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

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

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

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

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

```
470     pop dpl
471     pop dph
472
473     jmp koniecMiesiacePozaZakresem
474     miesiacePozaZakresemPrzyDniuPozaZakresem:
475
476     mov a, #00h
477     push dph
478     push dpl
479     mov dptr, #RTCnx
480     movx @dptr, a
481
482     mov a, #01h
483     mov dptr, #RTCxn
484     movx @dptr, a
485     pop dpl
486     pop dph
487
488     jmp koniecMiesiacePozaZakresem
489
490 //-----------------------------------------------------------------------
491
492
493     koniecDniPozaZakresem:
494     inc dptr
495     clr a
496     movc a, @a+dptr ; separator
497     inc dptr
498
499     clr a
500     movc a, @a+dptr ; dziesiatki miesiaca
501     clr c
502     subb a, #30h
503
504     mov r4, a     ;zapisz cyfre dziesiatek w r4
505     mov b, #10
506     mul ab       ;pomnoz cyfre dziesiatek przez 10...
507     mov r1, a     ;...i odloz wynik do r1
508
509     inc dptr     ;przesun dptr na kolejny adres w stringu "Dzien"
510     clr a
511     movc a, @a+dptr ; jednosci miesiaca
512     clr c
513     subb a, #30h    ; konwersja ascii->liczba
514
515     mov r5, a     ;zapisz cyfre jednosci w r5
516
517     clr c
518     addc a, r1     ;w akumulatorze jest teraz "cala" liczba miesiecy
519     mov r1, a     ;odloz cala liczbe miesiecy do akumulatora (dodatkowy
        backup)
520
```

```
521     clr c
522     subb a, #01    ; zmiejszamy liczbe dni o 1, by uzyskac zakres
           <0;11> zamiast <1;12>
523     jc misc
524 
525     clr c
526     subb a, #12
527     jnc misc
528     jc omit
529 
530     misc:
531     ljmp miesiacePozaZakresem
532 
533     omit:
534 
535     mov a, r1   ;przywrocenie wartosci miesiecy
536 
537     clr c
538     subb a, #02
539     jz przypadekLuty
540 
541     mov a, r1   ;przywrocenie wartosci miesiecy
542 
543     clr c
544     subb a, #04
545     jz przypadek30DniowyMiesiac
546 
547     mov a, r1   ;przywrocenie wartosci miesiecy
548 
549     clr c
550     subb a, #06
551     jz przypadek30DniowyMiesiac
552 
553     mov a, r1   ;przywrocenie wartosci miesiecy
554 
555     clr c
556     subb a, #09
557     jz przypadek30DniowyMiesiac
558 
559     mov a, r1   ;przywrocenie wartosci miesiecy
560 
561     clr c
562     subb a, #11
563     jz przypadek30DniowyMiesiac
564 
565     //przypadek 31dniowego miesiaca
566     mov a, r4
567     push dph
568     push dpl
569     mov dptr, #RTCnx
570     movx @dptr, a
571 
```

```
572     mov a, r5
573     mov dptr, #RTCxn
574     movx @dptr, a
575 
576     mov a, r2
577     mov dptr, #RTCdx
578     movx @dptr, a
579 
580     mov a, r3
581     mov dptr, #RTCxd
582     movx @dptr, a
583     pop dpl
584     pop dph
585 
586     jmp koniecMiesiacePozaZakresem
587 
588     przypadek30DniowyMiesiac:
589     mov a, r0    ;przywracamy wartosc dni
590     clr c
591     subb a, #31
592     jnc miesiacOKAleDzienNiedobry
593 
594     mov a, r4
595     push dph
596     push dpl
597     mov dptr, #RTCnx
598     movx @dptr, a
599 
600     mov a, r5
601     mov dptr, #RTCxn
602     movx @dptr, a
603 
604     mov a, r2
605     mov dptr, #RTCdx
606     movx @dptr, a
607 
608     mov a, r3
609     mov dptr, #RTCxd
610     movx @dptr, a
611     pop dpl
612     pop dph
613 
614     jmp koniecMiesiacePozaZakresem
615 
616     miesiacOKAleDzienNiedobry:
617 
618     mov a, r4
619     push dph
620     push dpl
621     mov dptr, #RTCnx
622     movx @dptr, a
623 
```

```
624     mov a, r5
625     mov dptr, #RTCxn
626     movx @dptr, a
627 
628     mov a, #00h
629     mov dptr, #RTCdx
630     movx @dptr, a
631 
632     mov a, #01h
633     mov dptr, #RTCxd
634     movx @dptr, a
635     pop dpl
636     pop dph
637 
638     jmp koniecMiesiacePozaZakresem
639 
640     przypadekLuty:
641     mov a, r0    ;przywracamy wartosc dni
642     clr c
643     subb a, #29
644     jnc lutyOKAleDzienNiedobry
645 
646     mov a, #00h
647     push dph
648     push dpl
649     mov dptr, #RTCnx
650     movx @dptr, a
651 
652     mov a, #02h
653     mov dptr, #RTCxn
654     movx @dptr, a
655 
656     mov a, r2
657     mov dptr, #RTCdx
658     movx @dptr, a
659 
660     mov a, r3
661     mov dptr, #RTCxd
662     movx @dptr, a
663     pop dpl
664     pop dph
665 
666     jmp koniecMiesiacePozaZakresem
667 
668     lutyOKAleDzienNiedobry:
669 
670     mov a, #00h
671     push dph
672     push dpl
673     mov dptr, #RTCnx
674     movx @dptr, a
675 
```

```
676     mov a, #02h
677     mov dptr, #RTCxn
678     movx @dptr, a
679
680     mov a, #00h
681     mov dptr, #RTCdx
682     movx @dptr, a
683
684     mov a, #01h
685     mov dptr, #RTCxd
686     movx @dptr, a
687     pop dpl
688     pop dph
689
690     jmp koniecMiesiacePozaZakresem
691
692     miesiacePozaZakresem:
693
694     mov a, #00h
695     push dph
696     push dpl
697     mov dptr, #RTCnx
698     movx @dptr, a
699
700     mov a, #01h
701     mov dptr, #RTCxn
702     movx @dptr, a
703
704     mov a, r2
705     mov dptr, #RTCdx
706     movx @dptr, a
707
708     mov a, r3
709     mov dptr, #RTCxd
710     movx @dptr, a
711     pop dpl
712     pop dph
713
714     ;jmp koniecMiesiacePozaZakresem ;nadmiarowe, ale na wszelki
          wypadek, gdyby zaszlo kopiowanie tego fragmentu
715     koniecMiesiacePozaZakresem:
716     inc dptr
717     clr a
718     movc a, @a+dptr ; separator
719     inc dptr
720     clr a
721     movc a, @a+dptr ; cyfra tysiecy roku
722     inc dptr
723     clr a
724     movc a, @a+dptr ; cyfra setek roku
725     inc dptr
726     clr a
```

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