

Linux extra / 9.časť

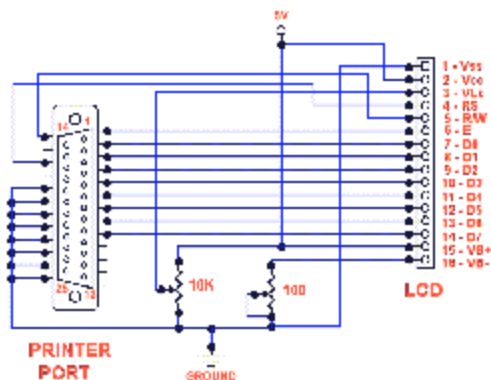
LCD displeje II

V minulej časti sme si povedali niečo o LCD displejoch a dnes pristúpime k samotnej realizácii.

Realizácia

Na Internete sa nachádza mnoho projektov, v ktorých je použitý LCD displej. Najčastejšie sa používa osembitové zapojenie, pripájajúce sa na paralelný port počítača, ktoré sa stalo určitým štandardom.

Pre naše potreby môžeme použiť zapojenie podľa pána *Andyho Rasa*, ktoré je na obrázku č.1:



Na ľavej strane je konektor CANON25M, ktorý sa zapája do paralelného portu počítača, na pravej strane je zapojenie na jednotlivé piny LCD displeja. Trimer o hodnote 10 kiloOhmov slúži na nastavenie kontrastu zobrazovaných znakov, trimer o hodnote 100 Ohmov nastavuje intenzitu podsvietenia celého displeja.

Paralelný port nemá vyvedené napájanie +5V, preto ho musíme získať inak.

Napájanie +5 Voltov môžeme získať z viacerých možností – jedna možnosť je použiť klávesnicový adaptér, čo je akýsi medzikus, ktorý sa zapojí medzi klávesnicu a počítač a má vyvedené napätie 5 Voltov. Tento adaptér môžeme kúpiť v obchode s počítačmi. Druhá možnosť je „vytiahnuť“ napájanie z *gameportu*, čo je konektor, do ktorého sa pripája joystick. Ak riešime jednotku ako interný modul, potom je najvýhodnejšie použiť konektor, aký sa používa na napájanie harddiskov alebo CD ROM mechaník, kde červený kábel je +5 Voltov a čierny je zem.

Taktiež môžeme použiť zapojenie typu **lcdmod**. Toto sa mierne líši od vyššie uvedeného zapojenia a popis jednotlivých prípojných bodov je v tabuľke č.2:

Vývod displeja	Označenie vývodu	Vstup/ Výstup	Význam	Paralelný port	
				lcdmod	Randy Rasa
1	Vss		0 Voltov (napájanie, zem)	18	18-25
2	Vcc		+5 Voltov (napájanie , plus)	+5V	+5V
3	Vee		nastavenie kontrastu	18	bežec potenciometra 10kOhm
4	RS	vstup	0 = vstup sú inštrukcie 1 = vstup sú dáta	14	16
5	R/W	vstup	0 = zápis dát do LCD 1 = čítanie dát z LCD	19	14
6	E	vstup	aktivácia displeja	1	1
7	DB0	V/V	dáta	2	2
8	DB1	V/V	dáta	3	3
9	DB2	V/V	dáta	4	4
10	DB3	V/V	dáta	5	5
11	DB4	V/V	dáta	6	6
12	DB5	V/V	dáta	7	7

13	DB6	V/V	dáta	8	8
14	DB7	V/V	dáta	9	9
15	anóda		podsvetlenie, anóda		+5 V
16	katóda		podsvetlenie, katóda		vývod potenciometra 100 Ohm

V prvom stĺpci sú uvedené poradové čísla vývodov LCD displeja, v druhom stĺpci je ich označenie. Tretí stĺpec popisuje, či sa jedná o vývod vstupu alebo obojsmerný vývod (V/V) a štvrtý stĺpec popisuje ich význam.

V stĺpcoch označených ako *lcdmod* a *Randy Rasa* sú uvedené čísla pinov konektora paralelného portu.

Všimnime si, že sa líšia len máličko, hlavne na vývode displeja č.4 a č.5.

Ak teda postavíme zariadenie podľa *Randyho Rasa*, spojíme vývod displeja č.4 s pinom paralelného portu č.16 a vývod displeja č.5 s pinom paralelného portu č.14, ak použijeme zapojenie typu *lcdmod*, tak spojíme vývod displeja č.4 s pinom paralelného portu č.14 a vývod displeja č.5 s pinom paralelného portu č.19.

Pozor!

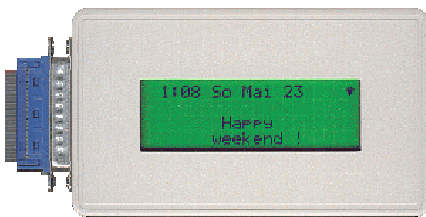
Uvedené zapojenia nie sú kompatibilné. Každé pracuje s iným ovládačom a softvérom, preto si najprv v dokumentácii k programu preštudujeme, aké zapojenie treba použiť.

Schému môžeme realizovať ako prípojný modul pomocou kábla k tlačiarňi, tak ako je to na obrázku č.3:



V takom prípade nesmieme zabudnúť, že dĺžka prípojného kábla nesmie prekročiť normu pre paralelný port, čo je asi 3 až 5 metrov.

Komu sa nepáči metóda „vrabčieho hniezda“, ten môže displej aj s príslušnou elektronikou umiestniť do elegantnej krabičky – obrázok č.4:



Ak chceme mať displej ako pevnú súčasť počítača, je najvýhodnejšie ho zabudovať ako modul do prázdnej šachty pre disketové alebo CD ROM mechaniky v skrinke počítača – obr.č.5:



Ovládač

Keď už máme zapojenie zostrojené, potrebujeme určitý program, ktorý bude naše zariadenie ovládať. Tomuto programu sa hovorí ovládač alebo driver (foneticky drajver).

Stiahneme si z internetovej stránky www.cevaro.sk alebo lcd-mod.sourceforge.net súbor **lcdmod-1.0.2.tgz** (táto verzia už podporuje jadro 2.6.x, ak by sme chceli súbor pre jadro 2.4.x, stiahneme si balíček **lcdmod-0.6.7.tgz**).

V príslušnom adresári balíček rozbalíme príkazom

```
[root@rubin root] # tar xvzf lcdmod-1.0.2.tgz
```

Vytvorí sa adresár **lcdmod-1.0.2** a v ňom vyhladáme súbor *config.h*.

V tomto súbore môžeme určiť spôsob zapojenia zariadenia. Jednotlivé položky začínajú znakom #. Niektoré položky sú už predpripravené a zakomentované znakmi // a preto stačí položku vybrať tým, že ju odkomentujeme (zmažeme //).

Ak teda máme zapojenie podľa Randyho Rasa, vyberieme položku **#define WIRING_RANDY_RASA** (všimnime si, že môžeme použiť aj zapojenie typu *lcdmod*).

Zároveň určíme adresu paralelného portu, ku ktorému je zariadenie pripojené (štandardne 378hex) položkou **#define DFLT_BASE 0x378**, počet riadkov displeja, napríklad 2 položkou **#define DFLT_DISP_ROWS 2** a počet znakov na riadku, napríklad 16, čo je položka **#define DFLT_DISP_COLS 16**.

Upravený súbor *config.h* uložíme.

Potom pristúpime ku kompilácii príkazmi

```
[root@rubin lcdmod-1.0.2] # make
[root@rubin lcdmod-1.0.2] # make install
```

Prekladom sa vytvorí súbor *lcd.o*, čo je nami žiadaný ovládač.

Keďže uvedený súbor je modulom jadra, zároveň sa upraví aj súbor */etc/modules.conf*, čím sa zabezpečí, že pri štarte počítača sa do pamäte natiahne súbor *lcd.o* a vytvorí sa nové zariadenie */dev/lcd*.

V prípade, že sa tak nestane, môžeme to vykonať ručne príkazom

```
[root@rubin lcdmod-1.0.2] # modprobe lcd
```

Môže sa stať, zvlášť na jadre 2.4.x, že vyhlási chybu:

```
LCD: parport busy
```

V takom prípade musíme odstrániť modul *lp.o*, napríklad príkazom

```
[root@rubin lcdmod-1.0.2] # rmmod lp
```

a zavedenie skúsime znovu.

Hlásenie na obrazovke typu:

```
LCD: init OK, iobase 0x378, rows: 2, columns: 16
```

potvrďuje, že sa modul zaviedol správne.

Niekedy sa môže stať, že chceme zmeniť parametre pripojenia, napríklad použijeme iný displej (elektronika zostáva rovnaká!), tak zadáme príkaz

```
[root@rubin lcdmod-1.0.2] # insmod lcd io=0x378 disp_rows=4 disp_cols=20
```

Či sa modul naozaj zaviedol do pamäte môžeme overiť aj príkazom

```
[root@rubin lcdmod-1.0.2] # lsmod
```

Teraz je ovládač správne nainštalovaný!

Test

Aby sme si overili, či naozaj všetko funguje, naše zariadenie otestujeme. Zadáme príkaz

```
[root@rubin lcdmod-1.0.2] # echo Ahoj > /dev/lcd
```

a na displeji nášho zariadenia sa objaví milé uvítanie. Ak tento pozdrav vidíme, môžeme sami seba pochváliť, lebo sme fakt, ale fakt dobrí....

Obsluha

Zariadenie je pripojené, ovládač nainštalovaný, takže teraz sa musíme naučiť toto všetko obsluhovať. Už s testu je vidieť, že obsluha je pomerne jednoduchá. Ak chceme niečo zobraziť na displeji, stačí ak to pomocou klasického linuxového príkazu **echo** presmerujeme na zariadenie `/dev/lcd`.

Riadiace kódy

Už minule sme si povedali, že do nášho inteligentného displeja môžeme zasielať nielen prostý text, ale aj riadiace kódy, ktorými môžeme určovať inteligentnú činnosť displeja, ako je výmaz, posun, zmena tvaru kurzora alebo definovanie vlastných užívateľských znakov.

Keďže modul lcd.o simuluje terminál typu VT52 (tí skorej narodení vedia, o čo sa jedná), tieto príkazy budeme zasielať ako *escape sekvencie*. Escape sekvencie sú znaky, ktorými sa ovládali terminály podľa normy ANSI. My sa teraz nimi nebudeme podrobnejšie zaoberať, v tabuľke č.6 je zoznam riadiacich kódov pre LCD displeje:

sekvencia	Význam
Znaky s ESC sekvenciou (musia začínať \033)	
[A	presunie kurzor o riadok hore
[B	presunie kurzor o riadok dole
[C	presunie kurzor o jednu pozíciu doprava
[D	presunie kurzor o jednu pozíciu doľava
[H	presunie kurzor "domov", teda do ľavého horného rohu
[J	zmaže všetko bez premiestnenia kurzoru domov
[K	zmaže od kurzoru do konca riadku bez premiestnenia kurzoru domov
[M	mapuje znaky (neskúšajte bez rozmyslu!)
[Y	priamy prístup na určitú pozíciu
[R	vybratie CGRAM
[V	zalamovanie riadkov zapnuté
[W	zalamovanie riadkov vypnuté
[b	prepína osvetlenie ZAP/VYP
Kódy bez ESC sekvencie (teda bez \033)	
\r	presunie kurzor na začiatok riadku
\n	presunie kurzor na nový riadok
\t	tabulátor (štandardne 3 znaky)

Príklad použitia

Príkaz

```
echo -ne "\033[H\033[J" > /dev/lcd
```

vymaže displej a presunie kurzor „domov“ (do ľavého horného rohu displeja).

Keď to spojíme s textom a ďalšími riadiacimi znakmi, napríklad

```
echo -ne "\033[J\033[HToto je prvý riadok \r\nToto je druhý riadok"> /dev/lcd
```

tak sa stane, že displej bude vymazaný, kurzor bude presunutý „domov“ a začne sa vypisovať text **Toto je prvý riadok**, potom sa kurzor vráti na začiatok riadku, prejde na druhý riadok a vypíše sa **Toto je druhý riadok**.

Po výpise zostáva text zobrazený až do najbližšieho nového výpisu alebo do odstránenia modulu `lcd.o` z pamäte počítača, čo môžeme využiť v našich skriptoch. Stačí, ak budeme cyklicky zbierať informácie z počítača, budeme ich nejako formátovať a potom vypisovať.

Príkladom môže byť skript `display_date.sh`, ktorý je súčasťou balíčka.

Jeho obsah je na výpise č.7:

```
#!/bin/sh
while true;
do
    echo -en "\033[J\033[H`date +"%A %e %b%"` \n`date +%r`" > /dev/lcd
    sleep 1s
done
```

Vidíme, že sa jedná o klasický shellovský skript, ktorý v pravidelnom intervale jednej sekundy vypisuje na obrazovku displeja dátum a čas, napríklad takto:

```
Nedela    4 sep
06:17:45
```

Pozor!

Nesmieme zabudnúť, že skript musí začínať riadkom `#!/bin/sh` a musí mať znak spustiteľnosti!

Tvorba vlastných zobrazovacích znakov

Už minule sme si povedali, že do LCD displeja môžeme na prvých osem pozícií (000 až 007) znakovkej mapy definovať vlastné znaky. Znaký majú rozmer 5x8 pixelov (8 na výšku a 5 na šírku). Definujeme riadky, kde sa berie do úvahy nižších 5 bitov. Plný riadok predstavuje binárne číslo 11111, čo je oktálovo 037.

Ak teda zadáme príkaz

```
echo -en "\033[R1\037\037\037\037\037\037\037" > /dev/lcd
```

vložíme na pozíciu č.1 znak, ktorý bude mať tvar plného obdĺžnika.

Ak zadáme príkaz

```
echo -en "\033[R0\037\000\037\000\037\000\037\000" > /dev/lcd
```

tak na pozíciu č.0 vložíme znak, ktorý sa bude skladať z vodorovných pruhov.

Ak chceme vypísať, aké znaky sú nadefinované na jednotlivých pozíciách, zadáme príkaz

```
echo -en "\000\001\002\003\004\005\006\007" > /dev/lcd
```

Účelové skripty

Ak nás teraz napadajú rôzne spôsoby využitia, nechajme sa inšpirovať nasledujúcimi príkladmi. Aj keď pri nich budeme využívať iné programy, ako `awk`, ktoré slúži na manipuláciu s textom, sú pomerne jednoduché a dajú sa jednoducho modifikovať bez hlbšej znalosti programovania.

Príklad č.1

Chceme vedieť, ako je vyťažený procesor a koľko má dostupnej operačnej pamäte?

Tu je výpis č.8:

```
#!/bin/sh
while true;
```

```
do
top b -n 2 -d 1 | grep "CPU state" | tail -1 | awk \
'{printf "\033[J\033[HCPU vytazenost: "100 - substr($9,1,2)"%"}' \> /dev/lcd
free | grep "Mem:" | awk '{printf "\nVolna pamat : " $4}' \> /dev/lcd
sleep 1s
done
```

Vytáženosť procesora získame z príkazu *top*, kde pomocou *awk* vyhladáme reťazec **CPU state**. Podobne z príkazu systému *free* získame údaj o voľnej pamäti. Všimnime si, že nemusíme použiť príkaz *echo*, samotný program *awk* má funkciu *printf*, ktorá *echo* nahradí. Zobrazenie na displeji bude asi takéto:

```
CPU vytazenost: 17%
Volna pamat    : 15192 kB
```

Príklad č.2:

Na druhom príklade si ukážeme, ako môžeme vypisovať na displeji objem dát prenesený na niektorej sieťovej karte.

Vzor skriptu je na výpise č.9:

```
#!/bin/sh
while true;
do
ifconfig eth1 | grep "RX bytes" | awk \
'{printf "\033[H\033[Jeth1 " $1 " "substr($2,7)"b"}' > /dev/lcd
ifconfig eth1 | grep "RX bytes" | awk \
'{printf "\r\n    " $5 " "substr($6,7)"b"}' > /dev/lcd
sleep 1s
done
```

V skripte využijeme jednoduchý príkaz systému *ifconfig eth1*, pomocou *grep* vytiahneme požadované údaje a odošleme na zariadenie */dev/lcd*.

Na displeji sa objaví niečo podobné:

```
eth0 RX 6567 KiB
      TX 1485 KiB
```

Príklad č.3

Na tomto poslednom príklade zobrazujeme teplotu procesora a otáčky ventilátora. Obsah skriptu je na výpise č.10:

```
#!/bin/sh
while true;
do
sensors | grep temp | tac | tail -1 | awk \
'{printf "\033[H\033[JTeplota : " $2 " " $3}' > /dev/lcd
sensors | grep fan1 | tac | tail -1 | awk \
'{printf "\r\nOtacky : " $2 " ot/min"}' > /dev/lcd
sleep 1s
done
```

Tentokrát sa obrátime na program **sensors**, vytiahneme teplotu a otáčky ventilátora procesora, preženieme programom *awk* a zobrazíme.

Na displeji uvidíme:

```
Teplota: 39,6 °C
Otacky : 4320 ot/min
```

A teraz sa môžeme pustiť do nášho vlastného projektu.

Miroslav Oravec