.56 33.02)    R90
R37    4k7      0207/10   rcl      (156.845 33.02)    R90
R38    82R      0207/10   rcl      (154.94 10.16)    R270
R39    82R      0207/10   rcl      (157.48 10.16)    R270
R40    82R      0207/10   rcl      (160.02 10.16)    R270
R41    82R      0207/10   rcl      (162.56 10.16)    R270
R42    82R      0207/10   rcl      (165.1 10.16)     R270
R43    82R      0207/10   rcl      (167.64 10.16)    R270
```

```
R44    82R      0207/10   rcl      (170.18 10.16)    R270
S1    P-B170H    9450-1    switch-misc  (19.05 33.655)    R0
S3    P-DT6RT    DT6     switch-misc  (8.89 45.085)     R90
S4    P-B170H    9450-1    switch-misc  (8.89 33.655)     R0
SV1    S1G20     MA03-1    con-lstb    (37.465 53.34)    R0
SV2           MA06-1    con-lstb    (41.91 95.25)     R180
T1    BC547     TO92     transistor   (163.83 64.135)    R180
T2    BC547     TO92     transistor   (157.48 53.975)    R180
T3    BC547     TO92     transistor   (157.48 60.325)    R180
T4    BC547     TO92     transistor   (163.83 51.435)    R180
T5    BC547     TO92     transistor   (163.83 57.785)    R180
U$1    74HC4017    DIL16    74xx-eu    (104.775 40.64)    R0
U$3    27128     DIL28@1   memory     (149.225 83.82)    R180
U$4    62128     DIL28@1   memory     (109.855 67.31)    R180
X1    CAN 9 Z 90   M09HP    con-subd    (116.84 8.89)     R0
X2    USB1X90    PN61729   con-berg    (66.675 3.175)    R0

+ takisto pätice na IO
```

8.7 Obslužný program pre ATMega16

Obslužný program pre procesor Atmega bol vyvíjaný v jazyku symbolických inštrukcií a kompilovaný v programe AVR Studio 4.13.

Väčšina programu je štruktúrovaná do interných neparametrických makier a podprogramov, ktoré sú volané v priebehu programu.

Program ošetruje aj dva externé prerušenia (INT0 a INT1), ktoré sú indikované tlačidlami. Prerušenie INT0 zavolá obslužný podprogram pre toto prerušenie na adrese 002H, ktorý rozsvieti všetky párne diódy.

Prerušenie INT1 zavolá obslužný podprogram pre toto prerušenie na adrese 004H, ktorý rozsvieti všetky nepárne diódy.

Prerušenie INT1 má väčšiu prioritu ako INT0, teda dokáže prerušiť aj jeho.

Program je rozdelený do blokov:

- testovanie zapojenia a funkcií procesora
- komunikácie s ovládacím programom cez sériovú linku

- blok jednotlivých podprogramov vykonávaných na základe príkazu o ovládacieho programu

8.7.1 Test zapojenia a funkcií procesora

Po štarte (resete) mikropočítača sa ako prvý vykoná vizuálny a zvukový test periférnych zariadení, a to tak že sa na určitú krátku dobu rozsvietia všetky diódy a bzučiak vydá krátky tón.

Nasleduje test funkcií procesora a to tak, že sa otestuje postupne pomocou operácií nad konštantami jeho sčítačka (inštrukciou ADD), komparátor (inštrukciou CP) logické hradlá (inštrukcie OR a AND) a pamäť (zápisom a prečítaním samých 1 a to isté s 0). V prípade chyby sa program zastaví (zacyklí v nekonečnej slučke) a na diódach sa rozsvieti prislúchajúci chybový vektor (viď tabuľku dole).

V prípade, že všetky testy prejdú bez chyby, tým je bootovanie dokončené.

inštrukcia	Funkcia	chybový vektor
CP	Komparátor	**ooooooo•**
ADD	Sčítanie	**ooooo•oo**
ADDC	Sčítanie s carry	**oooooo•o**
OR	logický súčet	**ooo•oooo**
AND	logický súčin	**oo•ooooo**
ST , LD	**Pamäť**	**oooo•ooo**

8.7.2 Komunikácia s ovládacím programom

Po úspešnom prebehnutí všetkých testov program v slučke čaká na prijatie znaku ‘a’. To je preňho indikácia, že je pripojený po sériovej linke k riadiacemu programu. Ako odpoveď pošle znak ‘[’, aby aj ovládací program dostal potvrdenie o spojení.

Po tomto sa program dostane do stavu, kde v slučke čaká na jednotlivé príkazy, ktoré majú tvar číslicových znakov.

8.7.3 Blok jednotlivých podprogramov

V tejto časti kódu sa nachádzajú podprogramy, ktoré sa vykonajú po stlačení na tlačítko v obslužnom programe, respektíve po vybratí si niektorej možnosti z menu, ktoré je dostupné pri práci s mikropočítačom cez konzolu. V menu sa pohybuje pomocou čísel, ktoré

označujú jednotlivé podprogramy. Menu s podprogramami, ktoré ponúka náš mikropočítač je nasledovné:

1. Knight rider
2. Párne / nepárne
3. Diódy + pípanie
4. Obsah registra SREG
5. Obsah registra UCSRA
6. Obsah registra UCSRB

8.7.4 DELAY

Táto procedúra sa využíva v každom programe. Sú to dva vnorené cykly, ktoré jednotlivými nastaveniami indexov vykonávajú veľké množstvo inštrukcií, čím sa spôsobí vyžiadané časové oneskorenie. Máme implementované dva typy delay procedúry, jedna je delay diody a druha delay bzučiak. Prvá slúži na oneskorenie pri postupnom rozsvecovaní diód v programoch a druhá slúži na vyrobenie frekvencia ktorou treba kmitať bzučiak aby vytvoril počuteľný tón.

8.7.5 KNIGHT RIDER

Po zvolení príslušnej možnosti sa vykoná skok na podprogram, ktorý simuluje slávny knight rider. Najskôr sa teda vysvieti prvá dióda zľava, zavolá sa podprogram delay diody, dióda sa zhasne a rozsvieti sa druhá dióda a znova sa zavolá podprogram delay diody atď. Takto to pokračuje až do času, kedy sa postupne rozsvieti každá dióda z ľava do prava. Potom sa to isté opakuje znova, no tentoraz z prava do ľava. Po skončení podprogramu ostávajú diódy zhasnuté.

8.7.6 PÁRNE / NEPÁRNE

Po zvolení príslušnej možnosti sa vykoná skok na podprogram, ktorý najskôr vysvieti všetky párne diódy, zavolá sa podprogram delay diody a všetky párne diódy zhasnú a vysvietia všetky nepárne diódy. Po skončení podprogramu ostanú diódy zhasnuté.

8.7.7 DIODY + PIPANIE

Po zvolení príslušnej možnosti sa vykoná skok na podprogram, ktorý najskôr vysvieti 4 a 5 diódu následne pípne bzučiakom zavolá podprogram delay diódy, ďalej rozsvieti 3 a 6

diódu pípne bzučiakom a zavolá podprogram delay diódy. Takýmto spôsobom sa pokračuje ďalej že sa rozsvietia 2 a 7 dióda pípne a nakoniec rozsvieti 1 a 8 diódu pípne a je koniec podprogramu. Vždy svietia len dve dané diódy a ostatné sú zhasnuté, na záver sú zhasnuté všetky diódy

8.7.8 VÝPIS KONFIGURÁCIÍ RIADIACICH REGISTROV

V menu je možné si vybrať výpis z nasledujúcich 8 bitových registrov mikroprocesora: SREG, UCSRA a UCSRB. Vybratý register je poslaný po sériovej linke obslužnému programu pre mikropočítač, ten ho rozdelí na jednotlivé bity a tak určí a vypíše konfiguráciu mikropočítača.

8.8 Obslužný program pre 8051

Obslužný program pre procesor 8051 bol vyvíjaný v jazyku symbolických inštrukcií a kompilovaný kompilátorom asm51.

Väčšina programu je štruktúrovaná do interných procedúr, ktoré sú volané v priebehu programu.

Program ošetruje aj dva externé prerušenia (INT0 a INT1), ktoré sú indikované tlačidlami. Prerušenie INT0 zavolá obslužný podprogram pre toto prerušenie na adrese 003H, ktorý vypíše na maticový displej ‘I.0‘ a rozbliká postupne pravú a ľavú časť diód.

Prerušenie INT1 zavolá obslužný podprogram pre toto prerušenie na adrese 013H, ktorý vypíše na maticový displej ‘I.1‘.

Prerušenie INT1 má väčšiu prioritu ako INT0, teda dokáže prerušiť aj jeho.

Program je rozdelený do blokov:

- testovanie zapojenia a funkcií procesora
- komunikácie s ovládacím programom cez sériovú linku
- blok jednotlivých podprogramov vykonávaných na základe príkazu o ovládacieho programu

8.8.1 Test zapojenia a funkcií procesora

Po štarte (resete) mikropočítača sa ako prvý vykoná vizuálny test vizuálnych periférnych zariadení, a to tak že sa na určitú krátku dobu rozsvietia všetky diody a všetky body maticového displeja.

Nasleduje test funkcií procesora a to tak, že sa otestuje postupne pomocou operácií nad konštantami jeho sčítačka (inštrukciou ADD), násobička (inštr. MUL), logické hradlá (inštrukcie ORL a ANL). V prípade chyby sa program zastaví (zacyklí v nekonečnej slučke) a na maticový displej sa ukáže znak výkričník a na diódach sa rozsvieti prislúchajúci chybový vektor (viď tabuľka dole).

V prípade, že všetky testy prejdú bez chyby, na displeji sa na sekundu ukáže nápis 'OK', a tým je bootovanie dokončené.

inštrukcia	Funkcia	chybový vektor
ADD	Sčítanie	•ooooooo
MUL	Násobenie	o•ooooooo
ORL	logický súčet	oo•ooooo
ANL	logický súčin	ooo•oooo

8.8.2 Komunikácia s ovládacím programom

Po uvítaní postupným vypísaním AHOJ na displej program v slučke čaká na prijatie znaku 'a'. To je preňho indikácia, že je pripojený po sériovej linke k riadiacemu programu. Ako odpoveď pošle znak ']', aby aj ovládací program dostal potvrdenie o spojení.

Po tomto sa program dostane do stavu, kde v slučke čaká na jednotlivé príkazy, ktoré majú tvar číslicových znakov a ovládajú menu so stromovou štruktúrou.

8.8.3 Blok jednotlivých podprogramov

V tejto časti kódu sa nachádzajú podprogramy, ktoré sa vykonajú po stlačení na tlačítko v obslužnom programe, respektíve po vybratí si niektorej možnosti z menu, ktoré je dostupné pri práci s mikropočítačom cez konzolu. V menu sa pohybuje pomocou čísel, ktoré označujú jednotlivé podprogramy, respektíve jednotlivé podmenu. Číslo nula slúži na pohyb v menu o krok vyššie. Menu s podprogramami, ktoré ponúka náš mikropočítač je nasledovné:

1. Výpis znakov na maticový displej
2. Rôzne blikanie diód

1) Syntetizátor
2) Párne/nepárne
3) Knight Rider 1
4) Knight Rider 3

3. Výpis konfiguračných registrov
   1) Register PSW
   2) Register SCON
   3) Register TMOD

8.8.4 DELAY

Táto procedúra sa využíva v každom programe. Sú to dva vnorené cykly, ktoré jednotlivými nastaveniami indexov vykonávajú veľké množstvo inštrukcií, čím sa spôsobí vyžiadané časové oneskorenie.

8.8.5 VÝPIS ZNAKOV NA MATICOVÝ DISPLEJ

Po zvolení príslušnej možnosti sa vykoná skok na podprogram, ktorý vypisuje znaky na maticový displej. Po stlačení znaku sa tento vyhľadá v zozname znakov na mikropočítači (obsahuje sadu veľkých písmen malej abecedy a interpunkčné znamienko bodku) a vysvieti sa na maticovom displeji 5x7, ktorý je k mikropočítaču pripojený. Podprogram okrem toho, že vysvieti znak na displeji, vyšle po sériovej linke aj špeciálny znak, ktorý hovorí o tom, že znak bol úspešne vysvietený na displej. Má to tú výhodu, že pri používaní ovládacieho programu pre tento mikropočítač je možné napísať celú vetu a odoslať ju na mikropočítač pre vypísanie. Program si vetu rozdelí na znaky, pošle prvý znak a čaká na špeciálny znak od mikropočítača. Keď ho dostane, vyšle mikropočítaču ďalší znak, až pokým sa nevysvieti celá veta.

8.8.6 SYNTETIZÁTOR

Po zvolení príslušnej možnosti sa vykoná skok na podprogram, ktorý najskôr vysvieti prostredné dve diódy, zavolá podprogram delay, pridá ďalšie dve diódy tak, že budú svietiť prostredné štyri diódy a znova zavolá podprogram delay atď. Takto to pokračuje, až sú všetky diódy vysvietené. Potom nasleduje postupné zhasínanie diód a to takým spôsobom, že najskôr zhasnú diódy po krajoch, zavolá sa podprogram delay, zhasnú ďalšie diódy atď. Je to inverzné

voči tomu, ako sa rozsvecovali diódy predtým, než sa všetky rozsvietili. Po skončení podprogramu ostávajú diódy zhasnuté.

8.8.7 PÁRNE / NEPÁRNE

Po zvolení príslušnej možnosti sa vykoná skok na podprogram, ktorý najskôr vysvieti všetky párne diódy, zavolá sa podprogram delay a všetky párne diódy zhasnú a vysvietia všetky nepárne diódy. Po skončení podprogramu ostanú diódy zhasnuté.

8.8.8 KNIGHT RIDER 1

Po zvolení príslušnej možnosti sa vykoná skok na podprogram, ktorý simuluje slávny knight rider. Najskôr sa teda vysvieti prvá dióda zľava, zavolá sa podprogram delay, dióda sa zhasne a rozsvieti sa druhá dióda a znova sa zavolá podprogram delay atď. Takto to pokračuje až do času, kedy sa postupne rozsvieti každá dióda z ľava do prava. Potom sa to isté opakuje znova, no tentoraz z prava do ľava. Po skončení podprogramu ostávajú diódy zhasnuté.

8.8.9 KNIGHT RIDER 3

Po zvolení príslušnej možnosti sa vykoná skok na podprogram, ktorý vykoná to isté čo podprogram s názvom Knight Rider 1, ibaže namiesto jednej diódy svietia súčasne 3 diódy.

8.8.10 VÝPIS KONFIGURÁCIÍ RIADIACICH REGISTROV

Po zvolení príslušnej možnosti v hlavnom menu, sa dostaneme do menu, v ktorom je možné si vybrať výpis z nasledujúcich 8 bitových registrov mikroprocesora: PSW, SCON a TMOD. Vybratý register je poslaný po sériovej linke obslužnému programu pre mikropočítač, ten ho rozdelí na jednotlivé bity a tak určí a vypíše konfiguráciu mikropočítača.

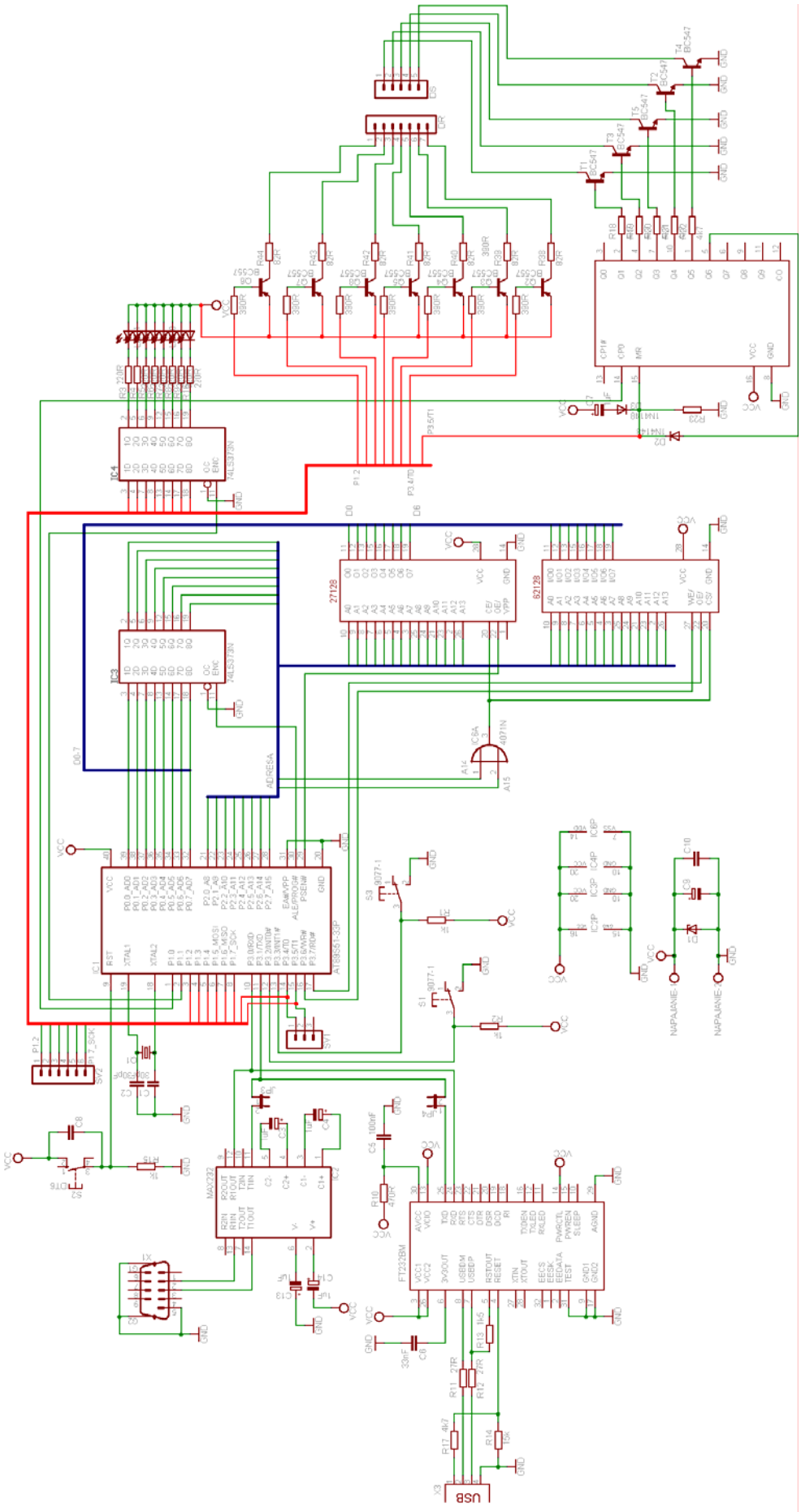
8.9 Zmeny oproti návrhu 8051

Podľa pôvodného návrhu sme obdržali vyrobenú dosku plošných spojov a osadili sme ju. Problémy nastali až po pripojení napájania a preto sme museli urobiť nasledujúce zmeny.

Prvým problémom bolo zapojenie dátovej zbernice. Chyba v návrhu spôsobila, že nebolo možné zabezpečiť v presne určený čas platné dáta na dátovej zbernici. Problém sa pokúšali odstrániť programovo, no tiež neúspešne. V pôvodnom návrhu nechali nepoužité porty procesora (P1.2, P1.3, P1.4, P1.5, P1.6, P1.7, P3.4, P3.5) a práve tie sme použili na

riešenie vyššie uvedeného problému. Od záchytného registra na diódy a aj od maticového displeja sme odpojili dátovú zbernicu a pripojili voľné porty procesora. Tým sa vyriešil problém s diódami, kedy sme vedeli vysvietiť požadovaný vektor. Pri maticovom nastal problém s intenzitou svietenia a preto sme prerobili vstup pre maticový displej. Odpory pripojené na kolektory a bázy tranzistorov BC557 sme vybrali a nahradili sme iba odpory pri bázach na veľkosť 390 Ohmov. Na bázy sme pripojili cez spomínané odpory porty procesora (iba sedem portov) a na kolektory tranzistorov sme pripojili napájacie napätie. Tým sme zlepšili svietivosť maticového displeja. Posledný problém, ktorý nastal sa opäť týkal maticového displeja. Programovo sme nevedeli zabezpečiť správne ovládanie svetelných stĺpcov displeja, čo súviselo s riadiacim obvodom pre displej (HC4017). Podľa pôvodného návrhu, spomínaný riadiaci obvod pre displej nemal pripojený vstup *reset* na procesor a preto sme ho nevedeli programom ovládať. Na riešenie tohto problému sme použili tiež jeden z voľných portov procesora (port P3.5).

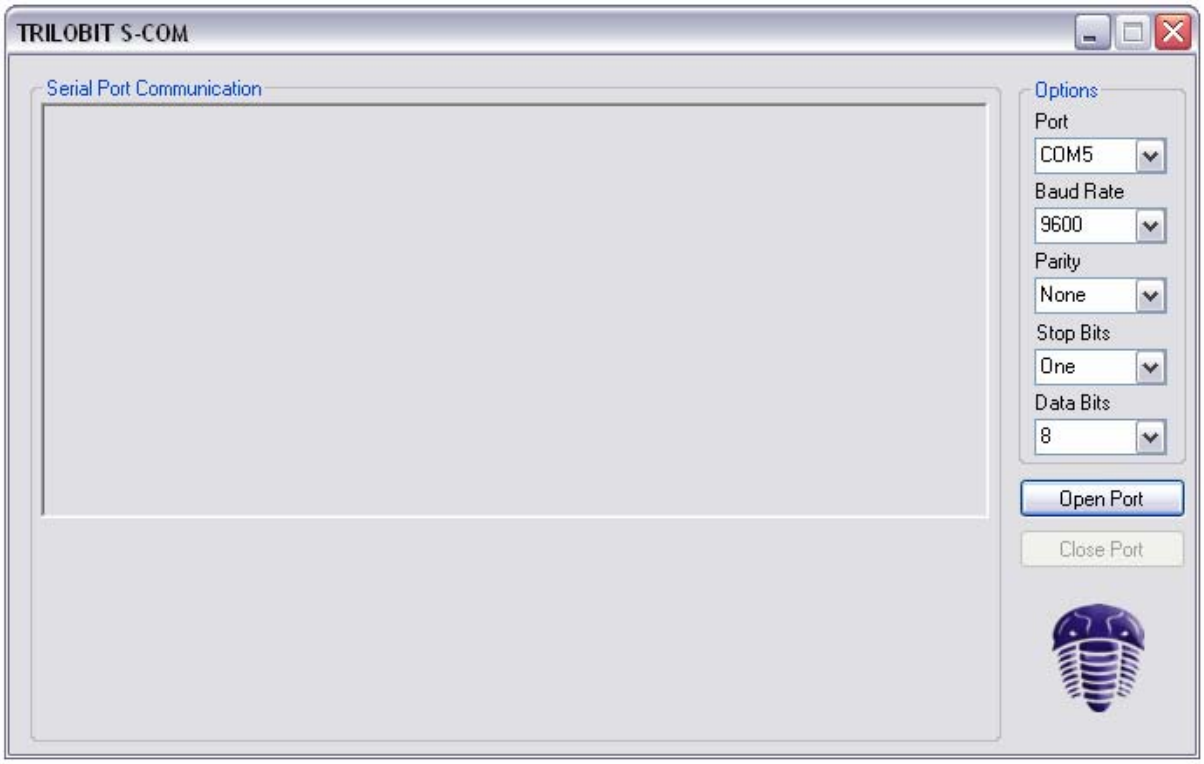
Všetky zmeny oproti návrhu sú vyznačené v schéme (nasledujúci obrázok) a sú zvýraznené červenou farbou.



8.10 Používateľská príručka obslužného programu pre PC

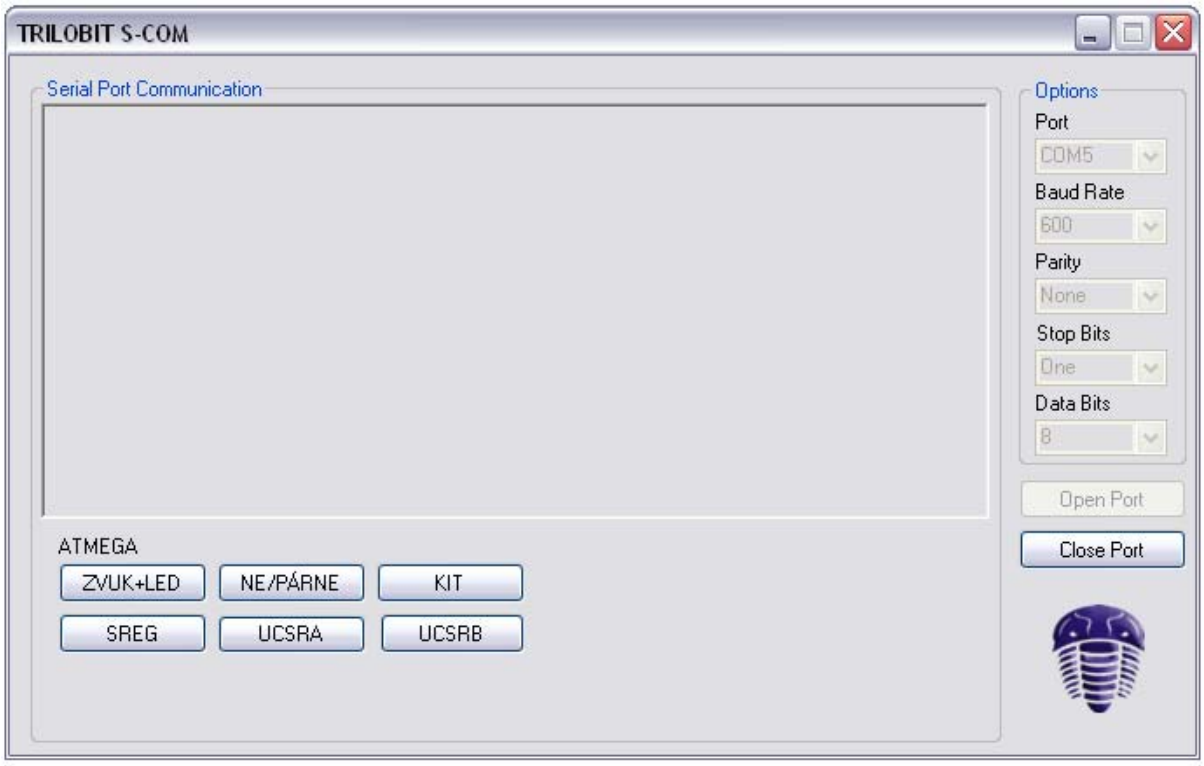
Pri prvotnom spustení programu sa zobrazí formulár, kde je potrebné vybrať procesor s ktorým budeme pracovať. To umožní programu nastaviť predvolené parametre na komunikáciu s daným mikropočítačom.

Po výbere sa zobrazí formulár v ktorom je potrebne stlačiť „Open Port“, čím sa otvorí port a nadväzuje sa komunikácia – čaká sa na odpoveď z mikropočítača.



Pokiaľ mikropočítač odpovedal na spojenie, formulár aktivuje ovládacie prvky pre daný mikropočítač.

8.10.1 Ovládanie ATMega16



Po úspešne nadviazanom spojení, sa odblokuje 6 tlačidiel, ktoré vykonávajú inštrukcie na procesore popísané v kapitole 8.7.3.

8.10.2 Ovládanie 8051

Po úspešne nadviazanom spojení, sa odblokuje 6 tlačidiel, ktoré vykonávajú inštrukcie na procesore popísané v kapitole 8.8.3.