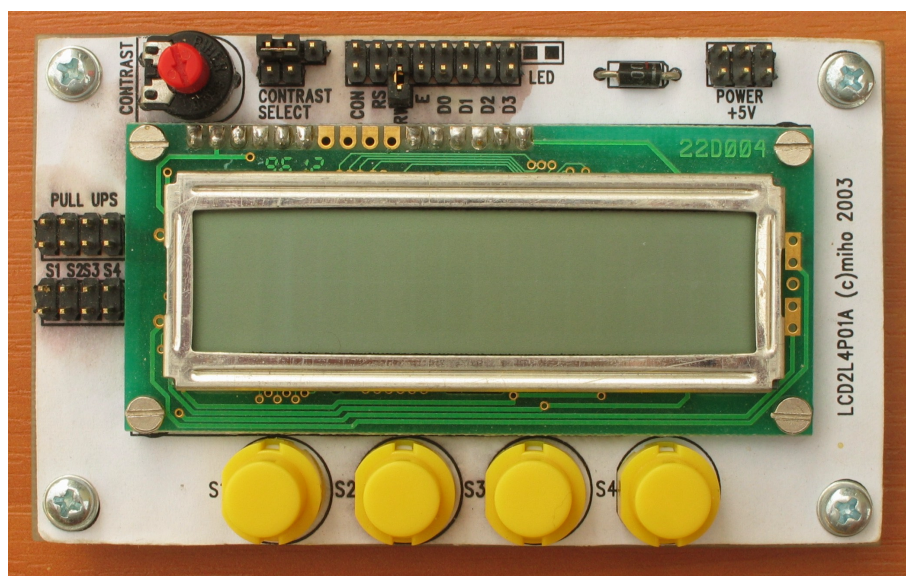


Modul LCD displeje se čtyřmi tlačítky

Milan Horkel

Modul LCD2L4P obsahuje dvouřádkový LCD displej s obvyklým Hitachi řadičem a čtveřicí tlačítek. Používá se jako univerzální uživatelský interfejs při pokusech s mikroprocesorovými konstrukcemi



1. Technické parametry

Parametr	Hodnota	Poznámka
Napájení	+5V nebo +3V/-2V	Záporné napětí pro řízení kontrastu
Spotřeba	cca 1mA	Při +5V
Displej	LCD, 2 řádky po 16 znacích	Hitachi řadič
Tlačítka	4 spínací tlačítka do země	Volitelný pull-up odpor
Rozměry	61 x 101 x 22 mm	Výška nad nosnou deskou

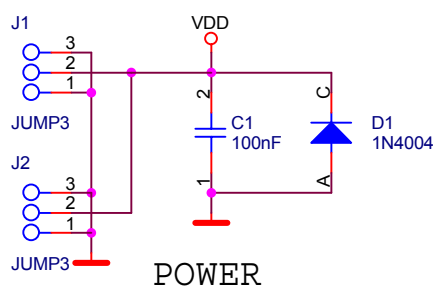
2. Popis konstrukce

2.1. Úvodem

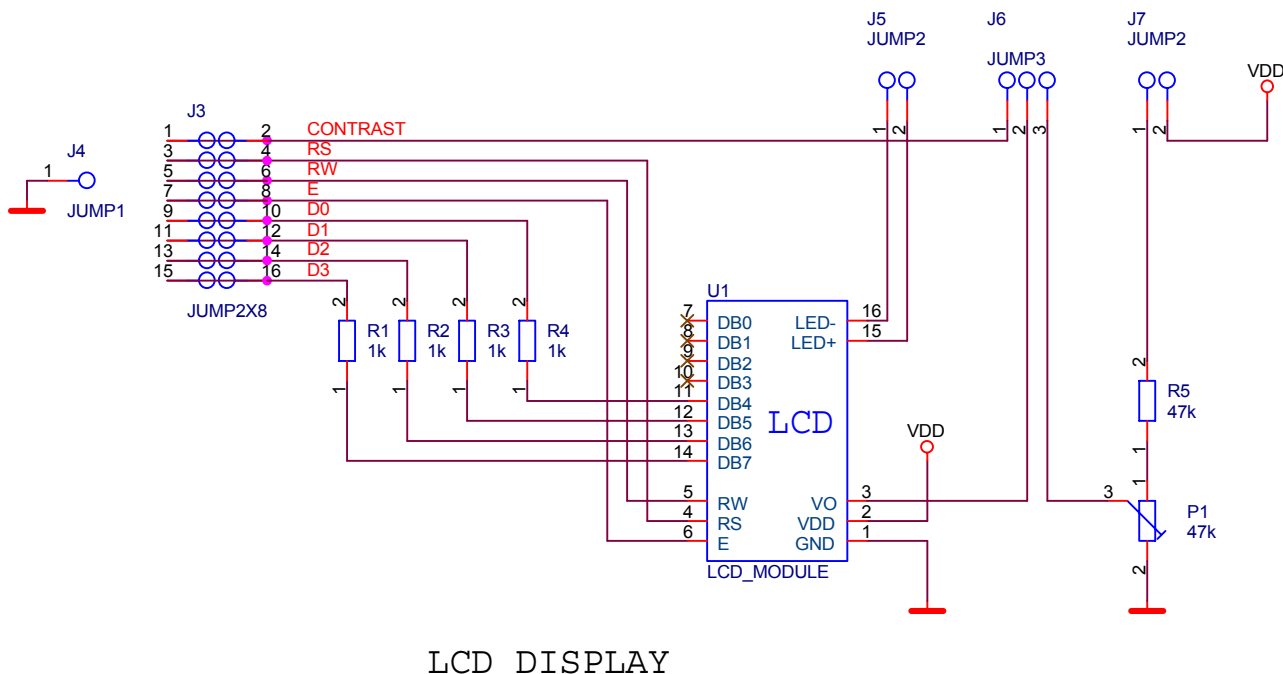
Na modulu je osazen obyčejný standardní dvouřádkový LCD displej s Hitachi řadičem (to jsou skoro všechny dostupné řádkové displeje). Z displeje je vyveden jen 4 bitový interfejs a příslušné řídicí signály. Pomocí propojek je možné připojit trimr pro řízení kontrastu displeje a trvale nastavit řídicí signál RW na režim jen zápis (v obvyklých konstrukcích se z LCD displeje nic nečte).

Pod displejem jsou umístěna 4 velká spínací tlačítka do země. Pomocí řady propojek označené „PULL UPS“ je možné připojit pull-up odpory do plusu (hodnota 10k).

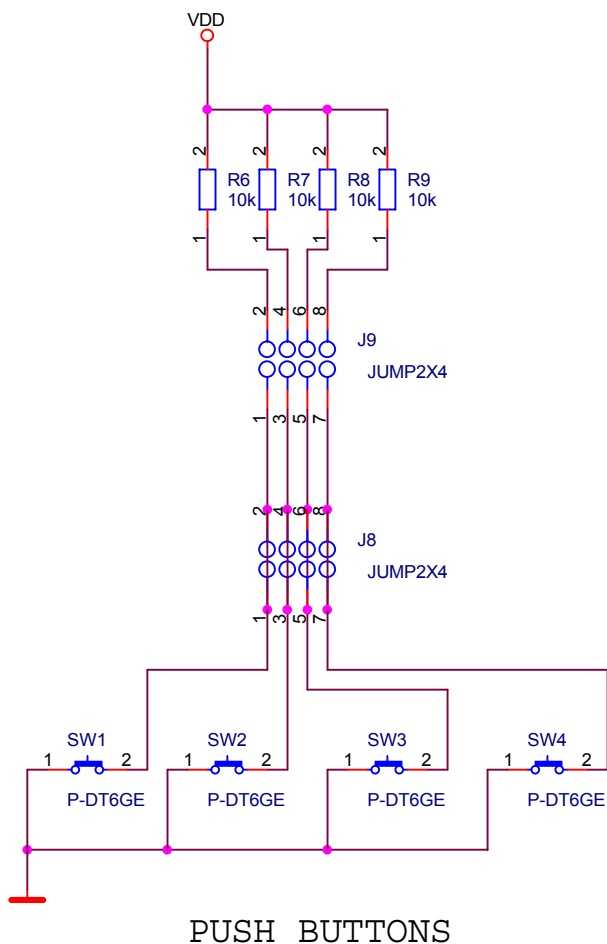
2.2. Zapojení modulu



Dioda D1 je ochranná dioda proti přepólování napájení. Standardní napájení je +5V ale v případě potřeby může být napájení menší ale je třeba použít záporného napětí pro řízení kontrastu LCD displeje (potřebný rozdíl mezi kladným napájením a záporným předpětím je cca 5V). Při sníženém napájecím napětí je podstatně snížena spotřeba. Napájecí proud pro řízení kontrastu je velmi malý a je možné snadno jej nastřídat.



LCD DISPLAY



Zapojení tlačítek je přímočaré. Odporů R6 až R9 jsou volitelné (propojkami J9) pull-up odporů. Tlačítka jsou napevno zapojená „do země“.

Reference

Název

Odporů

R1,R2,R3,R4

1k

R6,R7,R8,R9

10k

R5

47k

Odporové trimry

P1

47k

Keramické kondenzátory

C1

100nF

Polovodiče

D1

1N4004

U1

LCD SC1602A

Mechanické součástky

SW1,SW2,SW3,SW4

P-DT6GE

J4

JUMP1

J5,J7

JUMP2

J6

JUMP3

J1+J2

JUMP2X3

J8,J9

JUMP2X4

J3

JUMP2X8

2.3. Ovládání LCD displeje

LCD displej se ovládá zápisem do řídicích registrů (RS=0), do paměti pro zobrazovaný text (DD RAM) nebo do paměti definice znaků (CG RAM). Z paměti DD RAM a CG RAM je možné rovněž číst ale tato funkce nebývá potřebná a je možné trvale čtení deaktivovat (RW=0).

Zápis (i čtení) může buď probíhat najednou po 8 bitech nebo nadvakrát po 4 bitech. Druhý způsob je výhodný z hlediska snížení počtu portů procesoru, ke kterému je displej připojen. Pozor, pro čtyřbitový provoz se používají vývody D7 až D4.

Obsah paměti DD RAM definuje (v ASCII kódu + japonské znaky), co se bude zobrazovat. Pro jednotlivé řádky je vyhrazen určitý rozsah adres v DD RAM ze kterého je vidět jen výřez. Je možné volit, zda se při zápisu znaků do DD RAM automaticky zvyšuje adresa do které se zapisuje nebo zda se posouvá okno. Posouvání okna je možné řídit nezávisle a je tam možné snadno realizovat běžící text.

Instrukce	Řízení		Datová sběrnice							
	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0
Smaž displej a nuluj adresu DD	0	0	0	0	0	0	0	0	0	1
Nuluj adresu DD na 0 a zruš posunutí	0	0	0	0	0	0	0	0	1	*
Nastavení pohybu kurzoru	0	0	0	0	0	0	0	1	ID	S
Nastavení módu displeje	0	0	0	0	0	0	1	D	C	B
Okamžitý posuv kurzoru nebo textu	0	0	0	0	0	1	SC	RL	*	*
Nastavení funkce displeje	0	0	0	0	1	DL	N	F	*	*
Nastavení adresy CG	0	0	0	1	A5	A4	A3	A2	A1	A0
Nastavení adresy DD	0	0	1	A6	A5	A4	A3	A2	A1	A0
Čtení adresy a busy bitu	0	1	BF	A6	A5	A4	A3	A2	A1	A0
Zápis dat	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Čtení dat	1	1	D7	D6	D5	D4	D3	D2	D1	D0

Bit	Popis
S=1	Při zápisu/čtení dat se bude posouvat celý displej (směr určuje bit ID)
S=0	Při zápisu/čtení dat se bude posouvat jen kurzor (směr určuje bit ID)
ID=1	Při zápisu/čtení dat je ukazatel adresy inkrementován
ID=0	Při zápisu/čtení dat je ukazatel adresy dekrementován
B=1	Blikající znak na pozici kurzoru
B=0	Znak nebliká
C=1	Zapnutí kurzoru
C=0	Vypnutí kurzoru
D=1	Zapnutí displeje
D=0	Vypnutí displeje
RL=1	Okamžitý posuv doprava
RL=0	Okamžitý posuv doleva
SC=1	Okamžitý posuv textu
SC=0	Okamžitý posuv kurzoru
F=1	Znaky v rastru 5x10
F=0	Znaky v rastru 5x7 (běžné)
N=1	Počet řádek displeje 2 nebo 4
N=0	Počet řádek displeje 1 (některé jednořádkové displeje jsou dvouřádkové – místo řádek mají levou a pravou polovinu)
DL=1	Přenos dat 8 bitů
DL=0	Přenos dat 4 bity (používají se vodiče D7 až D4)
BF=1	Bussy – probíhá vnitřní operace
BF=0	Ready – je možné posílat instrukce

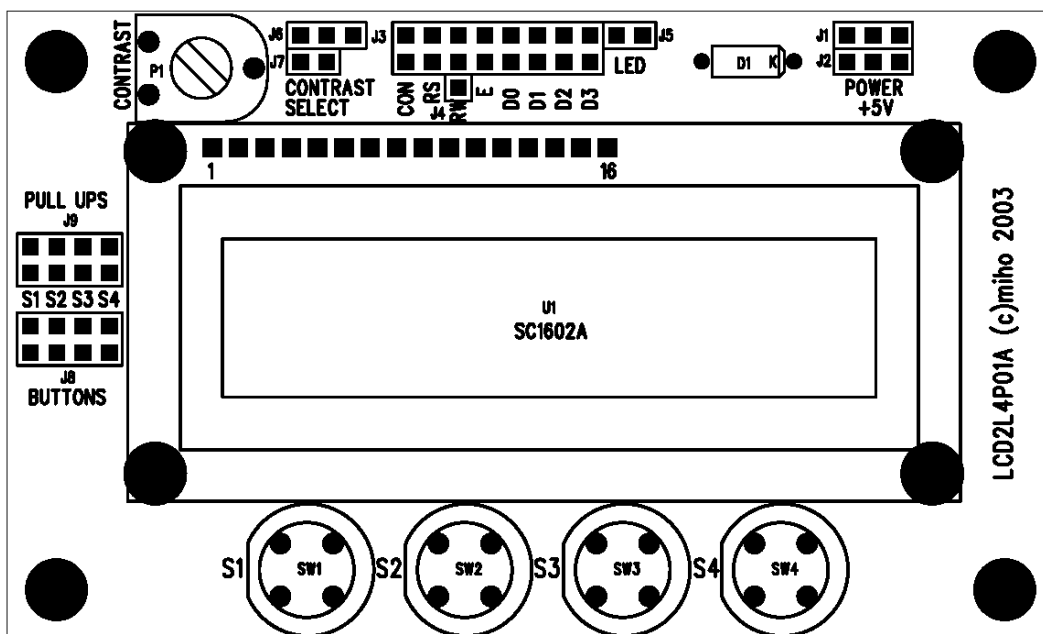
2.4. Mechanická konstrukce

Mechanická konstrukce je standardní. Displej je přišroubován šrouby M2.5, jako distanční podložky mezi displej a plošný spoj je možné použít maticky M3.

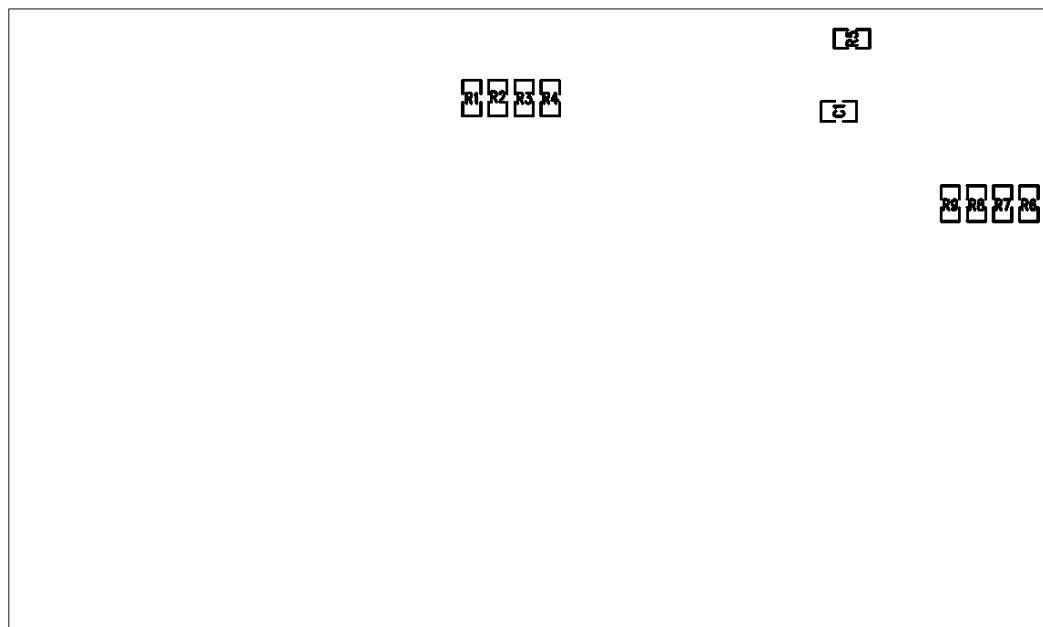
3. Osazení a oživení

3.1. Osazení

Strana součástí



Strana spojů



3.2. Oživení

Pokud jsou součásti, plošný spoj a pájení v pořádku není co oživovat.

4. Programové vybavení

Pro ovládání LCD displeje procesorem PIC je připraven knihovní modul LCD.C napsaný v jazyce C pro překladač CCS.

4.1. Uživatelský návod

V programu, který bude ovládat LCD displeje je třeba definovat připojení LCD displeje a včlenit knihovnu LCD.C do zdrojového programu. První příklad ukazuje jak se definuje připojení LCD displeje, když jsou datové bity displeje připojeny na libovolné porty procesoru:

```
#define LCD_RS          PIN_A2          // Signal RS
#define LCD_E          PIN_A3          // Signal E
#define LCD_D0         PIN_A4          // Data
#define LCD_D1         PIN_B1
#define LCD_D2         PIN_B2
#define LCD_D3         PIN_B3
#include <LCD.C>
```

Následující příklad zobrazuje připojení, kdy jsou jednotlivé bity datové sběrnice displeje připojeny na jediný port procesoru za sebou:

```
#define LCD_RS          PIN_A2          // Signal RS
#define LCD_E          PIN_A3          // Signal E
#define LCD_DATA_LSB   PIN_C2          // 4 bity za sebou
#include <LCD.C>
```

Pak je třeba LCD displej inicializovat (například ve funkci main()):

```
lcd_init();
```

Pak je možné buď posílat znaky na displej rovnou pomocí funkce LCD_putc() nebo prostřednictvím funkce printf():

```
LCD_Putc(0x40);
printf(LCD_Putc, "\n\rText1\r\nText2\n\r");
```

V případě potřeby je možné nadefinovat nové znaky (až 8 znaků najednou). Jejich tvar se uloží do CG RAM v displeji a je možné je zobrazit jako znaky s kódem 0x00 až 0x07 (nebo 0x08 až 0x0F, 0x10 až 0x17 a 0x18 až 0x1F). Tvary českých znaků a některých užitečných symbolů je již předpřipravena a používá se takto:

```
lcd_define_char(0,LCD_CHAR_BAT50);
    // Definice znaku baterie do pozice 0
lcd_define_char(2,LCD_CHAR_HLE LCD_CHAR_LUI);
    // Definice znaků ě a í od pozice 2
    // Mezi retezci s definicí není carka
```

```
printf(lcd_putc, "\fZnaky:\20\22\23");  
    // Vypis znaku z pozice 0, 2 a 3  
delay_ms(1000);  
    // Prodleva  
lcd_define_char(0, LCD_CHAR_BAT0);  
    // Predefinovani tvaru znaku v pozici 0  
delay_ms(1000);  
    // Prodleva
```

Do displeje je možné posílat řídicí příkazy nebo displej řídit pomocnými funkcemi takto:

<code>\f = \x0C</code>	nová stránka = smazání displeje
<code>\n = \x0A</code>	odřádkování = přechod na druhou řádku
<code>\b = \x08</code>	backspace = posunutí kurzoru o 1 pozici vlevo
<code>\r = \x0D</code>	přesune se na pozici 1,1
<code>\0 .. \7</code>	uživatelsky definovaný znak v pozici 0 až 7 v CG RAM
<code>\20 .. \27</code>	alternativní způsob jak vypsát uživatelsky definované znaky (funkce printf() chápe znak 0 jako ukončení tisku)
<code>lcd_gotoxy(x,y)</code>	přesuň kurzor na uvedenou pozici (1,1 znamená domu)
<code>lcd_cursor_on()</code>	zapne kurzor
<code>lcd_cursor_off()</code>	vypne kurzor

4.2. Popis programu

V případě nejasnosti je vhodné nahlédnout do zdrojového textu LCD.C a případně vyhledat podrobnosti. Na konci souboru je uveden příklad definování uživatelských LCD znaků a hned před ním je definice konstant pro české znaky a některé pomocné znaky včetně jejich popisu.

Aktuální verze LCD.C se může mírně lišit od výše uvedeného popisu, je proto třeba primárně vycházet z komentářů ve zdrojovém kódu.