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 oraz Zadanie 2

1.1 Polecenie

1.1.1 Zadanie 1

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.

1.1.2 Zadanie 2

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.

1.2 Rozwiązanie pełne

```
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 "13:70:70"
31   Dzien:  db "07:02:2021*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     clr c
205     subb a, #30h  ; konwersja ascii->liczba
206
207     mov r2, a     ;zapisz cyfre dziesiatek w r2
208     mov b, #10
209     mul ab       ;pomnoz cyfre dziesiatek przez 10...
210     mov r1, a     ;...i odloz wynik do r1
211
212     inc dptr     ;przesun dptr na kolejny adres w stringu "Czas"
213     clr a
214     movc a, @a+dptr ; jednosci godzin
215     clr c
216     subb a, #30h    ; konwersja ascii->liczba
217
218     mov r3, a     ;zapisz cyfre jednosci w r3
219
```

```
220     clr c
221     addc a, r1      ;w akumulatorze jest teraz "cala" liczba godzin
222
223     clr c
224     subb a, #24
225     jnc godzinyPozaZakresem
226
227     mov a, r2
228     push dph     ;zapisanie dptr  (wskazuje teraz na jednostki godzin!)
            na stosie
229     push dpl
230     mov dptr, #RTChx ;dptr=adres na rejestr
231     movx @dptr, a ;zaladuj rejestr zawartoscia wyjeta ze stringu "Czas
            "
232
233     mov a, r3
234     mov dptr, #RTCxh
235     movx @dptr, a
236     pop dpl
237     pop dph
238
239     jmp koniecGodzinyPozaZakresem
240     godzinyPozaZakresem:
241
242     mov a, #00h  ;ladujemy minimalna godzine
243     push dph     ;zapisanie dptr  (wskazuje teraz na jednostki godzin!)
            na stosie
244     push dpl
245     mov dptr, #RTChx ;dptr=adres na rejestr
246     movx @dptr, a ;zaladuj rejestr zawartoscia wyjeta ze stringu "Czas
            "
247
248     mov a, #00h ;ladujemy minimalna godzine
249     mov dptr, #RTCxh
250     movx @dptr, a
251     pop dpl
252     pop dph
253
254     koniecGodzinyPozaZakresem:
255     inc dptr
256     clr a
257     movc a, @a+dptr ; separator
258     inc dptr    ;teraz dptr pokazuje na dziesiatki minut
259
260     clr a
261     movc a, @a+dptr ; dziesiatki minut
262     clr c
263     subb a, #30h
264
265     mov r2, a     ;zapisz cyfre dziesiatek w r2
266     mov b, #10
267     mul ab       ;pomnoz cyfre dziesiatek przez 10...
```

```
268     mov r1, a     ;...i odloz wynik do r1
269
270     inc dptr     ;przesun dptr na kolejny adres w stringu "Czas"
271     clr a
272     movc a, @a+dptr ; jednosci minut
273     clr c
274     subb a, #30h    ; konwersja ascii->liczba
275
276     mov r3, a     ;zapisz cyfre jednosci w r3
277
278     clr c
279     addc a, r1     ;w akumulatorze jest teraz "cala" liczba minut
280
281     clr c
282     subb a, #60
283     jnc minutyPozaZakresem
284
285     mov a, r2
286     push dph  ;zapisanie dptr  (wskazuje teraz na jednostki minut!) na
            stosie
287     push dpl
288     mov dptr, #RTCmx
289     movx @dptr, a
290
291     mov a, r3
292     mov dptr, #RTCxm
293     movx @dptr, a
294     pop dpl
295     pop dph
296
297     jmp koniecMinutyPozaZakresem
298     minutyPozaZakresem:
299     mov a, #00h
300     push dph  ;zapisanie dptr  (wskazuje teraz na jednostki minut!) na
            stosie
301     push dpl
302     mov dptr, #RTCmx
303     movx @dptr, a
304
305     mov a, #00h
306     mov dptr, #RTCxm
307     movx @dptr, a
308     pop dpl
309     pop dph
310
311     koniecMinutyPozaZakresem:
312     inc dptr
313     clr a
314     movc a, @a+dptr ; separator
315     inc dptr     ; dptr pokazuje teraz na dziesiatki sekund
316
317     clr a
```

```
318     movc a, @a+dptr ; dziesiatki sekund
319     clr c
320     subb a, #30h
321
322     mov r2, a     ;zapisz cyfre dziesiatek w r2
323     mov b, #10
324     mul ab        ;pomnoz cyfre dziesiatek przez 10...
325     mov r1, a     ;...i odloz wynik do r1
326
327     inc dptr      ;przesun dptr na kolejny adres w stringu "Czas"
328     clr a
329     movc a, @a+dptr ; jednosci godzin
330     clr c
331     subb a, #30h    ; konwersja ascii->liczba
332
333     mov r3, a     ;zapisz cyfre jednosci w r3
334
335     clr c
336     addc a, r1     ;w akumulatorze jest teraz "cala" liczba sekund
337
338     clr c
339     subb a, #60
340     jnc sekundyPozaZakresem
341
342     mov a, r2
343     push dph
344     push dpl
345     mov dptr, #RTCsx
346     movx @dptr, a
347
348     mov a, r3
349     mov dptr, #RTCxs
350     movx @dptr, a
351     pop dpl
352     pop dph
353
354     jmp koniecSekundyPozaZakresem
355     sekundyPozaZakresem:
356
357     mov a, #00h   ;ladujemy minimalna godzine
358     push dph      ;zapisanie dptr  (wskazuje teraz na jednostki godzin!)
            na stosie
359     push dpl
360     mov dptr, #RTChx ;dptr=adres na rejestr
361     movx @dptr, a ;zaladuj rejestr zawartoscia wyjeta ze stringu "Czas
            "
362
363     mov a, #00h ;ladujemy minimalna godzine
364     mov dptr, #RTCxh
365     movx @dptr, a
366     pop dpl
367     pop dph
```

```
368
369     koniecSekundyPozaZakresem:
370     ret
371
372 // inicjalizacja daty
373 data_start:
374     clr c
375     clr a
376     mov dptr, #Dzien
377     movc a, @a+dptr ; dziesiatki dni
378     clr c
379     subb a, #30h
380
381     mov r2, a     ;zapisz cyfre dziesiatek w r2
382     mov b, #10
383     mul ab        ;pomnoz cyfre dziesiatek przez 10...
384     mov r1, a     ;...i odloz wynik do r1
385
386     inc dptr      ;przesun dptr na kolejny adres w stringu "Dzien"
387     clr a
388     movc a, @a+dptr ; jednosci dni
389     clr c
390     subb a, #30h    ; konwersja ascii->liczba
391
392     mov r3, a     ;zapisz cyfre jednosci w r3
393
394     clr c
395     addc a, r1     ;w akumulatorze jest teraz "cala" liczba dni
396
397     mov r0, a     ;zapisujemy dodatkowo liczbe dni, na potrzeby
        dodatkowych testow z numerem miesiaca
398
399     clr c
400     subb a, #01    ; zmiejszamy liczbe dni o 1, by uzyskac zakres
        <0;30> zamiast <1;31>
401     jc dniPozaZakresem
402
403     clr c
404     subb a, #31
405     jnc dniPozaZakresem
406
407     jmp koniecDniPozaZakresem
408     dniPozaZakresem:
409
410     mov a, #00h
411     push dph
412     push dpl
413     mov dptr, #RTCdx
414     movx @dptr, a
415
416     mov a, #01h
417     mov dptr, #RTCxd
```

```
418     movx @dptr, a
419     pop dpl
420     pop dph
421
422 //-------------------------------------------------------------------------
423
424     //ANALIZA MIESIECY, GDY dzien okazal sie poza zakresem
425     inc dptr
426     clr a
427     movc a, @a+dptr ; separator
428     inc dptr
429
430     clr a
431     movc a, @a+dptr ; dziesiatki miesiaca
432     clr c
433     subb a, #30h
434
435     mov r2, a     ;zapisz cyfre dziesiatek w r2
436     mov b, #10
437     mul ab        ;pomnoz cyfre dziesiatek przez 10...
438     mov r1, a     ;...i odloz wynik do r1
439
440     inc dptr      ;przesun dptr na kolejny adres w stringu "Dzien"
441     clr a
442     movc a, @a+dptr ; jednosci miesiaca
443     clr c
444     subb a, #30h    ; konwersja ascii->liczba
445
446     mov r3, a     ;zapisz cyfre jednosci w r3
447
448     clr c
449     addc a, r1     ;w akumulatorze jest teraz "cala" liczba miesiecy
450
451     clr c
452     subb a, #01    ; zmiejszamy liczbe dni o 1, by uzyskac zakres
          <0;11> zamiast <1;12>
453     jc miesiacePozaZakresemPrzyDniuPozaZakresem
454
455     clr c
456     subb a, #12
457     jnc miesiacePozaZakresemPrzyDniuPozaZakresem
458
459     mov a, r2
460     push dph
461     push dpl
462     mov dptr, #RTCnx
463     movx @dptr, a
464
465     mov a, r3
466     mov dptr, #RTCxn
467     movx @dptr, a
468     pop dpl
```

```
469     pop dph
470
471     jmp koniecMiesiacePozaZakresem
472     miesiacePozaZakresemPrzyDniuPozaZakresem:
473
474     mov a, #00h
475     push dph
476     push dpl
477     mov dptr, #RTCnx
478     movx @dptr, a
479
480     mov a, #01h
481     mov dptr, #RTCxn
482     movx @dptr, a
483     pop dpl
484     pop dph
485
486     jmp koniecMiesiacePozaZakresem
487
488 //-----------------------------------------------------------------------
489
490     koniecDniPozaZakresem:
491     inc dptr
492     clr a
493     movc a, @a+dptr ; separator
494     inc dptr
495
496     clr a
497     movc a, @a+dptr ; dziesiatki miesiaca
498     clr c
499     subb a, #30h
500
501     mov r4, a     ;zapisz cyfre dziesiatek w r4
502     mov b, #10
503     mul ab       ;pomnoz cyfre dziesiatek przez 10...
504     mov r1, a     ;...i odloz wynik do r1
505
506     inc dptr     ;przesun dptr na kolejny adres w stringu "Dzien"
507     clr a
508     movc a, @a+dptr ; jednosci miesiaca
509     clr c
510     subb a, #30h    ; konwersja ascii->liczba
511
512     mov r5, a     ;zapisz cyfre jednosci w r5
513
514     clr c
515     addc a, r1     ;w akumulatorze jest teraz "cala" liczba miesiecy
516     mov r1, a     ;odloz cala liczbe miesiecy do akumulatora (dodatkowy
        backup)
517
518     clr c
```

```
519     subb a, #01    ; zmiejszamy liczbe dni o 1, by uzyskac zakres
            <0;11> zamiast <1;12>
520     jc misc
521
522     clr c
523     subb a, #12
524     jnc misc
525     jc omit
526
527     misc:
528     ljmp miesiacePozaZakresem
529
530     omit:
531
532     mov a, r1   ;przywrocenie wartosci miesiecy
533
534     clr c
535     subb a, #02
536     jz przypadekLuty
537
538     mov a, r1   ;przywrocenie wartosci miesiecy
539
540     clr c
541     subb a, #04
542     jz przypadek30DniowyMiesiac
543
544     mov a, r1   ;przywrocenie wartosci miesiecy
545
546     clr c
547     subb a, #06
548     jz przypadek30DniowyMiesiac
549
550     mov a, r1   ;przywrocenie wartosci miesiecy
551
552     clr c
553     subb a, #09
554     jz przypadek30DniowyMiesiac
555
556     mov a, r1   ;przywrocenie wartosci miesiecy
557
558     clr c
559     subb a, #11
560     jz przypadek30DniowyMiesiac
561
562     //przypadek 31dniowego miesiaca
563     mov a, r4
564     push dph
565     push dpl
566     mov dptr, #RTCnx
567     movx @dptr, a
568
569     mov a, r5
```

```
570     mov dptr, #RTCxn
571     movx @dptr, a
572
573     mov a, r2
574     mov dptr, #RTCdx
575     movx @dptr, a
576
577     mov a, r3
578     mov dptr, #RTCxd
579     movx @dptr, a
580     pop dpl
581     pop dph
582
583     jmp koniecMiesiacePozaZakresem
584
585     przypadek30DniowyMiesiac:
586     mov a, r0    ;przywracamy wartosc dni
587     clr c
588     subb a, #31
589     jnc miesiacOKAleDzienNiedobry
590
591     mov a, r4
592     push dph
593     push dpl
594     mov dptr, #RTCnx
595     movx @dptr, a
596
597     mov a, r5
598     mov dptr, #RTCxn
599     movx @dptr, a
600
601     mov a, r2
602     mov dptr, #RTCdx
603     movx @dptr, a
604
605     mov a, r3
606     mov dptr, #RTCxd
607     movx @dptr, a
608     pop dpl
609     pop dph
610
611     jmp koniecMiesiacePozaZakresem
612
613     miesiacOKAleDzienNiedobry:
614
615     mov a, r4
616     push dph
617     push dpl
618     mov dptr, #RTCnx
619     movx @dptr, a
620
621     mov a, r5
```

```
622     mov dptr, #RTCxn
623     movx @dptr, a
624 
625     mov a, #00h
626     mov dptr, #RTCdx
627     movx @dptr, a
628 
629     mov a, #01h
630     mov dptr, #RTCxd
631     movx @dptr, a
632     pop dpl
633     pop dph
634 
635     jmp koniecMiesiacePozaZakresem
636 
637     przypadekLuty:
638     mov a, r0    ;przywracamy wartosc dni
639     clr c
640     subb a, #29
641     jnc lutyOKAleDzienNiedobry
642 
643     mov a, #00h
644     push dph
645     push dpl
646     mov dptr, #RTCnx
647     movx @dptr, a
648 
649     mov a, #02h
650     mov dptr, #RTCxn
651     movx @dptr, a
652 
653     mov a, r2
654     mov dptr, #RTCdx
655     movx @dptr, a
656 
657     mov a, r3
658     mov dptr, #RTCxd
659     movx @dptr, a
660     pop dpl
661     pop dph
662 
663     jmp koniecMiesiacePozaZakresem
664 
665     lutyOKAleDzienNiedobry:
666 
667     mov a, #00h
668     push dph
669     push dpl
670     mov dptr, #RTCnx
671     movx @dptr, a
672 
673     mov a, #02h
```

```
674     mov dptr, #RTCxn
675     movx @dptr, a
676 
677     mov a, #00h
678     mov dptr, #RTCdx
679     movx @dptr, a
680 
681     mov a, #01h
682     mov dptr, #RTCxd
683     movx @dptr, a
684     pop dpl
685     pop dph
686 
687     jmp koniecMiesiacePozaZakresem
688 
689     miesiacePozaZakresem:
690 
691     mov a, #00h
692     push dph
693     push dpl
694     mov dptr, #RTCnx
695     movx @dptr, a
696 
697     mov a, #01h
698     mov dptr, #RTCxn
699     movx @dptr, a
700 
701     mov a, r2
702     mov dptr, #RTCdx
703     movx @dptr, a
704 
705     mov a, r3
706     mov dptr, #RTCxd
707     movx @dptr, a
708     pop dpl
709     pop dph
710 
711     koniecMiesiacePozaZakresem:
712     inc dptr
713     clr a
714     movc a, @a+dptr ; separator
715     inc dptr
716     clr a
717     movc a, @a+dptr ; cyfra tysiecy roku
718     inc dptr
719     clr a
720     movc a, @a+dptr ; cyfra setek roku
721     inc dptr
722     clr a
723     movc a, @a+dptr ; dziesiatki roku
724     clr c
725     subb a, #30h
```

```
726     push dph
727     push dpl
728     mov dptr, #RTCyx
729     movx @dptr, a
730     pop dpl
731     pop dph
732     inc dptr
733     clr a
734     movc a, @a+dptr ; jednosci roku
735     clr c
736     subb a, #30h
737     push dph
738     push dpl
739     mov dptr, #RTCxy
740     movx @dptr, a
741     pop dpl
742     pop dph
743     inc dptr
744     clr a
745     movc a, @a+dptr ; separator
746     inc dptr
747     clr a
748     movc a, @a+dptr ; dzien tygodnia
749     clr c
750     subb a, #30h
751     push dph
752     push dpl
753     mov dptr, #RTCdw
754     movx @dptr, a
755     pop dpl
756     pop dph
757     ret
758
759     ; program główny
760 start:  init_LCD
761
762     acall czas_start
763     acall data_start
764
765 czas_plynie:  acall disp_time
766         acall disp_date
767         sjmp czas_plynie
768     NOP
769     NOP
770     NOP
771     JMP $
772 END START
```