

Maciej Byczko Bartosz Matysiak	Prowadzący: dr inż. Jacek Mazurkiewicz	Numer ćwiczenia 4
Cz 13:15 TN	Temat ćwiczenia: Klawiatura	Ocena:
Grupa: B	Data wykonania: 8 Kwietnia 2021	

## 1 Zadanie 1,2 oraz 3

### 1.1 Polecenie

1. Przygotować program umożliwiający wybór i przypisanie do klawiszy klawiatury jednego z trzech zestawów znaków, które będą wypisywane na wyświetlaczu LCD gdy klawisze klawiatury będą naciskane.

Przełączenie zestawów:

- D - cyfry od 0 do 9 oraz znaki A, B, C,
  - \* - małe litery od a do m,
  - # - duże litery do A do M.
2. Przygotować program wyświetlający na LCD znaki uzyskiwane przez naciśnięcia klawiszy klawiatury. Uzyskiwany z klawiatury tekst jest wyświetlany tak by w pierwszej linii LCD pokazanych zostało 16 znaków, po czym następuje automatyczne przejście do drugiej linii, gdzie wyświetlamy kolejne 16 znaków. Jeśli z klawiatury generujemy kolejne znaki, to kasujemy bieżącą zawartość wyświetlacza i znów w pierwszej linii wyświetlamy następne 16 znaków tekstu, w drugiej - kolejne 16 znaków tekstu, itd.
  3. Przygotować program realizujący wyłączenie repetycji naciśniętego klawisza - jego kod skaningowy jest poddawany dalszym operacjom dopiero po naciśnięciu i następującym potem zwolnieniu klawisza.

### 1.2 Rozwiązanie

Powyższe polecenia zostały połączone w sposób następujący:

- Zachowane zostały klawisze wyboru zestawu znaków tzn. D, \*, # nadal odpowiadają za zmianę używanych symboli.
- Gdy są wpisywane znaki to na ekranie zostają one "zawijane" do następnej linii a następnie kasowane gdy wyświetlacz się wypełni.
- Repetycja klawiszy została wyłączona
- Zamienione zostały znaki A, B, C z zestawu pod klawiszem D na znaki: ":", "-", ")" ponieważ znaki A, B, C można uzyskać w zestawie #.

```
1  ljmp start
2
3  P5 equ 0F8H
4  P7 equ 0DBH
5
6  LCDstatus equ 0FF2EH ; adres do odczytu gotowosci LCD
7  LCDcontrol equ 0FF2CH ; adres do podania bajtu sterujacego LCD
8  LCDdataWR equ 0FF2DH ; adres do podania kodu ASCII na LCD
9
10 // bajty sterujace LCD, inne dostepne w opisie LCD na stronie WWW
11 #define HOME 0x80 // put cursor to second line
12 #define INITDISP 0x38 // LCD init (8-bit mode)
13 #define HOM2 0xc0 // put cursor to second line
14 #define LCDON 0x0e // LCD nn, cursor off, blinking off
15 #define CLEAR 0x01 // LCD display clear
16
17 // linie klawiatury – sterowanie na port P5
18 #define LINE_1 0x7f // 0111 1111
19 #define LINE_2 0xbf // 1011 1111
20 #define LINE_3 0xdf // 1101 1111
21 #define LINE_4 0xef // 1110 1111
22 #define ALL_LINES 0x0f // 0000 1111
23
24 org 0100H
25
26 // macro do wprowadzenia bajtu sterujacego na LCD
27 LCDcntrlWR MACRO x ; x – parametr wywolania macra – bajt sterujacy
28     LOCAL loop ; LOCAL oznacza ze etykieta loop moze sie powtorzyc
        w programie
29 loop: MOV DPTR,#LCDstatus; DPTR zaladowany adresem statusu
30     MOVX A,@DPTR ; pobranie bajtu z biezacym statusem LCD
31     JB ACC.7,loop ; testowanie najstarszego bitu akumulatora
32     ; – wskazuje gotowosc LCD
33     MOV DPTR,#LCDcontrol ; DPTR zaladowany adresem do podania bajtu
        sterujacego
34     MOV A, x ; do akumulatora trafia argument wywolania macrabajt
        sterujacy
35     MOVX @DPTR,A ; bajt sterujacy podany do LCD – zadana akcja
        widoczna na LCD
36     ENDM
37 // macro do wypisania znaku ASCII na LCD,
38 //znak ASCII przed wywolaniem macra ma byc w A
39 LCDcharWR MACRO
40     LOCAL tutu ; LOCAL oznacza ze etykieta tutu moze sie
        powtorzyc w programie
41     PUSH ACC ; odlozenie biezacej zawartosci akumulatora na
        stos
42 tutu: MOV DPTR,#LCDstatus ; DPTR zaladowany adresem statusu
43     MOVX A,@DPTR ; pobranie bajtu z biezacym statusem LCD
44     JB ACC.7,tutu ; testowanie najstarszego bitu akumulatora
45     ; – wskazuje gotowosc LCD
46     MOV DPTR,#LCDdataWR ; DPTR zaladowany adresem do podania bajtu
```

```
sterujacego
47 POP ACC ; w akumulatorze ponownie kod ASCII znaku na LCD
48 MOVX @DPTR,A ; kod ASCII podany do LCD – znak widoczny na LCD
49 ENDM
50 // macro do inicjalizacji wyswietlacza – bez parametrów
51 init_LCD MACRO
52 LCDcntrlWR #INITDISP ; wywołanie macra LCDcntrlWR – inicjalizacja
LCD
53 LCDcntrlWR #CLEAR ; wywołanie macra LCDcntrlWR – czyszczenie LCD
54 LCDcntrlWR #LCDON ; wywołanie macra LCDcntrlWR – konfiguracja
kursora
55 ENDM
56
57 // funkcja opóźnienia
58
59 delay: mov r1, #0FFH
60 dwa: mov r2, #0FFH
61 trzy: djnz r2, trzy
62 djnz r1, dwa
63 ret
64
65 // funkcja wypisania znaku
66 putcharLCD: LCDcharWR
67 ret
68
69 //-----
70 // znak do wyswietlenia w akumulatorze,
71 //ktory jest uzywany – koniecznosc uzycia stosu lub innego rejestru
72 putkbdCharsin2Lines:
73 ;sprawdzenie, czy r5 == #20H, wtedy przenosimy kursor do pierwszej
linii
74 mov r4, a
75 mov a, r5
76 clr c
77 subb a, #20H
78 jnz nieUstawiajPoczatku1Linii ;
79
80 LCDcntrlWR #HOME
81
82 nieUstawiajPoczatku1Linii:
83 mov a, r5
84 clr c
85 subb a, #10H
86 jnz nieUstawiajPoczatku2Linii
87
88 LCDcntrlWR #HOM2
89
90 nieUstawiajPoczatku2Linii:
91 mov a, r5
92 clr c
93 subb a, #00H
94 jnz nieClear
```

```
95
96     LCDcntrlWR #CLEAR
97     LCDcntrlWR #HOME
98     mov r5, #20H
99
100     nieClear:
101     mov a, r4 ; a – wartosc znaku do wpisania na wyswietlacz
102     acall putCharLCD
103     dec r5
104
105     koniec: ret
106
107     //-----
108     // tablica przekodowania klawisze – ASCII w XRAM
109
110     keyascii:
111         ; znaki dla naciśnięcia *
112         ; Układ klawiatury:
113         /*
114             a b c d
115             e f g h
116             i j k l
117             m
118         */
119         mov dptr, #80EBH
120         mov a, #"m"
121         movx @dptr, a
122
123         mov dptr, #8077H
124         mov a, #"a"
125         movx @dptr, a
126
127         mov dptr, #807BH
128         mov a, #"b"
129         movx @dptr, a
130
131         mov dptr, #807DH
132         mov a, #"c"
133         movx @dptr, a
134
135         mov dptr, #80B7H
136         mov a, #"e"
137         movx @dptr, a
138
139         mov dptr, #80BBH
140         mov a, #"f"
141         movx @dptr, a
142
143         mov dptr, #80BDH
144         mov a, #"g"
145         movx @dptr, a
146
```

```
147     mov  dptr , #80D7H
148     mov  a , #"i"
149     movx @dptr , a
150
151     mov  dptr , #80DBH
152     mov  a , #"j"
153     movx @dptr , a
154
155     mov  dptr , #80DDH
156     mov  a , #"k"
157     movx @dptr , a
158
159     mov  dptr , #807EH
160     mov  a , #"d"
161     movx @dptr , a
162
163     mov  dptr , #80BEH
164     mov  a , #"h"
165     movx @dptr , a
166
167     mov  dptr , #80DEH
168     mov  a , #"l"
169     movx @dptr , a
170
171     ; znaki dla nacisniecia #
172     ; Układ klawiatury:
173     /*
174         A B C D
175         E F G H
176         I J K L
177         M
178     */
179     mov  dptr , #81EBH
180     mov  a , #"M"
181     movx @dptr , a
182
183     mov  dptr , #8177H
184     mov  a , #"A"
185     movx @dptr , a
186
187     mov  dptr , #817BH
188     mov  a , #"B"
189     movx @dptr , a
190
191     mov  dptr , #817DH
192     mov  a , #"C"
193     movx @dptr , a
194
195     mov  dptr , #81B7H
196     mov  a , #"E"
197     movx @dptr , a
198
```

```
199     mov  dptr , #81BBH
200     mov  a , #"F"
201     movx @dptr , a
202
203     mov  dptr , #81BDH
204     mov  a , #"G"
205     movx @dptr , a
206
207     mov  dptr , #81D7H
208     mov  a , #"I"
209     movx @dptr , a
210
211     mov  dptr , #81DBH
212     mov  a , #"J"
213     movx @dptr , a
214
215     mov  dptr , #81DDH
216     mov  a , #"K"
217     movx @dptr , a
218
219     mov  dptr , #817EH
220     mov  a , #"D"
221     movx @dptr , a
222
223     mov  dptr , #81BEH
224     mov  a , #"H"
225     movx @dptr , a
226
227     mov  dptr , #81DEH
228     mov  a , #"L"
229     movx @dptr , a
230
231     ; znaki dla nacisniecia D
232     ; Układ klawiatury:
233     /*
234         1 2 3 A
235         4 5 6 B
236         7 8 9 C
237         0
238     */
239     mov  dptr , #82EBH
240     mov  a , #"0"
241     movx @dptr , a
242
243     mov  dptr , #8277H
244     mov  a , #"1"
245     movx @dptr , a
246
247     mov  dptr , #827BH
248     mov  a , #"2"
249     movx @dptr , a
250
```

```
251     mov dptr, #827DH
252     mov a, #"3"
253     movx @dptr, a
254
255     mov dptr, #82B7H
256     mov a, #"4"
257     movx @dptr, a
258
259     mov dptr, #82BBH
260     mov a, #"5"
261     movx @dptr, a
262
263     mov dptr, #82BDH
264     mov a, #"6"
265     movx @dptr, a
266
267     mov dptr, #82D7H
268     mov a, #"7"
269     movx @dptr, a
270
271     mov dptr, #82DBH
272     mov a, #"8"
273     movx @dptr, a
274
275     mov dptr, #82DDH
276     mov a, #"9"
277     movx @dptr, a
278
279     mov dptr, #827EH
280     mov a, #";"
281     movx @dptr, a
282
283     mov dptr, #82BEH
284     mov a, #"—"
285     movx @dptr, a
286
287     mov dptr, #82DEH
288     mov a, #")"
289     movx @dptr, a
290     ret
291 //-----
292 // program główny
293 start:
294     init_LCD
295     acall keyascii
296
297     mov r3, #80h; zakładamy, że na początku działamy w trybie *
298
299 key_1:
300     mov r0, #LINE_1
301     mov a, r0
302     mov P5, a
```

```
303     mov a, P7
304     anl a, r0
305     mov r2, a
306     clr c
307     subb a, r0
308     jz key_2
309     mov a, r2
310     mov dph, r3
311     mov dpl, a
312     movx a,@dptr
313     mov P1, a
314     acall putcharLCD
315     //zatrzymanie repetycji klawisza
316 pushed_1:
317     //nawiasy z lewej=wcisniete to samo, z prawej=klawisz puszczone
318     mov a, P7; wczytanie informacji o wcisnietej kolumnie
319     //[P7=1111 0111 -> a=1111 0111][P7=1111 1111 -> a=1111 1111]
320     anl a, r0
321     //[a = 0111 0111] [a = 0111 1111]
322     clr c; wyczyszczenie aby nie przeszkadzal
323     subb a, r0; odejmij, jezeli jest to samo to beda wszystkie bity=0
324     //[0111 0111-0111 1111=~0000 1000][0111 1111-0111 1111=0000 0000]
325     jnz pushed_1; to nie przechodz dalej
326     acall delay; poczekaj chwile zanim kontynuujesz
327
328 key_2:
329     mov r0, #LINE_2
330     mov a, r0
331     mov P5, a
332     mov a, P7
333     anl a, r0
334     mov r2, a
335     clr c
336     subb a, r0
337     jz key_3
338     mov a, r2
339     mov dph, r3
340     mov dpl, a
341     movx a,@dptr
342     mov P1, a
343     acall putcharLCD
344     //zatrzymanie repetycji klawisza
345 pushed_2:
346     //nawiasy z lewej=wcisniete to samo, z prawej=klawisz puszczone
347     mov a, P7; wczytanie informacji o wcisnietej kolumnie
348     //[P7=1111 0111 -> a=1111 0111][P7=1111 1111 -> a=1111 1111]
349     anl a, r0
350     //[a = 0111 0111] [a = 0111 1111]
351     clr c; wyczyszczenie aby nie przeszkadzal
352     subb a, r0; odejmij, jezeli jest to samo to beda wszystkie bity=0
353     //[0111 0111-0111 1111=~0000 1000][0111 1111-0111 1111=0000 0000]
354     jnz pushed_2; to nie przechodz dalej
```



```
355     a call delay; poczekaj chwile zanim kontynuujesz
356
357 key_3:
358     mov r0, #LINE_3
359     mov a, r0
360     mov P5, a
361     mov a, P7
362     anl a, r0
363     mov r2, a
364     clr c
365     subb a, r0
366     jz key_4
367     mov a, r2
368     mov dph, r3
369     mov dpl, a
370     movx a, @dptr
371     mov P1, a
372     a call putcharLCD
373     //zatrzymanie repetycji klawisza
374 pushed_3:
375     //nawiasy z lewej=wcisniete to samo, z prawej=klawisz puszczoney
376     mov a, P7; wczytanie informacji o wcisnietej kolumnie
377     //[P7=1111 0111 -> a=1111 0111][P7=1111 1111 -> a=1111 1111]
378     anl a, r0
379     //[a = 0111 0111] [a = 0111 1111]
380     clr c; wyczyszczenie aby nie przeszkadzal
381     subb a, r0; odejmij, jezeli jest to samo to beda wszystkie bity=0
382     //[0111 0111-0111 1111=~0000 1000][0111 1111-0111 1111=0000 0000]
383
384     jnz pushed_3; to nie przechodz dalej
385     a call delay; poczekaj chwile zanim kontynuujesz
386
387 key_4:
388     mov r0, #LINE_4
389     mov a, r0
390     mov P5, a
391     mov a, P7
392     anl a, r0
393     mov r2, a
394     clr c
395     subb a, r0
396     jz key_1
397     //Analiza czy uzytkownik chce zmienic tryb dzialania
398     //+ sprawdzanie wcisniecia znaku 0
399     mov a, r2
400     ;mamy teraz backup w r2
401     ;sprawdzenie *
402     clr c
403     subb a, #0E7H ;kod skaningowy *
404     jz modeStar
405     ;jesli nie, to odtworz a i sprawdz #
406     mov a, r2
```

```
407     clr c
408     subb a, #0EDH ;kod skaningowy #
409     jz modeHash
410     ;jesli nie, to odtworz i sprawdz D
411     mov a, r2
412     clr c
413     subb a, #0EEH ;kod skaningowy D
414     jz modeD
415     //jesli nic sie nie spelnilo, to znaczy ze mamy naciniety klawisz
416     //i mozemy bezpiecznie nawiazac do zainicjowanej wczesniej komorki
417     //pamieci
418     mov a, r2
419     mov dph, r3
420     mov dpl, a
421     movx a, @dptr
422     mov P1, a
423     acall putcharLCD
424     //zatrzymanie repetycji klawisza
425
426 pushed_4:
427     //nawiasy z lewej=wcisniete to samo, z prawej=klawisz puszczone
428     mov a, P7; wczytanie informacji o wcisnietej kolumnie
429     // [P7=1111 0111 -> a=1111 0111] [P7=1111 1111 -> a=1111 1111]
430     anl a, r0
431     // [a = 0111 0111] [a = 0111 1111]
432     clr c; wyczyszczenie aby nie przeszkadzal
433     subb a, r0; odejmij, jezeli jest to samo to beda wszystkie bity=0
434     // [0111 0111-0111 1111=~0000 1000] [0111 1111-0111 1111=0000 0000]
435     jnz pushed_4; to nie przechodz dalej
436     acall delay; poczekaj chwile zanim kontynuujesz
437
438 zapetl:
439     jmp key_1
440     ;minifunkcje ustawiajace odpowiedni ukklad klawiatury
441 modeStar:
442     mov r3, #080h
443     sjmp zapetl
444
445 modeHash:
446     mov r3, #081h
447     sjmp zapetl
448
449 modeD:
450     mov r3, #082h
451     sjmp zapetl
452
453 nop
454 nop
455 nop
456 jmp $
457 end start
```