

| | | |
|-----------------------------------|---|----------------------|
| Maciej Byczko Bartosz Matysiak | Prowadzący: dr inż. Jacek Mazurkiewicz | Numer ćwiczenia 5 |
| Cz 13:15 TN | Temat ćwiczenia: Timery | Ocena: |
| Grupa: B | Data wykonania: 22 Kwietnia 2021 | |

1 Zadanie 1

1.1 Polecenie

Program "zegar", który powstał w finale zajęć laboratoryjnych proszę wzbogacić o możliwość "startu" i "stopu" odmierzania czasu. Od obsługi tych dodatkowych funkcjonalności proszę użyć wybranych guzików klawiatury przystawki oznaczonych literami A, B, C, D.

1.2 Rozwiązanie

```

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 000BH           ; obsluga przerwania
25   MOV TH0, #3CH      ; przeladowanie
26   MOV TL0, #0B0H    ; stalej timera na 50ms
27   DEC R0            ; korekta licznika
28   RETI             ; powrót z przerwania
29
30 org 0100H
31
32 // macro do wprowadzenia bajtu sterujacego na LCD

```

```

33 LCDcntrlWR MACRO x          ; x – parametr wywołania macra – bajt
    sterujący
34         LOCAL loop          ; LOCAL oznacza że etykieta loop może się
    powtórzyć w programie
35 loop: MOV DPTR,#LCDstatus    ; DPTR załadowany adresem statusu
36     MOVX A,@DPTR             ; pobranie bajtu z bieżącym statusem LCD
37     JB ACC.7,loop            ; testowanie najstarszego bitu akumulatora
38                                     ; – wskazuje gotowość LCD
39     MOV DPTR,#LCDcontrol    ; DPTR załadowany adresem do podania bajtu
    sterującego
40     MOV A, x                 ; do akumulatora trafia argument wywołania
    –macrabajt sterujący
41     MOVX @DPTR,A            ; bajt sterujący podany do LCD – zadana
    akcja widoczna na LCD
42     ENDM
43
44 // macro do wypisania znaku ASCII na LCD, znak ASCII przed wywołaniem
    macra ma być w A
45 LCDcharWR MACRO
46     LOCAL tutu              ; LOCAL oznacza że etykieta tutu może się
    powtórzyć w programie
47     PUSH ACC                ; odłożenie bieżącej zawartości
    akumulatora na stos
48 tutu: MOV DPTR,#LCDstatus    ; DPTR załadowany adresem statusu
49     MOVX A,@DPTR             ; pobranie bajtu z bieżącym statusem LCD
50     JB ACC.7,tutu            ; testowanie najstarszego bitu akumulatora
51                                     ; – wskazuje gotowość LCD
52     MOV DPTR,#LCDdataWR      ; DPTR załadowany adresem do podania bajtu
    sterującego
53     POP ACC                  ; w akumulatorze ponownie kod ASCII znaku
    na LCD
54     MOVX @DPTR,A            ; kod ASCII podany do LCD – znak widoczny
    na LCD
55     ENDM
56
57 // macro do inicjalizacji wyświetlacza – bez parametrów
58 init_LCD MACRO
59     LCDcntrlWR #INITDISP    ; wywołanie macra LCDcntrlWR –
    inicjalizacja LCD
60     LCDcntrlWR #CLEAR       ; wywołanie macra LCDcntrlWR –
    czyszczenie LCD
61     LCDcntrlWR #LCDON       ; wywołanie macra LCDcntrlWR –
    konfiguracja kursora
62     ENDM
63
64 // funkcja wypisania liczby dla potrzeb zegara
65 putdigitLCD: mov b, #10
66     div ab                  ; uzyskanie cyfry dziesiątek
67     add a, #30H             ; konwersja cyfry na kod ASCII
68     acall putcharLCD
69     mov a, b                 ; ładowanie cyfry jedności
70     add a, #30H             ; konwersja na LCD

```

```
71         acall putcharLCD
72         ret
73
74 // funkcja wypisywania znaku na LCD
75 putcharLCD: LCDcharWR
76         ret
77
78
79 // wyznaczanie bieżącej wartości zegara i jego wyświetlanie na LCD
80 ZEGAR:   INC R7           ; licznik sekund
81         MOV A, R7        ; obsługa sekund
82         CLR C
83         SUBB A, #60       ; przepelnienie sekund
84         JZ MINUTY
85         LCDcntrlWR #HOME  ; wyświetlenie całego zegara
86         MOV A, R5        ; godziny
87         ACALL putdigitLCD
88         MOV A, #": "     ; separator
89         ACALL putcharLCD
90         MOV A, R6        ; minuty
91         ACALL putdigitLCD
92         MOV A, #": "     ; separator
93         ACALL putcharLCD
94         MOV A, R7        ; sekundy
95         ACALL putdigitLCD
96         JMP FINAL
97 MINUTY:  MOV R7, #00H    ; zerowanie sekund
98         INC R6           ; licznik minut
99         MOV A, R6        ; obsługa minut
100        CLR C
101        SUBB A, #60       ; przepelnienie minut
102        JZ GODZINY
103        LCDcntrlWR #HOME  ; wyświetlenie całego zegara
104        MOV A, R5        ; godziny
105        ACALL putdigitLCD
106        MOV A, #": "     ; separator
107        ACALL putcharLCD
108        MOV A, R6        ; minuty
109        ACALL putdigitLCD
110        MOV A, #": "     ; separator
111        ACALL putcharLCD
112        MOV A, R7        ; sekundy
113        ACALL putdigitLCD
114        JMP FINAL
115 GODZINY: MOV R6, #00H    ; zerowanie minut
116        INC R5           ; licznik godzin
117        MOV A, R5
118        CLR C
119        SUBB A, #24       ; przepelenienie godzin – doba
120        JNZ EKRAN
121        MOV R5, #00H     ; zerowanie godzin
122 EKRAN:   LCDcntrlWR #HOME ; wyświetlenie całego zegara
```

```

123     MOV A, R5      ; godziny
124     ACALL putdigitLCD
125     MOV A, #": "   ; separator
126     ACALL putcharLCD
127     MOV A, R6      ; minuty
128     ACALL putdigitLCD
129     MOV A, #": "   ; separator
130     ACALL putcharLCD
131     MOV A, R7      ; sekundy
132     ACALL putdigitLCD
133 FINAL:    RET
134
135           ; program główny
136 START:    init_LCD
137     MOV TMOD, #01H   ; konfiguracja timera
138     MOV TH0, #3CH    ; ładowanie
139     MOV TL0, #0B0H   ; stałej timera na 50ms
140     SETB TR0         ; timer start
141     MOV IE, #82H     ; przerwania włącz
142     MOV R5, #00H     ; inicjacja zegara
143     MOV R6, #00H
144     MOV R7, #0FFH
145     MOV R0, #20      ; licznik odmierzen 20 x 50ms
146     MOV R3, #20      ; backup licznika odmierzen
147     ACALL ZEGAR      ; wyświetlenie zainicjowanego zegara
148
149 KEY_A:
150     mov r1, #LINE_1
151     mov a, r1
152     mov P5, a
153     mov a, P7
154     anl a, r1
155     mov r2, a
156     clr c
157     subb a, r1 ;WYKRYCIE jakiegokolwiek klawisza
158     jz KEY_B   ;jezli NIE zostal nacisniety zadn klawisz
159     mov a, r2   ;przywrocenie backupu kodu skaningowego
160     clr c
161     subb a, #7Eh ;WYKRYCIE A
162     jnz KEY_B   ;Jezeli NIE zostal nacisniety klawisz A
163     ;reakcja na A:
164     mov a, r3
165     mov r0, a   ;przywrocenie r0 z backupa
166     mov IE, #82H;właczenie obsługi przerwan
167     ;mozliwe, ze trzeba dorobic petle, ktora nie wypusci, dopoki A
        zostanie "odcisniete"
168     nieWypuszczajA:
169     mov a, P7
170     anl a, r1
171     clr c
172     subb a, r1
173     jnz nieWypuszczajA

```

```
174
175
176 KEY_B:
177     mov r1, #LINE_2
178     mov a, r1
179     mov P5, a
180     mov a, P7
181     anl a, r1
182     mov r2, a
183     clr c
184     subb a, r1 ;WYKRYCIE jakiegokolwiek klawisza
185     jz CZEKAM ;jezli NIE zostal nacisniety zaden klawisz
186     mov a, r2 ;przywrocenie backupu kodu skaningowego
187     clr c
188     subb a, #0BEh ;WYKRYCIE B
189     jnz CZEKAM ;Jezeli NIE zostal nacisniety klawisz B
190     ;reakcja na B:
191     mov IE, #00H ;wylaczenie obslugi przerwan
192     mov a, r0
193     mov r3, a ;odlozenie r0 do r3 (backup)
194
195     ;mozliwe, ze trzeba dorobic petle, ktora nie wypusci, dopoki B
196     ; zostanie "odcisniete"
197     nieWypuszczajB:
198     mov a, P7
199     anl a, r1
200     clr c
201     subb a, r1
202     jnz nieWypuszczajB
203
204     jmp KEY_A
205
206     MOV A, #0FH
207     MOV P1, A ; zapalenie diód
208
209
210 CZEKAM: MOV A, R0 ; czekam, a timer
211           ;nie jestem pewien, ale tu chyba moze zajsc
212           ;dekrementacja zera w r0 "nie w pore", tj. gdy jest
213           ;sprawdzane wcisniecie klawisza
214           ;, co sprawi, ze program zapetli sie tam, gdzie nie
215           ;powinien
216           ;aby to rozwiazac, przebuduje to:
217     JZ WYPISZ
218     clr c
219     add a, #20h ; daje zegarowi zapas, zwiazany z opisana
220     ;powyzej sytuacja
221     jc WYPISZ
222     jmp KEY_A
```

```

220 ;JNZ KEY_A ; mierzy laczny czas 1s (chociaz po dodaniu
      obsługi klawiatury , pewnie troche wiecej)
221 ;add a, #20h ; daje zegarowi zapas , zwiazany z opisana
      powyzej sytuacja
222 ;jnc KEY_A
223
224 WYPISZ:
225
226 MOV R0, #20 ; po zgloszeniu przerwania — ustawiam na nowo
      licznik odmierzen 20 x 50ms
227 ACALL ZEGAR ; uruchomienie procedury obsługi i wyswietlenia
      zegara
228
229 MOV A, P1 ; zmiana
230 CPL A ; swiecenia
231 MOV P1, A ; diód
232
233 JMP KEY_A ; powrot do sprawdzania klawiszy
234
235 NOP
236 NOP
237 NOP
238 JMP $
239 END START

```

2 Zadanie 2

Program "zegar", który powstał w finale zajęć laboratoryjnych proszę wzbogacić o możliwość ustawienia pozycji godzin i minut przed uruchomieniem odmierzania czasu. Proszę pamiętać, że zarówno pozycja godzin, jak i minut wymaga wprowadzenia liczby dwucyfrowej w zakresie od 00 do 23 dla godzin i - odpowiednio - od 00 do 59 dla minut. Trzeba zatem "zbudować" z naciśniętych "po sobie" cyfr od 0 do 9 liczbę dwucyfrową z kontrolą czy uzyskana liczba mieści się w wymaganym zakresie. Zatwierdzenie wprowadzonej godziny guzikiem #, wprowadzonych minut - guzikiem *.

2.1 Rozwiązanie

```

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 000BH ; obsługa przerwania
25     MOV TH0, #3CH ; przeładowanie
26     MOV TL0, #0B0H ; stalej timera na 50ms
27     DEC R0 ; korekta licznika
28     RETI ; powrót z przerwania
29
30 org 0100H
31
32 // macro do wprowadzenia bajtu sterującego na LCD
33 LCDcntrlWR MACRO x ; x – parametr wywołania macra – bajt
    sterujący
34     LOCAL loop ; LOCAL oznacza ze etykieta loop może się
        powtórzyć w programie
35 loop: MOV DPTR,#LCDstatus ; DPTR załadowany adresem statusu
36     MOVX A,@DPTR ; pobranie bajtu z bieżącym statusem LCD
37     JB ACC.7,loop ; testowanie najstarszego bitu akumulatora
38     ; – wskazuje gotowość LCD
39     MOV DPTR,#LCDcontrol ; DPTR załadowany adresem do podania bajtu
        sterującego
40     MOV A, x ; do akumulatora trafia argument wywołania
        –macrabajt sterujący
41     MOVX @DPTR,A ; bajt sterujący podany do LCD – zadana
        akcja widoczna na LCD
42     ENDM
43
44 // macro do wypisania znaku ASCII na LCD, znak ASCII przed wywołaniem
    macra ma być w A
45 LCDcharWR MACRO
46     LOCAL tutu ; LOCAL oznacza ze etykieta tutu może się
        powtórzyć w programie
47     PUSH ACC ; odłożenie bieżącej zawartości
        akumulatora na stos
48 tutu: MOV DPTR,#LCDstatus ; DPTR załadowany adresem statusu
49     MOVX A,@DPTR ; pobranie bajtu z bieżącym statusem LCD
50     JB ACC.7,tutu ; testowanie najstarszego bitu akumulatora
51     ; – wskazuje gotowość LCD
52     MOV DPTR,#LCDdataWR ; DPTR załadowany adresem do podania bajtu
        sterującego
53     POP ACC ; w akumulatorze ponownie kod ASCII znaku
        na LCD
54     MOVX @DPTR,A ; kod ASCII podany do LCD – znak widoczny
        na LCD
55     ENDM
56
57 // macro do inicjalizacji wyświetlacza – bez parametrów

```

```
58 init_LCD MACRO
59     LCDcntlWR #INITDISP ; wywołanie macra LCDcntlWR –
        inicjalizacja LCD
60     LCDcntlWR #CLEAR    ; wywołanie macra LCDcntlWR –
        czyszczenie LCD
61     LCDcntlWR #LCDON    ; wywołanie macra LCDcntlWR –
        konfiguracja kursora
62     ENDM
63
64 // funkcja wypisania liczby dla potrzeb zegara
65 putdigitLCD:  mov b, #10
66               div ab      ; uzyskanie cyfry dziesiątek
67               add a, #30H  ; konwersja cyfry na kod ASCII
68               acall putcharLCD
69               mov a, b     ; ładowanie cyfry jedności
70               add a, #30H  ; konwersja na LCD
71               acall putcharLCD
72               ret
73
74 // funkcja wypisywania znaku na LCD
75 putcharLCD:  LCDcharWR
76             ret
77
78 delay:      mov r3, #0FFH
79 dwa:        mov r4, #0FFH
80 trzy:       djnz r4, trzy
81             djnz r3, dwa
82             ret
83
84 keyascii:   mov dptr, #80EBH
85             mov a, #"0"
86             movx @dptr, a
87
88             mov dptr, #8077H
89             mov a, #"1"
90             movx @dptr, a
91
92             mov dptr, #807BH
93             mov a, #"2"
94             movx @dptr, a
95
96             mov dptr, #807DH
97             mov a, #"3"
98             movx @dptr, a
99
100            mov dptr, #80B7H
101            mov a, #"4"
102            movx @dptr, a
103
104            mov dptr, #80BBH
105            mov a, #"5"
106            movx @dptr, a
```



```
107
108     mov dptr, #80BDH
109     mov a, #"6"
110     movx @dptr, a
111
112     mov dptr, #80D7H
113     mov a, #"7"
114     movx @dptr, a
115
116     mov dptr, #80DBH
117     mov a, #"8"
118     movx @dptr, a
119
120     mov dptr, #80DDH
121     mov a, #"9"
122     movx @dptr, a
123
124     mov dptr, #807EH
125     mov a, #"A"
126     movx @dptr, a
127
128     mov dptr, #80BEH
129     mov a, #"B"
130     movx @dptr, a
131
132     mov dptr, #80DEH
133     mov a, #"C"
134     movx @dptr, a
135
136     mov dptr, #80EEH
137     mov a, #"D"
138     movx @dptr, a
139
140     mov dptr, #80E7H
141     mov a, #"*"
142     movx @dptr, a
143
144     mov dptr, #80EDH
145     mov a, #"#"
146     movx @dptr, a
147
148     ret
149
```

INPUT:

FirstDigit:

```
152     LCDcntrlWR #CLEAR      ; wywołanie macra LCDcntrlWR – czyszczenie
153                               LCD i przesunięcie kursora na początek
```

```
153     LCDcntrlWR #HOME
```

```
154     mov r1, #0              ; zerujemy bufor na testowaną wartość
```

```
156     Line1for1Digit:
```

```
157     mov r0, #LINE_1
```

```

158     mov a, r0
159     mov P5, a
160     mov a, P7
161     anl a, r0
162     mov r2, a
163     clr c
164     subb a, r0
165     jz Line2for1Digit           ;jezeli nic nie zostalo wcisniete
166     mov a, r2
167     clr c
168     subb a, #7Eh               ;jezeli zostal nacisniety klawisz A, skocz
169     jz Line2for1Digit
170     mov a, r2
171     mov dph, #80h
172     mov dpl, a
173     movx a,@dptr              ;teraz w akumulatorze jest kod ascii danej
                                cyfry
174     acall putcharLCD
175     clr c
176     subb a, #30h              ;konwertujemy kod ascii na cyfre
177     mov r1, a                 ;dodajemy cyfre do bufora
178     czekaj11:                 ; nie wypuszczaj, dopoki klawisz nie
                                zostanie "odcisniety"
179     mov a, P7
180     anl a, r0
181     clr c
182     subb a, r0
183     jnz czekaj11
184     acall delay
185     jmp SecondDigit
186
187     Line2for1Digit:
188     mov r0, #LINE_2
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     jz Line3for1Digit         ;jezeli nic nie zostalo wcisniete
197     mov a, r2
198     clr c
199     subb a, #0BEh             ;jezeli zostal nacisniety klawisz B, skocz
200     jz Line3for1Digit
201     mov a, r2
202     mov dph, #80h
203     mov dpl, a
204     movx a,@dptr              ;teraz w akumulatorze jest kod ascii danej
                                cyfry
205     acall putcharLCD
206     clr c

```

```
207     subb a, #30h           ;konwertujemy kod ascii na cyfre
208     mov r1, a             ;dodajemy cyfre do bufora
209     czekaj21:             ; nie wypuszczaj, dopoki klawisz nie
                           zostanie "odcisniety"
210     mov a, P7
211     anl a, r0
212     clr c
213     subb a, r0
214     jnz czekaj21
215     acall delay
216     jmp SecondDigit
217
218     Line3for1Digit:
219     mov r0, #LINE_3
220     mov a, r0
221     mov P5, a
222     mov a, P7
223     anl a, r0
224     mov r2, a
225     clr c
226     subb a, r0
227     jz Line4for1Digit      ;jezeli nic nie zostalo wcisniete
228     mov a, r2
229     clr c
230     subb a, #0DEh          ;jezeli zostal nacisniety klawisz C, skocz
231     jz Line4for1Digit
232     mov a, r2
233     mov dph, #80h
234     mov dpl, a
235     movx a, @dptr          ;teraz w akumulatorze jest kod ascii danej
                           cyfry
236     acall putcharLCD
237     clr c
238     subb a, #30h           ;konwertujemy kod ascii na cyfre
239     mov r1, a             ;dodajemy cyfre do bufora
240     czekaj31:             ; nie wypuszczaj, dopoki klawisz nie
                           zostanie "odcisniety"
241     mov a, P7
242     anl a, r0
243     clr c
244     subb a, r0
245     jnz czekaj31
246     acall delay
247     jmp SecondDigit
248
249     Line4for1Digit:        ;TU BEDZIE TROCHE INACZEJ!!!!!!
250     mov r0, #LINE_4
251     mov a, r0
252     mov P5, a
253     mov a, P7
254     anl a, r0
255     mov r2, a
```

```

256     clr c
257     subb a, r0
258     jz goback1Digit           ;jezeli nic nie zostalo wcisniete
259     mov a, r2
260     clr c
261     subb a, #0EBh           ;jezeli nie zostal nacisniety klawisz 0,
        skocz
262     jnz goback1Digit
263     mov a, r2
264     mov dph, #80h
265     mov dpl, a
266     movx a, @dptr           ;teraz w akumulatorze jest kod ascii danej
        cyfry
267     acall putcharLCD
268     clr c
269     subb a, #30h           ;konwertujemy kod ascii na cyfre
270     mov r1, a             ;dodajemy cyfre do bufora
271     czekaj41:             ; nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisniety"
272     mov a, P7
273     anl a, r0
274     clr c
275     subb a, r0
276     jnz czekaj41
277     acall delay
278     jmp SecondDigit
279
280 goback1Digit:
281     ljmp Line1for1Digit
282
283 SecondDigit:
284     mov b, #10             ;mnozmy wartosc pierwszej cyfry przez 10
285     mov a, r1
286     mul ab                 ;teraz w akumulatorze powinna byc
        dziesieciokrotnosc cyfry dziesiatek
287     mov r1, a
288     Line1for2Digit:
289     mov r0, #LINE_1
290     mov a, r0
291     mov P5, a
292     mov a, P7
293     anl a, r0
294     mov r2, a
295     clr c
296     subb a, r0
297     jz Line2for2Digit       ;jezeli nic nie zostalo wcisniete
298     mov a, r2
299     clr c
300     subb a, #7Eh           ;jezeli zostal nacisniety klawisz A, skocz
301     jz Line2for2Digit
302     mov a, r2
303     mov dph, #80h

```

```
304     mov dpl, a
305     movx a, @dptr           ;teraz w akumulatorze jest kod ascii danej
        cyfry
306     acall putcharLCD
307     clr c
308     subb a, #30h           ;konwertujemy kod ascii na cyfre
309     add a, r1
310     mov r1, a              ;dodajemy bufor do cyfry jednosci, a
        nastepnie kopiujemy z powrotem do bufora
311     czekaj12:              ; nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisniety"
312     mov a, P7
313     anl a, r0
314     clr c
315     subb a, r0
316     jnz czekaj12
317     acall delay
318     jmp Hashtag
319
320     Line2for2Digit:
321     mov r0, #LINE_2
322     mov a, r0
323     mov P5, a
324     mov a, P7
325     anl a, r0
326     mov r2, a
327     clr c
328     subb a, r0
329     jz Line3for2Digit      ;jezeli nic nie zostalo wcisniete
330     mov a, r2
331     clr c
332     subb a, #0BEh          ;jezeli zostal nacisniety klawisz B, skocz
333     jz Line3for2Digit
334     mov a, r2
335     mov dph, #80h
336     mov dpl, a
337     movx a, @dptr           ;teraz w akumulatorze jest kod ascii danej
        cyfry
338     acall putcharLCD
339     clr c
340     subb a, #30h           ;konwertujemy kod ascii na cyfre
341     add a, r1
342     mov r1, a              ;dodajemy bufor do cyfry jednosci, a
        nastepnie kopiujemy z powrotem do bufora
343     czekaj22:              ; nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisniety"
344     mov a, P7
345     anl a, r0
346     clr c
347     subb a, r0
348     jnz czekaj22
349     acall delay
```

```
350     jmp Hashtag
351
352     Line3for2Digit:
353     mov r0, #LINE_3
354     mov a, r0
355     mov P5, a
356     mov a, P7
357     anl a, r0
358     mov r2, a
359     clr c
360     subb a, r0
361     jz Line4for2Digit           ;jezeli nic nie zostalo wcisniete
362     mov a, r2
363     clr c
364     subb a, #0DEh             ;jezeli zostal nacisniety klawisz C, skocz
365     jz Line4for2Digit
366     mov a, r2
367     mov dph, #80h
368     mov dpl, a
369     movx a, @dptr             ;teraz w akumulatorze jest kod ascii danej
                                cyfry
370     acall putcharLCD
371     clr c
372     subb a, #30h              ;konwertujemy kod ascii na cyfre
373     add a, r1
374     mov r1, a                 ;dodajemy bufor do cyfry jednosci, a
                                nastepnie kopiujemy z powrotem do bufora
375     czekaj32:                 ; nie wypuszczaj, dopoki klawisz nie
                                zostanie "odcisniety"
376     mov a, P7
377     anl a, r0
378     clr c
379     subb a, r0
380     jnz czekaj32
381     acall delay
382     jmp Hashtag
383
384     Line4for2Digit:
385     mov r0, #LINE_4
386     mov a, r0
387     mov P5, a
388     mov a, P7
389     anl a, r0
390     mov r2, a
391     clr c
392     subb a, r0
393     jz goback2Digit           ;jezeli nic nie zostalo wcisniete
394     mov a, r2
395     clr c
396     subb a, #0EBh             ;jezeli nie zostal nacisniety klawisz 0,
                                skocz
397     jnz goback2Digit
```

```
398     mov a, r2
399     mov dph, #80h
400     mov dpl, a
401     movx a, @dptr           ; teraz w akumulatorze jest kod ascii danej
                             cyfry
402     acall putcharLCD
403     clr c
404     subb a, #30h           ; konwertujemy kod ascii na cyfre
405     add a, r1
406     mov r1, a              ; dodajemy bufor do cyfry jednosci, a
                             nastepnie kopiujemy z powrotem do bufora
407     czekaj42:              ; nie wypuszczaj, dopoki klawisz nie
                             zostanie "odcisniety"
408     mov a, P7
409     anl a, r0
410     clr c
411     subb a, r0
412     jnz czekaj42
413     acall delay
414     jmp Hashtag
415
416 goback2Digit:
417     ljmp Line1for2Digit
418
419
420 Hashtag:
421 ; dwie petle, czekajace na wpisanie znaku hash
422     mov r0, #LINE_4
423     mov a, r0
424     mov P5, a
425     mov a, P7
426     anl a, r0
427     mov r2, a
428     clr c
429     subb a, r0
430     jz Hashtag            ; jezeli nic sie nie wpisalo
431     mov a, r2
432     clr c
433     subb a, #0EDh         ; wpisane znaku #
434     jnz Hashtag
435 ; jesli wcisnelo sie #, czekaj na jego "odcisnienie"
436     czekajHash:           ; nie wypuszczaj, dopoki klawisz nie
                             zostanie "odcisniety"
437     mov a, P7
438     anl a, r0
439     clr c
440     subb a, r0
441     jnz czekajHash
442     acall delay
443     ; walidacja godziny
444     mov a, r1
445     clr c
```

```

446     subb a, #24
447     jnc hourValidationIncorrect
448     mov a, r1
449     mov r5, a
450     mov a, #': '
451     acall putcharLCD
452
453     jmp ThirdDigit
454
455 hourValidationIncorrect:
456     ljmp FirstDigit
457
458 ThirdDigit:
459     LCDcntlWR #CLEAR      ; wywołanie macra LCDcntlWR – czyszczenie
                           ; LCD i przesunięcie kursora na początek
460     LCDcntlWR #HOME
461     mov a, r5              ; przywróć godzinę na wyświetlacz
462     acall putdigitLCD
463     mov a, #': '
464     acall putcharLCD
465     mov r1, #0             ; zerujemy bufor na testowaną wartość
466
467     Line1for3Digit:
468     mov r0, #LINE_1
469     mov a, r0
470     mov P5, a
471     mov a, P7
472     anl a, r0
473     mov r2, a
474     clr c
475     subb a, r0
476     jz Line2for3Digit     ; jeżeli nic nie zostało wcisnięte
477     mov a, r2
478     clr c
479     subb a, #7Eh          ; jeżeli został naciśnięty klawisz A, skocz
480     jz Line2for3Digit
481     mov a, r2
482     mov dph, #80h
483     mov dpl, a
484     movx a, @dptr         ; teraz w akumulatorze jest kod ascii danej
                           ; cyfry
485     acall putcharLCD
486     clr c
487     subb a, #30h          ; konwertujemy kod ascii na cyfry
488     mov r1, a             ; dodajemy cyfry do bufora
489     czekaj13:             ; nie wypuszczaj, dopóki klawisz nie
                           ; zostanie "odcisnięty"
490     mov a, P7
491     anl a, r0
492     clr c
493     subb a, r0
494     jnz czekaj13

```



```

495     acall delay
496     jmp FourthDigit
497
498     Line2for3Digit:
499     mov r0, #LINE_2
500     mov a, r0
501     mov P5, a
502     mov a, P7
503     anl a, r0
504     mov r2, a
505     clr c
506     subb a, r0
507     jz Line3for3Digit           ;jezeli nic nie zostalo wcisniete
508     mov a, r2
509     clr c
510     subb a, #0BEh             ;jezeli zostal nacisniety klawisz B, skocz
511     jz Line3for3Digit
512     mov a, r2
513     mov dph, #80h
514     mov dpl, a
515     movx a, @dptr             ;teraz w akumulatorze jest kod ascii danej
                                cyfry
516     acall putcharLCD
517     clr c
518     subb a, #30h              ;konwertujemy kod ascii na cyfre
519     mov r1, a                 ;dodajemy cyfre do bufora
520     czekaj23:                 ; nie wypuszczaj, dopoki klawisz nie
                                zostanie "odcisniety"
521     mov a, P7
522     anl a, r0
523     clr c
524     subb a, r0
525     jnz czekaj23
526     acall delay
527     jmp FourthDigit
528
529     Line3for3Digit:
530     mov r0, #LINE_3
531     mov a, r0
532     mov P5, a
533     mov a, P7
534     anl a, r0
535     mov r2, a
536     clr c
537     subb a, r0
538     jz Line4for3Digit         ;jezeli nic nie zostalo wcisniete
539     mov a, r2
540     clr c
541     subb a, #0DEh             ;jezeli zostal nacisniety klawisz C, skocz
542     jz Line4for3Digit
543     mov a, r2
544     mov dph, #80h

```

```

545     mov dpl, a
546     movx a, @dptr           ;teraz w akumulatorze jest kod ascii danej
        cyfry
547     acall putcharLCD
548     clr c
549     subb a, #30h           ;konwertujemy kod ascii na cyfre
550     mov r1, a              ;dodajemy cyfre do bufora
551     czekaj33:              ; nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisniety"
552     mov a, P7
553     anl a, r0
554     clr c
555     subb a, r0
556     jnz czekaj33
557     acall delay
558     jmp FourthDigit

559
560     Line4for3Digit:        ;TU BEDZIE TROCHE INACZEJ!!!!!!
561     mov r0, #LINE_4
562     mov a, r0
563     mov P5, a
564     mov a, P7
565     anl a, r0
566     mov r2, a
567     clr c
568     subb a, r0
569     jz goback3Digit        ;jezeli nic nie zostalo wcisniete
570     mov a, r2
571     clr c
572     subb a, #0EBh          ;jezeli nie zostal nacisniety klawisz 0,
        skocz
573     jnz goback3Digit
574     mov a, r2
575     mov dph, #80h
576     mov dpl, a
577     movx a, @dptr           ;teraz w akumulatorze jest kod ascii danej
        cyfry
578     acall putcharLCD
579     clr c
580     subb a, #30h           ;konwertujemy kod ascii na cyfre
581     mov r1, a              ;dodajemy cyfre do bufora
582     czekaj43:              ; nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisniety"
583     mov a, P7
584     anl a, r0
585     clr c
586     subb a, r0
587     jnz czekaj43
588     acall delay
589     jmp FourthDigit

590
591 goback3Digit:

```

```
592 ljmp Line1for3Digit
593
594 FourthDigit:
595     mov b, #10           ;mnozmy wartosc pierwszej cyfry przez 10
596     mov a, r1
597     mul ab               ;teraz w akumulatorze powinna byc
                           dziesieciokrotnosc cyfry dziesiatek
598     mov r1, a
599     Line1for4Digit:
600     mov r0, #LINE_1
601     mov a, r0
602     mov P5, a
603     mov a, P7
604     anl a, r0
605     mov r2, a
606     clr c
607     subb a, r0
608     jz Line2for4Digit    ;jezeli nic nie zostalo wcisniete
609     mov a, r2
610     clr c
611     subb a, #7Eh        ;jezeli zostal naciśnięty klawisz A, skocz
612     jz Line2for4Digit
613     mov a, r2
614     mov dph, #80h
615     mov dpl, a
616     movx a, @dptr       ;teraz w akumulatorze jest kod ascii danej
                           cyfry
617     acall putcharLCD
618     clr c
619     subb a, #30h        ;konwertujemy kod ascii na cyfre
620     add a, r1
621     mov r1, a           ;dodajemy bufor do cyfry jednosci, a
                           nastepnie kopiujemy z powrotem do bufora
622     czekaj14:           ; nie wypuszczaj, dopoki klawisz nie
                           zostanie "odcisniety"
623     mov a, P7
624     anl a, r0
625     clr c
626     subb a, r0
627     jnz czekaj14
628     acall delay
629     jmp Star
630
631     Line2for4Digit:
632     mov r0, #LINE_2
633     mov a, r0
634     mov P5, a
635     mov a, P7
636     anl a, r0
637     mov r2, a
638     clr c
639     subb a, r0
```

```
640     jz Line3for4Digit                ;jezeli nic nie zostalo wcisniete
641     mov a, r2
642     clr c
643     subb a, #0BEh                    ;jezeli zostal naciśnietý klawisz B, skocz
644     jz Line3for4Digit
645     mov a, r2
646     mov dph, #80h
647     mov dpl, a
648     movx a,@dptr                    ;teraz w akumulatorze jest kod ascii danej
        cyfry
649     acall putcharLCD
650     clr c
651     subb a, #30h                    ;konwertujemy kod ascii na cyfre
652     add a, r1
653     mov r1, a                        ;dodajemy bufor do cyfry jednosci, a
        nastepnie kopiujemy z powrotem do bufora
654     czekaj24:                        ; nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisnietý"
655     mov a, P7
656     anl a, r0
657     clr c
658     subb a, r0
659     jnz czekaj24
660     acall delay
661     jmp Star
662
663     Line3for4Digit:
664     mov r0, #LINE_3
665     mov a, r0
666     mov P5, a
667     mov a, P7
668     anl a, r0
669     mov r2, a
670     clr c
671     subb a, r0
672     jz Line4for4Digit                ;jezeli nic nie zostalo wcisniete
673     mov a, r2
674     clr c
675     subb a, #0DEh                    ;jezeli zostal naciśnietý klawisz C, skocz
676     jz Line4for4Digit
677     mov a, r2
678     mov dph, #80h
679     mov dpl, a
680     movx a,@dptr                    ;teraz w akumulatorze jest kod ascii danej
        cyfry
681     acall putcharLCD
682     clr c
683     subb a, #30h                    ;konwertujemy kod ascii na cyfre
684     add a, r1
685     mov r1, a                        ;dodajemy bufor do cyfry jednosci, a
        nastepnie kopiujemy z powrotem do bufora
```

```
686     czekaj34:                ; nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisniety"
687     mov a, P7
688     anl a, r0
689     clr c
690     subb a, r0
691     jnz czekaj34
692     acall delay
693     jmp Star
694
695     Line4for4Digit:          ;TU BEDZIE TROCHE INACZEJ!!!!!!
696     mov r0, #LINE_4
697     mov a, r0
698     mov P5, a
699     mov a, P7
700     anl a, r0
701     mov r2, a
702     clr c
703     subb a, r0
704     jz goback4Digit          ;jezeli nic nie zostalo wcisniete
705     mov a, r2
706     clr c
707     subb a, #0EBh            ;jezeli nie zostal nacisniety klawisz 0,
        skocz
708     jnz goback4Digit
709     mov a, r2
710     mov dph, #80h
711     mov dpl, a
712     movx a, @dptr            ;teraz w akumulatorze jest kod ascii danej
        cyfry
713     acall putcharLCD
714     clr c
715     subb a, #30h              ;konwertujemy kod ascii na cyfre
716     add a, r1
717     mov r1, a                ;dodajemy bufor do cyfry jednosci, a
        nastepnie kopiujemy z powrotem do bufora
718     czekaj44:                ; nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisniety"
719     mov a, P7
720     anl a, r0
721     clr c
722     subb a, r0
723     jnz czekaj44
724     acall delay
725     jmp Star
726
727 goback4Digit:
728     ljmp Line1for4Digit
729
730 Star:
731     ;dwie petle, czekajace na wpisanie znaku star
732     mov r0, #LINE_4
```

```

733     mov a, r0
734     mov P5, a
735     mov a, P7
736     anl a, r0
737     mov r2, a
738     clr c
739     subb a, r0
740     jz Star ;jezeli nic sie nie wpisalo
741     mov a, r2
742     clr c
743     subb a, #0E7h ;wpisane znaku *
744     jnz Star
745     ;jesli wcisnelo sie *, czekaj na jego "odcisniecie"
746     czekajStar: ;nie wypuszczaj, dopoki klawisz nie
        zostanie "odcisniety"
747     mov a, P7
748     anl a, r0
749     clr c
750     subb a, r0
751     jnz czekajStar
752     acall delay
753     ;walidacja minuty
754     mov a, r1
755     clr c
756     subb a, #60
757     jnc minuteValidationIncorrect
758     mov a, r1
759     mov r6, a ;przechowanie wartosci minuty
760     LCDcntlWR #CLEAR ;wywołanie macra LCDcntlWR – czyszczenie
        LCD i przesuniecie kursora na poczatek
761     LCDcntlWR #HOME
762     jmp FINALLY
763
764 minuteValidationIncorrect:
765     ljmp ThirdDigit
766     FINALLY: RET
767
768 // wyznaczanie biezacej wartosci zegara i jego wyswietlanie na LCD
769 ZEGAR: INC R7 ;licznik sekund
770     MOV A, R7 ;obsługa sekund
771     CLR C
772     SUBB A, #60 ;przepelnienie sekund
773     JZ MINUTY
774     LCDcntlWR #HOME ;wyswietlenie calego zegara
775     MOV A, R5 ;godziny
776     ACALL putdigitLCD
777     MOV A, #": " ;separator
778     ACALL putcharLCD
779     MOV A, R6 ;minuty
780     ACALL putdigitLCD
781     MOV A, #": " ;separator
782     ACALL putcharLCD

```

```

783     MOV A, R7      ; sekundy
784     ACALL putdigitLCD
785     JMP FINAL
786 MINUTY:  MOV R7, #00H      ; zerowanie sekund
787     INC R6          ; licznik minut
788     MOV A, R6       ; obsluga minut
789     CLR C
790     SUBB A, #60      ; przepelnienie minut
791     JZ GODZINY
792     LCDcntrlWRR #HOME ; wyswietlenie calego zegara
793     MOV A, R5        ; godziny
794     ACALL putdigitLCD
795     MOV A, #": "    ; separator
796     ACALL putcharLCD
797     MOV A, R6        ; minuty
798     ACALL putdigitLCD
799     MOV A, #": "    ; separator
800     ACALL putcharLCD
801     MOV A, R7        ; sekundy
802     ACALL putdigitLCD
803     JMP FINAL
804 GODZINY:  MOV R6, #00H      ; zerowanie minut
805     INC R5          ; licznik godzin
806     MOV A, R5
807     CLR C
808     SUBB A, #24      ; przepelenienie godzin - doba
809     JNZ EKRAN
810     MOV R5, #00H     ; zerowanie godzin
811 EKRAN:    LCDcntrlWRR #HOME ; wyswietlenie calego zegara
812     MOV A, R5        ; godziny
813     ACALL putdigitLCD
814     MOV A, #": "    ; separator
815     ACALL putcharLCD
816     MOV A, R6        ; minuty
817     ACALL putdigitLCD
818     MOV A, #": "    ; separator
819     ACALL putcharLCD
820     MOV A, R7        ; sekundy
821     ACALL putdigitLCD
822 FINAL:    RET
823
824     ; program glówny
825 START:    init_LCD
826     acall keyascii
827     acall INPUT      ; inicjacja zegara
828     MOV TMOD, #01H   ; konfiguracja timera
829     MOV TH0, #3CH    ; ladowanie
830     MOV TL0, #0B0H   ; stalej timera na 50ms
831     SETB TR0         ; timer start
832     MOV IE, #82H     ; przerwania wlacz
833     MOV R7, #0FFH
834     ACALL ZEGAR      ; wyswietlenie zainicjowanego zegara

```

```
835     MOV A, #0FH
836     MOV P1, A           ; zapalenie diód
837     MOV R0, #20         ; licznik odmierzen 20 x 50ms
838 CZEKAM: MOV A, R0       ; czekam, a timer
839     JNZ CZEKAM          ; mierzy łączny czas 1s
840     MOV R0, #20         ; po zgłoszeniu przerwania — ustawiam na nowo
                        ; licznik odmierzen 20 x 50ms
841     ACALL ZEGAR         ; uruchomienie procedury obsługi i wyświetlenia
                        ; zegara
842     MOV A, P1           ; zmiana
843     CPL A               ; świecenia
844     MOV P1, A           ; diód
845     JMP CZEKAM          ; czekam na kolejna sekunde
846     NOP
847     NOP
848     NOP
849     JMP $
850 END START
```