

Maciej Byczko Bartosz Matysiak	Prowadzący: dr inż. Jacek Mazurkiewicz	Numer ćwiczenia 3
Cz 13:15 TN	Temat ćwiczenia: Wyświetlacz LCD	Ocena:
Grupa: B	Data wykonania: 25 Marca 2021	

1 Zadanie 1

1.1 Polecenie

Przygotować program wiążący guziki przypięte do P3 z wyświetlaczem LCD. W kodzie programu przygotowujemy 4 różne łańcuchy znaków przypisane do każdego z 4 guzików. Wyświetlanie danego tekstu następuje po naciśnięciu związanego z nim guzika. Program działa w pętli, a jednocześnie naciśnięcie dwóch skrajnych guzików powoduje wyjście z programu.

1.2 Rozwiązanie

```
1  ljmp start
2
3  LCDstatus equ 0FF2EH      ; adres do odczytu gotowosci LCD
4  LCDcontrol equ 0FF2CH     ; adres do podania bajtu sterujacego LCD
5  LCDdataWR equ 0FF3DH     ; adres do podania kodu ASCII na LCD
6
7  // bajty sterujace LCD, inne dostępne w opisie LCD na stronie WWW
8  #define HOME 0x80 // put cursor to second line
9  #define INITDISP 0x38 // LCD init (8-bit mode)
10 #define HOM2 0xc0 // put cursor to second line
11 #define LCDON 0x0e // LCD on, cursor off, blinking off
12 #define CLEAR 0x01 // LCD display clear
13
14 org 0100h
15
16 // deklaracja tekstów
17 text_przycisk: db "przycisk", 00
18 text_button: db "Wcisnieto", 00
19 text_button1: db "Przycisk 1", 00
20 text_button2: db "Przycisk 2", 00
21 text_button3: db "Przycisk 3", 00
22 text_button4: db "Przycisk 4", 00
23 text_exit: db "1+4 aby wyjsc", 00
24 text_end: db "Do widzenia ;)", 00
25 text_test: db "wypisz 16 znaków", 00
26
27 // macro do wprowadzenia bajtu sterujacego na LCD
28 LCDcntrlWR MACRO x      ; x - parametr wywolania macra - bajt
    sterujacy
```

```

29     LOCAL loop      ; LOCAL oznacza ze etykieta loop moze sie
                        powtorzyc w programie
30 loop: MOV  DPTR, #LCDstatus ; DPTR zaladowany adresem statusu
31     MOVX A, @DPTR      ; pobranie bajtu z biezacym statusem LCD
32     JB  ACC.7, loop    ; testowanie najstarszego bitu akumulatora
33                        ; - wskazuje gotowosc LCD
34     MOV  DPTR, #LCDcontrol ; DPTR zaladowany adresem do podania bajtu
                        sterujacego
35     MOV  A, x          ; do akumulatora trafia argument wywolania macra,
                        bajt sterujacy
36     MOVX @DPTR, A      ; bajt sterujacy podany do LCD - zadana akcja
                        widoczna na LCD
37     ENDM
38
39 // macro do wypisania znaku ASCII na LCD, znak ASCII przed
                        wywolaniem macra ma byc w A
40 LCDcharWR MACRO
41     LOCAL tutu      ; LOCAL oznacza ze etykieta tutu moze sie
                        powtorzyc w programie
42     PUSH ACC        ; odlozenie biezacej zawartosci akumulatora na
                        stos
43 tutu: MOV  DPTR, #LCDstatus ; DPTR zaladowany adresem statusu
44     MOVX A, @DPTR      ; pobranie bajtu z biezacym statusem LCD
45     JB  ACC.7, tutu    ; testowanie najstarszego bitu akumulatora
46                        ; - wskazuje gotowosc LCD
47     mov  83h, 06h      ; DPH - 83h, r6 - 06h czyli MOV DPH, R6
48     mov  82h, 07h      ; DPL - 82h, r7 - 07h czyli MOV DPL, R7
49     ;MOV  DPTR, #LCDdataWR ; DPTR zaladowany adresem do podania
                        bajtu sterujacego
50     POP  ACC           ; w akumulatorze ponownie kod ASCII znaku na LCD
51     MOVX @DPTR, A      ; kod ASCII podany do LCD - znak widoczny na
                        LCD
52     ENDM
53
54 // macro do inicjalizacji wyswietlacza - bez parametrów
55 init_LCD MACRO
56     LCDcntlWR #INITDISP ; wywołanie macra LCDcntlWR -
                        inicjalizacja LCD
57     LCDcntlWR #CLEAR    ; wywołanie macra LCDcntlWR - czyszczenie
                        LCD
58     LCDcntlWR #LCDON    ; wywołanie macra LCDcntlWR - konfiguracja
                        kursora
59     ENDM
60
61 write_str MACRO x
62     mov  dptr, x
63     acall putstrLCD
64     ENDM
65
66
67

```

```
68 // funkcja wypisania znaku
69 putcharLCD: LCDcharWR
70     ret
71
72 //funkcja wypisania lancucha znaków
73 putstrLCD:  mov r7, #30h ; DPL ustawiony tak by byl w DPTR adres
              FF30H
74 nextchar:  clr a
75             movc a, @a+dptr
76             jz koniec
77             push dph
78             push dpl
79             acall putcharLCD
80             pop dpl
81             pop dph
82             inc r7 ; dzieki temu mozliwa inkrementacja DPTR
83             inc dptr
84             sjmp nextchar
85 koniec:    ret
86
87 // funkcja opóznienia
88 delay:     mov r0, #15H
89 one:       mov r1, #0FFH
90 dwa:       mov r2, #0FFH
91 trzy:      djnz r2, trzy
92             djnz r1, dwa
93             djnz r0, one
94             ret
95
96
97
98 start:     init_LCD
99
100           mov r6, #0FFH ; adres LCDdataWR equ 0FF3DH jest w parze R6–R7
101           mov r7, #30H
102
103           LCDcntrlWR #CLEAR
104           write_str #text_test
105           LCDcntrlWR #HOM2
106           write_str #text_exit
107
108
109           select:
110
111           clr c
112           orl c, p3.3
113           orl c, p3.4
114           jnc exit
115
116           mov a, p3
117           jnb acc.2, push_button1
```

```
118     jnb acc.3, push_button2
119     jnb acc.4, push_button3
120     jnb acc.5, push_button4
121     jmp select
122
123     push_button1:
124         LCDcntlWR #CLEAR
125         write_str #text_button
126         LCDcntlWR #HOM2
127         write_str #text_button1
128         jmp select
129     push_button2:
130         LCDcntlWR #CLEAR
131         write_str #text_button
132         LCDcntlWR #HOM2
133         write_str #text_button2
134         jmp select
135     push_button3:
136         LCDcntlWR #CLEAR
137         write_str #text_button
138         LCDcntlWR #HOM2
139         write_str #text_button3
140         jmp select
141     push_button4:
142         LCDcntlWR #CLEAR
143         write_str #text_button
144         LCDcntlWR #HOM2
145         write_str #text_button4
146         jmp select
147     exit:
148         LCDcntlWR #CLEAR
149         write_str #text_end
150
151     nop
152     nop
153     nop
154     jmp $
155     end start
```

2 Zadanie 2

2.1 Polecenie

Przygotować program wyświetlający na LCD przygotowany w programie łańcuch znaków o długości znacząco przekraczającej 16 znaków. 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. Następnie pojawia się pauza, a po jej zakończeniu kasujemy bieżącą zawartość wyświetlacza i znów w pierwszej linii wyświetlamy następne 16 znaków tekstu, w drugiej - kolejne 16 znaków tekstu, pauza, itd. Akcja dobiega końca gdy zostanie takimi etapami wyświetlony cały przygotowany łańcuch znaków.

2.2 Rozwiązanie