

```

#include "io430.h"
#include "stdint.h"
#include "intrinsics.h"

/* LCD port és vezérlő bitek megadása */
#define LCD_PORT      P2OUT
#define LCD_PORT_DIR  P2DIR
#define LCD_MASK      BIT2+BIT3+BIT4+BIT5
#define LCD_RS        P2OUT_bit.P1
#define LCD_RS_DIR    P2DIR_bit.P1
#define LCD_E         P2OUT_bit.P0
#define LCD_E_DIR     P2DIR_bit.P0

/**-----
 * Inline függvény, amely egy regiszter valamelyik
 * bitcsoportját atomi művelettel módosítja
 *-----
 * param reg a módosítandó regiszter neve
 * param val a beírandó érték (helyiérték-helyesen!)
 * param mask a módosítandó bitcsoportot kijelölő maszk
 */
#define ChangeBits(reg,val,mask) reg^=((reg^val)&mask)

/**-----
 * késleltető eljárás (1 - 65535 ms)
 *-----
 * param delay a késleltetés ms egységben megadva
 */
void delay_ms(uint16_t delay) {
    uint16_t i;
    for(i=0; i<delay; i++) {
        __delay_cycles(1000);
    }
}

/**-----
 * LCD Enable bemenet pulzálása
 *-----
 */
void lcd_toggle_E() {
    LCD_E = 1;
    __delay_cycles(500);
    LCD_E = 0;
    __delay_cycles(500);
}

/**-----
 * Egy bájt kiküldése az LCD vezérlőjének
 *-----
 * param val a kiírandó érték
 * param cmd regiszterválasztó bit (0: parancs, 1: adat)
 */
void lcd_write(uint8_t val, uint8_t cmd) {
    LCD_RS = cmd;
    ChangeBits(LCD_PORT,((val&0xF0)>>2),LCD_MASK);
    lcd_toggle_E();
    ChangeBits(LCD_PORT,((val&0x0F)<<2),LCD_MASK);
    lcd_toggle_E();
    __delay_cycles(1000);
}

/**-----
 * LCD kijelző inicializálása
 *-----
 */
void lcd_init (void){

```

Névtelen

```

/-- Az LCD port inicializálása
ChangeBits(LCD_PORT,0x00,LCD_MASK);
ChangeBits(LCD_PORT_DIR,LCD_MASK,LCD_MASK);
P2SEL = 0;
LCD_RS = 0;
LCD_RS_DIR = 1;
LCD_E = 0;
LCD_E_DIR = 1;
delay_ms(100); //40ms várakozás bekapcsolás után
/-- Az LCD vezérlőjének inicializálása:
/-- 1. szoftveres LCD reset: 0x30 (8-bites mód) (0x30 => 0x09 P2.2 és P2.3)
kiírása háromszor
/-- 2. 4-bites üzemmód beállítása
/-- 3. Képernyőtörlés, kurzor kikapcsolása
ChangeBits(LCD_PORT,0x09,LCD_MASK);
lcd_toggle_E ();
delay_ms(5); //várjunk legalább 4.1ms-ot
lcd_toggle_E ();
__delay_cycles(100); //várjunk legalább 100us-ot
__delay_cycles(40); //várjunk legalább 37us-ot
ChangeBits(LCD_PORT,0x08,LCD_MASK); //0x20 = 4-bites üzemmód (0x20 =>
0x08 P2.2 és P2.3)
lcd_toggle_E ();
delay_ms(5); //várjunk legalább 4.1ms-ot
/-- Innen kezdve minden bájt két félbájtként kell kiírni! -----
lcd_write(0x28,0); // display mód beállítás
delay_ms(5); //várjunk legalább 4.1ms-ot
lcd_write(0x08,0); // display letiltás
lcd_write(0x01,0); // képernyőtörlés
lcd_write(0x0C,0); // display be, cursor, és villogás
ki
}

/**-----
* karakterfüzér kiírása az LCD-re
*-----
* param p_str karakterfüzér mutató (nullával lezárt stringre mutat)
*/
void lcd_puts(char* p_str) {
    char c;
    while ((c=*p_str)) {
        lcd_write(c,1);
        p_str++;
    }
}

int main(void) {
    WDTCTL = WDTPW + WDTHOLD; //watchdog letiltása
    P1DIR |= BIT0;
    lcd_init();
    lcd_puts("<== 2x16 LCD ==>");
    lcd_write(0xC0,0); //A második sor elejére lép
    lcd_puts("Now in 4bit mode");
    while(1) {
        P1OUT ^= BIT0; //LED1 villogtatása
        __delay_cycles(250000);
    }
}

```