

Maciej Byczko Bartosz Matysiak	Prowadzący: dr inż. Jacek Mazurkiewicz	Numer ćwiczenia 6
Cz 13:15 TN	Temat ćwiczenia: Transmisja portem szeregowym	Ocena:
Grupa: B	Data wykonania: 6 Maja 2021	

1 Zadanie 1

1.1 Polecenie

1.2 Rozwiązanie

```

1  ljmp start
2
3  P5 equ 0F8H
4  P7 equ 0DBH
5  p6 equ 0FAH
6
7  LCDstatus equ 0FF2EH      ; adres do odczytu gotowosci LCD
8  LCDcontrol equ 0FF2CH    ; adres do podania bajtu sterujacego LCD
9  LCDdataWR equ 0FF2DH    ; adres do podania kodu ASCII na LCD
10
11 // bajty sterujace LCD, inne dostepne w opisie LCD na stronie WWW
12 #define HOME      0x80      // put cursor to second line
13 #define INITDISP  0x38      // LCD init (8-bit mode)
14 #define HOM2      0xc0      // put cursor to second line
15 #define LCDON      0x0e      // LCD nn, cursor off, blinking off
16 #define CLEAR      0x01      // LCD display clear
17
18 // linie klawiatury – sterowanie na port P5
19 #define LINE_1      0x7f      // 0111 1111
20 #define LINE_2      0xbf      // 1011 1111
21 #define LINE_3      0xdf      // 1101 1111
22 #define LINE_4      0xef      // 1110 1111
23 #define ALL_LINES  0x0f      // 0000 1111
24
25 // procedura obslugi przerwania od portu szeregowego
26 org 0023H
27     PUSH ACC          ; by nie zniszczyc czegos co jest wazne w A
28     JBC TI, wyslac    ; odbieramy czy wysylamy dana
29     MOV A, SBUF       ; czytanie z z portu szeregowego
30     MOV R7, A         ; zamiana odczytanego dziwolaga na ASCII
31     ANL 07H, #0FH     ; przyklad 1 – 71H zamiast 31H
32     CLR C             ; jest 0111 0001 a ma byc 0011 0001
33     RRC A
34     ANL A, #0F0H
35     ORL A, R7
36     MOV P1, A         ; kod ASCII kontrolnie na diody

```

```

37     ACALL putcharLCD ; odczytany znak z portu szeregowego na LCD
38     CLR RI          ; przerwanie powodowane odczytem z portu
                        ; szeregowego obsluzone
39     JMP final
40 wyslac: MOV A, R5    ; czy wazan dana do wysylki jest w buforze
41     JNZ final        ; jesli nie – to nic nie rób
42     MOV A, R6        ; dana do wyslania z bufora
43     MOV SBUF, A      ; zapis do bufora portu szeregowego
44     MOV R5, #0FFH    ; wyzerowanie flagi wyslania
45 final: POP ACC
46     RETI
47
48 // program główny
49 org 0100H
50
51 // macro do wprowadzenia bajtu sterujacego na LCD
52 LCDcntrlWR MACRO x          ; x – parametr wywolania macra – bajt
                        ; sterujacy
53     LOCAL loop            ; LOCAL oznacza ze etykieta loop moze sie
                        ; powtorzyc w programie
54 loop: MOV DPTR,#LCDstatus  ; DPTR zaladowany adresem statusu
55     MOVX A,@DPTR          ; pobranie bajtu z biezacym statusem LCD
56     JB ACC.7,loop         ; testowanie najstarszego bitu akumulatora
57                           ; – wskazuje gotowosc LCD
58     MOV DPTR,#LCDcontrol  ; DPTR zaladowany adresem do podania bajtu
                        ; sterujacego
59     MOV A, x              ; do akumulatora trafia argument wywolania
                        ; -macrabajt sterujacy
60     MOVX @DPTR,A          ; bajt sterujacy podany do LCD – zadana
                        ; akcja widoczna na LCD
61     ENDM
62
63 // macro do wypisania znaku ASCII na LCD, znak ASCII przed wywołaniem
        ; macra ma byc w A
64 LCDcharWR MACRO
65     LOCAL tutu            ; LOCAL oznacza ze etykieta tutu moze sie
                        ; powtorzyc w programie
66     PUSH ACC              ; odlozenie biezacej zawartosci
                        ; akumulatora na stos
67 tutu: MOV DPTR,#LCDstatus  ; DPTR zaladowany adresem statusu
68     MOVX A,@DPTR          ; pobranie bajtu z biezacym statusem LCD
69     JB ACC.7,tutu         ; testowanie najstarszego bitu akumulatora
70                           ; – wskazuje gotowosc LCD
71     MOV DPTR,#LCDdataWR   ; DPTR zaladowany adresem do podania bajtu
                        ; sterujacego
72     POP ACC               ; w akumulatorze ponownie kod ASCII znaku
                        ; na LCD
73     MOVX @DPTR,A          ; kod ASCII podany do LCD – znak widoczny
                        ; na LCD
74     ENDM
75
76 // macro do inicjalizacji wyswietlacza – bez parametrów

```

```
77 init_LCD MACRO
78     LCDcntrlWR #INITDISP ; wywołanie macra LCDcntrlWR –
        inicjalizacja LCD
79     LCDcntrlWR #CLEAR    ; wywołanie macra LCDcntrlWR –
        czyszczenie LCD
80     LCDcntrlWR #LCDON    ; wywołanie macra LCDcntrlWR –
        konfiguracja kursora
81     ENDM
82
83 // funkcja opóźnienia
84
85     delay:  mov r1, #0FFH
86     dwa:   mov r2, #0FFH
87     trzy:  djnz r2, trzy
88           djnz r1, dwa
89           ret
90
91 // funkcja wypisania znaku
92 putcharLCD: LCDcharWR
93           ret
94
95 // wyłączenie piszczyka 1kHz
96 BEEPOFF:  MOV A, P6
97           CLR ACC.4
98           MOV P6, A
99           RET
100
101 // tablica przekodowania klawisze – ASCII w XRAM
102
103 keyascii: mov dptr, #80EBH
104           mov a, #"0"
105           movx @dptr, a
106
107           mov dptr, #8077H
108           mov a, #"1"
109           movx @dptr, a
110
111           mov dptr, #807BH
112           mov a, #"2"
113           movx @dptr, a
114
115           mov dptr, #807DH
116           mov a, #"3"
117           movx @dptr, a
118
119           mov dptr, #80B7H
120           mov a, #"4"
121           movx @dptr, a
122
123           mov dptr, #80BBH
124           mov a, #"5"
125           movx @dptr, a
```

```
126
127     mov dptr, #80BDH
128     mov a, #"6"
129     movx @dptr, a
130
131     mov dptr, #80D7H
132     mov a, #"7"
133     movx @dptr, a
134
135     mov dptr, #80DBH
136     mov a, #"8"
137     movx @dptr, a
138
139     mov dptr, #80DDH
140     mov a, #"9"
141     movx @dptr, a
142
143     mov dptr, #807EH
144     mov a, #"A"
145     movx @dptr, a
146
147     mov dptr, #80BEH
148     mov a, #"B"
149     movx @dptr, a
150
151     mov dptr, #80DEH
152     mov a, #"C"
153     movx @dptr, a
154
155     mov dptr, #80EEH
156     mov a, #"D"
157     movx @dptr, a
158
159     mov dptr, #80E7H
160     mov a, #"*"
161     movx @dptr, a
162
163     mov dptr, #80EDH
164     mov a, #"#"
165     movx @dptr, a
166
167     ret
168
169 // program główny
170     start:  init_LCD
171
172     acall keyascii
173
174     ACALL BEEPOFF ; wylaczenie piszczyka
175     mov R5, #0FFH
176     mov R6, #20H
177
```

```
178 // ustawienie portu szeregowego i timera
179     MOV SCON, #50H    ; tryb 1, 8 bitów danych, brak bitu parzystosci
180     MOV TMOD, #20H    ; konfiguracja Timera 1, tryb 2
181     MOV TH1, #0FDH    ; konfiguracja szybkości transmisji szeregowej
182     MOV TL1, #0FDH    ; 9600 bit/s
183     SETB TR1          ; timer rusza
184     MOV IE, #90H      ; włączenie przerwan od portu szeregowego
185
186
187 check_key MACRO x
188     mov r0, x
189     mov a, r0
190     mov P5, a
191     mov a, P7
192     anl a, r0
193     mov r2, a
194     clr c
195     subb a, r0
196
197 display_key:
198     mov a, r2
199     mov dph, #80h
200     mov dpl, a
201     movx a, @dptr
202     mov P1, a
203     mov r6, a        ; bufor dla znaku ASCII do wysyłki
204     mov r5, #00H      ; ustawienie znacznika że ważna dana w buforze
205                       R6
206     setb TI           ; wymuszamy przerwanie do wysyłki znaku
207     acall putcharLCD
208     acall hover
209
210     ret
211
212 hover:                ; nie wypuszczaj, dopoki klawisz nie zostanie "
213                       odcisniety"
214     mov a, P7
215     anl a, r0
216     clr c
217     subb a, r0
218     jnz hover
219     acall delay
220
221     ret
222
223 // obsługa klawiatury i wysyłki do portu szeregowego
224
225 key_1:    ;mov r0, #LINE_1
226           ;    mov a, r0
227           ;    mov P5, a
228           ;    mov a, P7
229           ;    anl a, r0
230           ;    mov r2, a
231           ;    clr c
```

```
228 ;      subb a, r0
229      check_key #LINE_1
230      jz key_2
231      acall display_key
232
233
234
235 key_2:  mov r0, #LINE_2
236      mov a, r0
237      mov P5, a
238      mov a, P7
239      anl a, r0
240      mov r2, a
241      clr c
242      subb a, r0
243      jz key_3
244      mov a, r2
245      mov dph, #80h
246      mov dpl, a
247      movx a, @dptr
248      mov P1, a
249      mov r6, a
250      mov r5, #00H
251      setb TI
252      acall putcharLCD
253      acall delay
254
255 key_3:  mov r0, #LINE_3
256      mov a, r0
257      mov P5, a
258      mov a, P7
259      anl a, r0
260      mov r2, a
261      clr c
262      subb a, r0
263      jz key_4
264      mov a, r2
265      mov dph, #80h
266      mov dpl, a
267      movx a, @dptr
268      mov P1, a
269      mov r6, a
270      mov r5, #00H
271      setb TI
272      acall putcharLCD
273      acall delay
274
275 key_4:  mov r0, #LINE_4
276      mov a, r0
277      mov P5, a
278      mov a, P7
279      anl a, r0
```

```
280     mov r2, a
281     clr c
282     subb a, r0
283     jz key_1
284     mov a, r2
285     mov dph, #80h
286     mov dpl, a
287     movx a, @dptr
288     mov P1, a
289     mov r6, a
290     mov r5, #00H
291     setb TI
292         acall putcharLCD
293     acall delay
294
295     jmp key_1
296
297     nop
298     nop
299     nop
300     jmp $
301     end start
```

2 Zadanie 2

2.1 Polecenie

2.2 Rozwiązanie