

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2017

Petr Jirčík

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

MATICOVÝ DISPLEJ

Petr Jirčík

Bakalářská práce

2017

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr Jirčík**
Osobní číslo: **I13017**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**
Název tématu: **Maticový displej**
Zadávající katedra: **Katedra elektrotechniky**

Z á s a d y p r o v y p r a c o v á n í :

Vytvořte mikroprocesory řízený systém pro zobrazení textových zpráv.
Zobrazované textové zprávy mají délku 8 znaků. Každý znak je zobrazen pomocí maticového displeje 5x7 bodů s tím, že systém podporuje kódování znaků ASCII (v rozsahu kódů 32 až 126) a dovoluje také definici 8 uživatelských znaků.
Hlavní mikroprocesor komunikuje s počítačem pomocí sériového portu a s jednotlivými místy displeje pak vhodnou komunikační sběrnici, místa displeje jsou řízena dalšími mikroprocesory nebo jinými obvody, napájení displeje je zajištěno klasickým 5V adaptérem.
Z počítače se přes sériový port zadává textová zpráva pro zobrazení a uživatelsky definované znaky.

Rozsah grafických prací:

Rozsah pracovní zprávy: **40 stran A4**

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

MATOUŠEK, David. Aplikace mikrokontrolérů ATmega644. 1. vyd. Praha: BEN - technická literatura, 2013, ca 200 s. v různém stránkování. ISBN 978-80-7300-492-7.

VÁŇA, Vladimír. Mikrokontroléry ATMEL AVR: programování v jazyce C : popis a práce ve vývojovém prostředí CodeVisionAVR C. 1. vyd. Praha: BEN - technická literatura, 2003, 215 s. ISBN 80-730-0102-0.

Vedoucí bakalářské práce:

Ing. Bc. David Matoušek

Katedra elektrotechniky

Datum zadání bakalářské práce: **31. října 2016**

Termín odevzdání bakalářské práce: **12. května 2017**



Ing. Zdeněk Němec, Ph.D.
děkan



Ing. Jan Pídaní, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2016

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 4. 5. 2017

Petr Jirčík

PODĚKOVÁNÍ

Velice rád bych poděkoval vedoucímu mé bakalářské práce Ing. Bc. Davidu Matouškovi za odborné rady a pomoc, která mi byla velice nápomocná při zhotovení této bakalářské práce.

ANOTACE

Cílem této práce je návrh maticového displeje, který zobrazí textovou zprávu o délce 8 znaků. Každý znak je zobrazen na displeji o rozměrech 5x7 bodů. Systém podporuje zobrazení znaků ASCII v rozsahu kódů 32 až 126 a umožňuje definici 8 uživatelských znaků. Komunikace mezi počítačem a hlavním mikroprocesorem probíhá pomocí sériového portu. Další mikroprocesory slouží k ovládání jednotlivých míst displeje. Napájení je řešeno klasickým 5V adaptérem. V teoretické části je popsána funkce a řízení displeje a dále vlastnosti použitých mikroprocesorů. V praktické části je popsána realizace maticového LED displeje a programování mikroprocesorů.

KLÍČOVÁ SLOVA

matice, displej, LED dioda, mikroprocesor, sériový port, znak, programátor

TITLE

TheDot-matrix Display

ANNOTATION

Theaimofthisworkis design matrix display, whichdisplays a text messagewith a lengthof 8 characters. Eachcharacterisdisplayed on thescreenwithdimensions 5x7 points. Thesystemsupportsthe display of ASCII codes in therangeof 32 to 126 and allowsthedefinitionof user-8 characters. Communicationbetweencomputer and main microprocessor is via serial port. Othermicroprocessorsused to controlthe display ofindividualsites. Powerissolvedclassic 5V adapter.Thetheoretical part describesthefunction and controlthe display and propertiesofmicroprocessors.Thepractical part describestherealization matrix LED display and microprocessor programming.

KEYWORDS

matrix, display, LED diode, microprocessor, serial port, character, programmer

OBSAH

SEZNAM OBRÁZKŮ	10
SEZNAM TABULEK	11
SEZNAM ZKRATEK	12
ÚVOD	12
1 TEORETICKÁ ČÁST	14
1.1. LED MATICOVÝ DISPLEJ	14
1.2. MIKROPROCESOR ATMEGA 8.....	15
1.2.1. ZÁKLADNÍ VLASTNOSTI	16
1.2.2. OBVODY V MIKROPROCESORU ATMEGA8	16
1.3. MIKROPROCESOR ATMEGA16.....	17
1.4. TRANZISTOROVÁ POLE	18
1.4.1. DARLINGTONOVO ZAPOJENÍ TRANZISTORŮ	19
1.5. PROGRAMOVÁNÍ AVR.....	20
1.5.1. TYPY ROZHRAŇÍ	20
1.5.2. ZPŮSOBY PŘIPOJENÍ.....	20
1.5.4. ISP PROGRAMOVÁNÍ.....	21
1.6. JEDNOTKA USART (UNIVERSAL SYNCHRONOUS AND ASYNCHRONOUS SERIAL RECEIVER AND TRANSMITTER)	22
1.6.1. POPIS JEDNOTLIVÝCH REGISTRŮ	23
1.7. SBĚRNICE TWI (TWO WIRE INTERFACE).....	30
1.7.1. PRAVIDLA PŘENOSU	31
1.7.2. PRACOVNÍ POSTUP SBĚRNICE TWI	32
1.7.3. REGISTRY SBĚRNICE TWI.....	34
2 PRAKTICKÁ ČÁST	38
2.1. OBECNÝ POPIS ŘEŠENÍ	38
2.2. POSTUP PROGRAMOVÁNÍ MIKROPROCESORU	39
2.3. KOMUNIKACE PŘES SÉRIOVÝ PORT	41
2.3.1. NASTAVENÍ TERMINÁLU	41
2.4. POPIS SCHÉMAT	42
2.5. ODESÍLÁNÍ ZNAKU NA DISPLEJ.....	47
2.6. UŽIVATELSKÉ ZNAKY	48
2.7. POPIS KONEČNÉHO ŘEŠENÍ.....	48

2.8.	POPIS PROGRAMU	49
2.8.1.	PROGRAM PRO MASTER.....	49
2.8.2.	PROGRAM PRO SLAVE	51
2.9.	DESKA PLOŠNÝCH SPOJŮ (DPS).....	52
ZÁVĚR		53
POUŽITÁ LITERATURA		55
SEZNAM PŘÍLOH.....		56

SEZNAM OBRÁZKŮ

OBRÁZEK 1 – LED MATICOVÝ DISPLEJ 5X7	15
OBRÁZEK 2 - VNITŘNÍ SCHÉMA ZAPOJENÍ LED MATICOVÉHO DISPLEJE	15
OBRÁZEK 3 - ATMEGA8 V POUZDŘE PDIP28	17
OBRÁZEK 4 – ATMEGA16 V POUZDŘE PDIP40	18
OBRÁZEK 5 - POUZDRO ULN2003A	19
OBRÁZEK 6 - ROZMÍSTĚNÍ VÝVODŮ A VNITŘNÍ STRUKTURA SEDMINÁSOBNÉHO TRANZISTOROVÉHO POLE ULN2003A.....	19
OBRÁZEK 7 - DARLINGTONOVO ZAPOJENÍ TRANZISTORU	20
OBRÁZEK 8 - ISP KONEKTOR	21
OBRÁZEK 9 - DATOVÝ REGISTR UDR	23
OBRÁZEK 10 - REGISTR UCSRA	25
OBRÁZEK 11 - REGISTR UCSRB	26
OBRÁZEK 12 - REGISTR UCSRC	28
OBRÁZEK 13 - REGISTR UBRR.....	29
OBRÁZEK 14 - SBĚRNICE TWI	33
OBRÁZEK 15 - REGISTR TWBR.....	34
OBRÁZEK 16 - REGISTR TWSR	35
OBRÁZEK 17 - REGISTR TWCR.....	36
OBRÁZEK 18 - REGISTR TWDR.....	37
OBRÁZEK 19 - REGISTR TWAR.....	37
OBRÁZEK 20 - BLOKOVÉ SCHÉMA ŘEŠENÍ ZAPOJENÍ LED MATICOVÉHO DISPLEJE.....	39
OBRÁZEK 21 - UKÁZKA Z PROGRAMU „ATMEL STUDIO 7.0“	40
OBRÁZEK 22 - UKÁZKA Z PROGRAMU „EXTREME BURNER – AVR“	40
OBRÁZEK 23 - NASTAVENÍ TERMINÁLU REALTERM	41
OBRÁZEK 24 - SCHÉMA NAPÁJENÍ	42
OBRÁZEK 25 - SCHÉMA ZAPOJENÍ ATMEGA8 V OBVODU	42
OBRÁZEK 26 - SCHÉMA ZAPOJENÍ ATMEGA16 (SLAVE1) V OBVODU	44
OBRÁZEK 27 - SCHÉMA ZAPOJENÍ ATMEGA16 (SLAVE2) V OBVODU	46
OBRÁZEK 28 - OSAZENÁ DESKA PLOŠNÝCH SPOJŮ (DPS)	53

SEZNAM TABULEK

TABULKA 1 - TABULKA S OZNAČENÍM JEDNOTLIVÝCH BITŮ V REGISTRU	
UCSRA.....	24
TABULKA 2 - TABULKA S OZNAČENÍM JEDNOTLIVÝCH BITŮ V REGISTRU	
UCSRB.....	25
TABULKA 3 - TABULKA S OZNAČENÍM JEDNOTLIVÝCH BITŮ V REGISTRU	
UCSRC.....	26
TABULKA 4 - TABULKA S MOŽNOSTMI VÝBĚRU PARITY POMOCÍ BITŮ UPM1 A	
UPM0	27
TABULKA 5 - TABULKA S VÝBĚREM POČTU BITŮ POMOCÍ BITŮ UCSZ2, UCSZ1 A	
UCSZ0.....	27
TABULKA 6 - TABULKA S VÝBĚREM REŽIMU ZA POMOCI BITU UCPOL	27
TABULKA 7 - TABULKA S JEDNOTLIVÝMI VÝPOČTY PŘENOSOVÉ RYCHLOSTI A	
HODNOTOU REGISTRU UBRR.....	28
TABULKA 8 - TABULKA S OZNAČENÍM JEDNOTLIVÝCH BITŮ V REGISTRU	
UBRRH	29
TABULKA 9 - TABULKA S OZNAČENÍM JEDNOTLIVÝCH BITŮ V REGISTRU	
UBRRL.....	29
TABULKA 10 - PRESCALER BITY PRO TWI.....	34
TABULKA 11 - TABULKA S OZNAČENÍM JEDNOTLIVÝCH BITŮ V REGISTRU	
TWSR (STAVOVÝ REGISTR TWI)	35
TABULKA 12 - TABULKA S OZNAČENÍM JEDNOTLIVÝCH BITŮ V REGISTRU	
TWCR (KONTROLNÍ REGISTR TWI)	36

SEZNAM ZKRATEK

LED	dioda emitující světlo
ASCII	americký standardní kód pro výměnu informací
USB	univerzální sériová sběrnice
USART	sériová sběrnice se synchronním / asynchronním přenosem
I2C	dvouvodičová sběrnice
PCB	deska plošných spojů
PC	uživatelský počítač

ÚVOD

Displej je jednou z možností, jak se dá zobrazit námi požadovaná informace. V současné době se s tímto druhem zobrazení můžeme setkat například u televizoru, monitoru počítače nebo různých měřicích zařízení, kde slouží k zobrazení naměřených hodnot a třeba i grafů. Využitelnost displejů je velmi vysoká a jedná se tak o nejpoužívanější zobrazovací techniku vůbec. V některých aplikacích nám postačí jednodušší LED nebo číslicové displeje, ale u náročnějších aplikací, kde klademe důraz na vyšší kvalitu zobrazení, je nutné použít náročnější technologie.

Mezi nejrozšířenější typy displejů patří segmentové displeje, které se dělí například nasedmisegmentové, devítisegmentové nebo šestnáctisegmentové, a maticové LED displeje, které dělíme podle zapojení na dva druhy. Se společnou anodou a se společnou katodou. Konstrukčně se řadí mezi nejjednodušší, jelikož nejde o nic jiného než jen o spojené LED diody do matice. Maticový LED displej slouží většinou k zobrazení textového řetězce, například na informační tabuli, nebo k zobrazení času, například v MHD.

Cílem této práce je navrhnout a následně zrealizovat maticový LED displej, který dokáže zobrazit textovou zprávu o velikostiosmi znaků, kde je každý znak zobrazen na jednom displeji o rozměrech 5x7 bodů. Těchto displejů je použito osm, takže výsledný maticový LED displej má rozměry 40x7 bodů. Znaky, které můžeme na displeji zobrazit, jsou v rozsahu ASCII tabulky od 32 do 126. Dále si můžeme libovolně nadefinovat až osm uživatelských znaků. Takovým znakem se rozumí například srdíčko, domeček apod.

Komunikace mezi počítačem a hlavním mikroprocesorem probíhá pomocí sériového portu přes USB, kde je použit převodník USB/TTL. Jako hlavní mikroprocesor je použit programovatelný mikroprocesor od firmy Atmel Atmega8. Dále jsou použity dva programovatelné mikroprocesory od firmy Atmel Atmega16, pomocí kterých se dá komunikovat prakticky s každým jednotlivým místem displeje.

Mezi jednotlivými místy displeje a programovatelnými mikroprocesory se nachází bipolární tranzistorová pole ULN2003A. Jsou zde použity z důvodu ovládání většího počtu obvodů, které vyžadují větší budící proud než je výstupní proud z mikroprocesoru. Darlingtonovo zapojení tranzistorů dosahuje velmi vysokého proudového zesílení. Jedno tranzistorové pole je sedminásobné, tedy obsahuje sedm zapojení tranzistorů. Všechny těchto sedm zapojení tranzistorů může fungovat nezávisle na sobě jeden po druhém.

První část práce je teoretická. Zabývá se popisem displejů, jejich dělením a také jejich řízením a funkcí. Dále zde najdeme popis použitých tranzistorových polí ULN2003A a

také teoretický popis použitých programovatelných mikroprocesorů. Konkrétně Atmega8 a Atmega16. Je zde i popis sériové linky, přes kterou probíhá komunikace mezi hlavním mikroprocesorem a počítačem, a popis I2C komunikace, neboli TWI, přes kterou probíhá komunikace mezi jednotlivými mikroprocesory.

Druhá část práce je praktická. Zabývá se popisem celkové realizace práce od návrhu až po finální realizaci. Dále je zde popsán způsob komunikace mezi jednotlivými mikroprocesory a také finální program, kterým ovládáme, jak již zmíněnou komunikaci mezi mikroprocesory, tak i maticové displeje a způsob zobrazování znaků na nich.

1 TEORETICKÁ ČÁST

1.1. LED MATICOVÝ DISPLEJ

LED maticový displej je součástka, která slouží například k zobrazení textového řetězce a jeho posuvu nebo obrázků. Obsahuje LED diody uspořádané do dvojrozměrného pole tak, že dohromady tvoří matici. Rozměrově mezi nepoužívanější maticové displeje patří matice 8x8 a 5x7 diod.

Zobrazení jednotlivých znaků, ať už to jsou číslice, písmena nebo jiné jakékoliv znaky, se provádí postupným rozsvěcováním LED diod v jednotlivých sloupcích. Zobrazení se provádí tedy postupně sloupec po sloupci. Rychlost zobrazení je tak vysoká, že pouhým lidským okem vidíme, jak svítí celý displej, přitom ve skutečnosti svítí vždy jen jeden sloupec. Jednotlivé LED diody jsou umístěny ve speciálním plastovém pouzdře, které obsahuje vnitřní propojení LED diod. Ve spodní části maticového displeje jsou umístěny ve dvou řadách vývody pro připojení. Existují dva druhy zapojení maticových displejů: se společnou anodou nebo se společnou katodou.

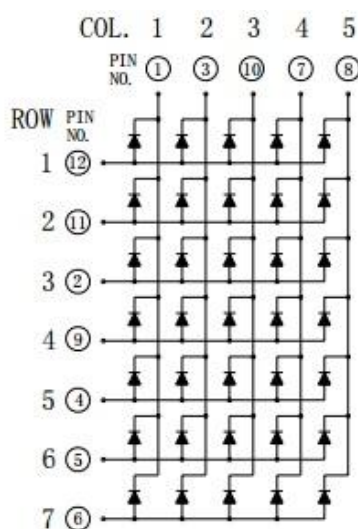
Samotný maticový displej je pouze maticové uspořádání a propojení LED diod. Proto je nutné použití mikroprocesoru pro řízení jednotlivých připojených vývodů.

V zapojení maticového displeje podle obrázku 2 je vidět, že katody LED diod jsou propojeny ve sloupcích a anody LED diod jsou společné v řádcích. Z tohoto zapojení je tedy patrné, že zobrazení znaku nebo nějakého obrazce není možné v jeden okamžik uskutečnit z důvodu rozsvícení více diod než je na požadovaný konkrétní znak potřeba. Tento problém se dá vyřešit tak, že LED diody budeme rozsvěcovat postupně buď po řádcích, nebo po sloupcích a to s dostatečně velkou frekvencí. Postupné rozsvěcování je řízeno předřazeným mikroprocesorem. Princip postupného rozsvěcování spočívá tedy v tom, že rozsvěcujeme sloupce postupně od prvního k poslednímu tak, že vždy svítí jen jeden sloupec. Nikdy nesvítí dva zároveň. Tohle celé se provádí s tak velkou frekvencí, aby lidské oko nezaznamenalo blikání LED diod. Efekt je tedy takový, že lidské oko si myslí, že je znak zobrazen konstantně, ale přitom tomu tak není.



1

Obrázek 1 – LED maticový displej 5x7



2

Obrázek 2 - Vnitřní schéma zapojení LED maticového displeje

1.2. MIKROPROCESOR ATMEGA 8

Mikroprocesor Atmega8 patří mezi 8-bitové nízkopříkonové mikroprocesory a jeho základem je AVR RISC architektura. Označení AVR je pro generaci mikroprocesorů typu RISC s harvardskou architekturou od firmy Atmel. Většina instrukcí se vykonává v průběhu hodinového cyklu a z toho vyplývá, že mikroprocesor Atmega8 dosahuje výkonu 1 MIPS na 1MHz.

¹KWM-20571ASB [online]. In: . [cit. 2017-05-02]. Dostupné z: <http://www.tme.eu/cz/details/kwm-20571asb/led-displeje-matrice/lucky-light/>

²1.9mm (0.7") 5x7 Super BrightRedDot Matrix LED Displays Technical Data Sheet [online]. In: . [cit. 2017-05-02]. Dostupné z: <http://www.tme.eu/cz/Document/51466a0713c1daa0e3d4f7e516813a89/KWM-20571ASB.pdf>

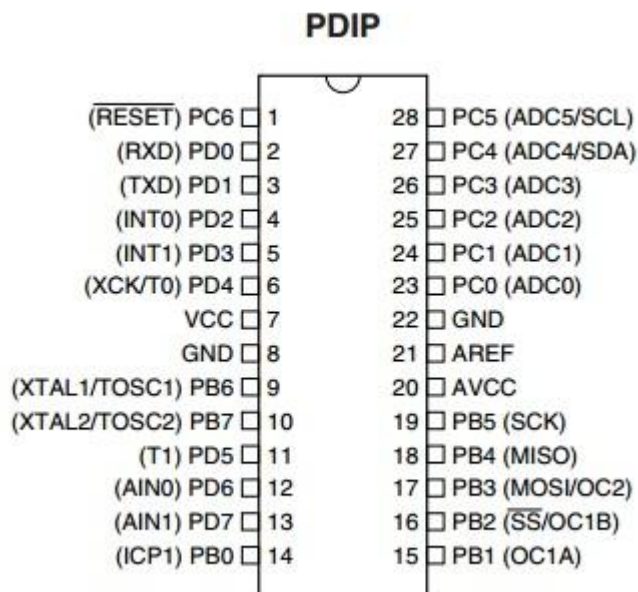
Programovatelný mikroprocesor Atmega8 je tzv. jednočipový počítač. To znamená, že jsou v něm obsaženy veškeré komponenty, které můžeme nalézt ve stolním počítači, akorát ve zmenšené verzi.

1.2.1. ZÁKLADNÍ VLASTNOSTI

- Mikroprocesor AVR:
 - 8-bitový
 - pokročilá RISC architektura
 - 32x8 univerzálních registrů
 - pracovní frekvence 1MHz
- 8kB In-System-Programmed (ISP) FLASH paměť:
 - FLASH paměť slouží k uložení programu
 - zachová se i po vypnutí
 - až 10 000 možností přepsání
- 1kB RAM (SRAM):
 - paměť volně přístupná pro aktuální provozní data
 - po odpojení napájení se smaže
- 512B EEPROM:
 - paměť sloužící pro dlouhodobé ukládání provozních dat
 - až 100 000 možností přepsání

1.2.2. OBVODY V MIKROPROCESORU ATMEGA8

- 2x 8-bit čítač/časovač
- 16-bit čítač/časovač
- Čítač reálného času s nezávislým oscilátorem
- 3x PWM obvody pro střídavou modulaci
- 4x 10-bit A/D převodník
- 2x 8-bit A/D převodník
- Programovatelný sériový USART – komunikace přes sériový port
- Watchdogtimer (WDT) – programovatelný dohlížecí obvod
- Analogový komparátor
- 23x programovatelný I/O obvod



3

Obrázek 3 - Atmega8 v pouzdře PDIP28

Mikroprocesor je napájen napětím o velikosti +5V a spotřeba proudu je velmi malá. Například při frekvenci 1MHz je spotřeba kolem 3mA. S rostoucí frekvencí roste spotřeba proudu a z toho vyplývá, že spotřeba je přímo úměrná frekvenci. Je patrné, že 8kB programovatelná paměť je zcela oddělená od 1kB RAM paměti a to se označuje jako Harvardská architektura. Obvody Atmegy mají více I/O obvodů, než má mikroprocesor vývodů. Je zde totiž předpoklad, že současně nevyužijeme všechny vývody mikroprocesoru najednou.

1.3. MIKROPROCESOR ATMEGA16

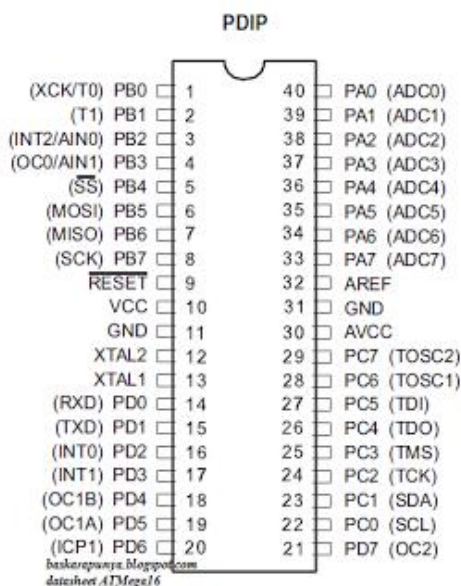
Stejně jako mikroprocesor Atmega8, tak i mikroprocesor Atmega16 patří mezi nízkopříkonové mikroprocesory, které jsou založeny na AVR RISC architektuře. Většina instrukcí se vykonává v průběhu hodinového cyklu a z toho vyplývá, že mikroprocesor Atmega16 dosahuje výkonu 1 MIPS na 1MHz.

Vyrábí se jak v klasickém THT provedení v pouzdře PDIP, tak i v SMD provedení v patici TQFP. Obsahuje vnitřní paměť FLASH o velikosti 16kB, paměť pro ukládání provozních dat EEPROM o velikosti 512B a paměť pro ukládání přechodových dat SRAM o velikosti 1kB. Mikroprocesor dále obsahuje 32 (4x8) I/O obvodů, které můžeme libovolně

³Atmega8A [online]. In: . [cit. 2017-05-02]. Dostupné z: <http://www.tme.eu/cz/Document/abe8b508d9fd51e5f8a0bd36702582e0/ATmega8A-DTE.pdf>

programovat a ovládat. Pracovní frekvence hodinového signálu mikroprocesoru Atmega16 je maximálně 16MHz. Dále mikroprocesor obsahuje další obvody jako například: A/D převodník, 16-bitový čítač, přerušení atd.

Mikroprocesor Atmega16 je v podstatě stejný jako mikroprocesor Atmega8, až na pár odlišností. Liší se vnitřní paměť RAM, kterou místo 8kB má 16kB, tedy dvojnásobnou. Dále v počtu vývodů, kterých má 40 oproti 28, tedy o 12 více a nakonec v počtu PWM kanálů, kterých má místo 3 o jeden více, tedy 4.

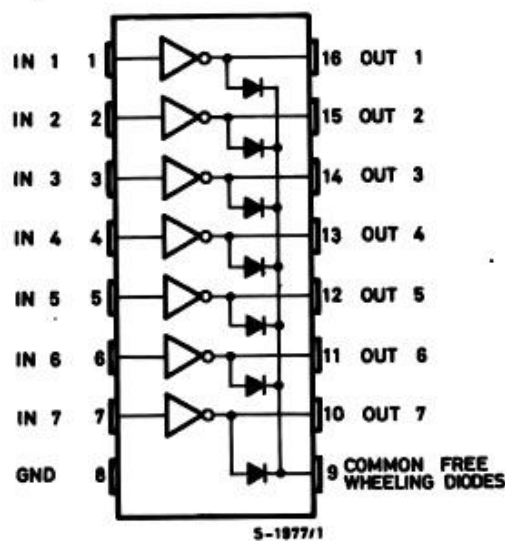


Obrázek 4 – Atmega16 v pouzdře PDIP40

1.4. TRANZISTOROVÁ POLE

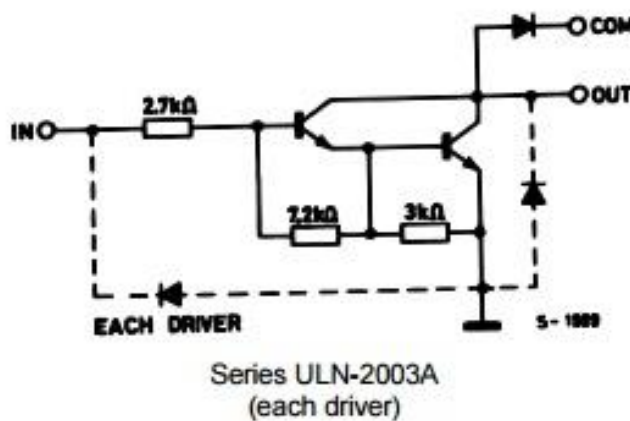
U aplikací číslicových obvodů je většinou podmínka na ovládání většího počtu obvodů, které potřebují větší budicí proud, než je například mikroprocesor schopen dodat. U takovýchto aplikací je vhodné použít tranzistorové pole, tedy sedminásobný spínací obvod. Vstupy jsou určeny pro běžné číslicové obvody např. TTL, CMOS, atd. Jejich výstupy jsou dimenzované napětově přibližně na 50V a proudově kolem 500mA. Na obrázku 5 je příklad tranzistorového pole ULN2003A a na obrázku 6 jeho vnitřní struktura.

⁴ATmega16A [online]. In: . [cit. 2017-05-02]. Dostupné z: <http://www.tme.eu/cz/Document/5b85697329d1e824e1960863439f91fd/ATmega16A-DTE.pdf>



5

Obrázek 5 - Pouzdro ULN2003A



6

Obrázek 6 - Rozmístění vývodů a vnitřní struktura sedminásobného tranzistorového pole ULN2003A

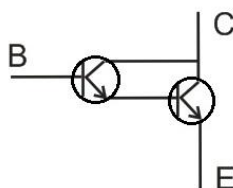
1.4.1. DARLINGTONOVO ZAPOJENÍ TRANZISTORŮ

Darlingtonovo zapojení tranzistorů se používá v aplikacích, kdy je potřeba velkého proudového zesílení. Toto zapojení si můžeme představit jako jeden tranzistor, který má velký

⁵ULN2003A [online]. In: . [cit. 2017-05-02]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.380-005.1.pdf>

⁶ULN2003A - SCHEMATIC DIAGRAM [online]. In: . [cit. 2017-05-02]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.380-005.1.pdf>

vstupní odpor způsobený velmi malým proudem přechozího tranzistoru, a také má velkou hodnotu proudového zesilovacího činitele, který je dán součinem jednotlivých proudových zesilovacích činitelů jednotlivých tranzistorů. Jednou z nevýhod zapojení je 2x větší napětí báze-emitor (U_{BE}), které je dáno součtem dílčích napětí $U_{BE1} + U_{BE2}$. Další nevýhodou je také větší hodnota saturačního napětí U_{CES} . Jako jednu z nevýhod můžeme považovat i velikostně větší zbytkové proudy.



Obrázek 7 - Darlingtonovo zapojení tranzistoru

1.5. PROGRAMOVÁNÍ AVR

V dnešní době je na výběr z několika typů rozhraní, přes která mohou být mikroprocesory AVR připojeny k programátorům a být naprogramovány. Každý mikroprocesor umí spolupracovat s několika rozhraními.

1.5.1. TYPY ROZHRAŇÍ

- ISP programování
- debugWIRE
- JTAG ladění
- HVPP (High-Voltage paralelní programování)
- HVSP (High-Voltage sériové programování)

Nezávisle na typu rozhraní existuje několik možností jak programátor připojit k PC, například pomocí USB portu nebo RS232 sériového portu, případně jejich vzájemnou kombinací.

1.5.2. ZPŮSOBY PŘIPOJENÍ

- USB port
- RS232 sériový port
- LPT paralelní port

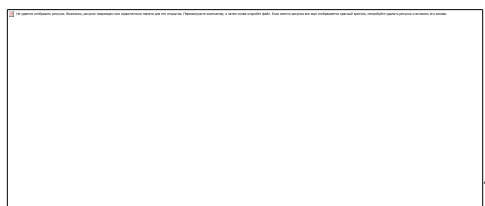
Volbou jedné nebo kombinací více možností propojení programátoru s PC a mikroprocesorem AVR je dáno jaké bude mít programátor možnosti. V jistých případech se programátor může skládat dokonce jen z několika pasivních součástek. Oproti tomu složitější programátory jsou realizovány pomocí jednoho mikroprocesoru AVR, v některých případech pomocí více mikroprocesorů AVR.

1.5.4. ISP PROGRAMOVÁNÍ

ISP programování neboli In-System Programming je vlastnost některých mikroprocesorů být naprogramovány přímo v zapojení.

Programátor pro ISP programování využívá 6 kontaktů:

- GND (zem)
- VTG (napájecí napětí od 1,8 do 5V)
- MOSI (Master Out Slave In, sériová linka od PC k AVR)
- MISO (Master In Slave Out, sériová linka od AVR k PC)
- SCK (hodiny sériových linek)
- RST (reset)



Obrázek 8 - ISP konektor

ISP programování mohou používat všechny mikroprocesory AVR k naprogramování. Naopak ISP programování neumožňuje ladění, k tomu se využívá například JTAG. Nejčastěji se používá ISP konektor s 2x3 piny a roztečí velikosti 2,54mm.

V dnešní době se používají ISP programátory s USB, což je velice výhodné, protože dnes každý počítač disponuje USB porty, na rozdíl například od COM portu, který třeba u notebooků najdeme stěží. USB port s sebou nese i další výhodu a to takovou, že je z něj zajištěno napájení.

⁷ISP Konektor [online]. In: . [cit. 2017-05-02]. Dostupné z: http://thomaspfeifer.net/einfaches_atmel_programmierkabel.htm_3.jpg

1.6. JEDNOTKA USART (UNIVERSAL SYNCHRONOUS AND ASYNCHRONOUS SERIAL RECEIVER AND TRANSMITTER)

Jedná se o jednu z nejpoužívanějších jednotek pro komunikaci, nejčastěji pro komunikaci mezi mikroprocesorem a uživatelským počítačem. Pro takovou komunikaci musíme použít vhodný převodník z rozhraní UART na rozhraní USB.

Základní schopností jednotky USART je ta, že umí pracovat v plně duplexním režimu. To znamená, že může data přijímat a zároveň vysílat. Pro způsob komunikace je na výběr ze dvou režimů, přičemž jeden je synchronní a druhý je asynchronní. Častěji používaným pracovním režimem je asynchronní režim, u kterého si můžeme dokonce nastavit různá kritéria, jako například zvolit si délku bitů od 5 do 9, počet tzv. stop bitů od 1 do 2 nebo nastavení parity.

Parita je vlastně paritní bit, který obsahuje informaci o celkovém počtu jedničkových bitů v řetězci a je především určen k detekci chyb v něm. Při komunikaci po sériové lince je možné nastavit paritu na jednu z více možností jako například:

- žádná parita (žádný paritní bit není posílán)
- lichá parita (lichý počet jedniček)
- sudá parita (sudý počet jedniček)
- 1 (hodnota paritního bitu je 1)
- 0 (hodnota paritního bitu je 0)

Ze struktury jednotky USART je patrné, že se skládá ze tří částí a to z: generátoru hodin, vysílače a přijímače. Generátor hodin je tvořen ze synchronizace externích hodin, které se používají v synchronním režimu „slave“ a z generátoru přenosové rychlosti.

Vysílač je tvořen jedním bufferem, sériovým posuvným registrem, generátorem parity a řídicí logikou pro více formátů sériových rámců. Bufferem rozumíme vyrovnávací paměť, která slouží k dočasnému uchovávání dat před jejich přesunutím na finální místo.

Nejsložitější část této struktury je přijímač, protože obsahuje jednotku, která zajišťuje obnovu dat a hodin. Přijímač i vysílač podporují stejné formáty rámců. Přijímač navíc detekuje chyby rámců, ztráty znaků a chyby parity.

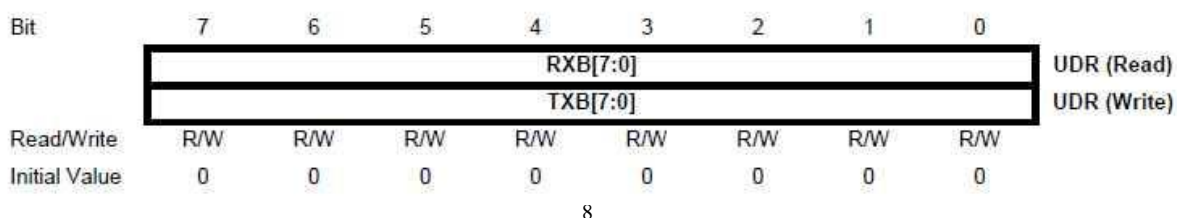
Při práci s jednotkou USART pracujeme s registry UDR, UCSRA, UCSRB, UCSRC a UBRR, který se skládá z registrového páru UBRRH a UBRRL. Datový registr UDR slouží k uložení a následnému přečtení dat. Registry UCSRA, UCSRB, UCSRC a UBRR slouží k nastavení sériového přenosu mikroprocesoru.

1.6.1. POPIS JEDNOTLIVÝCH REGISTRŮ

UDR (UART DATA REGISTER)

Při odeslání znaku nebo textového řetězce z počítače do mikroprocesoru pomocí sériového portu se data načtou do registru UDR, což je datový buffer. Z tohoto datového registru mohou být data následně přečtena. Datový registr UDR má tu vlastnost, že v jeden okamžik dokáže v sobě udržet pouze jeden datový byte, ten v něm zůstává až do doby, dokud není z datového registru UDR přečten. Po následném přečtení datového bytu se do datového registru UDR načte další datový byt, který byl uchováván v druhé úrovni bufferu. Z tohoto důvodu se tedy používá dvouúrovňový datový buffer.

Aby byl možný zápis dat do datového bufferu vysílače, tak se musí provést nastavení bitu UDRE v registru UCSRA na stav logické 1.



Obrázek 9 - Datový registr UDR

UCSRA (UART CONTROL AND STATUS REGISTER)

V registru UCSRA jsou obsaženy jak komunikační příznakové bity tak i bity, které informují o chybách.

⁸Registr UDR [online]. In: . [cit. 2017-05-02]. Dostupné z: <http://www.electroons.com/electroons/images/avr/UDR.jpg>

Tabulka 1 - Tabulka s označením jednotlivých bitů v registru UCSRA

Pozice bitu	Označení bitu
0	MPCM
1	U2X
2	PE
3	DOR
4	FE
5	UDRE
6	TXC
7	RXC

Bity na pozicích číslo 7 a 6, tedy RXC a TXC, obsahují informace o ukončení přijímání nebo vysílání znaku. Pokud je bit RXC nastaven na hodnotu logického stavu 1 znamená to, že v přijímací vyrovnávací paměti (bufferu) jsou data. Když dojde k následnému vyprázdnění bufferu, tak se bit RXC automaticky vynuluje. Pokud je bit TXC nastaven na hodnotu logického stavu 1 znamená to, že všechna data byla odeslána a v bufferu vysílače již žádná nejsou.

Bit na pozici číslo 5, tedy UDRE, nese informaci o tom, zda je vysílací vyrovnávací paměť (buffer) UDR prázdná a tím pádem i připravená pro zápis.

Jako další jsou popsány bity na pozicích 4,3 a 2, tedy FE, DOR a PE. Tyto tři bity jsou příznakové a nesou informaci o jednotlivých chybových stavech.

Další bity, které jsou na pozicích 4, 3 a 2, tedy FE, DOR a PE, jsou tzv. chybové bity. První z nich, tedy chybový bit FE, nese informaci o stavu chyby rámce. Pokud je jeho hodnota logického stavu nastavena na logickou 1 znamená to, že nebyl zaznamenán platný „stop bit“. Další chybový bit DOR nese informaci o ztracených datech. Třetí a poslední chybový bit PE je naplněn informací o stavu parity. Pokud je chybový bit PE nastaven na hodnotu logického stavu 1, tak to znamená, že hodnota vypočítané a přijaté parity se neshoduje.

Bit na pozici 1, tedy U2X, slouží k nastavení dvojnásobné přenosové rychlosti, avšak dojde i k zhoršení přesnosti. Nastavením tohoto bitu do stavu logické 1 se změní hodnota dělicího poměru pro výpočet přenosové rychlosti z hodnoty 16 na hodnotu 8.

Poslední bit na pozici 0, tedy MPCM, je pro nastavení tzv. víceprocesorového režimu, kdy se všechny rámce, které byly přijaty a neobsahují adresu, ignorují.

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

9

Obrázek 10 - Registr UCSRA

UCSRB (UART CONTROL AND STATUS REGISTER B)

Tento registr umožňuje nastavit povolení pro jednotlivá přerušení, tak i pro jednotlivé bloky.

Tabulka 2 - Tabulka s označením jednotlivých bitů v registru UCSRB

Pozice bitu	Označení bitu
0	TXB8
1	RXB8
2	UCSZ2
3	TXEN
4	RXEN
5	UDRIE
6	TXCIE
7	RXCIE

Různým nastavením bitů do stavu logické 1 na pozicích 7,6 a 5, tedy RXCIE, TXCIE a UDRIE, povolíme různá přerušení. Nastavením bitu RXCIE do stavu logické 1 se povolí přerušení po příjmu dat a dojde k nastavení RXC v registru UCSRA. Nastavením bitu TXCIE do stavu logické 1 se povolí přerušení po dokončení vysílání dat a dojde k nastavení TXC v registru UCSRA. Nastavením bitu UDRIE do stavu logické 1 se povolí přerušení po vyprázdnění registru UDR a dojde k nastavení UDRE v registru UCSRA. K tomu, aby mohlo být vyvoláno jakékoliv přerušení, musí být povolené tzv. globální přerušení, což se uskuteční příkazem „sei ()“.

Další bity, které jsou na pozicích 4 a 3, tedy RXEN a TXEN, slouží k povolení funkce přijímače a vysílače. Znamená to tedy, že pokud je stav bitu RXEN nastaven na hodnotu

⁹Registr UCSRA [online]. In: . [cit. 2017-05-02]. Dostupné z: <http://www.electrooons.com/electrooons/images/avr/UCSRA.jpg>

logické 1, tak je povolen přijímač u jednotky USART a pokud je stav bitu TXEN nastaven na hodnotu logické 1, tak je povolen i vysílač jednotky USART.

Bit na pozici 2, tedy UCSZ2, společně s bity na pozicích 2 a 1, tedy UCSZ1 a UCSZ0 u registru UCSRC, určuje konečnou délku datového znaku.

Bit na pozici 1, tedy RXB8, je tzv. devátý přijatý datový bit a musí být přečten ještě před přečtením registrem UDR.

Bit na pozici 0, tedy TXB8, je tzv. devátý vysílací bit a musí být zapsán ještě před zápisem do registru UDR.

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10

Obrázek 11 - Registr UCSRB

UCSRC (UART CONTROL AND STATUS REGISTER C)

Pro nastavení režimu komunikace slouží řídicí registr UCSRC, který je zároveň posledním řídicím registrem jednotky USART.

Tabulka 3 - Tabulka s označením jednotlivých bitů v registru UCSRC

Pozice bitu	Označení bitu
0	UCPOL
1	UCSZ0
2	UCSZ1
3	USBS
4	UPM0
5	UPM1
6	UMSEL
7	URSEL

Bit na pozici 7, tedy URSEL, rozhoduje o zápisu do registru UCSRC nebo do registru UBBRH. Tyto dva registry mají totiž stejnou adresu. Pokud má být proveden zápis do registru UCSRC, je potřeba nastavit hodnotu bitu URSEL do stavu logické 1.

¹⁰Registr UCSRB [online]. In: . [cit. 2017-05-02]. Dostupné z: <http://www.electroons.com/electroons/images/avr/UCSRB.jpg>

Bit na pozici 6, tedy UMSEL, slouží k výběru vhodného režimu komunikace.

Bity na pozicích 5 a 4, tedy UPM1 a UPM0, určují paritu.

Tabulka 4 - Tabulka s možnostmi výběru parity pomocí bitů UPM1 a UPM0

UPM1	UPM0	Parita
0	0	Žádná parita
0	1	Rezervováno
1	0	Sudá parita
1	1	Lichá parita

Bit na pozici 3, tedy USBS, určuje počet použitých „stop bitů“. Při nastavení hodnoty bitu na stav logické 1, jsou zvoleny dva „stop bity“ a při nastavení hodnoty bitu na stav logické 0, je zvolen pouze jeden „stop bit“.

Bity na pozicích 2 a 1, tedy UCSZ1 a UCSZ0, umožňují nastavit počet datových bitů.

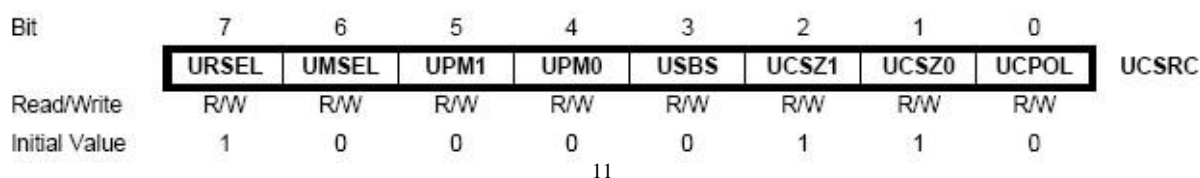
Tabulka 5 - Tabulka s výběrem počtu bitů pomocí bitů UCSZ2, UCSZ1 a UCSZ0

UCSZ2	UCSZ1	UCSZ0	Počet bitů
0	0	0	5
0	0	1	6
0	1	0	7
0	1	1	8
1	0	0	Rezervováno
1	0	1	Rezervováno
1	1	0	Rezervováno
1	1	1	9

Bit na pozici 0, tedy UCPOL, se používá pouze u synchronního režimu. Z toho vyplývá, že v asynchronním režimu je hodnota tohoto bitu nastavena na stav logické 0.

Tabulka 6 - Tabulka s výběrem režimu za pomoci bitu UCPOL

UCPOL	Změna dat na výstupu TXD	Navzorkování vstupních dat na RXD
0	Náběžná hrana XCK	Sestupná hrana XCK
1	Sestupná hrana XCK	Náběžná hrana XCK



11

Obrázek 12 - Registr UCSRC

UBRR (USART BAUD RATE REGISTER)

Registr UBRR je složen z registrového páru UBRRH a UBRRL. Pomocí těchto registrů se nastavuje přenosová rychlost. Výpočty přenosových rychlostí pro jednotlivé pracovní režimy a pro hodnoty registru UBRR se nachází v následující tabulce.

Tabulka 7 - Tabulka s jednotlivými výpočty přenosové rychlosti a hodnotou registru UBRR

Pracovní režim	Rovnice pro výpočet přenosové rychlosti	Rovnice pro výpočet hodnoty UBRR
Asynchronní normální režim (U2X = 0)	$BAUD = \frac{f_{osc}}{16 (UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16 BAUD} - 1$
Asynchronní režim s dvojnásobnou rychlostí (U2X = 1)	$ABUD = \frac{f_{osc}}{8 (UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8 BAUD} - 1$
Synchronní režim Master	$BAUD = \frac{f_{osc}}{2 (UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2 BAUD} - 1$

V dalších tabulkách je popis jednotlivých bitů v registru UBRRH a popis jednotlivých bitů v registru UBRRL.

¹¹Registr UCSRC [online]. In: . [cit. 2017-05-02]. Dostupné z: http://www.ermicro.com/blog/wp-content/uploads/2008/12/rs232_06.jpg

Tabulka 8 - Tabulka s označením jednotlivých bitů v registru UBRRH

Pozice bitu	Označení bitu
0	UBRR8
1	UBRR9
2	UBRR10
3	UBRR11
4	-
5	-
6	-
7	URSEL

Tabulka 9 - Tabulka s označením jednotlivých bitů v registru UBRRL

Pozice bitu	Označení bitu
0	UBRR0
1	UBRR1
2	UBRR2
3	UBRR3
4	UBRR4
5	UBRR5
6	UBRR6
7	UBRR7

Bit v registru UBRRH na pozici 7, tedy URSEL, rozhoduje zapsání dat. Data se zapisují do registru UBRRH nebo do registru UCSRC. Pro volbu zápisu dat do registru UBRRH musí být stav tohoto bitu na hodnotě logické 0.

Bit	15	14	13	12	11	10	9	8	
	URSEL	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

12

Obrázek 13 - Registr UBRR

¹²Registr UBRR [online]. In: . [cit. 2017-05-02]. Dostupné z: <http://www.electroons.com/electroons/images/avr/UBRR.jpg>

1.7. SBĚRNICE TWI (TWO WIRE INTERFACE)

Sběrnice TWI (TwoWire Interface – dvou vodičové rozhraní), kterou používají ke komunikaci mikroprocesory od firmy Atmel, je z praktického hlediska zcela totožná s tzv. sběrnici I2C (Inter Integrated Circuit) a liší se pouze názvem, jelikož název I2C je právně licencovaný.

Sběrnice I2C se skládá z jednoho páru vodičů, přes které se realizuje datové propojení mezi, ať už jedním nebo více, mikroprocesory, které se značí „Master“ a mezi dalšími podřazenými mikroprocesory, která se značí „Slave“. Tyto všechny mikroprocesory jsou připojené na tu samou sběrnici a komunikace mezi nimi je řízena pomocí adres jednotlivých mikroprocesorů. Je tedy jasné, že jak veškeré adresy mikroprocesorů, tak i veškerá data pro ně určená, se přenáší po stejných vodičích. Jednou z hlavních výhod této sběrnice je velice snadné propojení několika mikroprocesorů i vzhledem k návaznosti na rozšiřování dalšími mikroprocesory.

Další výhodou je již zmiňovaný dvou vodičový provoz, který je ke všemu navíc obousměrný. Dokáže data přijímat i vysílat. Jeden z vodičů je datový „SDA“ (serial data) a druhý z vodičů je hodinový „SCL“ (serial clock). Na jednu konkrétní sběrnici se může připojit několik mikroprocesorů. Kolik jich bude připojeno, však závisí na tom, jak jsou mikroprocesory adresovány. Pokud je adresace zvolena standardně, tedy 7 bitově, může být připojeno najednou až 128 mikroprocesorů. Avšak pokud je adresace zvolena rozšířeně, tedy 10 bitově, může být připojeno najednou až 1024 mikroprocesorů. Takového počtu připojených mikroprocesorů v praxi nikdy nedosáhneme a to hlavně z toho důvodu, že adresa mikroprocesoru se určí například pouze 3 bity (v takovém případě se na sběrnici může připojit maximálně 8 mikroprocesorů) a ne již zmíněnými 7 nebo 10 bity. Ve zvláštních případech adresa dokonce nelze určit a to z toho důvodu, že je již předem definovaná pro daný mikroprocesor. V tomto případě na sběrnici může být připojen pouze tento jeden mikroprocesor.

Další velice důležitou vlastností je přenosová rychlost TWI sběrnice. Možností pro volbu přenosové rychlosti je opět více. Přenosové rychlosti s frekvencí hodin o velikosti 100kHz se říká základní a používá se ve většině aplikací. Další možností je přenosová rychlost s frekvencí hodin o velikosti od 400kHz až do 1MHz. Je to tzv. vylepšená verze přenosové rychlosti, kterou však nepodporují všechny typy mikroprocesorů. Z toho vyplývá, že konečná přenosová rychlost musí brát zřetel na tzv. „nejpomalejší“ mikroprocesor, který je připojen ke sběrnici a jemu být přizpůsobena.

Pro správnou funkčnost sběrnice musí být oba vodiče, tedy jak „SDA“ tak i „SCL“ nastaveny na hodnotu logické 1. Tato podmínka bývá předem splněna použitými pull-up rezistory. Hodnota těchto rezistorů bývá z pravidla v řádech kiloohmů. Platí zde také tzv. nepřímá úměra, kdy za předpokladu zvyšující se frekvence klesá hodnota těchto pull-up rezistorů. Například při použití základní frekvence 100kHz se používají pull-up rezistory o velikosti 4k7.

Z praktického hlediska jsou dva způsoby jak zajistit komunikaci po sběrnici TWI. Jedním z nich je softwarová realizace a druhým je hardwarová realizace. Se softwarovou realizací, tedy tzv. „bit-banging“, která je mnohem jednodušší, se v praktických aplikacích setkáváme velice často. Výhoda této realizace spočívá v tom, že si uživatel může vybrat libovolné dva piny mikroprocesoru, které budou označeny jako SDA a SCL a následně po nich bude probíhat komunikace přes datovou sběrnici TWI. Nevýhodou této realizace je menší datová rychlost. Softwarová realizace se mnohem více používá u mikroprocesoru v pozici MASTER než pro mikroprocesor v pozici SLAVE, kde se skoro nepoužívá. Druhou možností je již výše zmíněná hardwarová realizace. Veškeré mikroprocesory AVR od firmy Atmel podporují komunikaci po sběrnici TWI, která je řízena za pomoci 5 registrů. Piny SDA a SCL, pomocí kterých probíhá přenos dat po sběrnici TWI se nachází u Atmega16 na pinech PC1 a PC0 a u Atmega8 na pinech PC4 a PC5.

1.7.1. PRAVIDLA PŘENOSU

Princip přenosu sběrnice spočívá v tom, že jeden mikroprocesor, který je hlavní, je nastaven jako tzv. „MASTER“ a další mikroprocesory jsou nastaveny jako tzv. „SLAVE“. Může nastat i ta varianta, kdy je více mikroprocesorů Master, je to tzv. multi-master varianta.

U mikroprocesoru MASTER je velice důležitým parametrem přenosová rychlost dat a u mikroprocesoru SLAVE je důležitým parametrem jeho adresa. Pro správnou funkčnost sběrnice musí mikroprocesor MASTER generovat hodinový signál o určité frekvenci vodičem SCL při každém přenosu. Jednoduše řečeno, pokud jeden mikroprocesor vysílá data, tak všechny ostatní mikroprocesory připojené na sběrnici data přijímají a pomocí adres se jednoduše určí, jakému mikroprocesoru jsou jaká data doručena. Ze všeho nejdříve se tedy musí zvolit adresa mikroprocesoru, kterému mají být doručena nebo z něho mají být přečtena data a také proces který se bude vykonávat, zda půjde o zápis nebo o čtení dat. To se určí bitem, který je součástí adresy a jde o tzv. bit „read“ nebo „write“ (R/W).

Obecně se dá říci, že pomocí mikroprocesoru MASTER se zadávají jednotlivé úkony, které se mají provést a přes sběrnici TWI se posílají do mikroprocesoru SLAVE, který jednotlivé úkony průběžně zpracovává a následně se sleduje stav vykonávání jednotlivých úkonů a zároveň se čeká na odezvu.

Celkem máme čtyři různé možnosti, jakými může každý mikroprocesor komunikovat a fungovat.

1. MASTER TRANSMITTER

- Mikroprocesor je ve funkci MASTER vysílače, kdy může vysílat data a adresu SLAVE mikroprocesoru, do kterého mají být data zapsána.

2. MASTER RECEIVER

- Mikroprocesor je ve funkci MASTER přijímače, kdy může přijímat data.

3. SLAVE RECEIVER

- Mikroprocesor je ve funkci SLAVE přijímače, kdy může přijímat jak adresu, tak i data.

4. SLAVE TRANSMITTER

- Mikroprocesor je ve funkci SLAVE vysílače, kdy může pouze vysílat data.

1.7.2. PRACOVNÍ POSTUP SBĚRNICE TWI

Zde jsou vysvětleny jednotlivě všechny stavy, se kterými se pracuje na sběrnici TWI. Jejich vhodnou kombinací se uskutečňuje přenos dat po sběrnici TWI.

1. Klidový stav

- Klidový stav znamená, že po TWI sběrnici neprobíhá žádný přenos dat ani není generován žádný hodinový signál. Tento stav je nastaven hodnotou logické 1, jak na vodiči SDA, tak i na vodiči SCL, a to díky tzv. pull-up rezistorům, které se nacházejí v zapojení mezi oběma vodiči a napájecím napětím.

2. START

- Pomocí tzv. start bitu se spustí přenos dat po sběrnici TWI. Prakticky to znamená, že logická hodnota vodiče SDA se změní ze stavu logické 1 do stavu logické 0

3. STOP

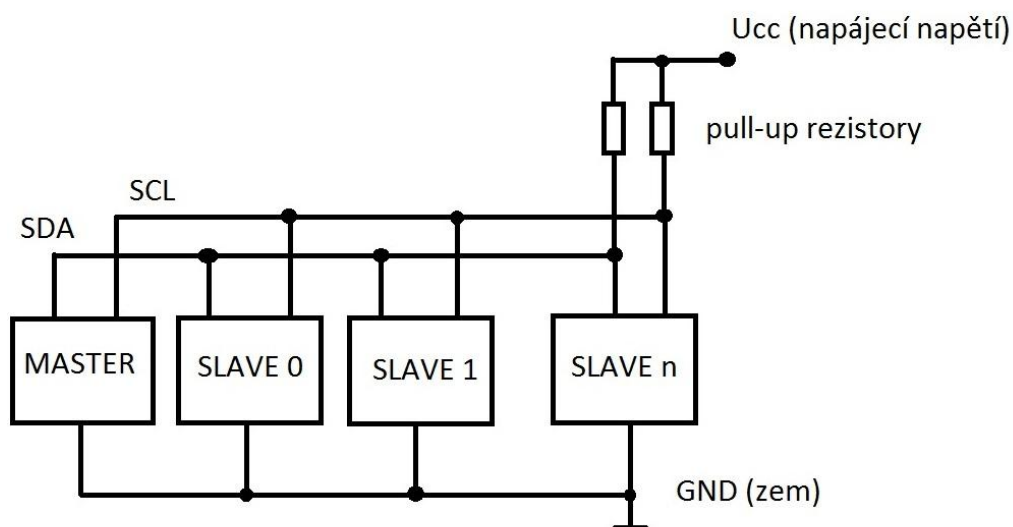
- Pomocí tzv. stop bitu se ukončí přenos po sběrnici TWI. Prakticky to znamená, že logická hodnota vodiče SDA se změní ze stavu logické 0 zpět do stavu logické 1.

4. Datový přenos

- Jednotlivá data se přenáší postupně sběrnici TWI po 1 bytu, tedy po 8 bitech od nejvyššího po nejnižší. Přenos je ovlivněn hodinovým signálem a to tak, že k přenosu jednoho bitu dojde vždy při každém pulzu SCL.

5. ACK

- ACK (acknowledge) je potvrzovací bit, který se používá pro potvrzení při přijímání dat. Tento bit je generován a zároveň odeslán mikroprocesorem, který přijímá vysílaná data, tedy přijímačem.



Obrázek 14 - Sběrnice TWI

1.7.3. REGISTRY SBĚRNICE TWI

I když je datový přenos po sběrnici TWI určen celkem pěti registry, tak ke správnému fungování sběrnice TWI je zapotřebí pracovat pouze se třemi. Zbývající dva, se kterými není potřeba neustále pracovat, se využívají pro nastavení rychlosti datového přenosu po sběrnici TWI a pro zadání SLAVE adresy, pokud je mikroprocesor ve stavu SLAVE.

TWBR (TWI BIT RATE REGISTER)

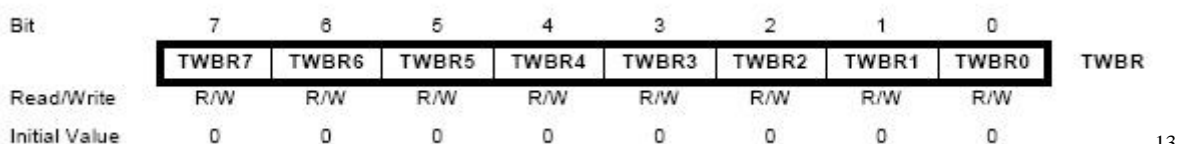
Pro nastavení rychlosti datového přenosu se využívá registr TWBR. Výsledná rychlost je dána vztahem:

$$f_{SCL} = \frac{f_{CPU}}{(16 + 2 * (TWBR) * 4^{(TWPS)})}$$

,kde f_{CPU} je frekvence mikroprocesoru a TWPS je hodnota prescaler bitů stavového registru TWI, tedy TWSR, která se nastavuje dvěma bity a to konkrétně bitem TWPS0 a bitem TWPS1 a jsou obsaženy v registru TWSR.

Tabulka 10 - Prescaler bity pro TWI

TWPS0	TWPS1	Prescaler bity
0	0	1
1	0	4
0	1	16
1	1	64



Obrázek 15 - Registr TWBR

TWSR (TWI STATUS REGISTER)

V registru TWSR, kromě již výše zmíněných bitů TWPS0 a TWPS1 pro nastavení prescaler bitů, se nachází dalších pět bitů, které jsou stavové a jsou značeny od TWS3 do

¹³Registr TWBR [online]. In: . [cit. 2017-05-02]. Dostupné z: http://2.bp.blogspot.com/-tGpoBp4-Yu0/UiFZckDKDsI/AAAAAAAAAECY/NBA10DrcS4Y/s1600/TWI_TWBR.png

TWS7. Tyto stavové bity se používají pro určení stavu sběrnice TWI a je možné z nich jednoduše odvodit, jaký pokyn byl po sběrnici TWI proveden naposledy a v jakém stavu je sběrnice TWI aktuálně.

Tabulka 11 - Tabulka s označením jednotlivých bitů v registru TWSR (stavový registr TWI)

Pozice bitu	Označení bitu
0	TWPS0
1	TWPS1
2	-
3	TWS3
4	TWS4
5	TWS5
6	TWS6
7	TWS7

Bit	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	14

Obrázek 16 - Registr TWSR

TWCR (TWI CONTROL REGISTER)

Registr TWCR je tzv. kontrolně řídicí registr a skrz něj se uskutečňuje řízení celé TWI komunikace.

Bit na pozici 7, tedy TWINT, poskytuje uživateli informaci o ukončení komunikace a čeká, jak uživatel zareaguje. Pokud se uživatel rozhodne pro pokračování v komunikaci, nastaví se stavový bit TWINT do stavu logické 1, kdy dojde k jeho vymazání.

Bit na pozici 6, tedy TWEA, slouží k nastavení odeslání informace o potvrzení přijetí zprávy či nikoliv.

Bity na pozicích 5 a 4, tedy TWSTA a TWSTO, se používají k vysílání pokynu START a následně pokynu STOP.

¹⁴Registr TWSR [online]. In: . [cit. 2017-05-02]. Dostupné z: http://www.ermicro.com/blog/wp-content/uploads/2009/02/i2c_05.jpg

Bit na pozici 3, tedy TWWC, je informační bit a informuje o výskytu chyby v komunikaci.

Bit na pozici 2, tedy TWEN, rozhoduje o zapnutí nebo následném vypnutí komunikace po sběrnici TWI.

Tabulka 12 - Tabulka s označením jednotlivých bitů v registru TWCR (kontrolní registr TWI)

Pozice bitu	Označení bitu
0	TWIE
1	-
2	TWEN
3	TWWC
4	TWSTO
5	TWSTA
6	TWEA
7	TWINT

Bit	7	6	5	4	3	2	1	0	
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	15

Obrázek 17 - Registr TWCR

TWDR (TWI DATA REGISTER)

Registr TWDR je datový registr sběrnice TWI. Pomocí tohoto registru dochází k vysílání dat po sběrnici i ke čtení dat ze sběrnice TWI.

¹⁵Registr TWCR [online]. In: . [cit. 2017-05-02]. Dostupné z: http://www.ermicro.com/blog/wp-content/uploads/2009/02/i2c_07_02.jpg

Bit	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	16

Obrázek 18 - Registr TWDR

TWAR (TWI ADDRESS REGISTER)

Registrem TWAR se nastavuje SLAVE adresa mikroprocesoru na sběrnici TWI, pokud je mikroprocesor ve stavu SLAVE.

Bit	7	6	5	4	3	2	1	0	
	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	TWAR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	0	17

Obrázek 19 - Registr TWAR

¹⁶Registr TWDR [online]. In: . [cit. 2017-05-02]. Dostupné z: http://3.bp.blogspot.com/-1nrFNJ-gf58/UiFafMISAAI/AAAAAAAAAEC0/SYA0lmhwgk/s1600/TWI_TWDR.png

¹⁷Registr TWAR [online]. In: . [cit. 2017-05-02]. Dostupné z: http://3.bp.blogspot.com/-AHD4XD-bHho/UiFatJnP5dI/AAAAAAAAAEC8/yAdxldclZUw/s1600/TWI_TWAR.png

2 PRAKTICKÁ ČÁST

2.1. OBECNÝ POPIS ŘEŠENÍ

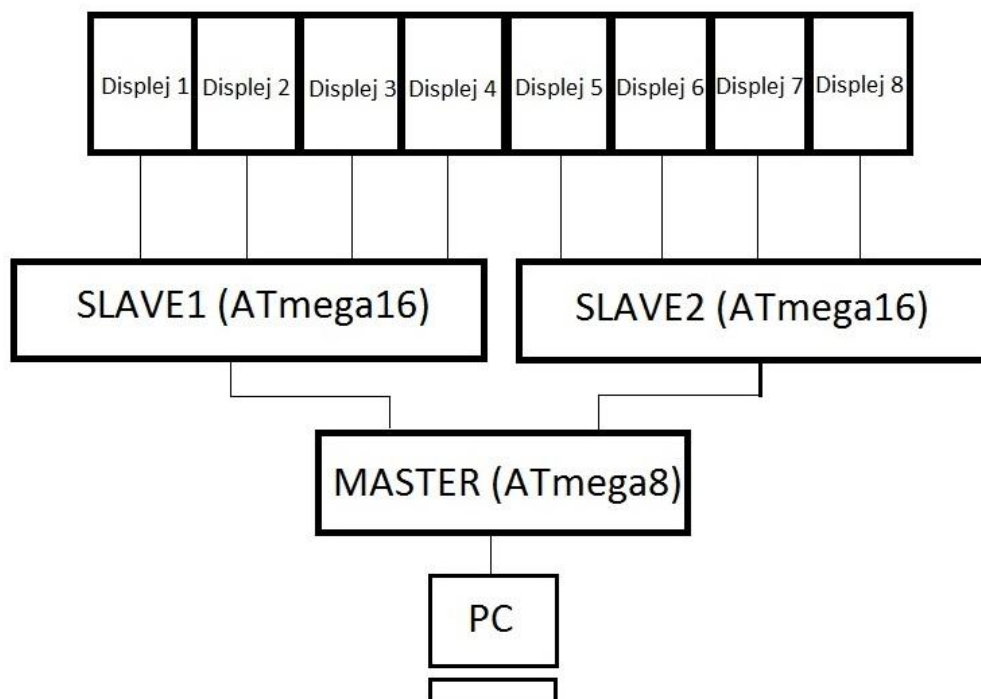
V praktické části práce je popsáno obecné řešení a následná realizace celé bakalářské práce. LED maticový displej je možné prakticky zrealizovat hned několika možnými způsoby za použití různých součástek. Výběr záleží pouze na požadavcích uživatele a na finančním omezení.

Jedním z nejpoužívanějších řešení pro realizaci LED maticového displeje je zapojení s řadičem MAX7219, které se používá v kombinaci s modulem Arduino. Nevýhodou u tohoto zapojení je cena řadiče MAX7219, vzhledem k tomu, že každý jednotlivý LED maticový displej má k sobě připojen právě jeden tento řadič. Z hlediska zadání této bakalářské práce, kde je zapotřebí osmi LED maticových displejů, by bylo toto řešení velmi neekonomické. Dalším z možných řešení realizace by bylo použití mikroprocesoru s dostatečným počtem vývodů a dostatečnou pamětí pro ovládání displejů. Tímto způsobem by nebylo splněno zadání bakalářské práce, vzhledem k požadovanému počtu mikroprocesorů a vzájemné komunikace mezi nimi.

Vzhledem k požadavku uživatelského řízení LED maticového displeje je nutné použít mikroprocesor pro jeho ovládání. Mikroprocesor musí mít dostatečný počet vývodů pro ovládání LED maticových displejů i dostatečně velkou paměť pro ukládání příkazů k jejich řízení. K takovýmto účelům jsou vhodné mikroprocesory od firmy Atmel z řady ATmega. K řešení této bakalářské práce, vzhledem k jejímu zadání, nebylo možné použít ani jednu z výše uváděných možností a pro realizaci bylo nutné zkonstruovat zcela jiné řešení.

Jako hlavní mikroprocesor, který je označován také jako „MASTER“, je zde použit mikroprocesor ATmega8, který vyrábí firma Atmel. Tento mikroprocesor za pomoci sériového portu řídí komunikaci mezi ním a počítačem. Z počítače jsou přes sériový port uživatelem odeslány znaky do hlavního mikroprocesoru, ať už to jsou znaky z ASCII tabulky nebo uživatelsky nadefinované znaky. Z mikroprocesoru „MASTER“ jsou znaky posílány dále po komunikační sběrnici do jednoho ze dvou podřadných mikroprocesorů, které jsou zde označovány jako „SLAVE1“ a „SLAVE2“. Těmto dvěma mikroprocesorům je jeden hlavní nadřazen, pro tyto účely zde jsou použity mikroprocesory ATmega16 od firmy Atmel. Každý z těchto dvou mikroprocesorů má na starost ovládání celkem čtyř jednotlivých displejů. Po odeslání první čtveřice znaků, jsou data odeslána pouze do prvního mikroprocesoru „SLAVE1“ a po odeslání další čtveřice znaků, jsou data odeslána do druhého mikroprocesoru „SLAVE2“. Program rozlišuje odesílání do mikroprocesorů „SLAVE1“ a „SLAVE2“ podle

adresy příslušného mikroprocesoru, kde adresa prvního mikroprocesoru je 0x20 a adresa druhého mikroprocesoru je 0x80. Pokud dojde ke shodě odesílané a očekávané adresy na straně přijímače, je uskutečněn přenos dat.



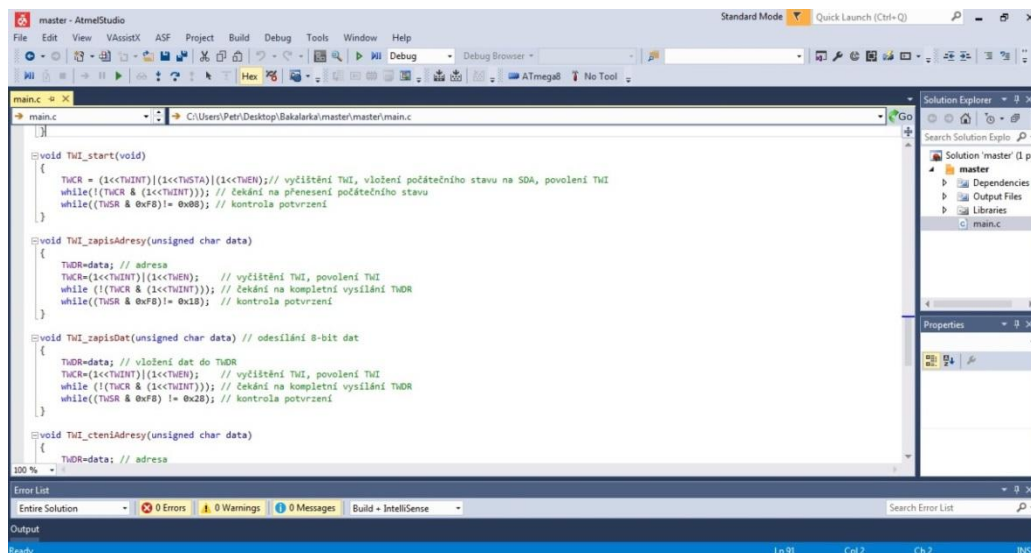
Obrázek 20 - Blokové schéma řešení zapojení LED maticového displeje

2.2. POSTUP PROGRAMOVÁNÍ MIKROPROCESORU

Mikroprocesory použité v zapojení maticového displeje jsou programovány jednotlivě a to za pomoci rozhraní ISP. Tento způsob programování je v dnešní době zcela běžný a má své určité výhody.

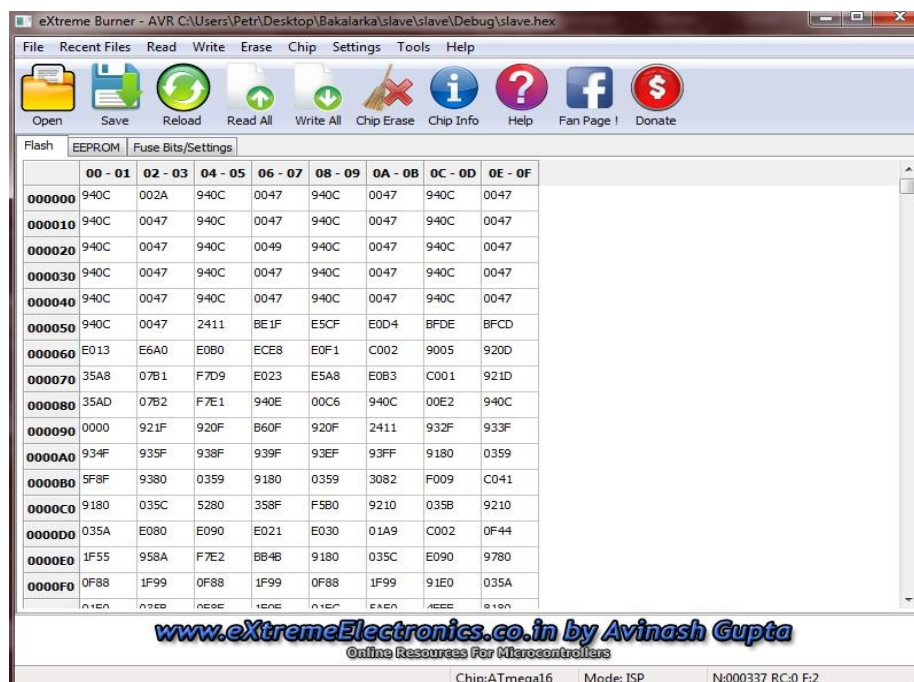
Jako zdroj dat pro zobrazení jednotlivých znaků na LED maticových displejích je zde používán počítač, přes který se uživatelem zadávají znaky, které se mají zobrazit. Je tedy nutné, aby byl počítač vždy připojen při zadávání znaků. Tímto způsobem lze pokaždé na displeji zobrazit jiné, uživatelem zvolené znaky a libovolně je měnit. Pokud by pro uživatele bylo dostačující zobrazovat na displeji pouze stále stejný text, lze mikroprocesory přeprogramovat, aby byly samy zdrojem dat a tím pádem by nebylo potřeba propojení s počítačem.

Pro vytváření jednotlivých řídicích programů, které se dále nahrávají do mikroprocesorů, se používá vývojové prostředí „Atmel Studio 7.0, ve kterém se používá programovací jazyk C.



Obrázek 21 - Ukázka z programu „Atmel Studio 7.0“

Jednotlivé mikroprocesory se programují pomocí programu „eXtremeBurner - AVR“ a programátoru „USBasp“, který je přímo určen na programování mikroprocesorů z řady ATmega od firmy Atmel a do počítače je připojen pomocí rozhraní USB.



Obrázek 22 - Ukázka z programu „eXtremeBurner – AVR“

2.3. KOMUNIKACE PŘES SÉRIOVÝ PORT

Přikomunikaci mezi počítačem a hlavním mikroprocesorem přes sériový port se v těchto aplikacích standardně používá tzv. Hyperterminál. Dokáže vysílat data na sériový port i data zněj přijímat. Pro účely této bakalářské práce byl zvolen terminál „Realterm“.

Před samotným použitím terminálu je nutné provést jeho základní nastavení pro správnou funkčnost. Toto nastavení se provádí v záložce „Port“. Dále už jen stačí vybrat správný port, kde je sériový port přes USB připojen a komunikace může být zahájena.

2.3.1. NASTAVENÍ TERMINÁLU

- Přenosová rychlost (počet bitů za sekundu - bit/s) – **9600**
- Počet datových bitů – **8 datových bitů**
- Parita - **žádná**
- Počet stop bitů - **1**
- Řízení toku hardwaru –**žádné**



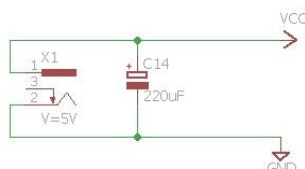
Obrázek 23 - Nastavení terminálu Realterm

Po provedení kompletního nastavení je důležité všechny změny potvrdit stiskem tlačítka „Change“. V záložce „Display“ je na výběr z různých formátů pro zobrazení dat, kde je vybrána možnost zobrazení formou ASCII znaků. K odesílání uživatelsky zvolených dat

přes sériový port slouží záložka „Send“. Je na výběr k odeslání buď číselné hodnoty, nebo přímo znaku z ASCII tabulky a je zde možnost i nastavení opakování odesílání dat.

2.4. POPIS SCHÉMAT

Po obvodové stránce se celkové schéma bakalářské práce skládá ze čtyř částí. První z nich je napájení maticového displeje.



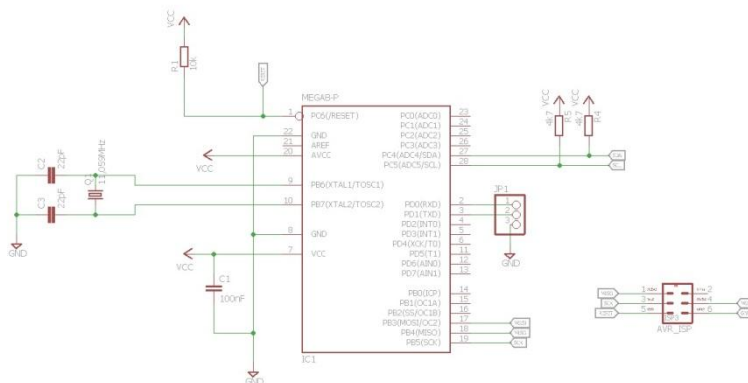
Obrázek 24 - Schéma napájení

Napájení je řešeno klasickým napájecím adaptérem o velikosti výstupního napětí 5V. Vzhledem k součástkám použitých v zapojení maticového displeje, byl zvolen napájecí adaptér o velikosti výstupního proudu 2A a výstupního výkonu 10W.

K DPS je adaptér připojen pomocí konektoru, k němuž je paralelně zapojen kondenzátor o velikosti 220uF a je označován jako tzv. blokovací kondenzátor. Je zde hlavně kvůli potlačení vysokofrekvenčního rušení signálů, které vzniká při připojení napájení a zabraňuje kmitání. Dále zkracují cestu střídavých proudů, které se neuzavírají přes zdroj, ale právě přes tento blokovací kondenzátor. Blokovací kondenzátor je zpravidla umístěn na DPS co nejbližší napájecímu konektoru, aby plošné spoje mezi nimi byly co nejkratší.

Díky tomuto blokovacímu kondenzátoru je zvýšena celková stabilita obvodu a dochází k potlačení vzájemného ovlivňování jednotlivých obvodů v zapojení skrz napájení.

Druhá část je schéma zapojení mikroprocesoru ATmega8 v obvodu a jeho příslušných vývodů, které jsou používány při realizaci bakalářské práce.



Obrázek 25 - Schéma zapojení ATmega8 v obvodu

Každý mikroprocesor potřebuje ke své správné funkčnosti zdroj hodinových impulzů o určité frekvenci, podle kterého obvody v mikroprocesoru pracují. Pro tyto účely je možné použít, v mikroprocesoru již zabudovaný, vnitřní obvod oscilátoru nebo je zde možnost připojit k mikroprocesoru externí obvod oscilátoru v podobě krystalu o určité uživatelem zvolené frekvenci.

V tomto případě je vnitřní obvod oscilátoru u ATmega8 nastaven od výrobce na hodnotu 1MHz. Pro správnou funkčnost mikroprocesoru v tomto zapojení je nutné použít externí zdroj frekvence, v podobě krystalu, a to o velikosti 11,059MHz. Připojení je provedeno pomocí vývodů mikroprocesoru XTAL1 a XTAL2. Dále jsou na obou přívodech krystalu umístěny kondenzátory o velikosti 22pF a to z důvodu stabilní frekvence. Tato realizace je také naznačená v datasheetu mikroprocesoru. Je tedy jasné, že v tomto případě se frekvence oscilátoru rovná frekvenci krystalu.

Pro samotné programování paměti mikroprocesoru jsou zde použity vývody MOSI, MISO, SCK a GND. Jedná se tedy o variantu ISP programování, která je již popsána v teoretické části.

Vývody SDA a SCL se v tomto případě používají pro komunikaci mezi jednotlivými mikroprocesory. Jedná se o již dříve popsanou TWI komunikační sběrnici. K těmto vývodům jsou připojeny ještě navíc tzv. pull-up rezistory, jejichž funkce je důležitá při vzájemné komunikaci mezi mikroprocesory.

Komunikace mezi hlavním mikroprocesorem a uživatelským počítačem je realizována pomocí sériové linky, pro kterou se používají vývody RXD, TXD a GND. Pin RXD slouží jako klasický přijímač. Přijímá data odeslaná z počítače uživatelem a posílá je dále do mikroprocesoru. Pin TXD má naopak funkci vysílače. Pokud je potřeba, vysílá data z mikroprocesoru do počítače, kde mohou být následně zobrazena nebo jinak zpracována.

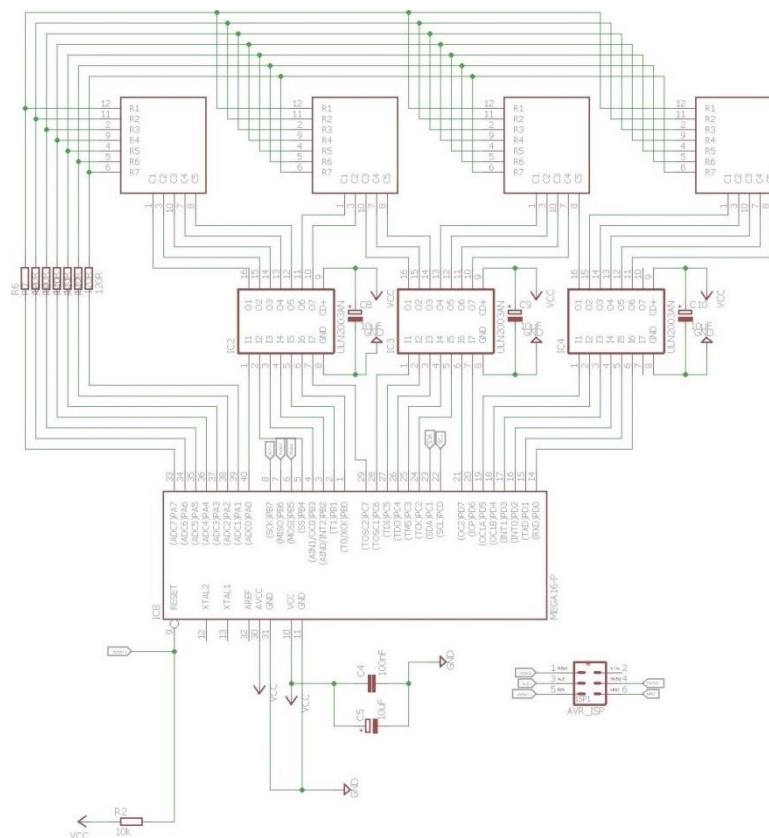
Pro správnou funkčnost sériové linky je nutné také vhodně nastavit tzv. fuse bity. V základní verzi jsou nastaveny na hodnoty LOW: 0xE1 a HIGH: 0xD9. Při tomto nastavení, ale nedochází k přenosu, který požadujeme, a sériová linka je tak nefunkční. Pro správnou funkčnost je tedy zapotřebí tyto „fuse bity“ změnit a to přesně na hodnoty LOW: 0xFF a HIGH: 0xC9. K této změně dochází s ohledem na změnu oscilátoru z interního na externí a na vnitřní nastavení mikroprocesoru.

Třetí část je schéma zapojení mikroprocesoru ATmega16 (SLAVE1) v obvodu a všech jeho příslušných obvodů, které jsou používány při realizaci.

I tento mikroprocesor musí být pro svoji správnou funkčnost naprogramován, jako každý jiný, a to za pomoci vývodů MISO, MOSI, SCK a GND. Těmito vývody je připojen programátor, kterým se mikroprocesor programuje. Používá se opět varianta ISP programování.

Na rozdíl od mikroprocesoru ATmega8, kde je použit externí oscilátor v podobě krystalu, je zde plně dostačující vnitřní oscilátor o velikosti frekvence 1MHz, který je v mikroprocesoru od výrobce předem zabudovaný.

Vývody SCL a SDA jsou zde opět určeny, jako u předchozího mikroprocesoru, pro vzájemnou komunikaci mezi jednotlivými mikroprocesory. Jedné se TWI komunikační sběrnici.



Obrázek 26 - Schéma zapojení ATmega16 (SLAVE1) v obvodu

Nejdůležitější funkcí mikroprocesoru ATmega16 v tomto zapojení je ovládání jednotlivých LED maticových displejů a jak již bylo dříve zmíněno, ovládá 4 displeje najednou.

Pomocí jednotlivých vývodů mikroprocesoru jsou ovládána konkrétní místa na displejích nezávisle na sobě. Pro ovládání jednotlivých řádků displejů jsou zde piny PA1 až PA7, přes které jsou ovládány řádky všech čtyř displejů. Nastavením těchto pinů do stavu

logické 1, jsou displeje připraveny zobrazovat data z mikroprocesoru o jednotlivých znacích. LED diody v displejích jsou zapojeny v režimu se společnou anodou a čekají na data, která přijdou pro každý sloupec zvlášť.

Například první displej je ovládán piny PA0, PB4, PB3, PB2 a PB1, kde každý pin posílá data, z mikroprocesoru na displej, pro jeden určitý sloupec. Pomocí funkce `data`, která jsou pro každý znak jiná, se postupně vykreslí znak na displeji a zůstane na něm vyobrazen. Jednotlivé displeje jsou označovány číselně od 1 do 4.

Každému mikroprocesoru je předřazeno tranzistorové pole ULN2003A, přes které prochází postupně všechna data určená pro příslušné sloupce. Jsou zde vloženy z důvodu většího budícího proudu, který je potřebný pro zobrazení znaků na displejích, než je schopen dodat samotný mikroprocesor.

Výpis používaných pinů pro jednotlivé displeje:

1. displej – PA0, PB4, PB3, PB2, PB1
2. displej – PB0, PC7, PC6, PC5, PC4
3. displej – PC3, PC2, PD7, PD6, PD5
4. displej – PD4, PD3, PD2, PD1, PD0

V základní verzi mikroprocesoru ATmega16 jsou tzv. fuse bity nastaveny na hodnoty LOW: 0xE1 a HIGH: 0x99. Při takovém nastavení jsou piny PC2, PC3, PC4 a PC5 blokovány z důvodu použití JTAGu, který je povolen. Pro povolení těchto pinů je nutné nastavit „fuse bity“ tak, aby se JTAG nepoužíval a to přesně na hodnoty LOW: 0xE1 a HIGH: 0xD9. Tímto nastavením přestanou být piny blokovány a je možné s nimi pracovat.

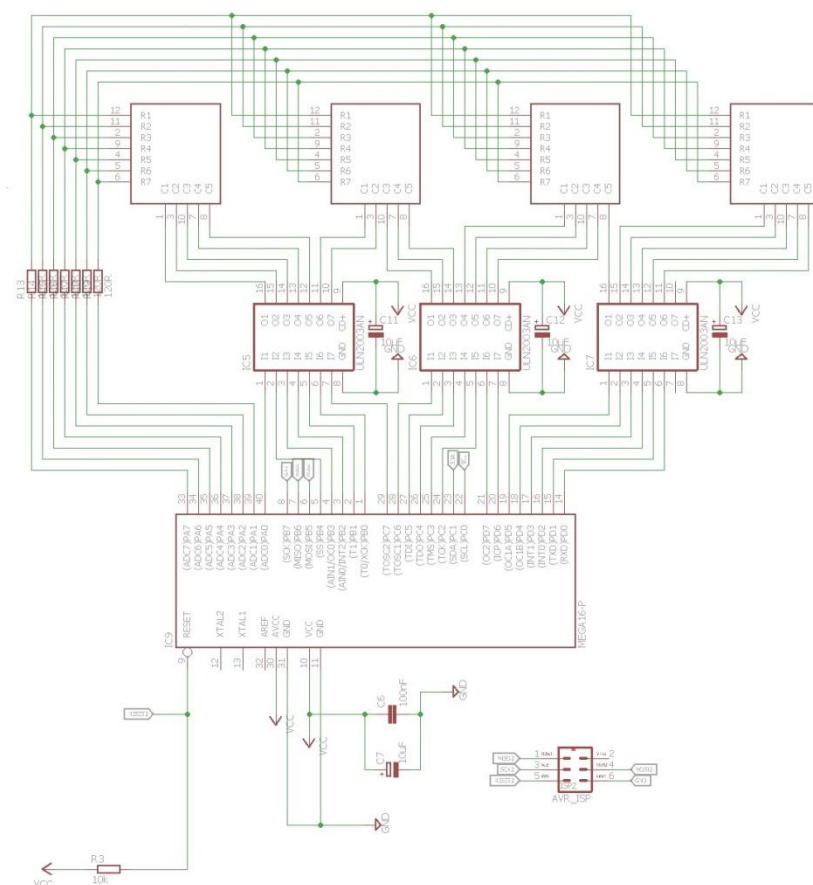
Čtvrtá část je schéma zapojení mikroprocesoru ATmega16 (SLAVE2) v obvodu a všech jeho příslušných obvodů, které jsou používány při realizaci.

Tato část je téměř shodná s předchozí třetí částí. Pro správnou funkčnost musí být také naprogramován a to pomocí ISP programování, ke kterému jsou potřeba vývody MISO, MOSI, SCK a GND.

Plně dostačující je zde také použití vnitřního oscilátoru o velikosti frekvence 1MHz, který je výrobcem již zabudován v mikroprocesoru.

Pro vzájemnou komunikaci mezi mikroprocesory je zde také použita TWI sběrnice, která využívá vývody SDA a SCL.

Princip ovládání displejů je také shodný s předešlou částí. Jednotlivé displeje jsou označovány číselně od 5 do 8.



Obrázek 27 - Schéma zapojení ATmega16 (SLAVE2) v obvodu

Stejně je zde i zapojení a funkčnost tranzistorových polí ULN2003A. Samotný mikroprocesor není schopen sám dodat dostatečně velký budící proud, a proto zde musí být tranzistorová pole použita pro jeho zvětšení.

Výpis používaných pinů pro jednotlivé displeje:

- 5. displej – PA0, PB4, PB3, PB2, PB1
- 6. displej – PB0, PC7, PC6, PC5, PC4
- 7. displej – PC3, PC2, PD7, PD6, PD5
- 8 displej – PD4, PD3, PD2, PD1, PD0

I v tomto případě je nutná změna tzv. fuse bitů pro správnou funkčnost mikroprocesoru a všech jeho pinů, které byly použity. Původně jsou nastaveny na hodnoty LOW: 0xE1 a HIGH: 0x99 a je potřeba je změnit na hodnoty: LOW: 0xE1 a HIGH: 0xD9.

2.5. ODESÍLÁNÍ ZNAKU NA DISPLEJ

Jednou z nejdůležitějších vlastností obvodu je schopnost odesílat data, která charakterizují určitý znak, a následně znak zobrazit na displeji. Tento celý postup je rozdělen do jednotlivých kroků, které na sebe navzájem navazují.

Nejdříve je nutné, aby skrz počítač uživatel zadal požadovaný znak, který chce zobrazit a následně jej odeslat. V tomto případě se může jednat o libovolný znak z ASCII tabulky v rozsahu kódu od 32 do 126, nebo o již předem nadefinovaný uživatelský znak. Odeslaný znak je následně přijat hlavním mikroprocesorem ATmega8, který rozhodne na který displej je znak odeslán a to podle aktuálního pořadí odeslaného znaku. Nejdříve jsou odeslané čtyři znaky na mikroprocesor SLAVE1 a další čtyři znaky jsou odeslané na mikroprocesor SLAVE2.

Veškeré znaky z ASCII tabulky, které se dají na displejích zobrazit, jsou uloženy v poli znaků s názvem „font1“, kde každý jednotlivý znak má svá data pro jeho následné zobrazení.

Po přijetí znaku mikroprocesorem ATmega16 je funkcí rozhodnuto o výběru dat pro požadovaný znak. Pokud je tento znak obsažen v ASCII tabulce kódů, je následně vykreslen na displej, který je aktuálně v pořadí.

Aby znaky zůstaly vykreslené na displejích, musí zde být použita funkce pro opakování, jinak by znaky na displejích pouze „problikly“. Nabízí se zde možnost použít nekonečnou smyčku opakování ve tvaru „while(1)“. Při použití této varianty je sice docíleno efektu, že znaky na displejích zůstávají vykresleny, ale nadále již nejdou „přepsat“ jinými znaky. To jde pouze možností úplného resetu, tedy odpojením a opětovným připojením napájení. Tato možnost je tedy velice nepraktická a nepoužívá se. Jako mnohem lepší možnost řešení tohoto problému se jeví použití tzv. časovače, který se v těchto případech zcela běžně používá a musí být u něj provedena různá nastavení jako například nastavení prescaleru, povolení přerušování přetečení a povolení globálního přerušování. V tomto případě je prescalernastaven na hodnotu F_CPU/1024. Následuje funkce pro indikaci přetečení časovače, která zajistí opakování celé funkce vždy v případě přetečení předem stanovené hodnoty.

Do programu bylo nutné zahrnout i bitové posuny z důvodu správného zobrazení znaků na displejích. Většina displejů je totiž ovládána více porty, například displej 1 je ovládán porty PORTA a PORTB, přes které se následně posílají data pro zobrazení znaků.

Tímto způsobem se postupně odesílají data jednotlivých znaků na všechny displeje, které na nich zůstávají zobrazeny. Po zaplnění všech displejů se funkce přesouvá opět na samotný začátek a odesláním dalších znaků se předchozí data přemazávají a nahrazují se postupně novými. Takto lze přepsat všechny jednotlivé displeje a zobrazit zcela odlišný text než byl předchozí. V případě potřeby úplného vynulování displejů a zahájení celého procesu od začátku, je nutné provést restart celého zapojení a to odpojením a následným připojením napájení.

2.6. UŽIVATELSKÉ ZNAKY

Nejen že tento LED maticový displej dokáže zobrazit jednotlivé znaky z ASCII tabulky, ale je zde také možnost, aby si uživatel, který se zařízením skrz počítač komunikuje, nadefinoval libovolný znak na kterémkoliv z displejů sám podle svého vlastního uvažování.

Definice uživatelských znaků musí být nějakým způsobem odlišena od standartního zadávání ASCII znaků. Program je naprogramován tak, že po odeslání znaku \$ (dolar) je režim zobrazování přepnut do režimu definování uživatelského znaku a čeká na jednotlivé hodnoty znaku. Následně uživatel zadává jednotlivé hodnoty znaku po řádcích, které se ihned po odeslání vykreslí na požadovaném displeji. Takto uživatel postupně odesílá hodnoty znaku pro všechny řádky displeje. Po odeslání posledního (sedmého) řádku se definice uživatelského znaku ukončí a uživatel může dále pokračovat v zadávání ASCII znaků z tabulky na dalších displejích nebo opět poslat znak \$ (dolaru) pro opětovné zadávání uživatelského znaku, ale tentokrát na následujícím displeji. Tímto způsobem můžou být uživatelské znaky vyobrazeny na všech 8 displejích najednou, nebo může dojít ke vzájemné kombinaci ASCII znaků a uživatelsky definovaných znaků. Tato volba již záleží pouze na uživateli. Příklad dat pro uživatelský znak může být následující: 0x04, 0x0E, 0x1F, 0x1F, 0x1F, 0x0E, 0x04. Data se odesílají postupně po jedné hodnotě.

2.7. POPIS KONEČNÉHO ŘEŠENÍ

Konečné řešení této práce je zrealizováno pomocí mikroprocesorů z řady ATmega a vzájemné komunikace mezi nimi. Znaky, které se mají zobrazit na displeji, se uživatelem zadávají do počítače a skrz hyperterminál, který je v něm, se data dále posílají pomocí

převodníku USB-USART a sériového portu do hlavního mikroprocesoru ATmega8, který se označuje jako „MASTER“ a má na starost řízení celého zapojení, aby fungovalo dle předpokladů. Z tohoto hlavního mikroprocesoru se data dále posílají pomocí komunikační sběrnice TWI do „podřadných“ mikroprocesorů ATmega16, které se označují jako „SLAVE1“ a „SLAVE2“. Jejich výběr je ovládán pomocí adresy jednotlivých mikroprocesorů. Každý z těchto mikroprocesorů ovládá 4 LED maticové displeje, na kterých se zobrazují uživatelem zadané znaky. Mezi mikroprocesory ATmega16 a LED maticovými displeji jsou umístěna tranzistorová pole, která jsou připojena jednotlivě ke všem LED maticovým displejům.

Pro kompletní realizaci práce je navržena a zhotovena deska plošných spojů (DPS). Tato DPS slouží jak k propojení všech jednotlivých modulů a propojení komunikace mezi nimi, tak i jako nosný substrát všech modulů zařízení. Celá deska je navržena a zrealizována jako dvouvrstvá DPS a jsou na ní tzv. prokovy, což jsou prokovené otvory, které slouží k propojení obou vrstev vzájemně mezi sebou.

2.8. POPIS PROGRAMU

2.8.1. PROGRAM PRO MASTER

V popisu programu pro hlavní mikroprocesor ATmega8, nazývaným jako tzv. „MASTER“, jsou vysvětlena ty nejdůležitější jednotlivá nastavení pro komunikační sběrnici TWI, pro komunikaci přes sériovou linku USART a samotný řídicí program.

Základní nastavení:

`#define F_CPU 11059200UL`

- z důvodu použití externího krystalu o velikosti frekvence 11,095MHz u hlavního mikroprocesoru ATmega8, je nutné tuto frekvenci nastavit i v programu, aby všechny potřebné výpočty byly počítány s touto hodnotou.

`#define F_SCL 100000L`

- nastavení frekvence potřebné pro výpočet přenosové rychlosti komunikační sběrnice TWI. V tomto případě je frekvence 100kHz.

Nastavení komunikace přes sériovou linku USART:

`UCSRB=(1<<RXEN)|(1<<TXEN);`

- příkaz pro spuštění obvodů jak pro vysílání dat, tak i pro přijímání dat

`UCSRC=(1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1);`

- příkaz pro nastavení přenosu po 8bitových znacích

`UBRRH=(BAUD_PRESCALE>>8);UBRRL=BAUD_PRESCALE;`

- příkaz pro nastavení přenosové rychlosti sériové linky USART. Pro nastavení přenosové rychlosti musíme nejprve provést její výpočet pomocí těchto operací.

- `#defineBAUD_PRESCALE(((F_CPU/(USART_BAUDRATE*16UL)))`
- `#defineUSART_BAUDRATE9600`

Nastavení komunikační sběrnice TWI:

Podrobný popis všech funkcí pro nastavení komunikační sběrnice TWI je obsažen v kódu.

`while((UCSRA&(1<<RXC))==0){};`

- příkaz, kdy mikroprocesor čeká na to, až budou přijatá data připravena ke čtení z registru UDR

`TWI_start();`

- funkce, kdy je zahájen přenos dat po vyslání tzv. start bitu

`TWI_zapisAdresy(adresa1+zapis);`

- funkce, která odešle adresu mikroprocesoru (SLAVE1), do kterého chceme zapisovat data, a zároveň se pošle příkaz, který říká, že poslaná data se budou zapisovat

`TWI_zapisDat(ReceivedByte);`

- funkce, která odešle zvolená data do mikroprocesoru, který byl již předem určen pomocí adresy

TWI_stop();

- funkce, kdy je ukončen přenos dat po vyslání tzv. stop bitu

Těmito funkcemi je nastavena komunikace přes komunikační sběrnici TWI s mikroprocesorem ATmega16, nazývaným jako tzv. „SLAVE1“. Pro nastavení té stejné komunikace akorát s mikroprocesorem ATmega16, nazývaným jako „SLAVE2“, se používají úplně stejné funkce a stejný postup. Jediná změna je v nastavení adresy mikroprocesoru, protože každý „SLAVE“ mikroprocesor má vlastní unikátní adresu, aby se rozlišily.

TWI_zapisAdresy(adresa2+zapis);

- funkce, která odešle adresu mikroprocesoru (SLAVE2), do kterého chceme zapisovat data, a zároveň se pošle příkaz, který říká, že poslaná data se budou zapisovat

2.8.2. PROGRAM PRO SLAVE

V popisu programu pro „podřadný“ mikroprocesor ATmega16, nazývaným jako tzv. „SLAVE“, jsou vysvětlena ty nejdůležitější jednotlivá nastavení pro komunikační sběrnici TWI, nastavení časovače a samotný řídicí program.

Základní nastavení:

#define F_CPU 1000000UL

- frekvence mikroprocesoru je přednastavena od výrobce na hodnotu 1MHz a musí být v programu zapsána hlavně kvůli správným výpočtům, které s touto frekvencí souvisí

Nastavení časovače:

TCCR0|=(1<<CS01)|(1<<CS00);

- prescaler je určen ke generování hodinového odpočtu pomocí frekvence F_CPU. V tomto případě je prescaler nastaven na F_CPU/1024.

TIMSK|=(1<<TOIE0);

- příkaz pro povolení přerušení přetečení

TCNT0=0;

- příkaz pro počáteční inicializaci počítadla

`odpocet=0;`

- příkaz pro počáteční inicializaci proměnné určené pro hodinový odpočet

`sei();`

- příkaz pro povolení globálního přerušení čítače

`ISR(TIMERO0_OVF_vect){};`

- funkce pro indikaci přetečení časovače. Ve funkci probíhá stálé navyšování hodnoty proměnné „odpocet“ a po dosažení předem stanovené hodnoty dojde k přetečení. Následně je tato proměnná vynulována a celý proces se opakuje.

Nastavení komunikační sběrnice TWI:

`TWI_nastaveniSlave();`

- funkce pro nastavení adresy mikroprocesoru. Funkce je stejná pro mikroprocesor SLAVE1 i SLAVE2.

`TWI_shodaCteniSlave();`

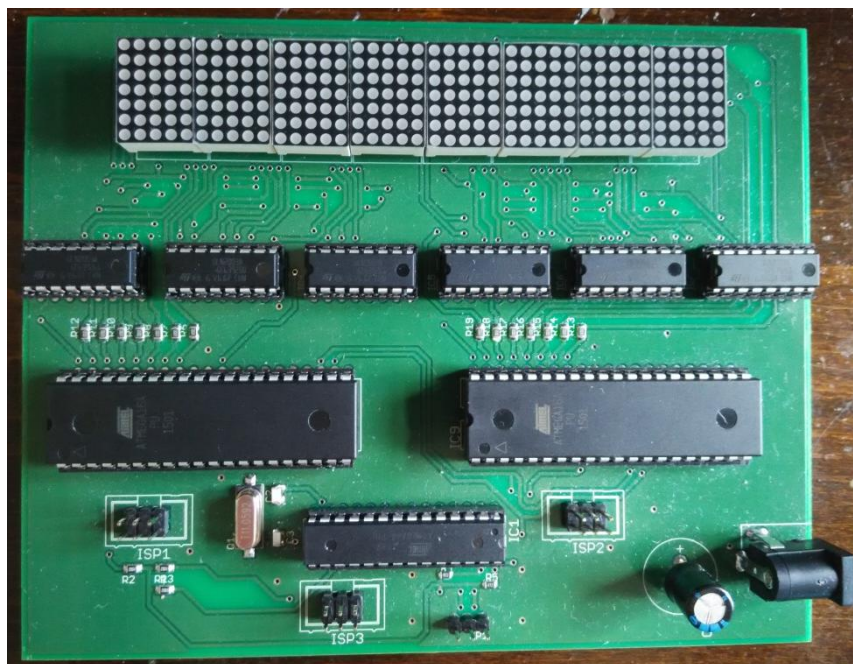
- funkce pro přijímání a rozpoznání odeslané adresy z hlavního mikroprocesoru. Pokud se přijímaná adresa shoduje s adresou příslušného mikroprocesoru, tak přenos dat bude probíhat skrz něj.

`TWI_ctiSlave();`

- funkce pro přijímání odeslaných dat z hlavního mikroprocesoru. Poté, co je rozpoznána adresa mikroprocesoru, jsou jím přijímána data.

2.9. DESKA PLOŠNÝCH SPOJŮ (DPS)

Celková výsledná realizace bakalářské práce se nakonec skládá z jedné dvouvrstvé desky plošných spojů (DPS). Jednotlivé součástky jsou na desce podle potřeby napájené z obou stran a jejich propojení mezi vrstvami je zajištěno prokovenými otvory, tedy prokovy.



Obrázek 28 - Osazená deska plošných spojů (DPS)

Jednotlivé programovatelné mikroprocesory a také tranzistorová pole nejsou napájeny přímo na DPS, ale jsou uloženy v patičkách, které jsou na DPS napájeny, z důvodu lepší obsluhovatelnosti a lepšímu přístupu k jednotlivým součástkám. Například v případě poškození součástky se dají z patice jednoduše vyjmout a následně nahradit novou či jinou se stejnými vlastnostmi a rozměry.

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a zrealizovat LED maticový displej, který dokáže zobrazit textovou zprávu o délce až 8 znaků. Každý jednotlivý znak se má zobrazit na maticovém displeji o rozměrech 5x7 bodů. Maticový displej má podporovat zobrazení znaků z tabulky ASCII v rozsahu od kódu 32 až po 126 a také zobrazení 8 předem nadefinovaných uživatelských znaků. Hlavní komunikace mezi uživatelským počítačem a hlavním ovládacím mikroprocesorem má probíhat pomocí sériového portu USART. Řízení a komunikace s displejem má probíhat pomocí dalších mikroprocesorů v zapojení. Napájení celého zapojení má být řešeno klasickým 5V adaptérem.

V teoretické části práce je nejprve popsán popis a princip fungování LED maticového displeje s ohledem na jeho vnitřní zapojení. Dále je zde popis jednotlivých mikroprocesorů, které byly v práci použity. Jde tedy o mikroprocesory ATmega8 a ATmega16 a jsou zde popsány jejich základní vlastnosti i seznam všech obvodů, které mikroprocesory obsahují.

V praktické části je nejprve popsán obecný postup řešení LED maticového displeje. Jsou zde okrajově popsány různé možnosti řešení práce a detailněji je popsáno konečné zvolené řešení LED maticového displeje, které obsahuje kombinaci mikroprocesorů a vzájemnou komunikaci mezi nimi, aby bylo splněno zadání bakalářské práce. Mikroprocesory jsou dále rozděleny na „MASTER“, „SLAVE1“ a „SLAVE2“. Nakreslené blokové schéma zapojení naznačuje způsob řízení a vzájemnou komunikaci mezi nimi. Dále je zde popsán postup programování mikroprocesorů. Tento postup se skládá hned z několika kroků. Zaprvé musely být vytvořeny a sepsány programy ve vývojovém prostředí Atmel Studio 7.0, které mají na starost veškerou komunikaci mezi jednotlivými součástkami, jejich ovládání a celkové řízení celé práce. Dále byly tyto programy postupně nahrány do příslušných mikroprocesorů pomocí programátoru a speciálního nahrávacího programu, který je určen pro nahrávání programů do mikroprocesorů. Jako další je popsána komunikace přes sériový port USART mezi hlavním mikroprocesorem a uživatelským počítačem. Pro tuto komunikaci je důležité mít nastaveny všechny potřebné vlastnosti terminálu na požadovaných hodnotách, aby komunikace probíhala dle předpokladů. Stručně je zde popsán i konečný program pro správnou funkčnost práce, například různé funkce, které jsou v programu obsaženy, nebo různé příkazy pro nastavení komunikací použitých v konečném řešení.

Napájení maticových displejů je realizováno pomocí PORTA mikroprocesoru ATmega16 a zároveň je proud omezen pomocí odporu o velikosti 120R. Přes odpory projde přibližně proud o velikosti 41mA, který se následně rozdělí mezi několik LED diod, které mají být rozsvíceny. Při největším zatížení, tj. 20 LED diod v řádku, je svítivost jednotlivých LED diod mnohem menší než při jakémkoliv nižším zatížení. Znamená to tedy, že čím více LED diod v řádku je rozsvíceno, tím menší je celková svítivost LED diod v řádku. V tomto případě by bylo lepší nenapájet LED diody přes mikroprocesor, ale přes PNP tranzistory (nebo P-FETy) přímo z napájení a jednotlivé báze (gate-y) připojit jako v tomto zapojení na PORTA. Zvolit omezovací odpory o velikosti 330R a připojit je na porty D1-D7 u každého ULN2003A. Tímto zapojením by se nastavil stálý proud a celý LED maticový displej by svítil jasněji než doposud.

Nakonec lze tedy konstatovat, že zhotovená bakalářská práce splňuje všechny jednotlivé body zadání, které byly předem stanoveny. Byl vytvořen, podle zadání, LED maticový displej s možností zobrazení textu o velikosti až 8 znaků, kdy komunikace s uživatelským počítačem je zrealizována pomocí sériového portu a celé zařízení je napájeno klasickým adaptérem na 5V.

POUŽITÁ LITERATURA

1. VÁŇA, Vladimír. *Mikrokontroléry ATMEL AVR - Programování v jazyce C: popis a práce ve vývojovém prostředí CodeVisionAVR C*. Praha: BEN - technická literatura, 2003. ISBN 80-7300-102-0.
2. MATOUŠEK, David. *Aplikace mikrokontrolérů ATmega644*. Praha: BEN - technická literatura, 2013. ISBN 978-80-7300-492-7.
3. MATOUŠEK, David. *Práce s mikrokontroléry ATMEL*. Praha: BEN - technická literatura, 2004. μ C. ISBN 80-730-0094-6.
4. *Programování mikrokontrolérů architektury AVR (I)* [online]. [cit. 2017-04-24]. Dostupné z: <https://www.soom.cz/clanky/1170--Programovani-mikrokontroleru-architektury-AVR-1>
5. *Programujeme AVR v jazyku C - 8. část* [online]. [cit. 2017-04-24]. Dostupné z: <https://svetelektro.com/clanky/programujeme-avr-v-jazyku-c-8-cast-511.html>
6. *Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877* [online]. [cit. 2017-04-24]. Dostupné z: <http://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeeprom-24lc256>
7. *Mikrokontroléry rady AVR – čítač/časovač I* [online]. [cit. 2017-04-24]. Dostupné z: <http://lubosweb.php5.sk/mikrokontrolery-rady-avr-citaccasovac-1/>
8. *Datasheet ATmega8(L)* [online]. [cit. 2017-04-24]. Dostupné z: http://www.atmel.com/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_datasheet.pdf
9. *Datasheet ATmega16(L)* [online]. [cit. 2017-04-24]. Dostupné z: <http://www.atmel.com/images/doc2466.pdf>
10. *Datasheet KWM-2057IASB* [online]. [cit. 2017-04-24]. Dostupné z: http://www.luckylight.cn/UploadFiles/pdf_2013112413853449131.pdf
11. *Datasheet ULN2003A* [online]. [cit. 2017-04-24]. Dostupné z: <https://www.diodes.com/assets/Datasheets/ULN200xA.pdf>

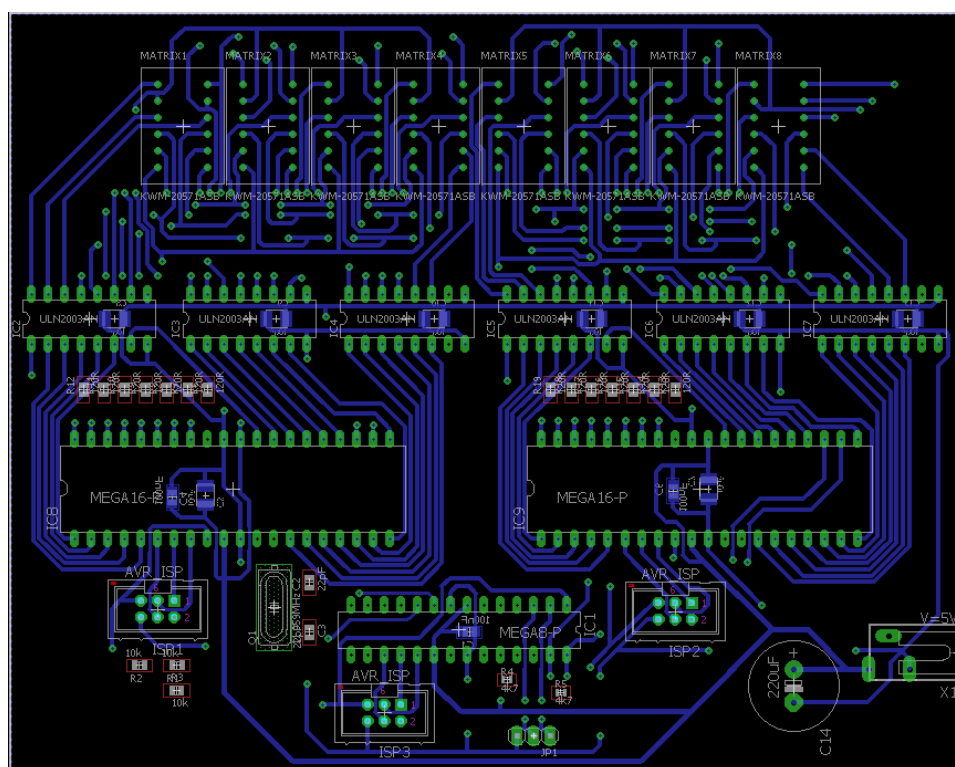
SEZNAM PŘÍLOH

Příloha A – Návrh DPS

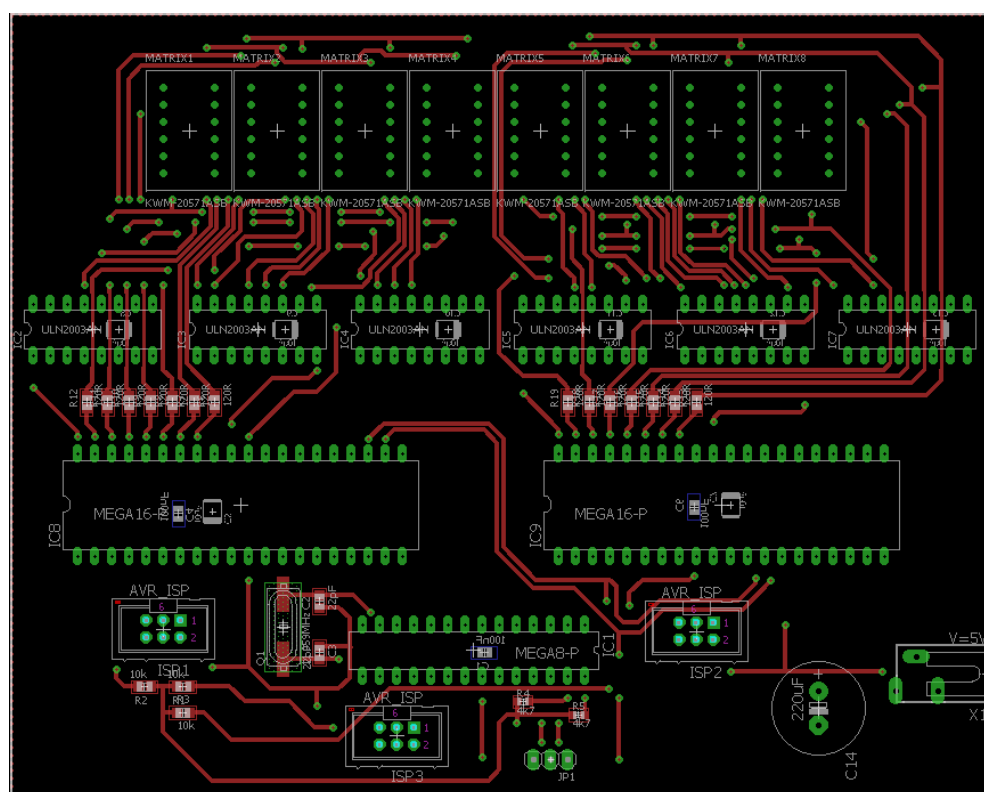
Příloha B – Výsledný maticový displej

Příloha C - CD

Příloha A – Návrh DPS



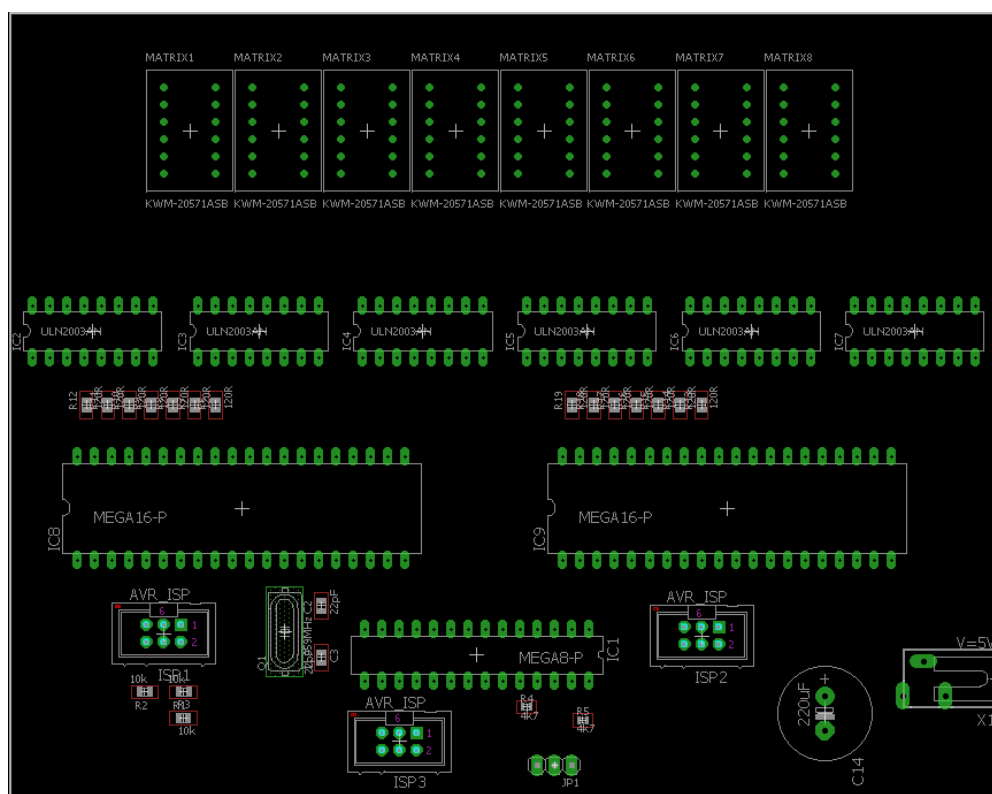
Obrázek 1 – Návrh DPS, spodní strana



Obrázek 2 – Návrh DPS, vrchní strana



Obrázek 3 – Rozmístění součástek na DPS, spodní strana



Obrázek 4 – Rozmístění součástek na DPS, vrchní strana

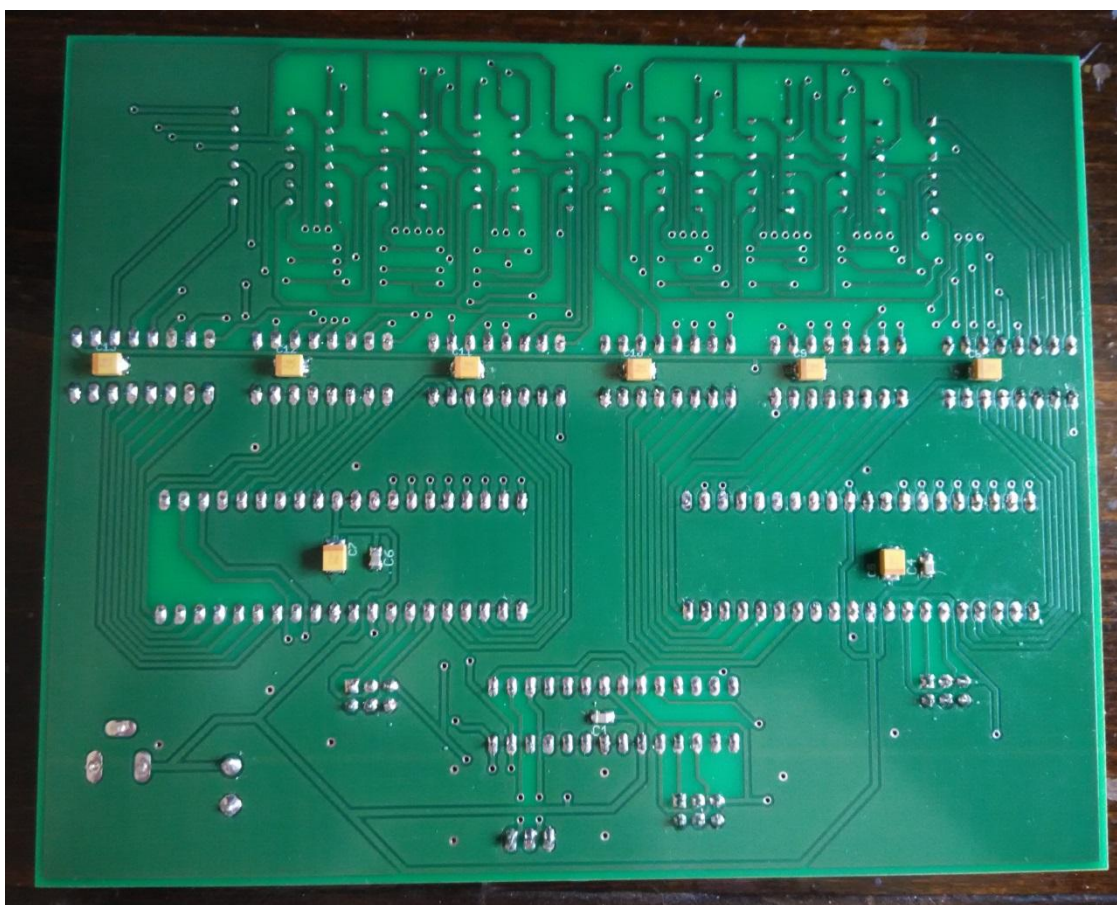
Tabulka 1 – Seznam součástek na DPS

Pořadové číslo	Označení na DPS	Název/hodnota	Pouzdro
1	V=5V	PC-GK1.3	SCD-014-A
2	ISP1	S2G06	MLW06G
3	ISP2	S2G06	MLW06G
4	ISP3	S2G06	MLW06G
5	JP1	ZL201-03G	1X03
6	IC1	ATmega8	DIL28-3
7	IC2	ULN2003AN	DIL16
8	IC3	ULN2003AN	DIL16
9	IC4	ULN2003AN	DIL16
10	IC5	ULN2003AN	DIL16
11	IC6	ULN2003AN	DIL16
12	IC7	ULN2003AN	DIL16
13	IC8	ATmega16	DIL40
14	IC9	ATmega16	DIL40
15	MATRIX1	KWM-20571ASB	KWM-20571CSB/ASB
16	MATRIX2	KWM-20571ASB	KWM-20571CSB/ASB
17	MATRIX3	KWM-20571ASB	KWM-20571CSB/ASB
18	MATRIX4	KWM-20571ASB	KWM-20571CSB/ASB
19	MATRIX5	KWM-20571ASB	KWM-20571CSB/ASB
20	MATRIX6	KWM-20571ASB	KWM-20571CSB/ASB
21	MATRIX7	KWM-20571ASB	KWM-20571CSB/ASB
22	MATRIX8	KWM-20571ASB	KWM-20571CSB/ASB

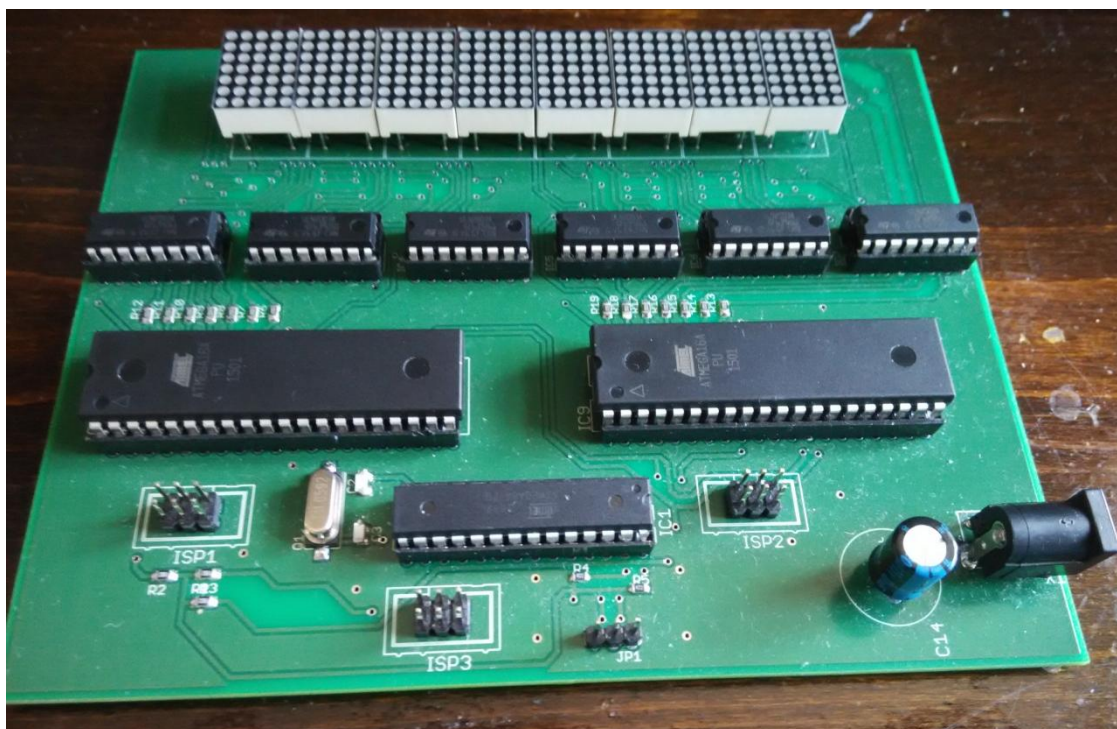
23	C1	100nF	C0805
24	C2	22pF	C0805
25	C3	22pF	C0805
26	C4	100nF	C0805
27	C5	10uF	SMC_B
28	C6	100nF	C0805
29	C7	10uF	SMC_B
30	C8	10uF	SMC_B
31	C9	10uF	SMC_B
32	C10	10uF	SMC_B
33	C11	10uF	SMC_B
34	C12	10uF	SMC_B
35	C13	10uF	SMC_B
36	C14	220uF	E5-13
37	R1	10k	R0805
38	R2	10k	R0805
39	R3	10k	R0805
40	R4	4k7	R0603
41	R5	4k7	R0603
42	R6	120R	R0805
43	R7	120R	R0805
44	R8	120R	R0805
45	R9	120R	R0805
46	R10	120R	R0805

47	R11	120R	R0805
48	R12	120R	R0805
49	R13	120R	R0805
50	R14	120R	R0805
51	R15	120R	R0805
52	R16	120R	R0805
53	R17	120R	R0805
54	R18	120R	R0805
55	R19	120R	R0805
56	Q1	11,059MHz	SM49

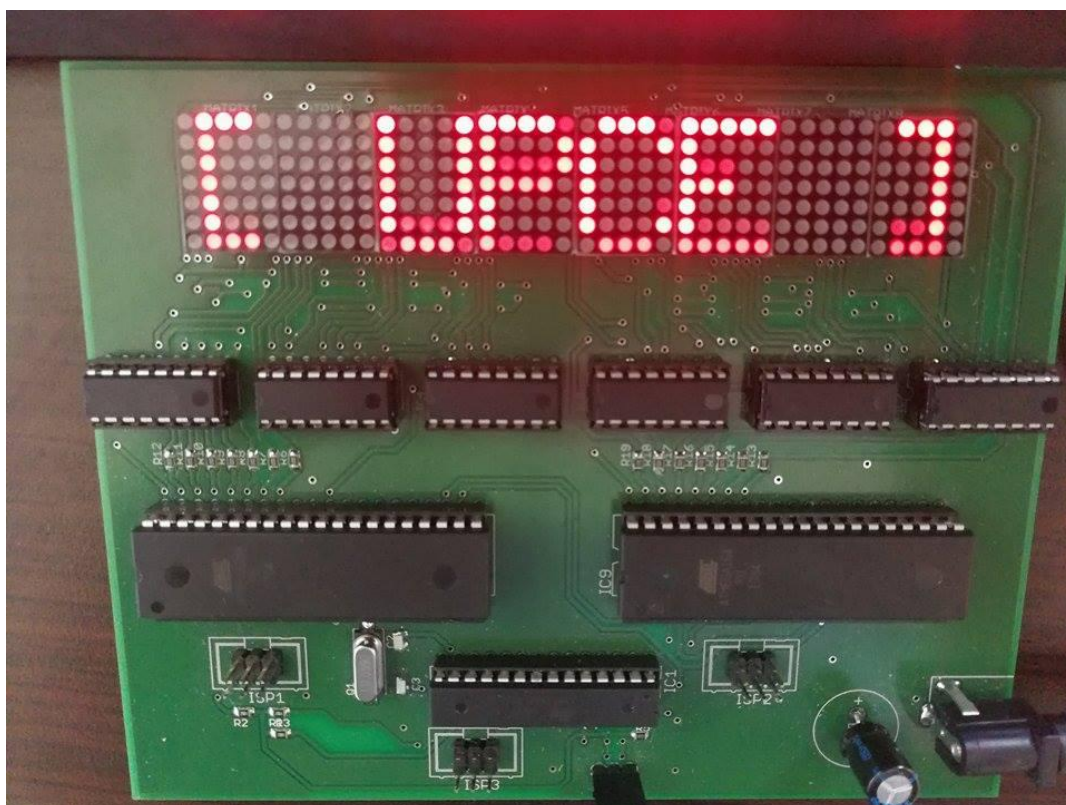
Příloha B – Výsledný maticový displej



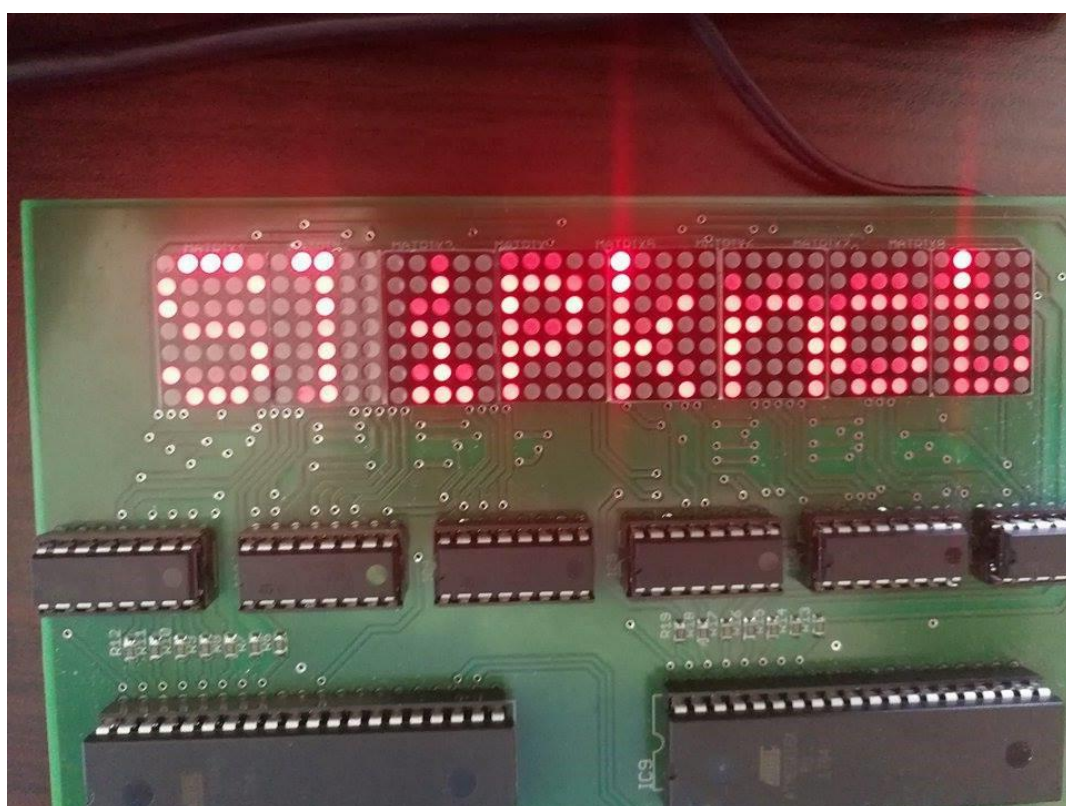
Obrázek 1 – LED maticový displej, spodní strana



Obrázek 2 – LED maticový displej, vrchní strana



Obrázek 3 – LED maticový displej, ukázka 1



Obrázek 4 – LED maticový displej, ukázka 2



Obrázek 5 – USBASP programátor



Obrázek 6 – Převodník FTDI232

Příloha C - CD

Obsah CD:

DPS_BP -Schéma a návrh DPS

MASTER - Zdrojový kód pro mikroprocesor MASTER

SLAVE1 - Zdrojový kód pro mikroprocesor SLAVE1

SLAVE2 - Zdrojový kód pro mikroprocesor SLAVE2

JircikP_MaticovyDisplej_2017.pdf – Vlastní text práce