Εργαστήριο Μικροϋπολογιστών
Θοδωρής Παπαρρηγόπουλος
el18040
Ομάδα 21

3ο Εργαστήριο

Άσκηση 1

```
#define F_CPU 8000000
#include <avr/io.h>
#include <util/delay.h>

/* Functions Declarations */
void SUCCESS();
void BLINK_FAIL();

unsigned char scan_row(int i);
unsigned char swap(unsigned char x);
void scan_keypad();
int scan_keypad_rising_edge();
unsigned char keypad_to_ascii();

/* Global Variables */
unsigned char mem[2], key_reg[2];
unsigned char first, second; // first: First key and second: Second Key;

int main(void)
{
    DDRB = 0xFF;        // PORTB => OUTPUT
    DDRC = 0xF0;        // KEYPAD: PORTC[7:4] => OUTPUT, PORTC[3:0] => INPUT

    while (1) {
          mem[0] = 0;    // INITIALIZE RAM
          mem[1] = 0;
          PORTB = 0;

          while (!scan_keypad_rising_edge()) {}
          first = keypad_to_ascii();

          // GET SECOND DIGIT
          while(!scan_keypad_rising_edge()){}
          second = keypad_to_ascii();
          if (first == '2' && second == '1') {
            SUCCESS();
          } else {
            BLINK_FAIL();
          }
    }
}

void SUCCESS() {
      PORTB = 0xFF;
      _delay_ms(4000);
      PORTB = 0x00;
}

void BLINK_FAIL() {
      for (int i = 0; i < 4; ++i) {
            PORTB = 0xff;
            _delay_ms(500);
            PORTB = 0x00;
            _delay_ms(500);
      }
}

unsigned char scan_row(int i) {   // i = 1,2,3,4
      unsigned char a = ( 1 << 3 ); // SKIP 3 LSB
      a = (a << i);                  // SELECT ROW ACCORDING TO FUNCTION INPUT i
      PORTC = a;                     // WE SELECT ROW BY SETTING CORRESPONDING BIT TO 1
      _delay_us(500);                // DELAY FOR REMOTE USAGE
      return PINC & 0x0F;            // WE READ THE 4 LSB, '1' INDICATES SWITCH PUSHED
}

/* FUNCTION TO SWAP LO WITH HO BITS */
unsigned char swap(unsigned char x) {
      return ((x & 0x0F) << 4 | (x & 0xF0) >> 4);
}

/* SCAN ROWS(1..4) *DIFFERENT ORDER FROM EXERSISE DOCUMENT*
* FIRST ROW: PC4->PC0: 1,  PC4->PC1: 2,  PC4->PC2: 3, PC4->PC3: A
* SECOND ROW: PC5->PC0: 4,  PC5->PC1: 5,  PC5->PC2: 6, PC5->PC3: B
* THIRD ROW: PC6->PC0: 7,  PC6->PC1: 8,  PC6->PC2: 9, PC6->PC3: C
* FOURTH ROW: PC7->PC0: *,  PC7->PC1: 0,  PC7->PC2: #, PC7->PC3: D
*/
void scan_keypad() {
      unsigned char i;

      // check row 1, 0b0001-ROW CORRESPONDING TO PC4
      i = scan_row(1);
      key_reg[1] = swap(i);    //key_reg[1] = first_row(4 MSB)-0000

      // check row 2, 0b0010-ROW CORRESPONDING TO PC5
      i = scan_row(2);
      key_reg[1] += i;         //key_reg[1] = first_row(4 MSB)-second_row(4 LSB)

      // check row 3, 0b0100-ROW CORRESPONDING TO PC6
      i = scan_row(3);
      key_reg[0] = swap(i);    //key_reg[0] = third_row(4 MSB) -0000

      // check row 4, 0b1000-ROW CORRESPONDING TO PC7
      i = scan_row(4);
      key_reg[0] += i;         //key_reg[0] = third_row(4 MSB)-fourth_row(4 LSB)
      PORTC = 0x00;                  // added for remote usage
}

int scan_keypad_rising_edge() {
      // CHECK KEYPAD
      scan_keypad();                                    // RETURNS RESULTS IN key_reg
      // ADD TEMPORARY VARIABLES
      unsigned char tmp_keypad[2];
      tmp_keypad[0] = key_reg[0];                 //tmp_keypad HOLD ACQUIRED DATA FROM
SCAN_KEYPAD()
      tmp_keypad[1] = key_reg[1];

      _delay_ms(0x15);                            // APOFYGH SPINTHIRISMOU


      scan_keypad();
      key_reg[0] &= tmp_keypad[0];         // APPORIPSE TIS TIMES POU EMFANISAN
SPINTHIRISMO
      key_reg[1] &= tmp_keypad[1];

      tmp_keypad[0] = mem[0];                     // BRING LAST STATE OF SWITCHES FROM
RAM TO tmp_keypad
      tmp_keypad[1] = mem[1];

      mem[0] = key_reg[0];                        // STORE NEW KEYPAD STATE IN RAM FOR
FUTURE CALL
      mem[1] = key_reg[1];


      key_reg[0] &= ~tmp_keypad[0];        // FIND KEYPAD SWITCHES THAT HAVE JUST BEEN
PRESSED
      key_reg[1] &= ~tmp_keypad[1];

      return (key_reg[0] || key_reg[1]); // 16 BIT VALUE INDICATING FRESHLY PRESSED
SWITCHES - RETURNS 0 IF NO SWITCH PRESSED
}

/* CONVERT VALUE TO ASCII CODE *CHECK COMMENT ABOVE SCAN_KEYPAD FOR CORRESPONDENCE
* key_reg[0] = third_row(4 MSB)-fourth_row(4 LSB)
* key_reg[1] = first_row(4 MSB)-second_row(4 LSB)
* LSB -> MSB == LEFT -> RIGHT IN KEYPAD */
unsigned char keypad_to_ascii() {
      if (key_reg[0] & 0x01)
      return '*';

      if (key_reg[0] & 0x02)
      return '0';

      if (key_reg[0] & 0x04)
      return '#';

      if (key_reg[0] & 0x08)
      return 'D';

      if (key_reg[0] & 0x10)
      return '7';

      if (key_reg[0] & 0x20)
      return '8';

      if (key_reg[0] & 0x40)
      return '9';

      if (key_reg[0] & 0x80)
      return 'C';

      if (key_reg[1] & 0x01)
      return '4';

      if (key_reg[1] & 0x02)
      return '5';

      if (key_reg[1] & 0x04)
      return '6';

      if (key_reg[1] & 0x08)
      return 'B';
      if (key_reg[1] & 0x10)
      return '1';

      if (key_reg[1] & 0x20)
      return '2';

      if (key_reg[1] & 0x40)
      return '3';

      if (key_reg[1] & 0x80)
      return 'A';

      // Nothing Found
      return 0;
}
```

Άσκηση 2

```
.DSEG
_tmp_: .byte 2

.CSEG
.include "m16def.inc"

.def temp=r20
.def cnt=r21
.macro SET_LEDS_ON
; MACRO: SET ALL LEDS OF PORTA TO ON
; AFFECTED REGISTER:
ser r18
out PORTB,r18
.endm


.macro SET_LEDS_OFF
; MACRO: SET ALL LEDS OF PORTA TO ON
; AFFECTED REGISTER: r20
clr r20
out PORTB,r20
.endm

.org 0x00
rjmp init

init:
        clr temp
        ; initialization  stack pointer
        ldi r24, low(RAMEND)
        out SPL, r24

        ldi r24, high(RAMEND)
        out SPH, r24

        ser r24 ; r24 = FF
        out DDRB, r24 ; initialize  port b
        out DDRD, r24 ; and d for output

        ldi r24, (1 << PC7) | (1 << PC6) | (1 << PC5) | (1 << PC4)
        out DDRC, r24 ;


first_digit:
        ldi r24,0xf0 ; pernaw asso se ola ta pliktra
        rcall scan_keypad_rising_edge_sim ; elegxw tis eksodous
        clr r22 ; arxikopoiw sto 0
        or r22, r24 ; ta grafw ola ekei gia na dv an exw allages
        or r22, r25 ;
        cpi r22,0 ; an einai 0 shmainei den exw allages kai aksana elegxw
        breq first_digit
        cpi r25, 32 ; elegxw gia to 2
        brne wrong_first
        cpi r24,0
        brne wrong_first
        rjmp second_digit

wrong_first:
        ldi r21,1       ; flag that indicates first digit was incorrect

        second_digit:
        ldi r24,0xf0 ; pernaw asso se ola ta pliktra
        rcall scan_keypad_rising_edge_sim ; elegxw tis eksodous
        clr r22 ; arxikopoiw sto 0
        or r22, r24 ; ta grafw ola ekei gia na dv an exw allages
        or r22, r25
        cpi r22,0
        breq second_digit
        ;r24 = 0 r25 = 0b1000000
        cpi r21,1
        breq wrong_passwd
        cpi r24,0
        brne wrong_passwd
        cpi r25,16
        brne wrong_passwd
        ; an ftasw mexri edw tote exw swsto kwdiko ara

right_password: ; kanei ta 4-sec flashes kai grafei sthn othoni
        rcall scan_keypad_rising_edge_sim ; extra call for remote usage
        SET_LEDS_ON  ; ALL LEDS ON (MACRO)
        ; print  WELCOME 21
        rcall lcd_init_sim
        ldi r24,'W'
        rcall lcd_data_sim
        ldi r24,'E'
        rcall lcd_data_sim
        ldi r24,'L'
        rcall lcd_data_sim
        ldi r24,'C'
        rcall lcd_data_sim
        ldi r24,'O'
        rcall lcd_data_sim
        ldi r24,'M'
        rcall lcd_data_sim
        ldi r24,'E'
        rcall lcd_data_sim
        ldi r24,' '
        rcall lcd_data_sim
        ldi r24,'2'
        rcall lcd_data_sim
        ldi r24,'1'
        rcall lcd_data_sim
        ldi r24,low(4000)
        ldi r25,high(4000)
        rcall wait_msec                    ; DELAY 4 SECONDS (MACRO)
        SET_LEDS_OFF

        rjmp first_digit

wrong_passwd:
        rcall scan_keypad_rising_edge_sim ; extra call for remote usage
        ; print "ALARM ON"
        rcall lcd_init_sim
        ldi r24,'A'
        rcall lcd_data_sim ; ???????? ???? byte ????????? ???? ??????? ??? ?????? lcd
        ldi r24,'L'
        rcall lcd_data_sim
        ldi r24,'A'
        rcall lcd_data_sim
        ldi r24,'R'
        rcall lcd_data_sim
        ldi r24,'M'
        rcall lcd_data_sim
        ldi r24,' '
        rcall lcd_data_sim
        ldi r24,'O'
        rcall lcd_data_sim
        ldi r24,'N'
        rcall lcd_data_sim
        ldi cnt,0x04         ; iterate 4 times
        L1:
        SET_LEDS_ON                 ; set leds on (MACRO)
        ldi r24,low(500)
        ldi r25,high(500)
        rcall wait_msec                    ; delay 0.5sec (MACRO)

        SET_LEDS_OFF                ; set leds off (MACRO)
        ldi r24,low(500)
        ldi r25,high(500)
        rcall wait_msec

        dec cnt                     ; cnt--
        cpi cnt, 0x0         ; (cnt == 0) ?
        brne L1                     ; if cnt != 0 goto L1
        rjmp first_digit




scan_row_sim:
        out PORTC, r25 ; ? ?????????? ?????? ??????? ??? ??????    1
        push r24 ; ????? ?????? ??? ??????????? ??? ?? ?????
        push r25 ; ?????????? ??? ???????????? ??????????????
        ldi r24,low(500) ; ?????????
        ldi r25,high(500)
        rcall wait_usec
        pop r25
        pop r24 ; ????? ????? ??????
        nop
        nop ; ??????????? ??? ?? ???????? ?? ????? ? ?????? ??????????
        in r24, PINC ; ??????????? ?? ?????? (??????) ??? ????????? ??? ????? ?????????
        andi r24 ,0x0f ; ????????????? ?? 4 LSB ???? ??    1    ???????? ??? ????? ?????????
        ret ; ?? ?????????

scan_keypad_sim:
        push r26 ; ?????????? ???? ??????????? r27:r26 ????? ????
        push r27 ; ????????? ???? ???? ???????
        ldi r25 , 0x10 ; ?????? ??? ????? ?????? ??? ????????????? (PC4: 1 2 3 A)
        rcall scan_row_sim
        swap r24 ; ?????????? ?? ??????????
        mov r27, r24 ; ??? 4 msb ??? r27
        ldi r25 ,0x20 ; ?????? ?? ??????? ?????? ??? ????????????? (PC5: 4 5 6 B)
        rcall scan_row_sim
        add r27, r24 ; ?????????? ?? ?????????? ??? 4 lsb ??? r27
        ldi r25 , 0x40 ; ?????? ??? ????? ?????? ??? ????????????? (PC6: 7 8 9 C)
        rcall scan_row_sim
        swap r24 ; ?????????? ?? ??????????
        mov r26, r24 ; ??? 4 msb ??? r26
        ldi r25 ,0x80 ; ?????? ??? ??????? ?????? ??? ????????????? (PC7: * 0 # D)
        rcall scan_row_sim
        add r26, r24 ; ?????????? ?? ?????????? ??? 4 lsb ??? r26
        movw r24, r26 ; ???????? ?? ?????????? ????? ??????????? r25:r24
        clr r26 ; ?????????? ??? ??? ????????????? ????????
        out PORTC,r26 ; ?????????? ??? ??? ????????????? ????????
        pop r27 ; ????????? ???? ??????????? r27:r26
        pop r26
        ret

scan_keypad_rising_edge_sim:
        push r22 ; ?????????? ???? ??????????? r23:r22 ??? ????
        push r23 ; r26:r27 ????? ???? ????????? ???? ???? ???????
        push r26
        push r27
        rcall scan_keypad_sim ; ?????? ?? ???????????? ??? ?????????? ?????????
        push r24 ; ??? ?????????? ?? ??????????
        push r25
        ldi r24 ,15 ; ??????????? 15 ms (??????? ????? 10-20 msec ??? ??????????? ??? ???
        ldi r25 ,0 ; ???????????? ??? ?????????????     ????????????? ????????????)
        rcall wait_msec
        rcall scan_keypad_sim ; ?????? ?? ???????????? ???? ??? ????????
        pop r23 ; ??? ??????? ?????????? ???????????
        pop r22
        and r24 ,r22
        and r25 ,r23
        ldi r26 ,low(_tmp_) ; ??????? ??? ????????? ??? ????????? ????
        ldi r27 ,high(_tmp_) ; ??????????? ????? ??? ???????? ????? r27:r26
        ld r23 ,X+
        ld r22 ,X
        st X ,r24 ; ?????????? ??? RAM ?? ??? ?????????
        st -X ,r25 ; ??? ?????????
        com r23
        com r22 ; ???? ???? ????????? ??? ????? «?????» ???????
        and r24 ,r22
        and r25 ,r23
        pop r27 ; ????????? ???? ??????????? r27:r26
        pop r26 ; ??? r23:r22
        pop r23
        pop r22
        ret

keypad_to_ascii_sim:
        push r26 ; ?????????? ???? ??????????? r27:r26 ????? ????
        push r27 ; ????????? ???? ??? ???????
        movw r26 ,r24 ; ??????    1    ???? ?????? ??? ?????????? r26 ????????
        ; ?? ???????? ??????? ??? ????????
        ldi r24 ,'*'
        ; r26
        ;C 9 8 7 D # 0 *
        sbrc r26 ,0
        rjmp return_ascii
        ldi r24 ,'0'
        sbrc r26 ,1
        rjmp return_ascii
        ldi r24 ,'#'
        sbrc r26 ,2
        rjmp return_ascii
        ldi r24 ,'D'
        sbrc r26 ,3 ; ?? ??? ?????    1    ??????????? ??? ret, ?????? (?? ?????    1    )
        rjmp return_ascii ; ?????????? ?? ??? ?????????? r24 ??? ASCII ???? ??? D.
        ldi r24 ,'7'
        sbrc r26 ,4
        rjmp return_ascii
        ldi r24 ,'8'
        sbrc r26 ,5
        rjmp return_ascii
        ldi r24 ,'9'
        sbrc r26 ,6
        rjmp return_ascii ;
        ldi r24 ,'C'
        sbrc r26 ,7
        rjmp return_ascii
        ldi r24 ,'4' ; ??????    1    ???? ?????? ??? ?????????? r27 ????????
        sbrc r27 ,0 ; ?? ???????? ??????? ??? ????????
        rjmp return_ascii
        ldi r24 ,'5'
        ;r27
        ;? 3 2 1 B 6 5 4
        sbrc r27 ,1
        rjmp return_ascii
        ldi r24 ,'6'
        sbrc r27 ,2
        rjmp return_ascii
        ldi r24 ,'B'
        sbrc r27 ,3
        rjmp return_ascii
        ldi r24 ,'1'
        sbrc r27 ,4
        rjmp return_ascii ;
        ldi r24 ,'2'
        sbrc r27 ,5
        rjmp return_ascii
        ldi r24 ,'3'
        sbrc r27 ,6
        rjmp return_ascii
        ldi r24 ,'A'
        sbrc r27 ,7
        rjmp return_ascii
        clr r24
        rjmp return_ascii
        return_ascii:
        pop r27 ; ????????? ???? ??????????? r27:r26
        pop r26
        ret

write_2_nibbles_sim:
        push r24 ; ????? ?????? ??? ??????????? ??? ?? ?????
        push r25 ; ?????????? ??? ???????????? ??????????????
        ldi r24 ,low(6000) ; ?????????
        ldi r25 ,high(6000)
        rcall wait_usec
        pop r25
        pop r24 ; ????? ????? ??????
        push r24 ; ??????? ?? 4 MSB
        in r25, PIND ; ??????????? ?? 4 LSB ??? ?? ?????????????
        andi r25, 0x0f ; ??? ?? ??? ????????? ??? ????? ??????????? ?????????
        andi r24, 0xf0 ; ????????????? ?? 4 MSB ???
        add r24, r25 ; ???????????? ?? ?? ???????????? 4 LSB
        out PORTD, r24 ; ??? ???????? ???? ?????
        sbi PORTD, PD3 ; ????????????? ?????? Enable ???? ????????? PD3
        cbi PORTD, PD3 ; PD3=1 ??? ???? PD3=0
        push r24 ; ????? ?????? ??? ??????????? ??? ?? ?????
        push r25 ; ?????????? ??? ???????????? ??????????????
        ldi r24 ,low(6000) ; ?????????
        ldi r25 ,high(6000)
        rcall wait_usec
        pop r25
        pop r24 ; ????? ????? ??????
        pop r24 ; ??????? ?? 4 LSB. ????????? ?? byte.
        swap r24 ; ????????????? ?? 4 MSB ?? ?? 4 LSB
        andi r24 ,0xf0 ; ??? ?? ??? ????? ???? ?????????????
        add r24, r25
        out PORTD, r24
        sbi PORTD, PD3 ; ???? ?????? Enable
        cbi PORTD, PD3
        ret

lcd_data_sim:
        push r24
        push r25
        sbi PORTD,PD2
        rcall write_2_nibbles_sim
        ldi r24,43
        ldi r25,0
        rcall wait_usec
        pop r25
        pop r24
        ret

lcd_command_sim:
        push r24 ; ?????????? ???? ??????????? r25:r24 ????? ????
        push r25 ; ????????? ???? ??? ???????
        cbi PORTD, PD2 ; ??????? ??? ?????????? ??????? (PD2=0)
        rcall write_2_nibbles_sim ; ???????? ??? ??????? ??? ??????? 39?sec
        ldi r24, 39 ; ??? ??? ?????????? ??? ????????? ??? ??? ??? ??????? ??? lcd.
        ldi r25, 0 ; ???.: ???????? ??? ???????, ?? clear display ??? return home,
        rcall wait_usec ; ??? ???????? ????????? ?????????? ??????? ????????.
        pop r25 ; ????????? ???? ??????????? r25:r24
        pop r24
        ret

lcd_init_sim:
        push r24 ; ?????????? ???? ??????????? r25:r24 ????? ????
        push r25 ; ????????? ???? ??? ???????

        ldi r24, 40 ; ???? ? ???????? ??? lcd ????????????? ??
        ldi r25, 0 ; ????? ??????? ??? ???? ??? ????????????.
        rcall wait_msec ; ??????? 40 msec ????? ???? ?? ???????????.
        ldi r24, 0x30 ; ?????? ????????? ?? 8 bit mode
        out PORTD, r24 ; ?????? ??? ???????? ?? ??????? ???????
        sbi PORTD, PD3 ; ??? ?? ?????????? ??????? ??? ???????
        cbi PORTD, PD3 ; ??? ??????, ? ?????? ???????????? ??? ?????
        ldi r24, 39
        ldi r25, 0 ; ??? ? ???????? ??? ?????? ????????? ?? 8-bit mode
        rcall wait_usec ; ??? ?? ?????? ??????, ???? ?? ? ???????? ???? ??????????
        ; ??????? 4 bit ?? ??????? ?? ?????????? 8 bit
        push r24 ; ????? ?????? ??? ??????????? ??? ?? ?????
        push r25 ; ?????????? ??? ???????????? ??????????????
        ldi r24,low(1000) ; ?????????
        ldi r25,high(1000)
        rcall wait_usec
        pop r25
        pop r24 ; ????? ????? ??????
        ldi r24, 0x30
        out PORTD, r24
        sbi PORTD, PD3
        cbi PORTD, PD3
        ldi r24,39
        ldi r25,0
        rcall wait_usec
        push r24 ; ????? ?????? ??? ??????????? ??? ?? ?????
        push r25 ; ?????????? ??? ???????????? ??????????????
        ldi r24 ,low(1000) ; ?????????
        ldi r25 ,high(1000)
        rcall wait_usec
        pop r25
        pop r24 ; ????? ????? ??????
        ldi r24,0x20 ; ?????? ?? 4-bit mode
        out PORTD, r24
        sbi PORTD, PD3
        cbi PORTD, PD3
        ldi r24,39
        ldi r25,0
        rcall wait_usec
        push r24 ; ????? ?????? ??? ??????????? ??? ?? ?????
        push r25 ; ?????????? ??? ???????????? ??????????????
        ldi r24 ,low(1000) ; ?????????
        ldi r25 ,high(1000)
        rcall wait_usec
        pop r25
        pop r24 ; ????? ????? ??????
        ldi r24,0x28 ; ??????? ?????????? ???????? 5x8 ????????
        rcall lcd_command_sim ; ??? ???????? ??? ??????? ???? ?????
        ldi r24,0x0c ; ???????????? ??? ??????, ???????? ??? ???????
        rcall lcd_command_sim
        ldi r24,0x01 ; ?????????? ??? ??????
        rcall lcd_command_sim
        ldi r24, low(1530)
        ldi r25, high(1530)
        rcall wait_usec
        ldi r24 ,0x06 ; ???????????? ????????? ??????? ???? 1 ??? ??????????
        rcall lcd_command_sim ; ??? ????? ???????????? ???? ??????? ??????????? ???
        ; ?????????????? ??? ????????? ????????? ??? ??????
        pop r25 ; ????????? ???? ??????????? r25:r24
        pop r24
        ret

wait_msec:
        push r24                                 ; 2 ?????? (0.250 ?sec)
        push r25                                 ; 2 ??????
        ldi r24 , low(998)               ; ??????? ??? ?????. r25:r24 ?? 998 (1 ?????? - 0.125 ?sec)
        ldi r25 , high(998)              ; 1 ?????? (0.125 ?sec)
        rcall wait_usec                          ; 3 ?????? (0.375 ?sec), ???????? ???????? ???????????
998.375 ?sec
        pop r25                                          ; 2 ?????? (0.250 ?sec)
        pop r24                                          ; 2 ??????
        sbiw r24 , 1                     ; 2 ??????
        brne wait_msec                           ; 1 ? 2 ?????? (0.125 ? 0.250 ?sec)
        ret                                              ; 4 ?????? (0.500 ?sec)

wait_usec:
        sbiw r24 ,1                      ; 2 ?????? (0.250 ?sec)
        nop                                      ; 1 ?????? (0.125 ?sec)
        nop                                      ; 1 ?????? (0.125 ?sec)
        nop                                      ; 1 ?????? (0.125 ?sec)
        nop                                      ; 1 ?????? (0.125 ?sec)
        brne wait_usec                   ; 1 ? 2 ?????? (0.125 ? 0.250 ?sec)
        ret                                      ; 4 ?????? (0.500 ?sec)
```