

```
<div style="text-align: justify;"></div> <div
style="text-align: justify;"><br /></div> <div style="text-align: justify;">Kompatibiln
```

[LCD panel s ATmega8](index.php/koutek-avr/94-kit-2-lcd-panel-s-atmega8-zapojeni-a-programy/524-avr-lcd-pane-l-s-atmega8)

Ke stažení [LCD_002.c](http://www.ok1kvk.cz/upload/ok1wmr/clanky/avr/programy/LCD_002/LCD_002.c)

[LCD_002.pdf](http://www.ok1kvk.cz/upload/ok1wmr/clanky/avr/programy/LCD_002/LCD_002.pdf)

[LCD_002.htm](http://www.ok1kvk.cz/upload/ok1wmr/clanky/avr/programy/LCD_002/LCD_002.htm)

```
</div> <div style="text-align: justify;"><span style="color:
#ffffff;">.</span><br /></div> <div style="text-align: justify;"><span style="color:
#ffffff;">.</span><br /></div> <hr /> <span style="font: 10pt Courier New;"><span
class="cpp1-comment"></span></span><span style="font: 10pt Courier New;"><span
class="cpp1-comment"></span></span> Untitled &lt;!-- body { color: #000000;
background-color: #FFFFFF; } .cpp1-assembler { } .cpp1-brackets { } .cpp1-comment { color:
#008000; font-style: italic; } .cpp1-float { color: #000080; } .cpp1-hexadecimal { color: #000080;
} .cpp1-character { } .cpp1-identifier { } .cpp1-illegalchar { } .cpp1-number { color: #000080; }
.cpp1-octal { color: #0000FF; } .cpp1-preprocessor { } .cpp1-reservedword { font-weight: bold; }
.cpp1-space { color: #008080; } .cpp1-string { color: #800000; } .cpp1-symbol { } --&gt;
<span style="font: 10pt Courier New;"><span class="cpp1-comment">/* Minule jsme na
p?ravku ATmega_LCD_uni vyzkoueli displej, a nyn?se pokus?e "oivit" i kl?esnici. */ /*
Obslu?rutiny pro kl?esnici jsou ulo?eny v souboru "kbd_2x3dm.c". Tento soubor je op?t
z?isl? na konfiguraci ulo?en?v souboru "atmega_lcd_uni_cfg.c" a je proto nutn? aby soubor
"atmega_lcd_uni_cfg.c" byl vlo?en je?t? p?ed souborem "kbd_2x3dm.c". Soubor
"kbd_2x3dm.c" obsahuje n?leduj?dv? rutiny: void kbd_ini(void) //inicializace kl?esnice
(nastaven?vstupn?h a v?stupn?h pin?) unsigned char kbd_read(void) //tato funkce "p?e?te"
kl?esnici, kdykoli je vol?a, a vr?hodnotu pr?? stisknut?ch tla?ek Funkce
"kbd_read(void)" vrac?hodnotu kl?es kter?jsou stisknuty pr?? v tom okam?iku, kdy je vol?a.
Pokud tedy chceme na stisk kl?esy reagovat okam?it?, je vhodn?tuto funkci volat p?ibli?n?
10 - 20 kr? za vte?inu. Jinak by byla patrn?prodleva, kdy je pot?eba tla?ko chv?i dr?et,
ne? si toho procesor "v?imne". Soubor "kbd_2x3dm.c" tak?obsahuje dv?
zaj?vav?konstanty, kter????chov? kl?esnice, pokud je tla?ko dr?eno d?e. Tyto
konstanty jsou: KBD_SKIP KBD_ACC Konstanta "KBD_SKIP" ur?uje, kolik cykl? se
m?vynechat, pokud je stisknuto st?e tot?tla?ko. To znamen? ?e kdy? funkce "kbd_read()"
zjist? ?e je stisknuto stejn?tla?ko (nebo stejn?tla?ka), jako kdy? byla vol?a minule,
vr?m?to hodnoty tohoto tla?ka nulu (jako kdyby nebylo stisknuto ?n? tla?ko).
Konstanta "KBD_SKIP" tedy ur?uje, kolikr? m?funkce "kbd_read()" toto tla?ko ignorovat,
ne? op?t vr?jeho hodnotu. Pokud je tla?ko dr?eno i nad?e, pot?co funkce "kbd_read()"
ji? podru? vr?ila jeho hodnotu, za???se uplat?ovat konstanta "KBD_ACC". Tato konstanta
ur?uje, o kolik cykl? se m?vynechat m? p?ed ka?d?m dal??? navr?en? hodnoty t?o?
tla?ka. V?sledkem tedy je, ?e pokud u?vitel stiskne a dr?tla?ko, vr?se jeho
hodnota, ?ek?se podle "KBD_SKIP", znovu se vr?jeho hodnotu, ?ek?se o "KBD_ACC"
m?, zase se vr?hodnota, ?ek?se o dal??? "KBD_ACC" m?, a tak to jde a? do doby, kdy
```

se vrátí hodnota drženého tlačítka při každém cyklu (volá funkce "kbd_read()"), nebo dokud uživatel tlačíko opět neuvolní. Tato funkce se nazývá "autorepeat" a hodí se například, pokud je potřeba nastavit nějakou proměnnou ve velkém rozsahu. */ /* Nyní se tedy můžeme pustit do psaní programu. Náš program bude jednoduše vypisovat klávesy stisknuté tlačítkem na displej. Tento program bude pro nás velmi užitečný, protože ať budeme později potřebovat, aby program reagoval na dané tlačítko, budeme potřebovat vědět, jaké číslo (klávesa) toto tlačítko představuje. Ten klávesnice bude probíhat vždy jednou za 100 ms (10x za vteřinu), což zajistí pomocná časovač. pro ten klávesnice nám bude stačit jednoduchý časovač TC0. Znovu budeme na displej vypisovat proměnnou "opakování" která se pokaždé když je navržen klávesa zvětšuje o jednu. Je to proměnná typu unsigned char - bude se tedy zvyšovat jen do 255, protože pak přeteče a začne opět od nuly. */

```
#define F_CPU 1000000
// Taktovací frekvence procesoru 1 MHz
#include <avr/io.h>
// Knihovna vstupů a výstupů
#include <avr/interrupt.h>
// Knihovna přerušení
#include "atmega_lcd_uni_cfg.c"
// konfigurace vstupů a výstupů pro přímý řadek ATmega_lcd_uni
#include "lcd.c"
// Obsluhující rutiny pro display
#include "kbd_2x3dm.c"
// Obsluhující rutiny pro klávesnici
#define TCNT0VAL (256-(F_CPU/1024/10))
// Hodnota, od které bude časovač
// počítat, než přeteče a vyvolá přerušení
// 256 => maximum času + 1 = přetečení
// 1024 => délka,
// 10 => přerušení se vyvolá 10x za vteřinu
volatile unsigned char tlcitko = 0;
// Globální proměnná obsahující stisknuté tlačítko. Používá se v main() a přerušení //vektor přerušení vykoná vždy když přeteče časovač TC0 (cca 10x za vteřinu)
ISR(TIMER0_OVF_vect) {
    TCNT0=TCNT0VAL;
    // nastavení hodnoty, od které časovač počítá
    // (jinak by počítal od nuly)
    tlcitko=kbd_read();
    // přičte klávesnici a hodnotu uloží do "tlcitko"
}
```

```

class="cpp1-reservedword">int</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">main(</span><span class="cpp1-reservedword">void</span><span
class="cpp1-brackets">) { </span><span class="cpp1-reservedword">unsigned</span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">char</span><span
class="cpp1-space"> </span><span class="cpp1-identifier">opakovani=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">;</span><span
class="cpp1-space"> </span><span class="cpp1-comment">//prom?nn kter se poka d po
vr en k u </span><span class="cpp1-space"> </span><span
class="cpp1-comment">//tla? ka zv o 1 </span><span
class="cpp1-identifier">TCR0=0B00000101;</span><span class="cpp1-space">
</span><span class="cpp1-comment">//zapnout TC0 s p?edd?li?kou 1024 </span><span
class="cpp1-identifier">setb</span><span class="cpp1-space"> </span><span
class="cpp1-brackets">(TIMSK,</span><span class="cpp1-number">0</span><span
class="cpp1-brackets">);</span><span class="cpp1-space"> </span><span
class="cpp1-comment">//povol p?eru en p?i p?ete?en TC0 (TCNT0==0x100)
</span><span class="cpp1-identifier">SREG</span><span class="cpp1-space">
</span><span class="cpp1-symbol">|=</span><span class="cpp1-space"> </span><span
class="cpp1-brackets">(</span><span class="cpp1-number">1</span><span
class="cpp1-symbol"><<<</span><span class="cpp1-number">7</span><span
class="cpp1-brackets">);</span><span class="cpp1-space"> </span><span
class="cpp1-comment">//povolen glob n o p?eru en </span><span
class="cpp1-identifier">TCNT0=TCNT0VAL;</span><span class="cpp1-space">
</span><span class="cpp1-comment">//resetTC </span><span
class="cpp1-identifier">lcd_ini();</span><span class="cpp1-space"> </span><span
class="cpp1-comment">//inicializace displeje (nastaven v stupn h pin? a
zah en komunikace) </span><span class="cpp1-identifier">kbd_ini();</span><span
class="cpp1-space"> </span><span class="cpp1-comment">//inicializace kl esnice
(nastaven vstupn h pin?) </span><span class="cpp1-identifier">cls();</span><span
class="cpp1-space"> </span><span class="cpp1-comment">//smaz d displeje
</span><span class="cpp1-identifier">gotoxy(</span><span
class="cpp1-number">1</span><span class="cpp1-symbol">,</span><span
class="cpp1-number">1</span><span class="cpp1-brackets">);</span><span
class="cpp1-space"> </span><span class="cpp1-comment">//kurzor na za? ek prvn o ? ku
</span><span class="cpp1-identifier">lcd_text(</span><span
class="cpp1-string">"Tlacitko:"</span><span class="cpp1-brackets">);</span><span
class="cpp1-space"> </span><span class="cpp1-comment">//nap me text </span><span
class="cpp1-identifier">gotoxy(</span><span class="cpp1-number">2</span><span
class="cpp1-symbol">,</span><span class="cpp1-number">1</span><span
class="cpp1-brackets">);</span><span class="cpp1-space"> </span><span
class="cpp1-comment">//kurzor na za? ek druh o ? ku </span><span
class="cpp1-identifier">lcd_text(</span><span class="cpp1-string">"Opakovani:"</span><span
class="cpp1-brackets">);</span><span class="cpp1-space"> </span><span
class="cpp1-comment">//nap me text </span><span
class="cpp1-reservedword">while</span><span class="cpp1-brackets">(</span><span
class="cpp1-number">1</span><span class="cpp1-brackets">)</span><span
class="cpp1-space"> </span><span class="cpp1-comment">//hlavn smy?ka </span><span

```

```
class="cpp1-space"> </span><span class="cpp1-brackets">{ </span><span  
class="cpp1-space"> </span><span class="cpp1-reservedword">if</span><span  
class="cpp1-brackets">(tlacitko)</span><span class="cpp1-space"> </span><span  
class="cpp1-comment"> //pokud je stisknut n?jak tla?ko ("tlacitko" nen nula)  
</span><span class="cpp1-space"> </span><span class="cpp1-brackets">{ </span><span  
class="cpp1-space"> </span><span class="cpp1-identifier">gotoxy(</span><span  
class="cpp1-number">1</span><span class="cpp1-symbol">,</span><span  
class="cpp1-number">13</span><span class="cpp1-brackets">);</span><span  
class="cpp1-space"> </span><span class="cpp1-comment"> //nastaven kurzoru za text  
(abychom jej nep?epsali) </span><span class="cpp1-space"> </span><span  
class="cpp1-identifier">lcd_byt(tlacitko);</span><span class="cpp1-space"> </span><span  
class="cpp1-comment"> //vypi k tla?ka </span><span class="cpp1-space">  
</span><span class="cpp1-identifier">tlacitko=</span><span  
class="cpp1-number">0</span><span class="cpp1-symbol">;</span><span><span  
class="cpp1-space"> </span><span class="cpp1-comment"> //hodnotu tla?ka u  
nepot?ebujeme </span><span class="cpp1-space"> </span><span  
class="cpp1-identifier">gotoxy(</span><span class="cpp1-number">2</span><span><span  
class="cpp1-symbol">,</span><span class="cpp1-number">13</span><span><span  
class="cpp1-brackets">);</span><span class="cpp1-space"> </span><span  
class="cpp1-comment"> //nastaven kurzoru za text </span><span class="cpp1-space">  
</span><span class="cpp1-identifier">lcd_byt(opakovani);</span><span class="cpp1-space">  
</span><span class="cpp1-comment"> //vypi ?lo opakov<< </span><span  
class="cpp1-space"> </span><span class="cpp1-identifier">opakovani++; </span><span  
class="cpp1-space"> </span><span class="cpp1-brackets">} </span><span  
class="cpp1-space"> </span><span class="cpp1-brackets">} } </span><span  
class="cpp1-comment"> //konec programu //Pro radioklub OK1KVK napsal Vašek Kr  
</span></span>
```