Maciej Byczko Bartosz Matysiak	Prowadzący: dr inż. Jacek Mazurkiewicz	Numer ćwiczenia 7
Cz 13:15 TN	Temat ćwiczenia: RTC i inne atrakcje	Ocena:
Grupa: B	Data wykonania: 20 Maja 2021	

1 Zadanie 1

1.1 Polecenie

Rozbudować finalną postać programu o mechanizmy kontroli zakresu wpisanych w inicjujący łańcuch ASCII danych. Dopuszczalny zakres dla sekund i minut: od 00 do 59, dla godzin: od 00 do 23, dla miesięcy: od 01 do 12, dla dni: od 01 do 31. Mechanizm kontroli ma działać w zakresie procedury inicjalizacji czasu i daty. W przypadku wykrycia danych spoza wymaganego zakresu inicjalizacja ma wprowadzić minimalne dopuszczalne wartości dla danej pozycji czasu lub daty.

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 0FF2DH        ; adres do podania kodu ASCII na LCD
6
7  RTCxs equ 0FF00H   ; seconds
8  RTCsx equ 0FF01H
9  RTCxm equ 0FF02H   ; minutes
10 RTCmx equ 0FF03H
11 RTCxh equ 0FF04H   ; hours
12 RTChx equ 0FF05H
13 RTCxd equ 0FF06H   ; day
14 RTCdx equ 0FF07H
15 RTCxn equ 0FF08H   ; month
16 RTCnx equ 0FF09H
17 RTCxy equ 0FF0AH   ; year
18 RTCyx equ 0FF0BH
19 RTCdw equ 0FF0CH   ; day of week
20 RTCpf equ 0FF0FH
21
22 // bajty sterujace LCD, inne dostepne w opisie LCD na stronie WWW
23 #define  HOME     0x80     // put cursor to second line
24 #define  INITDISP 0x38     // LCD init (8-bit mode)
25 #define  HOM2     0xc0     // put cursor to second line
26 #define  LCDON    0x0e     // LCD nn, cursor off, blinking off
27 #define  CLEAR    0x01     // LCD display clear
28
29 org 0100H
30   Czas: db "13:40:00"
```

```
31    Dzien: db "20:05:2021*4"
32    Month: db "JanFebMarAprMayJunJulAugSepOctNovDec"
33    Week: db "SunMonTueWedThuFriSat"
34    TwentyH: db 02
35    TwentyL: db 00
36
37  // macro do wprowadzenia bajtu sterujacego na LCD
38  LCDcntrlWR MACRO x          ; x - parametr wywolania macra - bajt
       sterujacy
39          LOCAL loop          ; LOCAL oznacza ze etykieta loop moze sie
               powtórzyc w programie
40  loop: MOV  DPTR,#LCDstatus  ; DPTR zaladowany adresem statusu
41        MOVX A,@DPTR          ; pobranie bajtu z biezacym statusem LCD
42        JB   ACC.7,loop       ; testowanie najstarszego bitu akumulatora
43                              ; - wskazuje gotowosc LCD
44        MOV  DPTR,#LCDcontrol ; DPTR zaladowany adresem do podania bajtu
             sterujacego
45        MOV  A, x             ; do akumulatora trafia argument wywolania
             -macrabajt sterujacy
46        MOVX @DPTR,A          ; bajt sterujacy podany do LCD - zadana
             akcja widoczna na LCD
47        ENDM
48
49  // macro do wypisania znaku ASCII na LCD, znak ASCII przed wywolaniem
       macra ma byc w A
50  LCDcharWR MACRO
51        LOCAL tutu            ; LOCAL oznacza ze etykieta tutu moze sie
             powtórzyc w programie
52        PUSH ACC              ; odlozenie biezacej zawartosci
             akumulatora na stos
53  tutu: MOV  DPTR,#LCDstatus  ; DPTR zaladowany adresem statusu
54        MOVX A,@DPTR          ; pobranie bajtu z biezacym statusem LCD
55        JB   ACC.7,tutu       ; testowanie najstarszego bitu akumulatora
56                              ; - wskazuje gotowosc LCD
57        MOV  DPTR,#LCDdataWR  ; DPTR zaladowany adresem do podania bajtu
             sterujacego
58        POP  ACC              ; w akumulatorze ponownie kod ASCII znaku
             na LCD
59        MOVX @DPTR,A          ; kod ASCII podany do LCD - znak widoczny
             na LCD
60        ENDM
61
62  // macro do inicjalizacji wyswietlacza - bez parametrów
63  init_LCD MACRO
64            LCDcntrlWR #INITDISP ; wywolanie macra LCDcntrlWR -
                 inicjalizacja LCD
65            LCDcntrlWR #CLEAR    ; wywolanie macra LCDcntrlWR -
                 czyszczenie LCD
66            LCDcntrlWR #LCDON    ; wywolanie macra LCDcntrlWR -
                 konfiguracja kursora
67            ENDM
68
```

```
69  // macro do wypisywania polowki wskazania pozycji czasu lub daty
70  disp_nibble MACRO
71    movx A,@DPTR
72    anl A,#0Fh   ; select 4-bits
73    orl A,#30H   ; change to ASCII
74    call putcharLCD
75    ENDM
76
77  // funkcja wypisywania znaku na LCD
78  putcharLCD: LCDcharWR
79        ret
80
81  // wypisywanie czasu
82  disp_time:
83      LCDcntrlWR #HOME
84      mov DPTR,#RTChx ; get hours from RTC (higher nibble)
85      disp_nibble
86      mov DPTR,#RTCxh ; get hours from RTC (lower nibble)
87      disp_nibble
88      mov A,#':'
89      call putcharLCD
90      mov DPTR,#RTCmx ; get minutes from RTC (higher nibble)
91      disp_nibble
92      mov DPTR,#RTCxm ; get minutes from RTC (lower nibble)
93      disp_nibble
94      mov A,#':'
95      call putcharLCD;
96      mov DPTR,#RTCsx ; get seconds from RTC (higher nibble)
97      disp_nibble
98      mov DPTR,#RTCxs ; get seconds from RTC (lower nibble)
99      disp_nibble
100     RET
101
102 // wypisywanie dnia tygodnia slownie
103 week_word:
104     mov DPTR,#RTCdw ; get day of week from RTC
105     movx a, @DPTR
106     anl a, #0FH
107     mov b, #03
108     mul ab
109     mov r7,a
110     mov DPTR,#Week
111     movc a,@a+dptr
112     push dph
113     push dpl
114     acall putcharLCD
115     pop dpl
116     pop dph
117     inc dptr
118     mov a,r7
119     movc a,@a+dptr
120     push dph
```

```
121     push dpl
122     acall putcharLCD
123     pop dpl
124     pop dph
125     inc dptr
126     mov a,r7
127     movc a,@a+dptr
128     acall putcharLCD
129     ret
130 
131 // wypisywanie nazwy miesiaca slownie
132 month_word:
133     mov DPTR,#RTCmx ; get month from RTC (higher nibble)
134     movx a, @DPTR
135     anl a, #0FH
136     mov b, #10
137     mul ab
138     mov r7,a
139     mov DPTR,#RTCmn ; get month from RTC (lower nibble)
140     movx a, @DPTR
141     anl a, #0FH
142     add a,r7
143     clr c
144     subb a, #01
145     mov b, #03
146     mul ab
147     mov r7,a
148     mov DPTR,#Month
149     movc a,@a+dptr
150     push dph
151     push dpl
152     acall putcharLCD
153     pop dpl
154     pop dph
155     inc dptr
156     mov a,r7
157     movc a,@a+dptr
158     push dph
159     push dpl
160     acall putcharLCD
161     pop dpl
162     pop dph
163     inc dptr
164     mov a,r7
165     movc a,@a+dptr
166     acall putcharLCD
167     ret
168 
169 // wypisywanie daty
170 disp_date:
171   LCDcntrlWR #HOM2
172   mov DPTR,#RTCdx ; get day from RTC (higher nibble)
```

```
173   disp_nibble
174   mov DPTR,#RTCxd ; get day from RTC (lower nibble)
175   disp_nibble
176   mov A,#'-'
177   call putcharLCD
178   acall month_word
179   mov A,#'-'
180   call putcharLCD;
181   mov DPTR,#TwentyH
182   disp_nibble
183   mov DPTR,#TwentyL
184   disp_nibble
185   mov DPTR,#RTCyx ; get year from RTC (higher nibble)
186   disp_nibble
187   mov DPTR,#RTCxy ; get year from RTC (lower nibble)
188   disp_nibble
189   mov A,#" "
190   call putcharLCD;
191   acall week_word
192   RET
193
194 hourValidationIncorrect:
195   mov dptr, #RTChx
196   movx @dptr, #0
197   mov dptr, #RTCxh
198   movx @dptr, #0
199 ret
200
201 minuteValidationIncorrect:
202   mov dptr, #RTCmx
203   movx @dptr, #0
204   mov dptr, #RTCxm
205   movx @dptr, #0
206 ret
207
208 secondsValidationIncorrect:
209   mov dptr, #RTCsx
210   movx @dptr, #0
211   mov dptr, #RTCxs
212   movx @dptr, #0
213 ret
214
215 saveValue1:
216   mov r0, a ; zapamietanie świartoci a
217   clr a ;  wyczyszczenie rejestru r1 za pomoca a
218   mov r1, a
219   mov a, r0; przywrocenie wartosci a
220   add a, r1 ; suma z rejestrem r1
221   mov r1, a; zapisanie sumy
222 ret
223
224 saveValue2:
```

```
225   mov r0, a ; zapamietanie świartoci a
226   add a, r1 ; suma z rejestrem r1
227   mov r1, a; zapisanie sumy
228   mov a, r0; przywrocenie wartosci a
229 ret
230
231
232 // inicjalizacja czasu
233 czas_start:
234     mov DPTR, #RTCpf ; 24h zegar
235     movx a, @DPTR
236     orl a, #04H
237     movx @DPTR, a
238     clr c
239     clr a
240     mov dptr, #Czas
241     movc a, @a+dptr ; dziesiatki godzin
242     subb a, #30h
243
244     acall saveValue1
245
246     push dph
247     push dpl
248     mov dptr, #RTChx
249     movx @dptr, a
250     pop dpl
251     pop dph
252     inc dptr
253     clr a
254     movc a, @a+dptr ; jednosci godzin
255     subb a, #30h
256
257     acall saveValue2
258
259     push dph
260     push dpl
261     mov dptr, #RTCxh
262     movx @dptr, a
263
264     mov a, r1
265     clr c
266     subb a, #24
267     jc hourValidationCorrect
268     acall hourValidationIncorrect
269   hourValidationCorrect:
270     pop dpl
271     pop dph
272     inc dptr
273     clr a
274     movc a, @a+dptr ; separator
275     inc dptr
276     clr a
```

```
277     movc a, @a+dptr ; dziesiatki minut
278     subb a, #30h
279 
280     acall saveValue1
281 
282     push dph
283     push dpl
284     mov dptr, #RTCmx
285     movx @dptr, a
286     pop dpl
287     pop dph
288     inc dptr
289     clr a
290     movc a, @a+dptr ; jednosci minut
291     subb a, #30h
292 
293     acall saveValue2
294 
295     push dph
296     push dpl
297     mov dptr, #RTCxm
298     movx @dptr, a
299 
300     mov a, r1
301     clr c
302     subb a, #60
303     jc minuteValidationCorrect
304     acall minuteValidationIncorrect
305 minuteValidationCorrect:
306     pop dpl
307     pop dph
308     inc dptr
309     clr a
310     movc a, @a+dptr ; separator
311     inc dptr
312     clr a
313     movc a, @a+dptr ; dziesiatki sekund
314     subb a, #30h
315 
316     acall saveValue1
317 
318     push dph
319     push dpl
320     mov dptr, #RTCsx
321     movx @dptr, a
322     pop dpl
323     pop dph
324     inc dptr
325     clr a
326     movc a, @a+dptr ; jednosci sekund
327     subb a, #30h
328 
```

```
329     acall saveValue2
330
331     push dph
332     push dpl
333     mov dptr, #RTCxs
334     movx @dptr, a
335
336     mov a, r1
337     clr c
338     subb a, #60
339     jc secondsValidationCorrect
340     acall secondsValidationIncorrect
341   secondsValidationCorrect:
342     pop dpl
343     pop dph
344     ret
345 daysValidationIncorrect:
346   mov dptr, #RTCdx
347   movx @dptr, #0
348   mov dptr, #RTCxd
349   movx @dptr, #1
350   // jezeli poprawiamy miesiac to zapiszmy jego nowo wersje czyli
        styczen
351   mov r0, a
352   mov a, #01
353   mov r2, a
354   mov a, r0
355 ret
356
357 monthsValidationIncorrect:
358   mov dptr, #RTCnx
359   movx @dptr, #0
360   mov dptr, #RTCxn
361   movx @dptr, #1
362   mov r0, a
363   mov a, r1
364   mov r3, a
365   mov a, r0
366 ret
367
368 dayMonthValidation:
369   ; pod r2 kryje sie zapis dni
370   ; pod r3 kryje sie zapis miesiaca
371
372 ret
373
374 saveDays:
375   mov r0, a
376   mov a, r1
377   mov r2, a
378   mov a, r0
379 ret
```

```
380
381 saveMonth:
382   mov r0, a
383   mov a, r1
384   mov r3, a
385   mov a, r0
386 ret
387
388 // inicjalizacja daty
389 data_start: clr c
390     clr a
391     mov dptr, #Dzien
392     movc a, @a+dptr ; dziesiatki dni
393     subb a, #30h
394
395     acall saveValue1
396
397     push dph
398     push dpl
399     mov dptr, #RTCdx
400     movx @dptr, a
401     pop dpl
402     pop dph
403     inc dptr
404     clr a
405     movc a, @a+dptr ; jednosci dni
406     subb a, #30h
407
408     acall saveValue2
409
410     acall saveDays
411
412     push dph
413     push dpl
414     mov dptr, #RTCxd
415     movx @dptr, a
416
417     mov a, r1
418     clr c
419     subb a, #32
420     jc daysValidationCorrect
421     acall daysValidationIncorrect
422   daysValidationCorrect:
423
424     pop dpl
425     pop dph
426     inc dptr
427     clr a
428     movc a, @a+dptr ; separator
429     inc dptr
430     clr a
431     movc a, @a+dptr ; dziesiatki miesiaca
```

```
432     subb a, #30h
433
434     acall saveValue1
435
436     push dph
437     push dpl
438     mov dptr, #RTCnx
439     movx @dptr, a
440     pop dpl
441     pop dph
442     inc dptr
443     clr a
444     movc a, @a+dptr ; jednosci miesiaca
445     subb a, #30h
446
447     acall saveValue2
448
449     acall saveMonth
450
451     push dph
452     push dpl
453     mov dptr, #RTCxn
454     movx @dptr, a
455
456     mov a, r1
457     clr c
458     subb a, #13
459     jc monthsValidationCorrect
460     acall monthsValidationIncorrect
461     jmp noCheck
462   monthsValidationCorrect:
463   acall dayMonthValidation
464 noCheck:
465     pop dpl
466     pop dph
467     inc dptr
468     clr a
469     movc a, @a+dptr ; separator
470     inc dptr
471     clr a
472     movc a, @a+dptr ; cyfra tysiecy roku
473     inc dptr
474     clr a
475     movc a, @a+dptr ; cyfra setek roku
476     inc dptr
477     clr a
478     movc a, @a+dptr ; dziesiatki roku
479     subb a, #30h
480     push dph
481     push dpl
482     mov dptr, #RTCyx
483     movx @dptr, a
```

```
484     pop dpl
485     pop dph
486     inc dptr
487     clr a
488     movc a, @a+dptr ; jednosci roku
489     subb a, #30h
490     push dph
491     push dpl
492     mov dptr, #RTCxy
493     movx @dptr, a
494     pop dpl
495     pop dph
496     inc dptr
497     clr a
498     movc a, @a+dptr ; separator
499     inc dptr
500     clr a
501     movc a, @a+dptr ; dzien tygodnia
502     subb a, #30h
503     push dph
504     push dpl
505     mov dptr, #RTCdw
506     movx @dptr, a
507     pop dpl
508     pop dph
509     ret
510
511
512         ; program główny
513 start:  init_LCD
514
515     acall czas_start
516     acall data_start
517
518
519 czas_plynie:  acall disp_time
520         acall disp_date
521         sjmp czas_plynie
522     NOP
523     NOP
524     NOP
525     JMP $
526 END START
```

2 Zadanie 2

2.1 Polecenie

Zadanie dodatkowe. Opisany powyżej mechanizm kontroli rozbudować o sprawdzanie poprawnej korelacji danej dotyczącej miesiąca i dnia. Innymi słowy dopuszczalny zakres wartości dnia ma uwzględniać maksymalną liczbę dni danego miesiąca.

2.2 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 0FF2DH          ; adres do podania kodu ASCII na LCD
6
7  RTCxs equ 0FF00H   ; seconds
8  RTCsx equ 0FF01H
9  RTCxm equ 0FF02H   ; minutes
10 RTCmx equ 0FF03H
11 RTCxh equ 0FF04H   ; hours
12 RTChx equ 0FF05H
13 RTCxd equ 0FF06H   ; day
14 RTCdx equ 0FF07H
15 RTCxn equ 0FF08H   ; month
16 RTCnx equ 0FF09H
17 RTCxy equ 0FF0AH   ; year
18 RTCyx equ 0FF0BH
19 RTCdw equ 0FF0CH   ; day of week
20 RTCpf equ 0FF0FH
21
22 // bajty sterujace LCD, inne dostepne w opisie LCD na stronie WWW
23 #define  HOME     0x80      // put cursor to second line
24 #define  INITDISP 0x38      // LCD init (8-bit mode)
25 #define  HOM2     0xc0      // put cursor to second line
26 #define  LCDON    0x0e      // LCD nn, cursor off, blinking off
27 #define  CLEAR    0x01      // LCD display clear
28
29 org 0100H
30   Czas: db "13:40:00"
31   Dzien: db "20:05:2021*4"
32   Month: db "JanFebMarAprMayJunJulAugSepOctNovDec"
33   Week: db "SunMonTueWedThuFriSat"
34   TwentyH: db 02
35   TwentyL: db 00
36
37 // macro do wprowadzenia bajtu sterujacego na LCD
38 LCDcntrlWR MACRO x           ; x - parametr wywolania macra - bajt
      sterujacy
39           LOCAL loop         ; LOCAL oznacza ze etykieta loop moze sie
                powtórzyc w programie
40 loop: MOV  DPTR,#LCDstatus   ; DPTR zaladowany adresem statusu
41       MOVX A,@DPTR           ; pobranie bajtu z biezacym statusem LCD
42       JB   ACC.7,loop        ; testowanie najstarszego bitu akumulatora
43                              ; - wskazuje gotowosc LCD
44       MOV  DPTR,#LCDcontrol ; DPTR zaladowany adresem do podania bajtu
            sterujacego
45       MOV  A, x              ; do akumulatora trafia argument wywolania
            -macrabajt sterujacy
```

```
46        MOVX @DPTR,A            ; bajt sterujacy podany do LCD - zadana
              akcja widoczna na LCD
47        ENDM
48
49  // macro do wypisania znaku ASCII na LCD, znak ASCII przed wywolaniem
        macra ma byc w A
50  LCDcharWR MACRO
51        LOCAL tutu              ; LOCAL oznacza ze etykieta tutu moze sie
              powtórzyc w programie
52        PUSH ACC                ; odlozenie biezacej zawartosci
              akumulatora na stos
53  tutu: MOV  DPTR,#LCDstatus    ; DPTR zaladowany adresem statusu
54        MOVX A,@DPTR            ; pobranie bajtu z biezacym statusem LCD
55        JB   ACC.7,tutu         ; testowanie najstarszego bitu akumulatora
56                                ; - wskazuje gotowosc LCD
57        MOV  DPTR,#LCDdataWR    ; DPTR zaladowany adresem do podania bajtu
              sterujacego
58        POP  ACC                ; w akumulatorze ponownie kod ASCII znaku
              na LCD
59        MOVX @DPTR,A            ; kod ASCII podany do LCD - znak widoczny
              na LCD
60        ENDM
61
62  // macro do inicjalizacji wyswietlacza - bez parametrów
63  init_LCD MACRO
64            LCDcntrlWR #INITDISP ; wywolanie macra LCDcntrlWR -
                  inicjalizacja LCD
65            LCDcntrlWR #CLEAR    ; wywolanie macra LCDcntrlWR -
                  czyszczenie LCD
66            LCDcntrlWR #LCDON    ; wywolanie macra LCDcntrlWR -
                  konfiguracja kursora
67            ENDM
68
69  // macro do wypisywania polowki wskazania pozycji czasu lub daty
70  disp_nibble MACRO
71    movx A,@DPTR
72    anl A,#0Fh   ; select 4-bits
73    orl A,#30H   ; change to ASCII
74    call putcharLCD
75    ENDM
76
77  // funkcja wypisywania znaku na LCD
78  putcharLCD: LCDcharWR
79          ret
80
81  // wypisywanie czasu
82  disp_time:
83       LCDcntrlWR #HOME
84       mov DPTR,#RTChx ; get hours from RTC (higher nibble)
85       disp_nibble
86       mov DPTR,#RTCxh ; get hours from RTC (lower nibble)
87       disp_nibble
```

```
88      mov A,#':'
89      call putcharLCD
90      mov DPTR,#RTCmx ; get minutes from RTC (higher nibble)
91      disp_nibble
92      mov DPTR,#RTCxm ; get minutes from RTC (lower nibble)
93      disp_nibble
94      mov A,#':'
95      call putcharLCD;
96      mov DPTR,#RTCsx ; get seconds from RTC (higher nibble)
97      disp_nibble
98      mov DPTR,#RTCxs ; get seconds from RTC (lower nibble)
99      disp_nibble
100     RET
101
102 // wypisywanie dnia tygodnia slownie
103 week_word:
104     mov DPTR,#RTCdw ; get day of week from RTC
105     movx a, @DPTR
106     anl a, #0FH
107     mov b, #03
108     mul ab
109     mov r7,a
110     mov DPTR,#Week
111     movc a,@a+dptr
112     push dph
113     push dpl
114     acall putcharLCD
115     pop dpl
116     pop dph
117     inc dptr
118     mov a,r7
119     movc a,@a+dptr
120     push dph
121     push dpl
122     acall putcharLCD
123     pop dpl
124     pop dph
125     inc dptr
126     mov a,r7
127     movc a,@a+dptr
128     acall putcharLCD
129     ret
130
131 // wypisywanie nazwy miesiaca slownie
132 month_word:
133     mov DPTR,#RTCnx ; get month from RTC (higher nibble)
134     movx a, @DPTR
135     anl a, #0FH
136     mov b, #10
137     mul ab
138     mov r7,a
139     mov DPTR,#RTCxn ; get month from RTC (lower nibble)
```

```
140     movx a, @DPTR
141     anl a, #0FH
142     add a,r7
143     clr c
144     subb a, #01
145     mov b, #03
146     mul ab
147     mov r7,a
148     mov DPTR,#Month
149     movc a,@a+dptr
150     push dph
151     push dpl
152     acall putcharLCD
153     pop dpl
154     pop dph
155     inc dptr
156     mov a,r7
157     movc a,@a+dptr
158     push dph
159     push dpl
160     acall putcharLCD
161     pop dpl
162     pop dph
163     inc dptr
164     mov a,r7
165     movc a,@a+dptr
166     acall putcharLCD
167     ret
168
169 // wypisywanie daty
170 disp_date:
171   LCDcntrlWR #HOM2
172   mov DPTR,#RTCdx ; get day from RTC (higher nibble)
173   disp_nibble
174   mov DPTR,#RTCxd ; get day from RTC (lower nibble)
175   disp_nibble
176   mov A,#'-'
177   call putcharLCD
178   acall month_word
179   mov A,#'-'
180   call putcharLCD;
181   mov DPTR,#TwentyH
182   disp_nibble
183   mov DPTR,#TwentyL
184   disp_nibble
185   mov DPTR,#RTCyx ; get year from RTC (higher nibble)
186   disp_nibble
187   mov DPTR,#RTCxy ; get year from RTC (lower nibble)
188   disp_nibble
189   mov A,#" "
190   call putcharLCD;
191   acall week_word
```

```
192   RET
193
194 // inicjalizacja czasu
195 czas_start:
196     mov DPTR, #RTCpf ; 24h zegar
197     movx a, @DPTR
198     orl a, #04H
199     movx @DPTR, a
200     clr c
201     clr a
202     mov dptr, #Czas
203     movc a, @a+dptr ; dziesiatki godzin
204     subb a, #30h
205     push dph
206     push dpl
207     mov dptr, #RTChx
208     movx @dptr, a
209     pop dpl
210     pop dph
211     inc dptr
212     clr a
213     movc a, @a+dptr ; jednosci godzin
214     subb a, #30h
215     push dph
216     push dpl
217     mov dptr, #RTCxh
218     movx @dptr, a
219     pop dpl
220     pop dph
221     inc dptr
222     clr a
223     movc a, @a+dptr ; separator
224     inc dptr
225     clr a
226     movc a, @a+dptr ; dziesiatki minut
227     subb a, #30h
228     push dph
229     push dpl
230     mov dptr, #RTCmx
231     movx @dptr, a
232     pop dpl
233     pop dph
234     inc dptr
235     clr a
236     movc a, @a+dptr ; jednosci minut
237     subb a, #30h
238     push dph
239     push dpl
240     mov dptr, #RTCxm
241     movx @dptr, a
242     pop dpl
243     pop dph
```

```
244     inc dptr
245     clr a
246     movc a, @a+dptr ; separator
247     inc dptr
248     clr a
249     movc a, @a+dptr ; dziesiatki sekund
250     subb a, #30h
251     push dph
252     push dpl
253     mov dptr, #RTCsx
254     movx @dptr, a
255     pop dpl
256     pop dph
257     inc dptr
258     clr a
259     movc a, @a+dptr ; jednosci sekund
260     subb a, #30h
261     push dph
262     push dpl
263     mov dptr, #RTCxs
264     movx @dptr, a
265     pop dpl
266     pop dph
267     ret
268
269 // inicjalizacja daty
270 data_start: clr c
271     clr a
272     mov dptr, #Dzien
273     movc a, @a+dptr ; dziesiatki dni
274     subb a, #30h
275     push dph
276     push dpl
277     mov dptr, #RTCdx
278     movx @dptr, a
279     pop dpl
280     pop dph
281     inc dptr
282     clr a
283     movc a, @a+dptr ; jednosci dni
284     subb a, #30h
285     push dph
286     push dpl
287     mov dptr, #RTCxd
288     movx @dptr, a
289     pop dpl
290     pop dph
291     inc dptr
292     clr a
293     movc a, @a+dptr ; separator
294     inc dptr
295     clr a
```

```
296     movc a, @a+dptr ; dziesiatki miesiaca
297     subb a, #30h
298     push dph
299     push dpl
300     mov dptr, #RTCnx
301     movx @dptr, a
302     pop dpl
303     pop dph
304     inc dptr
305     clr a
306     movc a, @a+dptr ; jednosci miesiaca
307     subb a, #30h
308     push dph
309     push dpl
310     mov dptr, #RTCxn
311     movx @dptr, a
312     pop dpl
313     pop dph
314     inc dptr
315     clr a
316     movc a, @a+dptr ; separator
317     inc dptr
318     clr a
319     movc a, @a+dptr ; cyfra tysiecy roku
320     inc dptr
321     clr a
322     movc a, @a+dptr ; cyfra setek roku
323     inc dptr
324     clr a
325     movc a, @a+dptr ; dziesiatki roku
326     subb a, #30h
327     push dph
328     push dpl
329     mov dptr, #RTCyx
330     movx @dptr, a
331     pop dpl
332     pop dph
333     inc dptr
334     clr a
335     movc a, @a+dptr ; jednosci roku
336     subb a, #30h
337     push dph
338     push dpl
339     mov dptr, #RTCxy
340     movx @dptr, a
341     pop dpl
342     pop dph
343     inc dptr
344     clr a
345     movc a, @a+dptr ; separator
346     inc dptr
347     clr a
```

```
348     movc a, @a+dptr ; dzien tygodnia
349     subb a, #30h
350     push dph
351     push dpl
352     mov dptr, #RTCdw
353     movx @dptr, a
354     pop dpl
355     pop dph
356     ret
357
358
359         ; program główny
360 start:  init_LCD
361
362     acall czas_start
363     acall data_start
364
365
366 czas_plynie:  acall disp_time
367         acall disp_date
368         sjmp czas_plynie
369     NOP
370     NOP
371     NOP
372     JMP $
373 END START
```