

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů



Bakalářská práce

## **Programovatelný řadič displeje**

Armen Hajrapetjan

Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

Studijní program: Elektrotechnika a informatika, strukturovaný, bakalářský

Obor: Výpočetní technika

10. června 2008



## **Poděkování**

Děkuji svému vedoucímu Ing. Pavel Kubalík, Ph.D. za pomoc při tvorbě této bakalářské práce.



## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu. Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), v platném znění.

V Praze dne 10.6.2009 .....



## **Abstract**

The aim of this baccalaureate work is to produce a modulus that shall communicate with graphic display of definition 128x64 points. The modulus will be controlled through a serial line. With the assistance of instructions, a created protocol shall operate a graphic display. The protocol will ease working with graphic displays to such a level that everybody who can use serial lines (transmission and data receiving) and does not have knowledge of graphic displays will be able to utilise it.

## **Abstrakt**

Cílem této bakalářské práce je vyrobit modul, jenž bude komunikovat s grafickým displejem o rozlišení 128x64 bodů, a který se bude ovládat přes sériovou linku. Vytvořený protokol bude řídit grafický displej za pomoci instrukcí. Tento protokol usnadní práci s grafickými displeji na takovou úroveň, že s nimi bude moci pracovat každý, kdo umí pracovat se sériovou linkou (vysílání a přijímání dat) a nemá žádné znalosti grafických displejů





## **Obsah**

Úvod	1
Rešerše	3
Analýza	5
Návrh řešení	9
Závěr	23
Seznam použité literatury	25
Obsah CD	27
Příloha A - Seznam použitých součástí	29



## Seznam obrázků

- Obr. 2.1.0: Význam vývodu, první a dvacátý pin je označen na desce LCD.
- Obr. 2.1.1: Uspořádání paměti RAM v řadiči KS0108.
- Obr. 2.1.2: Instrukční sada LCD.
- Obr. 3.1.0: Napěťové úrovně dle standartu rs232.
- Obr. 3.1.1: Schéma převodníku úrovně standartu rs232.
- Obr. 3.1.2: Schéma zapojení konektoru pro komunikaci přes RS232 s PC. Toto zapojení je určeno pro přímý kabel.
- Obr. 3.1.3: Schéma zapojení konektoru pro komunikaci přes RS232 v úrovni TTL.
- Obr. 3.1.4: Schéma zapojení konektoru pro komunikaci přes SPI.
- Obr. 3.1.5: Zapojení komunikačního rozhraní mezi LCD a modulem.
- Obr. 3.2.0: Schéma zapojení zdroje.
- Obr. 3.2.1: Schéma zapojení napájecího konektoru.
- Obr. 3.2.2: Schéma zapojení programovacího konektoru.
- Obr. 3.3.0: Schéma zapojení mikroprocesoru PIC18F4520.
- Obr. 3.4.0: Opravené zapojení podsvícení displeje.
- Obr. 3.5.0: Zapojení resetovacího tlačítka.
- Obr. 3.6.0: Schéma zapojení dvou periferních LED diod.
- Obr. 3.6.1: Schéma zapojení periferních přepínačů a přepínačů pro nastavení komunikačního rozhraní a zobrazení loga.
- Obr. 3.6.2: Schéma zapojení piezoelektrického měniče.
- Obr. 3.6.3: Schéma zapojení teplotního čidla.
- Obr. 3.7.0: Schéma zapojení adresných přepínačů.
- Obr. 3.7.1: Naimplementovaná ASCII tabulka.
- Obr. 3.7.2: Doporučené zapojení pro komunikaci s více moduly.



## Seznam použitých zkratek

LCD	Liquid crystal display
RAM	Random-access memory
MIPS	Million Instructions per Second
EUSART	Enhanced universal synchronous receiver transmitter
SPI	Serial Peripheral Interface
TTL	Transistor-transistor-logic
ICSP	In-Circuit Serial Programming
FXGP	Flexible graphics protocol



# Úvod

V dnešním světě skoro každé zařízení obsahuje nějaký displej pro komunikaci s vnějším světem. Z tohoto důvodu jsem se rozhodl vytvořit modul, který usnadní práci s grafickým displejem, a tím umožní jiným výrobcům různých zařízení zefektivnit svoji produktivitu práce tím, že se nebudou muset jejich vývojáři zabývat tím, jak graficky displej funguje, a bude jim stačit jen znát komunikaci přes sériovou linku. Mým cílem je tedy vyrobit modul a vytvořit protokol pro komunikaci s grafickým displejem, který bude řídit grafický displej o rozlišení 128x64 bodů s níže uvedenými funkcemi:

- Modul bude obsahovat metodu pro :
  - Psaní znaků.
  - Kreslení grafů.
  - Mazání celého displeje.
  - Mazání řádků.
  - Kreslení obrázků nebo promítání snímků.
  - Také se bude moci zobrazovat invertovaně.
  
- Modul se bude řídit sadou jednoduchých instrukcí.

Kapitola 2 obsahuje analýzu návrhu.

Kapitola 3 podrobně popisuje navržené schéma a vytvořený protokol FXGP (flexible graphic protocol), který byl vytvořen pro efektivní a rychlou komunikaci přes sériovou linku, a také instrukční sadou podporující tento protokol pro grafický LCD displej.





## Rešerše

Na internetu jsem našel několik společností vyrábějících LCD displeje s rozhraním RS232

- 1) <<http://www.crystalfontz.com/products/index-ser.html>>. Výběr těchto displejů je však omezen pouze na alfanumerické displeje řady 20x4 znaků nebo 16x2 znaků, proto na nich nelze kvalitně zobrazovat snímky či grafy.
- 2) <[http://www.acscontrol.com/Index\\_LcdTerminal.asp?Page=/Pages/Products/Lcd/128x64%20LCD\\_Display\\_Terminal.htm](http://www.acscontrol.com/Index_LcdTerminal.asp?Page=/Pages/Products/Lcd/128x64%20LCD_Display_Terminal.htm)>. Tuto společnost jsem objevil v době, kdy už jsem měl hotový svůj prototyp. Tento řadič je podobný mému, ale v maličkostech se od sebe výrazně liší. Řadič této firmy je velký jako celý displej, pracuje také na protokolu, který má podobné, ale přece jen trochu jiné vlastnosti a rysy. Obsahuje základní znakovou sadu bez českých znaků. Obsahuje periférii speaker a je možné k němu připojit tlačítka.



# Analýza

## 2.1 Výběr vhodného grafického displeje

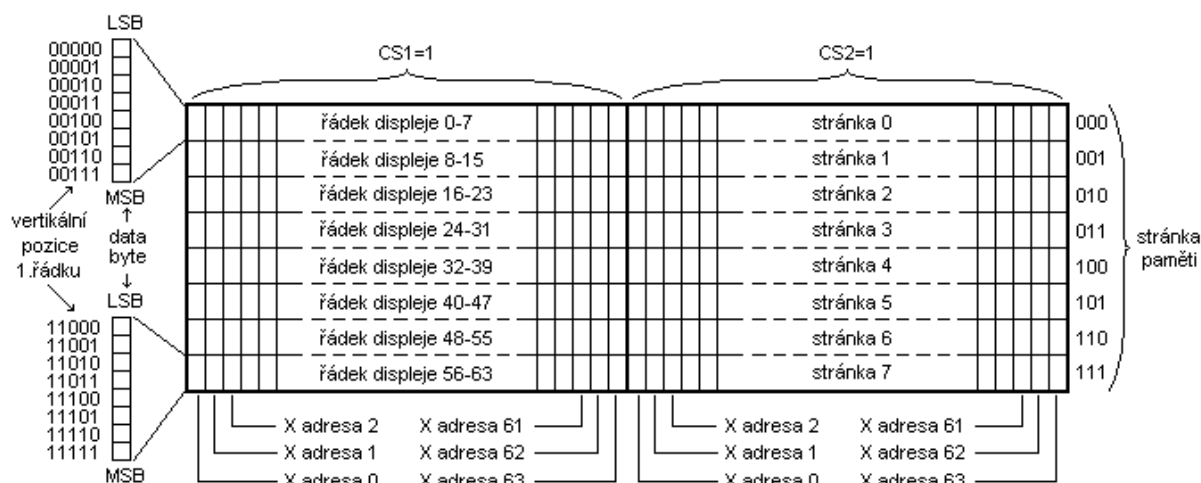
Při výběru modulu displeje jsem se rozhodoval mezi dvěma moduly. Prvním je modul s řadičem KS0108, který je nejvíce rozšířen a je snadno dostupný. Jako druhý přicházel v úvahu displej s řadičem S6B1713. Protože však tento řadič potřebuje i externí součástky a obsahuje o 10 vývodů více než řadič KS0108, rozhodl jsem se, že ho nepoužiji. Také je hůře dostupný. Oba tyto typy LCD jsou v rozlišení 128x64 bodu.

Displej pracuje s dvěma řadiči KS0108, z nichž každý obsluhuje jednu polovinu displeje v rozlišení 64x64 bodu. Veškeré vývody mimo CS1 a CS2 jsou spolu paralelně zapojeny. Pomocí vývodu CS1 a CS2 vybíráme, se kterou z částí LCD budeme pracovat, tedy buď levou či pravou, nebo oběma. LCD komunikuje po 8 bitové sběrnici. Výběr mezi instrukcí a daty se provádí signálem D/I a mezi zápisem a čtením signál R/W.

Pin	Signál	Popis
1	Vss	Napájení GND
2	Vdd	Napájení +5V
3	V0	Nastavení kontrastu
4	D/I	Volba mezi instrukcí a daty ( instrukce: D/I=0 , data: D/I=1 )
5	R/W	Volba mezi čtením a zápisem ( zápis: R/W=0, čtení: R/W=1 )
6	E	Hodiny
7	DB0	Datový bit 0
8	DB1	Datový bit 1
9	DB2	Datový bit 2
10	DB3	Datový bit 3
11	DB4	Datový bit 4
12	DB5	Datový bit 5
13	DB6	Datový bit 6
14	DB7	Datový bit 7
15	CS1	Výběr čipu pro levou stranu LCD
16	CS2	Výběr čipu pro pravou stranu LCD
17	RES	Reset negováno
18	VEE	Výstup záporného napětí pro nastavení kontrastu
19	A	Anoda podsvícení +
20	K	Katoda podsvícení -

Obr. 2.1.0: Význam vývodu, první a dvacátý pin je označen na desce LCD.

Pokud zapisujeme nebo čteme data, tak se nám automaticky inkrementuje pozice X. Pokud docílíme hodnoty 63, tak se nám hodnota vynuluje a začínáme opět od nuly na stejném řádku. Řádky se neinkrementují automaticky, přepínají se pomocí instrukce displeje. Vertikální pozice se dá využít pro posouvání obsahu grafického displeje. Uspořádání paměti RAM je na obrázku 2.1



Obr. 2.1.1: Uspořádání paměti RAM v řadiči KS0108.

Instrukce	DI	RW	DB	DB	DB	DB	DB	DB	DB	DB	Funkce
			7	6	5	4	3	2	1	0	
Čtení dat do LCD	1	1	DATA								
Zápis dat z LCD	1	0	DATA								
Čtení příznaku	0	1	B	0	O/F	R	0	0	0	0	
Nastavení pozice Y	0	0	1	1	Vertikální pozice 0-63						
Nastavení pozice X	0	0	0	1	X 0-63						
Nastavení stránky	0	0	1	0	1	1	1	Stránka 0-7			
Zapnutí vypnutí LCD	0	0	0	0	1	1	1	1	1	0/1	

Obr. 2.1.2: Instrukční sada LCD.

## 2.2 Výběr vhodného mikroprocesoru:

Protože jsem několik let pracoval s procesory od microchipů, rozhodl jsem se, že je využiji i pro svoji současnou práci. Lépe jim totiž rozumím, mám veškeré potřebné vybavení na jejich programování a jednodušeji se mi s nimi pracuje. Na výběr jsem měl z několika řad 8 bitových mikroprocesorů 10,12,16 a 18, rozhodl jsem se využít řadu 18, která je z 8 bitových procesorů nejvýkonnější a zároveň má větší instrukční sadu než jeho předchůdci. Z této řady jsem si vybral procesor PIC18F4520, který se vyznačuje vysokým výkonem (10MIPS), velkou kapacitou paměti program a paměti RAM a také obsahuje potřebné komunikační rozhraní EUSART a SPI pro budoucí rozšíření.

### **2.3 Přehled:**

Počet řádků: 8

Minimální počet znaků na řádek: 21

Maximální počet znaků na řádek: 64

Počet snímků za sekundu: 2,5 snímků/s

Minimální proudový odběr: 73 mA

Maximální proudový odběr: 90 mA

Napájecí napětí: 8 - 20 V

Rozměry modulu: 93x35 mm

Typ přenosu a její rychlost: RS232 bez parity, 1 stop bit, 19200 baud

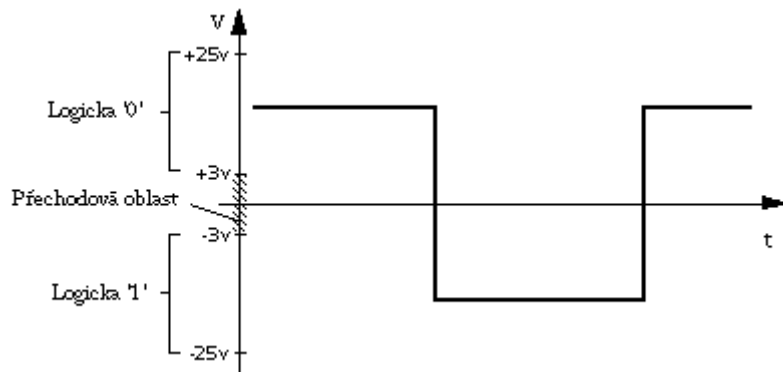


# Návrh řešení

## 3.1 Rozhraní modulu IQGLCDM

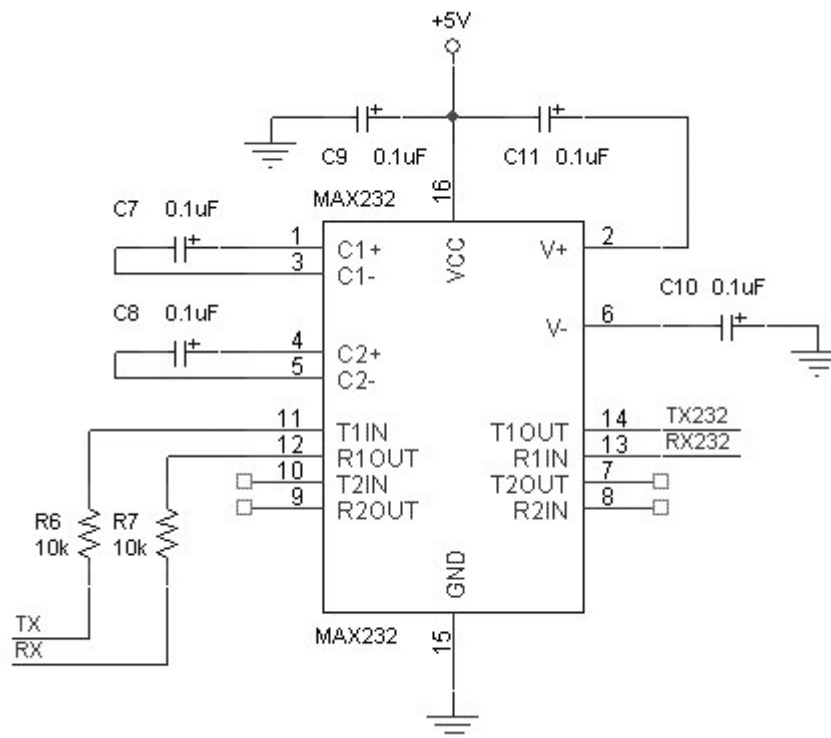
### Komunikace modulu s PC je pomocí standartu RS232

Modul s PC komunikuje asynchronně bez paritního bitu a full duplexně, což znamená, že ve stejném okamžiku můžeme data vysílat i číst. Logické úrovně pro komunikaci s PC jsou -12V pro log 1 a +12V pro log 0.

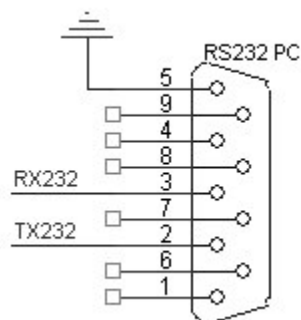


Obr. 3.1.0: Napěťové úrovně dle standartu rs232.

Jelikož modul pracuje s 5 voltovou logikou, musel jsem použít převodník, který tyto úrovně zajistí. Rozhodl jsem se použít převodníky od firmy Maxim a to konkrétně MAX232. Tento obvod obsahuje dvě dvojice oddělovačů, jednu dvojici pro příjem a druhou pro vysílání. Obvod MAX232 potřebuje pro svoji činnost externí kondenzátory, které slouží pro převodník jako nábojové pumpy, aby se mohly vytvořit logické úrovně pro rozhraní RS232.



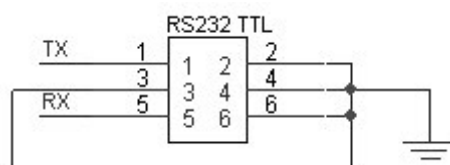
Obr. 3.1.1: Schéma převodníku úrovní standartu rs232.



Obr. 3.1.2: Schéma zapojení konektoru pro komunikaci přes RS232 s PC. Toto zapojení je určeno pro přímý kabel.

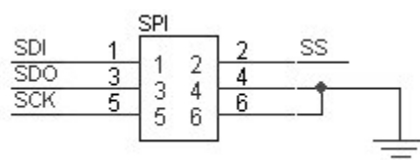
### Komunikace modulu s jiným nízkonapětovým zařízením pomocí sériové linky

Modul může s ostatními nízkonapětovými zařízeními komunikovat opět pomocí standardu RS232 tentokrát v úrovni TTL.



Obr. 3.1.3: Schéma zapojení konektoru pro komunikaci přes RS232 v úrovni TTL.

Do budoucna je zde připraven i konektor pro komunikaci přes rozhraní SPI. Výběr mezi rozhraním se bude vybírat pomocí přepínače RS232/SPI, který je také připraven. Zapojení přepínače lze vidět na obrázku 3.6.1.

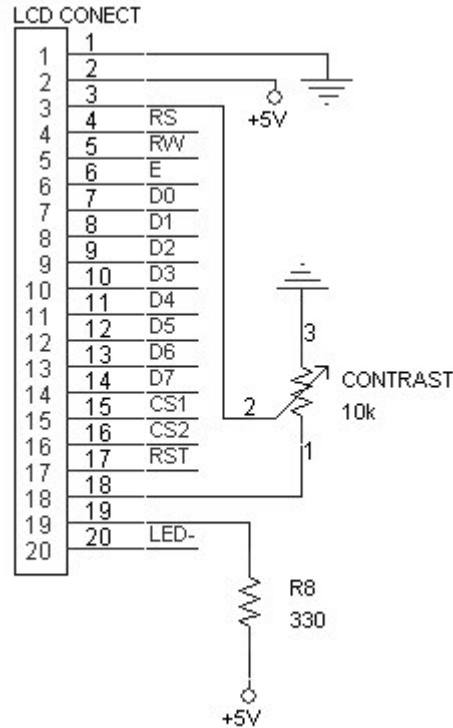


Obr. 3.1.4: Schéma zapojení konektoru pro komunikaci přes SPI.

### Komunikace modulu s LCD displejem:

Displej komunikuje s LCD displejem přes paralelní sběrnici. Zapojení komunikačního rozhraní mezi LCD a modulem je na obrázku 3.1.5.



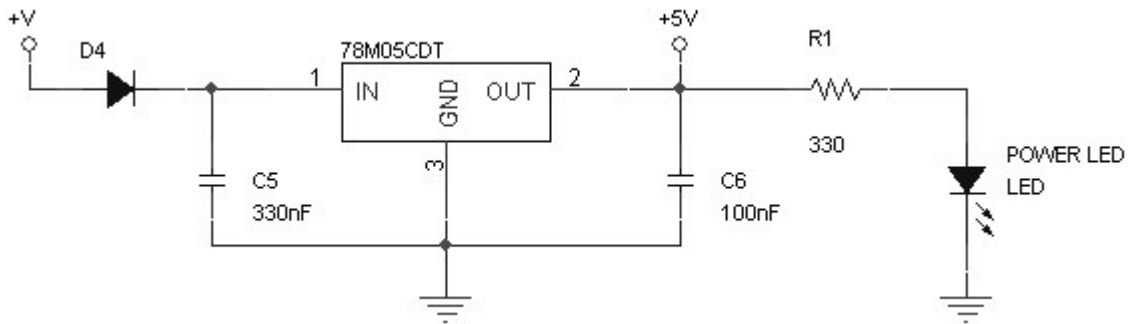


Obr. 3.1.5: Zapojení komunikačního rozhraní mezi LCD a modulem.

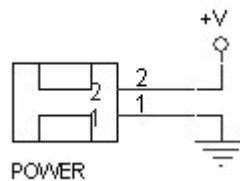
### 3.2 Napájení modulu:

Modul je možné napájet dvěma způsoby:

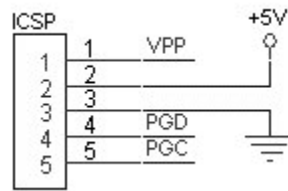
1. Přes napájecí konektor, který je zároveň chráněn proti přepólování, pomocí usměrňovací diody stabilizuje vstupní napětí na 5V přes stabilizátor 78M05CDT. Připojení napájecího napětí je indikováno pomocí POWER\_LED (zelená). Schéma zdroje je na obrázku 3.2.0
2. Přes programovací konektor ICSP, kde lze využít pin 2 pro stabilizované napětí 5V a pin 3 pro zem. Zapojení konektoru ICSP je na obrázku 3.2.2



Obr. 3.2.0: Schéma zapojení zdroje.



Obr. 3.2.1: Schéma zapojení napájecího konektoru.

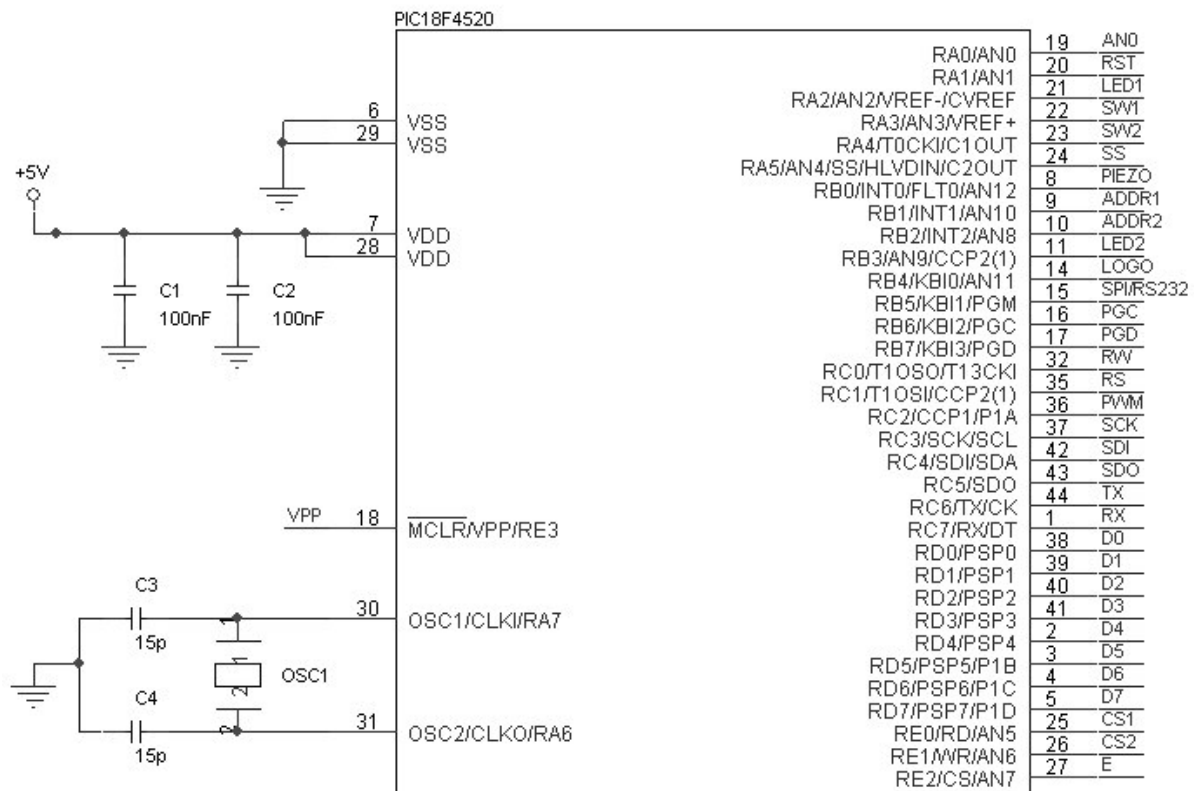


Obr. 3.2.2: Schéma zapojení programovacího konektoru.

### 3.3 Programování a zapojení mikroprocesoru:

Mikroprocesor se programuje přímo na modulu přes programovací konektor ICSP, tento způsob jsem zvolil z důvodu většího komfortu. K programování lze využít jakýkoliv programátor, který podporuje mikroprocesor PIC18F4520 a zároveň umožňuje programování přes rozhraní ICSP. Já osobně jsem zvolil pro programování programátor PICKIT2 od firmy MICROCHIP. Zapojení ICSP konektoru je zobrazeno na obrázku 3.2.2.

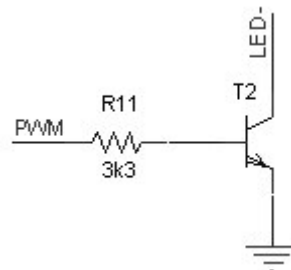
V blízkosti napojení mikroprocesoru jsou umístěny dva filtrační kondenzátory pro větší stabilitu napájecího napětí. K procesoru je zapojen 10MHz oscilátor, který je v procesoru násoben čtyřmi, tudíž procesor běží na frekvenci 40MHz. Na obrázku 3.3.0 jsou zobrazeny veškeré zapojení pinu mikroprocesoru.



Obr. 3.3.0: Schéma zapojení mikroprocesoru PIC18F4520.

### 3.4 Podsvícení displeje:

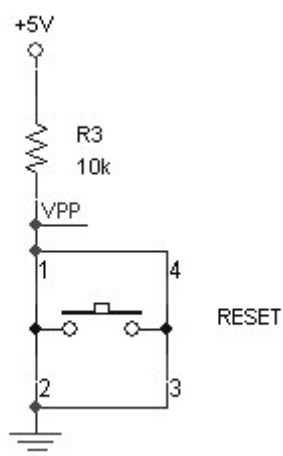
Podsvícení displeje jsem bohužel na hardwaru chybně naimplementoval, a proto podsvícení není tak úplně funkční. Chyba spočívala v tom, že jsem podsvícení rovnou zapojil na pin procesoru a ten nedokáže dodat dostatečný proud pro podsvícení. Novější zapojení je řešeno přes tranzistor, který může dodat pro podsvícení vyšší proud než pin procesoru. Zapojení opravené části hardware je na obrázku 3.4.0



Obr. 3.4.0: Opravené zapojení podsvícení displeje.

### 3.5 Reset procesoru:

Procesor se dá resetovat externím tlačítkem, který je umístěn na modulu. Reset proběhne pokud na pin VPP mikroprocesoru připojíme zem, pro normální chod mikroprocesoru je potřeba pin VPP zapojit na logickou jedničku. Zapojení resetovacího tlačítka je na obrázku 3.5.0



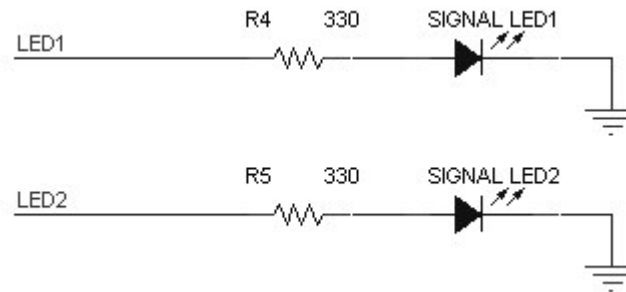
Obr. 3.5.0: Zapojení resetovacího tlačítka.

### 3.6 Periferie:

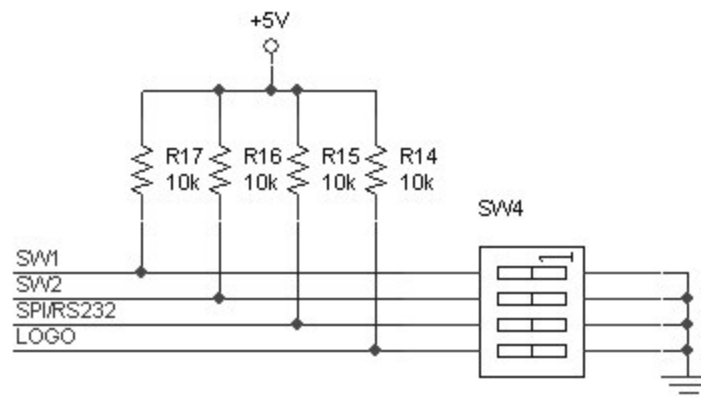
Modul je vybaven několika periferními zařízeními, které by mohly být vhodné pro některé aplikace:

1. Dvě modré LED diody zapojení je možné vidět na obrázku 3.6.0
2. Dva přepínače SW1 a SW2 zapojení je vidět na obrázku 3.6.1 . tato část je připravena pro softwarovou implementaci.

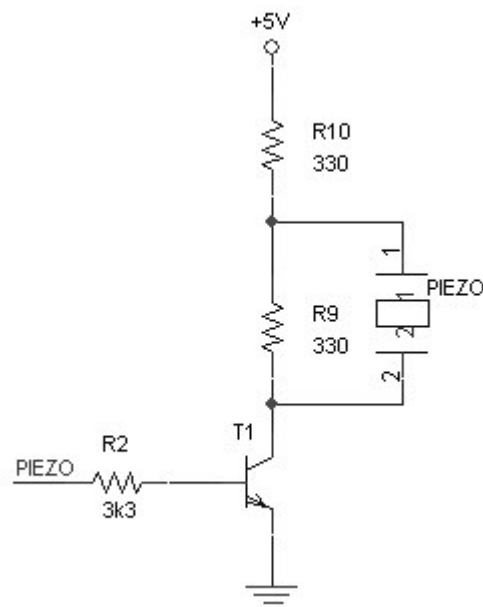
3. magnetodynamický měnič. U aktuálně použitého magnetodynamického měnič není zapojen odpor R9, jelikož tento magnetodynamický měnič už v sobě má zabudován obvod pro generování tonu , proto jej stačí připojit jen k napájení přes ochranný odpor.
4. Teplotní čidlo. Tato část je připravena pro softwarovou implementaci.



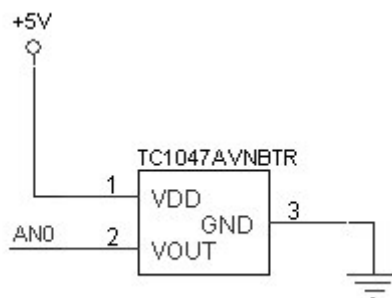
Obr. 3.6.0: Schéma zapojení dvou periferních LED diod.



Obr. 3.6.1: Schéma zapojení periferních přepínačů a přepínačů pro nastavení komunikačního rozhraní a zobrazení loga.



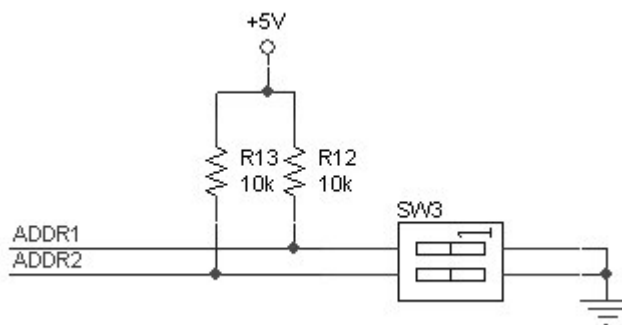
Obr. 3.6.2: Schéma zapojení magnetodynamický měnič.



Obr. 3.6.3: Schéma zapojení teplotního čidla.

### 3.7 Komunikační protokol:

Pro komunikaci s modulem slouží protokol FXGP (flexible graphics protocol), který jsem navrhl přímo pro tento účel. Tento protokol pracuje flexibilně, jeho délka se tedy mění podle toho, jakou instrukci vykonáváte. Umožňuje připojení až 4 LCD displejů na jeden sériový port, který se bude adresovat. Adresa displeje se nastavuje na přepínačích ADDR1 a ADDR2. Schéma zapojení je na obrázku 3.7.0, např.: pro odeslání znaků se musí odeslat dva byty, první byt je instrukce s adresou displeje, adresa se nastavuje volí dvěma bity prvního bytu, dalšími šesti bity se volí instrukce, druhý byt se bere jako data. Pro kreslení souřadnicového bodu se musí odeslat tři byty, první bude opět instrukce s adresou displeje a další byty budou x a y pozice. Pro vymazání celého displeje stačí odeslat jen jeden byt, který bude obsahovat opět adresu displeje s instrukcí. Při zadání neexistující instrukce se tato instrukce zahodí. Pokud se odesílá existující instrukce s neplatnou adresou displeje tak se zahodí i následující data pro tuto instrukci, které přijdou. Tímto se zabrání při komunikaci více displejů, aby některý z displejů špatně pochopil instrukci, kterou jsme ji neodeslali. Například pokud odešleme instrukci mazání řádku, tak displej, jemuž tato instrukce byla určena, bude čekat na data řádku, kterou bude mazat. Ostatní příchozí data budou ignorovat a zahodí je. Pokud by se tak nestalo, tak by displej, který by měl adresu 00, při příchodu požadavku na smazání řádku 6, pochopil tato data jako instrukci a smazal by celý obsah displeje. Pro plynulejší chod displeje je naimplementován softwarový buffer v podobě FIFO o velikosti 250 bytu, pokud se buffer zaplní, tak se vyšle příznak F jako full a jakmile se buffer uvolní o 200 bytu tak se pošle příznak E jako empty 50 bytu se zanechává kvůli plynulejšímu chodu displeje, aby data na sebe lépe navazovala například kvůli kreslení obrázku. Od odeslání příznaku F do odeslání příznaku E jsou veškerá příchozí data ignorována.



Obr. 3.7.0: Schéma zapojení adresných přepínačů.

## Psaní textu na displeji:

Z důvodu nedostatku času jsem naimplementoval jen základní ASCII tabulku, tento nedostatek v budoucnu hodlám odstranit a vytvořit plnohodnotnou ASCII tabulku s 256 znaky. Pokud se odešle znak, který není naimplementován, vypíše se jen mezera široka jeden pixel.

Znakům není přiřazená pevná šířka. To znamená, že pokud je znak široký jen jeden pixel jako například !, tak se vykreslí jen široký jeden pixel a poté hned mezera. Toto ušetří více místa na displeji, což znamená, že se na řádek může vejít různý počet znaků. Znak, který se nevejde na řádek je automaticky umístěn na další. Pokud se nacházíme na posledním řádku a už se nám do něj nevejde žádný další znak, obsah celého displeje se odskroluje o jeden řádek nahoru, a my tím získáme další prázdný řádek, do kterého můžeme psát text.

### Hight bits

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000		▶	MEZ	0	@	P	`	p	€				Ř	Đ		
0001	☺	◀	!	1	A	Q	a	q			˘	±	Á	Ñ	á	
0010	☹		"	2	B	R	b	r			˘		Â	Ň		ň
0011	♥		#	3	C	S	c	s					Ă	Ó		ó
0100	♦		\$	4	D	Y	d	t					Ä	Ô	ä	ô
0101	♣		%	5	E	U	e	u				μ		Ő		ő
0110	♠		&	6	F	V	f	v			ı		Ć	Ö		ö
0111	▪		'	7	G	W	g	w			§			x		÷
1000	■		(	8	H	X	h	x						Ř	č	ř
1001	TAB		)	9	I	Y	i	y					É	Ů	é	ů
1010			*	:	J	Z	j	z	Š	š				Ú		ú
1011	♂		+	;	K	[	k	{						Ů		ů
1100	♀		,	<	L	\	l		Ś				Ě	Ü	ě	ü
1101	ENT		-	=	M	]	m	}	Ť	ť			Í	Ý	í	ý
1110	♪		.	>	N	^	n	~	Ž	ž			Î			
1111	◇		/	?	O	_	o		Ž			ž	Ď		ď	

Obr. 3.7.1: Naimplementovaná ASCII tabulka.

TAB....tabulátor vytvoří mezeru širokou 12 pixelu  
 ENT....enter nastaví kurzor na začátek dalšího řádku  
 MEZ...mezerník vytvoří mezeru širokou 1 pixel

### Instrukce: 0b000001

#### Struktura pro odeslání jednoho znaku:

První byt		Druhý byt
Adresa LCD 2 bity	Instrukce 6 bitu	Data 1 byt ASCII (0-255)

#### Vzor pro odesílání přes rozhraní:

Adresa displeje je nastavena na 0b10

0b10000010=0x81

**Vstup:** 0x81 'A' 0x81 'H' 0x81 'O'0x81 'J'

**Vystup na displej:** AHOJ

**Vstup:** 0x81 'A' 0x81 ' ' 0x81 'H' 0x82 'O'0x81 'J'

**Vystup na displej:** A HOJ

// výpis A mezera HOJ

**Vstup:** 0x81 'A' (0x81 (0x0D nebo 0xA)) 0x81 'H' 0x81 'O'0x81 'J'

**Vystup na displej:** A

HOJ

// výpis A enter HOJ

**Vstup:** 0x81 'A' 0x81 " 0x81 'H' 0x81 'O'0x81 'J'

**Vystup na displej:** A HOJ

// výpis A tabulátor HOJ

### Kreslení souřadnicového bodu:

Souřadnice X a Y jsou v modulu. X v modulu 128 a Y v modulo 64. Při kreslení souřadnicového bodu se automaticky nastavují stránky displeje na defaultní nastavení. To znamená, že pokud budeme na displej psát tolik znaků, že se na něj tyto znaky nevejdou a displej bude tak odskrolovan, tak při kreslení souřadnicového bodu nastane to, že se poslední řádek dostane na první pozici a ostatní se odskrolují zpět o jeden řádek dolů. Proto je nejlepší postup buď displej celý smazat, nebo psát maximálně na 8 řádků displeje, aby nenastalo skrolování. Pozice 0,0 je na displeji vlevo dole a pozice 123,63 vpravo nahoře.

**Instrukce: 0b000100**

**Struktura pro vykreslení jednoho bodu:**

První byt		Druhý byt	Třetí byt
Adresa LCD 2 bity	Instrukce 6 bitu	Souřadnice X (0-127)	Souřadnice Y (0-63)

**Vzor pro odesílání přes rozhraní:**

**Adresa displeje je nastavena na 0b10**

0b10000100=0x84

**Vstup:** 0x84 0x00 0x00 0x84 0x7F 0x3F

**Vystup na displej:** bod na pozici 0,0 a 123,63

**Vstup:** 0x84 0x80 0x40 0x84 0xFF 0x7F

**Vystup na displej:** bod na pozici 0,0 a 123,63

### Mazání celého displeje:

Instrukce mazání celého displeje smaže celý displej a nastaví kurzor na začátek prvního řádku displeje.

**Instrukce: 0b000110**

**Struktura pro smazání celého displeje:**

1 byt	
Adresa LCD 2 bity	Instrukce 6 bitu

**Vzor pro odesílání přes rozhraní:**

**Adresa displeje je nastavena na 0b10**

0b10000110=0x86

**Vstup: 0x86**

**Vystup na displej:** smazaný celý displej a nastavený kurzor na začátek prvního řádku displeje

### Mazání řádků:

Data řádku jsou v modulu 8. Pokud dojde ke skrolování displeje tak řádek 0 je pořadovým řádkem 0. Po smazání řádku se kurzor nastaví na začátek tohoto řádku, po dosažení konce tohoto řádku se kurzor nastaví na začátek dalšího řádku. Pokud na tomto řádku budou nějaké znaky, tak se bude přepisovat novými znaky. Zde mám drobnou chybu, kterou ještě budu muset odstranit, jelikož pokud se obraz odskroluje, tak se při smazání řádku a vyplněním ho znaky nedostaneme na další řádek displeje, kde budeme přepisovat staré znaky, ale displej se opět odskroluje a další znaky se budou zapisovat na nový prázdný řádek.

**Instrukce: 0b000000**

**Struktura pro smazání jednoho řádku:**

První byt		Druhý byt	
Adresa LCD 2 bity	Instrukce 6 bitu	Data 1 byt (0-7)	

**Vzor pro odesílání přes rozhraní:**

**Adresa displeje je nastavena na 0b10**

0b10000000=0x80

**Vstup: 0x80 0x00**

**Vystup na displej:** smazaný řádek 0 a nastavený kurzor na začátek tohoto řádku

**Vstup: 0x80 0x0B**

**Vystup na displej:** smazaný řádek 3 a nastavený kurzor na začátek tohoto řádku

### Kreslení obrázků bez komprese:

Obrázek se začíná kreslit od místa nastavení kurzoru, tedy obrázek lze nakreslit za textem. Pokud kreslený obraz dosáhne konce displeje tak se začíná od začátku řádku 0 a původní obraz je překreslen novými vyslanými daty.



**Instrukce: 0b000010**

**Struktura pro vykreslení 8 svislých bodu:**

První byt		Druhý byt
Adresa LCD 2 bity	Instrukce 6 bitu	Data 1 byt (0-255)

**Vzor pro odesílání přes rozhraní:**

**Adresa displeje je nastavena na 0b10**

0b10000010=0x82

**Vstup:** 0x82 0xFF

**Vystup na displej:** výpise na displej svislých 8 černých bodu

**Vstup:** 0x82 0x00

**Vystup na displej:** výpise na displej svislých 8 bílých bodu

**Kreslení obrázků s kompresí:**

Kreslení obrázku s kompresí je velice podobné kreslení obrázku bez komprese. Rozdíl je v tom, že pokud chceme vykreslit 8 svislých černých nebo bílých bodu a potřebujeme jich vykreslit více než jen jeden, můžeme k tomu využít právě tuto instrukci, které jen stačí říci, zda chceme kreslit černé nebo bílé body, a kolik těchto bodů chceme vykreslit. Kreslení jiné kombinace bodů už funguje stejně jako při kreslení obrázku bez komprese. Kreslení obrázku s kompresí a bez komprese můžeme mezi sebou libovolně kombinovat.

**Instrukce: 0b000011**

**Struktura pro vykreslení 8 svislých bodu jiných než 8 svislých černých nebo bílých bodu :**

První byt		Druhý byt
Adresa LCD 2 bity	Instrukce 6 bitu	Data 1 byt (0-255)

**Struktura pro vykreslení 8 svislých černých nebo bílých bodu:**

První byt		Druhý byt	Třetí byt
Adresa LCD 2 bity	Instrukce 6 bitu	8 černých nebo bílých bodu 0xFF nebo 0x00	Počet těchto svislých bodu (0-255)

**Vzor pro odesílání přes rozhraní:**

**Adresa displeje je nastavena na 0b10**

0b10000011=0x83

**Vstup:** 0x82 0xFF 0x09

**Vystup na displej:** výpise na displej 9 krát 8 svislých černých bodu

**Vstup:** 0x82 0x00 0x09

**Vystup na displej:** výpise na displej 9 krát 8 svislých bílých bodu

**Vstup:** 0x82 0xF0

**Vystup na displej:** výpise na displej 4 černé a 4 bílé body shora černými začínající a bílými končící.

### Inverze:

Instrukce inverze zinvertuje výstup pro displej po vykonání této instrukce se výstup bude vypisovat obráceně. To znamená, že bílé body se budou vypisovat jako černé, a černé jako bílé. Pro opětovnou inverzi zpět je třeba tuto instrukci zopakovat.

#### **Instrukce: 0b000101**

##### **Struktura pro inverzi výstupu pro displej:**

1 byt	
Adresa LCD 2 bity	Instrukce 6 bitu

#### **Vzor pro odesílání přes rozhraní:**

##### **Adresa displeje je nastavena na 0b10**

0b10000101=0x85

**Vstup:** 0x85 0x81 'A' 0x81 'H' 0x81 'O' 0x81 'J'

**Vystup na displej:** AHOJ

**Vstup:** 0x85 0x81 'A' 0x81 'H' 0x81 'O' 0x81 'J' 0x85 0x81 'A' 0x81 'H' 0x81 'O' 0x81 'J'

**Vystup na displej:** AHOJ AHOJ

// tento vystup nenavazuje na předchozí pokud by navazoval tak by výpis proběhnul obráceně: AHOJ AHOJ

### Zapnout nebo vypnout SIGNAL LED1:

Toto je jedna z periférií, kterou může uživatel použít pro svoji činnost, například signalizovat nějakou událost modrou LED1 diodou.

#### **Instrukce: 0b000111**

##### **Struktura pro zapnutí nebo vypnutí SIGNAL LED1:**

1 byt	
Adresa LCD 2 bity	Instrukce 6 bitu

#### **Vzor pro odesílání přes rozhraní:**

##### **Adresa displeje je nastavena na 0b10**

0b10000111=0x87

**Vstup:** 0x87

**Vystup na displej:** Zapne LED1 pokud bylo vypnuté, vypne LED1 pokud bylo zapnuté

### Zapnout nebo vypnout SIGNAL LED2:

Toto je jedna z periférií, kterou může uživatel použít pro svoji činnost, například signalizovat nějakou událost modrou LED2 diodou.

#### **Instrukce: 0b001000**

##### **Struktura pro zapnutí nebo vypnutí SIGNAL LED2:**

1 byt	
Adresa LCD 2 bity	Instrukce 6 bitu

**Vzor pro odesílání přes rozhraní:**

**Adresa displeje je nastavena na 0b10**

0b10001000=0x88

**Vstup:** 0x88

**Výstup na displej:** Zapne LED2 pokud bylo vypnuté, vypne LED2 pokud bylo zapnuté

**Zapnout nebo vypnout magnetodynamický měnič:**

Toto je jedna z periférií, kterou může uživatel použít pro svoji činnost, například signalizovat nějakou událost zvukovou signalizací.

**Instrukce: 0b001001**

**Struktura pro zapnutí nebo vypnutí magnetodynamický měnič:**

1 byt	
Adresa LCD 2 bity	Instrukce 6 bitu

**Vzor pro odesílání přes rozhraní:**

**Adresa displeje je nastavena na 0b10**

0b10001001=0x89

**Vstup:** 0x89

**Výstup na displej:** Zapne sirénu pokud bylo vypnuté, vypne sirénu pokud bylo zapnuté

**Test periférii:**

Touto instrukci je možné rychle otestovat základní periférie na modulu. Test probíhá tak, že postupně za sebou zapíná a vypíná periférie v krátkých časových intervalech. Poslechem a pohledem se dá zjistit, zda periférie fungují.

**Instrukce: 0b001011**

**Struktura pro test periférii:**

1 byt	
Adresa LCD 2 bity	Instrukce 6 bitu

**Vzor pro odesílání přes rozhraní:**

**Adresa displeje je nastavena na 0b10**

0b10001011=0x8B

**Vstup:** 0x8B

**Výstup na displej:** Zapne LED1 vypne LED1, zapne LED2 vypne LED2, zapne sirénu vypne sirénu.

## Podsvícení:

Touto instrukcí umožňuje zapnout nebo vypnout podsvícení displeje.

### Instrukce: 0b001010

#### Struktura pro zapnutí nebo vypnutí podsvícení:

1 byt	
Adresa LCD 2 bity	Instrukce 6 bitu

#### Vzor pro odesílání přes rozhraní:

Adresa displeje je nastavena na 0b10

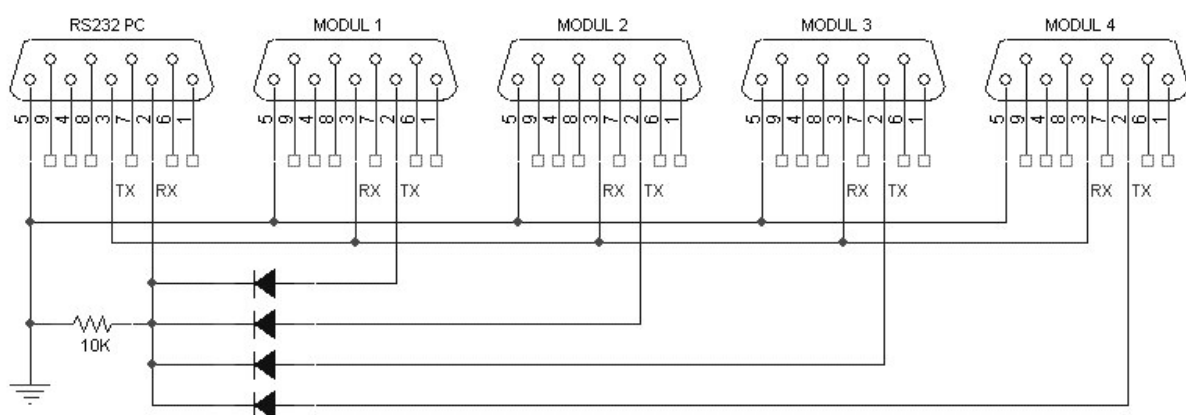
0b10001011=0x8A

Vstup: 0x8A

Výstup na displej: Zapne podsvícení pokud bylo vypnuté, vypne podsvícení pokud bylo zapnuté

#### Komunikace s více displeji:

Jelikož je displej vybaven adresou, je možné na jednu sériovou linku připojit až čtyř moduly. Zde nastává problém při odesílání příznaku od stran modulu, kdy se musí zamezit kolizím, aby ve stejný okamžik neodesílaly moduly příznak. Tento problém se dá odstranit pomocí softwarového řízení modulu. V okamžiku, kdy moduly zapneme k napájení můžeme bezpečně odeslat 249B do každého modulu, aniž bychom se báli naplnění jejich bufferu, a tím odesláním příznaku F. Pokud bychom vyslali 250B, mohlo by nastat to, že se buffer zaplní a vyšle příznak F. Po tomto kroku je potřeba před každým odesláním informace počkat určitý časový usek o velikosti přibližně 1 ms, zda nepřijde od modulu příznak. Pokud modul odešle příznak F, bude třeba vyčkat do doby, dokud se buffer neuvolní a nevyšle příznak R. Po té je opět možné odesílat data všem modulům.



Obr. 3.7.2: Doporučené zapojení pro komunikaci s více moduly.

Na obrázku 3.7.2 je doporučena zapojení pro komunikaci s více moduly. diody slouží jako ochrana proti zkratu.

## Závěr

Na závěr bych chtěl podotknout, že tento modul není zcela hotov. Mám v úmyslu tento modul rozšířit o mnoho dalších funkcí jako například rozhraní SPI, čtení hodnot z teplotního čidla a další rozmanité možnosti, které se vývojářům budou hodit a ulehčí to jejich práci. Také zamýšlím opravit chyby, kterých jsem se dopustil na prvním prototypu (jako je zapojení podsvícení, umístění otvoru šroubu a softwarových chyb). Tento modul bych chtěl dále používat u svých dalších výtvorů, kde budu potřebovat LCD displej.



## Zdroje

[ 1 ] <<http://hw.cz/rs-232>>

[ 2 ] <<http://hw.cz/novinky/art2725-ovladani-grafickych-lcd-modulu-s-radicem-ks0108-s6b0108.html>>

[ 3 ] <<http://www.tme.eu/cz/>>

[ 4 ] <<http://www.gme.cz/cz/index.php>>

[ 5 ] <<http://www.microchip.com>>





## **Obsah CD**

- [ 1 ] Kompletní zdrojový kód v jazyce C.
- [ 2 ] Přeložený kód pro nahrání do mikroprocesoru.
- [ 3 ] Kompletní schéma v programu OrCAD Capture.
- [ 4 ] Vyhotovený plošný spoj v programu OrCAD PCB Editor.
- [ 5 ] Vygenerovaná data pro výrobu plošného spoje.
- [ 6 ] Datasheet mikroprocesoru PIC18F4520.
- [ 7 ] Datasheet LCD displeje.
- [ 8 ] Tento dokument ve formátu pdf.
- [ 9 ] Tento dokument ve formátu doc.



## Příloha A - Seznam použitých součástek

R12-R17,R6,R7,R3	Odpor 0805	10	KOhm
R8,R10,R11?,R4,R5,R1	Odpor 0805	330	Ohm
R2	Odpor 0805	3K3	Ohm
C1,C2,C6	Keramický kondenzátor 0805	100	nF
C3,C4	Keramický kondenzátor 0805	15	pF
C5	Keramický kondenzátor 0805	330	nF
C7-C11	Tantalový kondenzátor	100	nF
OSC1	Krystal 10 Mhz	10.00M	SMDHC49R
PIC18F4520	Mikroprocesor	PIC18F4520-I/PT	
LCD CONECT	Lámací lišta 20 pinů		
CONTRAST	Trimer 10 KOhm	TS53YL10K	
RESET	Spínač	TACTM-63N	
T1	Tranzistor NPN	BC849C	
PIEZO	Magnetodynamický měnič	LD-BZEG-1305	
SIGNAL LED1, SIGNAL LED2	LED dioda modrá 0805		
SW4	Přepínač DIP-SWITCH SMD	ESD104	
SW3	Přepínač DIP-SWITCH SMD	ESD102	
RS232TTL,SPI	Male socket 6 pinů	09185066324	
ICSP	Vidlice zahnutá 5 pinů	PSH02-05WG	
POWER	Vidlice zahnutá 2 piny	PSH02-02WG	
POWER LED	LED dioda zelená 0805		
78M05CDT	Stabiliz. napětí +5V	78M05CDT	
MAX232	Transceiver RC232 SMD	ICL232-SMD	
RS232PC	Vidlice D-SUB 9 pinů	09651616711	
MCP9700?	Teplotní čidlo SMD	TC1047AVNBTR	
D4	Usměrňovací dioda SMD	MBR0530T1G	

