ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΜ&ΜΥ
Εργαστήριο Μικροϋπολογιστών

7η Εργαστηριακή Άσκηση
Ακ. έτος 2011-2012

Ομάδα C07:

Ελένη Ευαγγελάτου   Α.Μ.: 03108050
Γρηγόρης Λύρας   Α.Μ.: 03109687
Βασιλεία Φραγκιαδάκη   Α.Μ.: 03108026

16 Φεβρουαρίου 2012

Άσκηση (i)

Κυρίως κώδικας:

```
/*Includes for compatibility with GNU toolchain*/
#define __SFR_OFFSET 0
#include <avr/io.h>
#include <avr/interrupt.h>
.global main
/*
 * AVR_3h_ask1.asm
 *
 *  Created: 10/2/2012 10:01:52 ??
 *   Author: Valia
 */


/*
 * AVR_ask1.asm
 *
 *  Created: 10/2/2012 9:24:48 ??
 *   Author: Valia
 */

#define temp1   r24
#define input   r28
#define output  r27
#define temp2   r25
#define regC  r29
#define tempo  r26
#define output1  r30

main:
    ldi temp1, hi8(RAMEND)
    out SPH, temp1
    ldi temp1, lo8(RAMEND)
    out SPL, temp1

    ser temp1
    out DDRB, temp1

    clr temp1
    out DDRD, temp1

    clr regC
    out DDRC, regC
start:
    clr output1
    in regC, PINC

    lsr regC
    brcc syn1
etik1:      ;//perimenei mexri na ginei 0 (button)
    in tempo, PINC
    lsr tempo
    ;brcc syn1
    ori output1, 1

syn1:
    lsr regC
    brcc syn2
etik2:      ;//perimenei mexri na ginei 0 (button)
    in tempo, PINC
    lsr tempo
    lsr tempo
    ;brcc syn2
    ori output1, 2

syn2:
    lsr regC
    brcc syn3
etik3:      ;//perimenei mexri na ginei 0 (button)
    in tempo, PINC
    lsr tempo
    lsr tempo
    lsr tempo
```

1

```asm
     ;brcc syn3
     ori output1, 4

 syn3:
     lsr regC
     brcc syn4
etik4:      ;//perimenei mexri na ginei 0 (button)
     in tempo, PINC
     lsr tempo
     lsr tempo
     lsr tempo
     lsr tempo
     ;brcc syn4
     ori output1, 8

 syn4:
     lsr regC
     brcc syn5
etik5:      ;//perimenei mexri na ginei 0 (button)
     in tempo, PINC
     lsl tempo
     lsl tempo
     lsl tempo
     lsl tempo
     ;brcs etik5
     ori output1, 0x010

syn5:
     lsr regC
     brcc syn6
etik6:      ;//perimenei mexri na ginei 0 (button)
     in tempo, PINC
     lsl tempo
     lsl tempo
     lsl tempo
     ;brcs etik6
     ori output1, 0x020

 syn6:
     lsr regC
     brcc syn7
 etik7:      ;//perimenei mexri na ginei 0 (button)
     in tempo, PINC
     lsl tempo
     lsl tempo
     ;brcs etik7
     ori output1, 0x040

syn7:
     lsr regC
     brcc synexise
 etik8:      ;//perimenei mexri na ginei 0 (button)
     in tempo, PINC
     lsl tempo
     ;brcs etik8
     ori output1, 0x080

 synexise:

     in input, PIND
     clr temp2   ;//mhdenismos exodwn
     clr output

gate_1:
     lsr input


         brcc exit1_gate1  ;//an to lsb = 0 feugoume
     lsr input
     brcc exit_gate1  ;//an to 2o lsb = 0 feugoume
     inc temp2
         rjmp exit_gate1

exit1_gate1:
         lsr input;//dior8wsh
```

2

```asm
148    exit_gate1:
149
150    gate_2:
151        lsr input
152        brcc exit1_gate2 ;//an to 3o lsb = 0 feugoume
153        lsr input
154        brcc exit_gate2  ;//an to 4o lsb = 0 feugoume
155        ori output, 3
156            rjmp exit_gate2
157    exit1_gate2:
158            lsr input
159    exit_gate2:
160        or output, temp2    ;//exodos sta 2 prwta lsb bit
161
162    gate_3:
163        lsr input
164        brcs exit1_gate3 ;//an to 5o lsb = 1 feugoume
165        lsr input
166        brcs exit_gate3 ;//an to 6o lsb = 1 feugoume
167        ori output, 4  ;// or me 100
168            rjmp exit_gate3
169    exit1_gate3:
170            lsr input
171    exit_gate3:
172
173    gate_4:
174        lsr input
175        brcs  one    ;//an einai 1 psaxnoumt an kai to allo einai 1
176        lsr input    ;//an einai to 1o 0, tsekaroyme an kai to 2o einai 0
177        brcs exit_gate4
178        rjmp setting
179
180    one:
181        lsr input
182        brcc exit_gate4 ;// an to allo einai 0 feugoyme
183
184    setting:
185        ori output,8
186
187    exit_gate4:
188        eor output, output1  ;//xor
189        out PORTB, output
190        rjmp start
```

Άσκηση (ii)

Κυρίως κώδικας:

```c
1    /* -.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
2
3    * File Name : part2.c
4
5    * Purpose :
6
7    * Creation Date : 07-02-2012
8
9    * Last Modified : Wed 08 Feb 2012 12:09:50 AM EET
10
11   * Created By : Greg Liras <gregliras@gmail.com>
12
13   -.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.*/
14
15
16   #include<avr/io.h>
17
18   int f0 ( int c )
19   {
20       for( ; c > 0 ; c >>=1 )
21           if ( ( c & 3 ) == 3 )
22               return 0;
23       return 1;
24   }
25   int f1 ( int c )
26   {
```
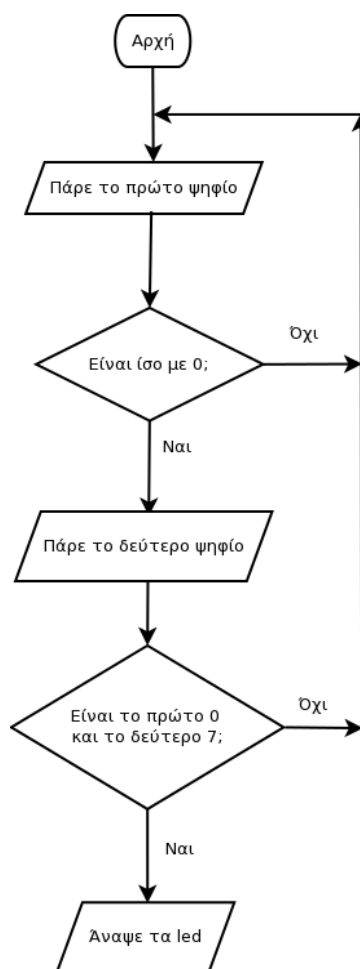
```
27      if (  c <= 15 || c == 31 )
28          return 1;
29      return 0;
30  }
31  int main(void)
32  {
33      DDRA = 0xff;
34      DDRC = 0x00;
35      int f_0;
36      int f_1;
37      int c;
38      for( ;; )
39      {
40          c = PINC & 31;
41          f_0 = f0( c );
42          f_1 = f1( c );
43          PORTA = f_0 | ( f_1 << 1) | ( ( f_0 | f_1 ) << 2);
44      }
45  }
```

Άσκηση (iii)

Σχήμα 1: Διάγραμμα ροής

Κυρίως κώδικας:

```
1   #define __SFR_OFFSET 0
2   #include <avr/io.h>
3
4   .data
5       _tmp_: .byte 2
6   .text
```

```asm
    .global main
main:
reset:
    ldi r24,lo8(RAMEND)
    out SPL,r24
    ldi r24,hi8(RAMEND)
    out SPH,r24
    ser r24
    out DDRA,r24
    ;// PORTA output
    ldi r24 ,(1 << PC7) | (1 << PC6) | (1 << PC5) | (1 << PC4)
    out DDRC ,r24
    ;// 4x4 pad input
    rcall scan_keypad_rising_edge
    ;// initialize
lock:
    ;//reset the memory
    ldi r24,0x00
    ldi r25,0x00
    ldi r26 ,lo8(_tmp_)
    ldi r27 ,hi8(_tmp_)
    st X+ ,r24
    st X ,r25
first_key:
    ldi r24, 0x14
    rcall scan_keypad_rising_edge
    cpi r24,0x00
    breq first_key
    cpi r24, 0x02
    ;//first test failed
    brne lock

    ldi r24,0x00
    ldi r25,0x00
    ldi r26 ,lo8(_tmp_)
    ldi r27 ,hi8(_tmp_)
    st X+ ,r24
    st X ,r25

second_key:
    ldi r24, 0x14
    rcall scan_keypad_rising_edge
    ;//second test failed
    cpi r24,0x00
    breq second_key
    cpi r24, 0x10
    brne lock
    rjmp pass

pass:
    ldi r24,0xff
    out PORTA, r24
    ldi r24,lo8(2000)
    ldi r25,hi8(2000)
    rcall wait_msec
    ;//reset the leds
    ldi r24,0x00
    out PORTA, r24
    rjmp lock


scan_row:
    ldi r25 ,0x08
back_: lsl r25
    dec r24
    brne back_
    out PORTC ,r25
    nop
    nop
    in r24 ,PINC
    andi r24 ,0x0f
    ret
```

```
82  scan_keypad:
83      ldi r24 ,0x01
84      rcall scan_row
85      swap r24
86      mov r27 ,r24
87      ldi r24 ,0x02
88      rcall scan_row
89      add r27 ,r24
90      ldi r24 ,0x03
91      rcall scan_row
92      swap r24
93      mov r26 ,r24
94      ldi r24 ,0x04
95      rcall scan_row
96      add r26 ,r24
97      movw r24 ,r26
98      ret
99
100
101 scan_keypad_rising_edge:
102     mov r22 ,r24
103     rcall scan_keypad
104     push r24
105     push r25
106     mov r24 ,r22
107     ldi r25 ,0
108     rcall wait_msec
109     rcall scan_keypad
110     pop r23
111     pop r22
112     and r24 ,r22
113     and r25 ,r23
114     ldi r26 ,lo8(_tmp_)
115     ldi r27 ,hi8(_tmp_)
116     ld r23 ,X+
117     ld r22 ,X
118     st X ,r24
119     st -X ,r25
120     com r23
121     com r22
122     and r24 ,r22
123     and r25 ,r23
124     ret
125
126
127
128 wait_usec:
129     sbiw r24,1
130     nop
131     brne wait_usec
132     ret
133
134 wait_msec:
135     in r27,PINB
136     ror r27
137     brcs wait_msec
138     push r24
139     push r25
140     ldi r24,lo8(998)
141     ldi r25,hi8(998)
142     rcall wait_usec
143     pop r25
144     pop r24
145     sbiw r24,1
146     brne wait_msec
147     ret
```

Άσκηση (iv)

Κυρίως κώδικας:

```
1  /*Includes for compatibility with GNU toolchain*/
2  #define __SFR_OFFSET 0
3  #include <avr/io.h>
```

```
4  #include <avr/interrupt.h>
5  .global main
6  /*
7   * AVRask4.asm
8   *
9   *  Created: 12/2/2012 10:31:36 ??
10  *   Author: Valia
11  */
12  .data
13  _tmp_ :.byte 2
14
15  .text
16  #define temp1  r24
17  #define temp2  r25
18
19  main:
20
21      ser r24
22      out DDRD,r24
23      ldi r24 ,(1 << PC7) | (1 << PC6) | (1 << PC5) | (1 << PC4)
24      out DDRC ,r24
25
26      rcall lcd_init ;//initialize
27
28
29
30      ldi r24,'T'  ;//'T'
31      rcall lcd_data
32      ldi r24,'E' ;//'E'
33      rcall lcd_data
34      ldi r24,'A' ;//'E'
35      rcall lcd_data
36      ldi r24,'M' ;//'E'
37      rcall lcd_data
38      ldi r24,' ' ;//'E'
39      rcall lcd_data
40      ldi r24,'0' ;//'E'
41      rcall lcd_data
42      ldi r24,'7' ;//'E'
43      rcall lcd_data
44
45  start:
46      ldi r24,0x14
47      rcall scan_keypad_rising_edge ;//apotelesma sto r24:r25
48      rcall keypad_to_ascii
49      cpi r24,0x0
50      breq start
51
52      push r24
53      ldi r24,0x01    ;//freskarisma
54      rcall lcd_command
55      ldi r24,lo8(1530)
56      ldi r25,hi8(1530)
57      rcall wait_usec
58      pop r24
59
60      rcall lcd_data  ;//emfanish sthn o8onh
61
62      ldi r26,lo8(_tmp_)  ;//ka8arisma mnhmhs
63      ldi r27,hi8(_tmp_)
64      ldi r24,0x0
65      st  X+,r24
66      st  X,r24
67
68      rjmp start
69
70  keypad_to_ascii:
71      movw r26,r24
72      ldi r24,'*'
73      sbrc r26,0
74      ret
75      ldi r24,'0'
76      sbrc r26,1
77      ret
78      ldi r24,'#'
```

```asm
        sbrc r26,2
        ret
        ldi r24,'D'
        sbrc r26,3
        ret
        ldi r24,'7'
        sbrc r26,4
        ret
        ldi r24,'8'
        sbrc r26,5
        ret
        ldi r24,'9'
        sbrc r26,6
        ret
        ldi r24,'C'
        sbrc r26,7
        ret
        ldi r24,'4'
        sbrc r27,0
        ret
        ldi r24,'5'
        sbrc r27,1
        ret
        ldi r24,'6'
        sbrc r27,2
        ret
        ldi r24,'B'
        sbrc r27,3
        ret
        ldi r24,'1'
        sbrc r27,4
        ret
        ldi r24,'2'
        sbrc r27,5
        ret
        ldi r24,'3'
        sbrc r27,6
        ret
        ldi r24,'A'
        sbrc r27,7
        ret
        clr r24
        ret

scan_keypad_rising_edge:
        mov r22,r24
        rcall scan_keypad

        push r24 ;//apo8ikeush apotelesmatos
        push r25
        mov r24,r22 ;//ka8ysterhsh r22msec (10-20  msec)
        ldi r25,0
        rcall wait_msec

        rcall scan_keypad ;//aporrispe osa plhktra emfanizoyn spin8hrismo
        pop r23
        pop r22
        and r24,r22
        and r25,r23
        ldi r26,lo8(_tmp_)
        ldi r27,hi8(_tmp_)
        ld r23,X+
        ld r22,X
        st X,r24
        st -X,r25
        com r23
        com r22
        and r24,r22
        and r25,r22
        ret

lcd_init:
        ldi r24,40
        ldi r25,0
        rcall wait_msec
```

```
154
155     ldi r24,0x30
156     out PORTD,r24
157     sbi PORTD,PD3
158     cbi PORTD,PD3
159     ldi r24,39
160     ldi r25,0
161     rcall wait_usec
162
163     ldi r24,0x30
164     out PORTD,r24
165     sbi PORTD,PD3
166     cbi PORTD,PD3
167     ldi r24,39
168     ldi r25,0
169     rcall wait_usec
170
171     ldi r24,0x20
172     out PORTD,r24
173     sbi PORTD,PD3
174     cbi PORTD,PD3
175     ldi r24,39
176     ldi r25,0
177     rcall wait_usec
178
179     ldi r24,0x28
180     rcall lcd_command
181
182     ldi r24,0x0c
183     rcall lcd_command
184
185     ldi r24,0x01
186     rcall lcd_command
187     ldi r24,lo8(1530)
188     ldi r25,hi8(1530)
189     rcall wait_usec
190
191     ldi r24,0x06
192     rcall lcd_command
193
194     ret
195
196 lcd_data:
197     sbi PORTD,PD2
198     rcall write_2_nibbles
199     ldi r24,43
200     ldi r25,0
201     rcall wait_usec
202     ret
203
204 lcd_command:
205     cbi PORTD,PD2
206     rcall write_2_nibbles
207     ldi r24,39
208     ldi r25,0
209     rcall wait_usec
210     ret
211
212 write_2_nibbles:
213     push r24
214     in r25,PIND
215     andi r25,0x0f
216     andi r24,0xf0
217     add r24,r25
218     out PORTD,r24
219     sbi PORTD,PD3
220     cbi PORTD,PD3
221     pop r24
222     swap r24
223     andi r24,0xf0
224     add r24,r25
225     out PORTD,r24
226     sbi PORTD,PD3
227     cbi PORTD,PD3
228     ret
```

9

```
229
230    scan_keypad:
231        ldi r24,0x01
232        rcall scan_row
233        swap r24
234        mov r27,r24
235        ldi r24,0x02
236        rcall scan_row
237        add r27,r24        ;//1h kai 2h grammh
238        ldi r24,0x03
239        rcall scan_row
240        swap r24
241        mov r26,r24
242        ldi r24,0x04
243        rcall scan_row
244        add r26,r24        ;//3h kai 4h grammh
245        movw r24,r26       ;//r25:r24 to apotelesma
246        ret
247
248    scan_row:
249        ldi r25,0x08
250    back_:
251        lsl r25
252        dec r24
253        brne back_
254        out PORTC,r25
255        nop
256        nop
257        in r24,PINC
258        andi r24,0x0f
259        ret
260
261    wait_msec:
262        push temp1
263        push temp2
264        ldi temp1, lo8(998)
265        ldi temp2, hi8(998)
266        rcall wait_usec
267        pop temp2
268        pop temp1
269        sbiw temp1, 1
270        brne wait_msec
271
272        ret
273
274    wait_usec:
275        sbiw temp1,1
276        nop
277        nop
278        nop
279        nop
280        brne wait_usec
281
282        ret
```

Άσκηση (v)

Κυρίως κώδικας:

```
1    #define __SFR_OFFSET 0
2    #include <avr/io.h>
3    #include <avr/interrupt.h>
4
5    .data
6        _tmp_ :.byte 2
7
8    .text
9
10   .global main
11   .org 0x000
12       rjmp main
13   .org 0x010
14       rjmp ovf_int_rout
15       reti
```

```
16
17    main:
18    reset:
19        ldi r24,lo8(RAMEND)
20        out SPL,r24
21        ldi r24,hi8(RAMEND)
22        out SPH,r24
23
24        ser r24
25        out DDRA,r24
26        out DDRD,r24
27
28        clr r24
29        out DDRB,r24
30
31        ldi r24 ,(1 << PC7) | (1 << PC6) | (1 << PC5) | (1 << PC4)
32        out DDRC ,r24
33
34
35        rcall lcd_init ;//initialize
36
37    loop:
38        in r24,PINB
39        cpi r24,0x0
40        breq loop
41
42
43        ;rcall set_timer
44        rcall getpass
45        rjmp loop
46
47    set_timer:
48        push r24
49        ;activate timer1
50        ;2 is TOIE1
51        ;avrStudio doesn ' t know about it
52        ldi r24,(1 << 2)
53        out TIMSK,r24
54        ldi r24,(1 << CS12)|(0<<CS11)|(1<<CS10)
55        out TCCR1B,r24
56
57        ldi r24,0xFF
58        sts TCNT1H,r24
59        ldi r24,0xFF
60        sts TCNT1L,r24
61
62        pop r24
63        sei
64        ret
65
66    ovf_int_rout:
67        rcall alarm_on
68        sei
69        reti
70
71    cls:
72        push r24
73        ldi r24,0x01
74        rcall lcd_command
75        ldi r24,lo8(1530)
76        ldi r25,hi8(1530)
77        rcall wait_usec
78        pop r24
79        ret
80
81
82    getpass:
83        rcall cls
84        rcall clear_tmp_
85        clr r24
86    dig0:
87        ldi r24,0x14
88        rcall scan_keypad_rising_edge
89
90        cpi r24,0x0
```

```
 91        breq dig0
 92        cpi r24,0x80
 93        brne alarm_on
 94
 95
 96        rcall clear_tmp_
 97        ldi r24,0x14
 98   dig1:
 99        ldi r24,0x14
100        rcall scan_keypad_rising_edge
101        cpi r24,0x0
102        breq dig1
103        cpi r24,0x40
104        brne alarm_on
105
106        rcall clear_tmp_
107        ldi r24,0x14
108   dig2:
109        rcall scan_keypad_rising_edge
110        cpi r24,0x0
111        breq dig2
112        cpi r24,0x02
113        brne alarm_on
114
115        rcall clear_tmp_
116        ldi r24,0x14
117   dig3:
118        rcall scan_keypad_rising_edge
119        cpi r24,0x0
120        breq dig3
121        cpi r24,0x10
122        brne alarm_on
123        rcall alarm_off
124        ret
125
126   clear_tmp_:
127        push r24
128        push r26
129        ldi r26,lo8(_tmp_)   ;//ka8arisma mnhmhs
130        ldi r27,hi8(_tmp_)
131        ldi r24,0x0
132        st  X+,r24
133        st  X,r24
134        pop r26
135        pop r24
136        ret
137
138   alarm_on:
139        push r24
140        ldi r24,0xff
141        out PORTA,r24
142        pop r24
143
144   write_alarm_on:
145        rcall lcd_init
146        ;rcall cls
147        ldi r24,'A'
148        rcall lcd_data
149        ldi r24,'L'
150        rcall lcd_data
151        ldi r24,'A'
152        rcall lcd_data
153        ldi r24,'R'
154        rcall lcd_data
155        ldi r24,'M'
156        rcall lcd_data
157        ldi r24,' '
158        rcall lcd_data
159        ldi r24,'O'
160        rcall lcd_data
161        ldi r24,'N'
162        rcall lcd_data
163        ret
164
165   alarm_off:
```

```asm
166         push r24
167         rcall write_alarm_off
168         ldi r24,0x00
169         out PORTA,r24
170         rcall set_timer_off
171         pop r24
172         ret
173
174 write_alarm_off:
175         rcall cls
176         ldi r24,'A'
177         rcall lcd_data
178         ldi r24,'L'
179         rcall lcd_data
180         ldi r24,'A'
181         rcall lcd_data
182         ldi r24,'R'
183         rcall lcd_data
184         ldi r24,'M'
185         rcall lcd_data
186         ldi r24,' '
187         rcall lcd_data
188         ldi r24,'O'
189         rcall lcd_data
190         ldi r24,'F'
191         rcall lcd_data
192         ldi r24,'F'
193         rcall lcd_data
194         ret
195
196 set_timer_off:
197         push r24
198         ldi r24,0x0
199         out TCCR1A,r24
200         out TCCR1B,r24
201         ret
202
203 keypad_to_ascii:
204         movw r26,r24
205         ldi r24,'*'
206         sbrc r26,0
207         ret
208         ldi r24,'0'
209         sbrc r26,1
210         ret
211         ldi r24,'#'
212         sbrc r26,2
213         ret
214         ldi r24,'D'
215         sbrc r26,3
216         ret
217         ldi r24,'7'
218         sbrc r26,4
219         ret
220         ldi r24,'8'
221         sbrc r26,5
222         ret
223         ldi r24,'9'
224         sbrc r26,6
225         ret
226         ldi r24,'C'
227         sbrc r26,7
228         ret
229         ldi r24,'4'
230         sbrc r27,0
231         ret
232         ldi r24,'5'
233         sbrc r27,1
234         ret
235         ldi r24,'6'
236         sbrc r27,2
237         ret
238         ldi r24,'B'
239         sbrc r27,3
240         ret
```

```
241        ldi r24,'1'
242        sbrc r27,4
243        ret
244        ldi r24,'2'
245        sbrc r27,5
246        ret
247        ldi r24,'3'
248        sbrc r27,6
249        ret
250        ldi r24,'A'
251        sbrc r27,7
252        ret
253        clr r24
254        ret
255
256    scan_keypad_rising_edge:
257        mov r22,r24
258        rcall scan_keypad
259
260        push r24 ;//apo8ikeush apotelesmatos
261        push r25
262        mov r24,r22 ;//ka8ysterhsh r22msec (10-20  msec)
263        ldi r25,0
264        rcall wait_msec
265
266        rcall scan_keypad ;//aporrispe osa plhktra emfanizoyn spin8hrismo
267        pop r23
268        pop r22
269        and r24,r22
270        and r25,r23
271        ldi r26,lo8(_tmp_)
272        ldi r27,hi8(_tmp_)
273        ld r23,X+
274        ld r22,X
275        st X,r24
276        st -X,r25
277        com r23
278        com r22
279        and r24,r22
280        and r25,r22
281        ret
282
283    lcd_init:
284        ldi r24,40
285        ldi r25,0
286        rcall wait_msec
287
288        ldi r24,0x30
289        out PORTD,r24
290        sbi PORTD,PD3
291        cbi PORTD,PD3
292        ldi r24,39
293        ldi r25,0
294        rcall wait_usec
295
296        ldi r24,0x30
297        out PORTD,r24
298        sbi PORTD,PD3
299        cbi PORTD,PD3
300        ldi r24,39
301        ldi r25,0
302        rcall wait_usec
303
304        ldi r24,0x20
305        out PORTD,r24
306        sbi PORTD,PD3
307        cbi PORTD,PD3
308        ldi r24,39
309        ldi r25,0
310        rcall wait_usec
311
312        ldi r24,0x28
313        rcall lcd_command
314
315        ldi r24,0x0c
```

```
316        rcall lcd_command
317
318        ldi r24,0x01
319        rcall lcd_command
320        ldi r24,lo8(1530)
321        ldi r25,hi8(1530)
322        rcall wait_usec
323
324        ldi r24,0x06
325        rcall lcd_command
326
327        ret
328
329  lcd_data:
330        sbi PORTD,PD2
331        rcall write_2_nibbles
332        ldi r24,43
333        ldi r25,0
334        rcall wait_usec
335        ret
336
337  lcd_command:
338        cbi PORTD,PD2
339        rcall write_2_nibbles
340        ldi r24,39
341        ldi r25,0
342        rcall wait_usec
343        ret
344
345  write_2_nibbles:
346        push r24
347        in r25,PINB
348        andi r25,0x0f
349        andi r24,0xf0
350        add r24,r25
351        out PORTD,r24
352        sbi PORTD,PD3
353        cbi PORTD,PD3
354        pop r24
355        swap r24
356        andi r24,0xf0
357        add r24,r25
358        out PORTD,r24
359        sbi PORTD,PD3
360        cbi PORTD,PD3
361        ret
362
363  scan_keypad:
364        ldi r24,0x01
365        rcall scan_row
366        swap r24
367        mov r27,r24
368        ldi r24,0x02
369        rcall scan_row
370        add r27,r24      ;//1h kai 2h grammh
371        ldi r24,0x03
372        rcall scan_row
373        swap r24
374        mov r26,r24
375        ldi r24,0x04
376        rcall scan_row
377        add r26,r24      ;//3h kai 4h grammh
378        movw r24,r26     ;//r25:r24 to apotelesma
379        ret
380
381  scan_row:
382        ldi r25,0x08
383  back_:
384        lsl r25
385        dec r24
386        brne back_
387        out PORTC,r25
388        nop
389        nop
390        in r24,PINC
```

15

```
391        andi r24,0x0f
392        ret
393
394  wait_msec:
395        push r24
396        push r25
397        ldi r24, lo8(998)
398        ldi r25, hi8(998)
399        rcall wait_usec
400        pop r25
401        pop r24
402        sbiw r24, 1
403        brne wait_msec
404
405        ret
406
407        wait_usec:
408        sbiw r24,1
409        nop
410        nop
411        nop
412        nop
413        brne wait_usec
414
415        ret
```

Άσκηση (vi)

Κυρίως κώδικας:

```
1    /*Includes for compatibility with GNU toolchain*/
2    #define __SFR_OFFSET 0
3    #include <avr/io.h>
4    #include <avr/interrupt.h>
5    .global main
6    /*
7     * AVRAssembler3_a.asm
8     *
9     *  Created: 14/2/2012 1:23:46 ??
10    *   Author: Eleni
11    */
12    /*
13    * AVRAssembler3.asm
14    *
15    *  Created: 13/2/2012 9:07:06 ??
16    *   Author: Eleni
17    */
18
19    /*
20    * bonus_part_2.asm
21    *
22    *  Created: 12/2/2012 1:08:30 ??
23    *   Author: Eleni
24    */
25
26   #define input  r18
27   #define sign  r21
28   #define tempo  r20
29   #define ekat   r22
30   #define dek   r23
31   #define mon  r19
32   #define quantum  r24
33    #define temp1  r24
34    #define temp2  r25
35
36   main:
37
38        ldi r24 ,lo8(RAMEND)
39        out SPL, r24
40        ldi r24, hi8(RAMEND)
41            out SPH, r24
42
43        clr r25              ;//bazw mhdenika gia eisodo sthn A
44        out DDRB, r25
```

16

```asm
45
46      ser r25                          ;//bazw 1 gia eksodo sthn D
47      out DDRD, r25
48
49
50
51          rcall lcd_init
52
53  start:
54
55
56
57      in input, PINB
58      mov tempo, input
59      lsl tempo
60      brcc positive            ;//an carry = 0 pame sto positive
61
62  negative:                                ;//edw einai arnhtikos o ari8mos
63      ;// bazw to arnhtiko proshmo
64      ldi sign, '-'
65      neg input                ;//to neg dinei kateu8eian to sumplhrwma ws pros 2
66      rjmp calculation
67
68  positive:
69      ;//an einai 8etikos bazw to + kai proxwrw sto calculation gia display
70      ldi sign, '+'
71
72  calculation:
73      clr ekat
74      clr dek
75      clr mon
76
77      cpi input, 0x64          ;//sugkrish me to 100
78      brlo count_dek           ;//branch if less sto metrhma dekadwn
79
80      ;//alliws exoume 1 ekatontada
81  count_ekat:
82      ldi ekat, 1              ;//giati mia ekatontada to polu 8a exoume
83      subi input, 0x64    ;//afairw 100 epeidh 8elw na sunexisw
84
85  count_dek:
86      cpi input, 10            ;//sugkrish me to 10
87      brlo count_mon           ;//an einai mikrotero tou 10 metrame tis monades
88      ;//alliws
89      subi input, 10           ;//afairw 10
90      inc dek
91          rjmp count_dek
92
93      ;//edw ston input exoun meinei pleon oi monades
94  count_mon:
95      mov mon, input
96      rjmp print_to_lcd
97
98  print_to_lcd:
99          ;//tupwnw to proshmo
100 lcd_clear:
101         ldi r24, 0x01            ;//ka8arismos o8onhs
102     rcall lcd_command
103     ldi r24, lo8(1530)
104     ldi r25, hi8(1530)
105     rcall wait_usec     ;//telos ka8arismos o8onhs
106     mov quantum, sign
107     rcall lcd_data
108
109     ;//elegxos gia an exw ekatontada
110     cpi ekat, 1
111     brlo check_dekades  ;//an <100 koitaw mhpws einai kai dekades0
112     subi ekat, -48              ;//pros8etw 48 gia na parw ton ascii
113     mov quantum, ekat
114     rcall lcd_data
115
116 check_dekades:
117     cpi dek, 0
118     breq check_monades  ;//an einai iso me 0
119     subi dek, -48               ;//alliws.
```

```
120        mov quantum, dek
121        rcall lcd_data
122
123    check_monades:
124        subi mon, -48
125        mov quantum, mon    ;//oi monades o,ti kai na nai tis deixnw, giati exw sumplhrwma ws pros 2. Opote 8a deiksw gia 0 to +0.
126        rcall lcd_data
127
128        ldi r24,0xAA
129        ldi r25,0x00
130        rcall wait_msec
131
132
133            rjmp start
134
135
136    ;//======routines gia thn o8onh=========
137    lcd_init:
138        ldi r24,40
139        ldi r25,0
140        rcall wait_msec
141
142        ldi r24,0x30
143        out PORTD,r24
144        sbi PORTD,PD3
145        cbi PORTD,PD3
146        ldi r24,39
147        ldi r25,0
148        rcall wait_usec
149
150        ldi r24,0x30
151        out PORTD,r24
152        sbi PORTD,PD3
153        cbi PORTD,PD3
154        ldi r24,39
155        ldi r25,0
156        rcall wait_usec
157
158        ldi r24,0x20
159        out PORTD,r24
160        sbi PORTD,PD3
161        cbi PORTD,PD3
162        ldi r24,39
163        ldi r25,0
164        rcall wait_usec
165
166        ldi r24,0x28
167        rcall lcd_command
168
169        ldi r24,0x0c
170        rcall lcd_command
171
172        ldi r24,0x01
173        rcall lcd_command
174        ldi r24,lo8(1530)
175        ldi r25,hi8(1530)
176        rcall wait_usec
177
178        ldi r24,0x06
179        rcall lcd_command
180
181        ret
182
183    lcd_data:
184        sbi PORTD,PD2
185        rcall write_2_nibbles
186        ldi r24,43
187        ldi r25,0
188        rcall wait_usec
189        ret
190
191    lcd_command:
192        cbi PORTD,PD2
193        rcall write_2_nibbles
194        ldi r24,39
```

```
195        ldi r25,0
196        rcall wait_usec
197        ret
198
199    write_2_nibbles:
200        push r24
201        in r25,PIND
202        andi r25,0x0f
203        andi r24,0xf0
204        add r24,r25
205        out PORTD,r24
206        sbi PORTD,PD3
207        cbi PORTD,PD3
208        pop r24
209        swap r24
210        andi r24,0xf0
211        add r24,r25
212        out PORTD,r24
213        sbi PORTD,PD3
214        cbi PORTD,PD3
215        ret
216
217    ;//=========wait routines=========
218        wait_msec:
219        push temp1
220        push temp2
221        ldi temp1, lo8(998)
222        ldi temp2, hi8(998)
223        rcall wait_usec
224        pop temp2
225        pop temp1
226        sbiw temp1, 1
227        brne wait_msec
228
229        ret
230
231        wait_usec:
232        sbiw temp1,1
233        nop
234        nop
235        nop
236        nop
237        brne wait_usec
238
239        ret
```