

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  Dzień: 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  LCDcntlWR MACRO x                ; x - parametr wywołania macra - bajt
    sterujacy
39      LOCAL loop                  ; LOCAL oznacza ze etykieta loop moze sie
        powtorzyc 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 wywołania
        -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 wywołaniem
    macra ma byc w A
50  LCDcharWR MACRO
51      LOCAL tutu                  ; LOCAL oznacza ze etykieta tutu moze sie
        powtorzyc 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      LCDcntlWR #INITDISP        ; wywołanie macra LCDcntlWR -
        inicjalizacja LCD
65      LCDcntlWR #CLEAR           ; wywołanie macra LCDcntlWR -
        czyszczenie LCD
66      LCDcntlWR #LCDON           ; wywołanie macra LCDcntlWR -
        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     a call 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     a call 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 b, #10
217     mov r0, a ; zapamietanie swartoci a
218     mul ab
219     mov r1, a ; wczytanie wartosci a do r1
220     mov a, r0; przywrocenie wartosci a
221
222     ret
223
224 saveValue2:
```

```
225     mov r0, a ; zapamietanie swartoci 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
238     movx @DPTR, a
239     clr c
240     clr a
241     mov dptr, #Czas
242     movc a, @a+dptr ; dziesiatki godzin
243     subb a, #30h
244
245     acall saveValue1
246
247     push dph
248     push dpl
249     mov dptr, #RTCChx
250     movx @dptr, a
251     pop dpl
252     pop dph
253     inc dptr
254     clr a
255     movc a, @a+dptr ; jednosci godzin
256     subb a, #30h
257
258     acall saveValue2
259
260     push dph
261     push dpl
262     mov dptr, #RTCxh
263     movx @dptr, a
264
265     mov a, r1
266     clr c
267     subb a, #24
268     jc hourValidationCorrect
269     acall hourValidationIncorrect
270 hourValidationCorrect:
271
272     pop dpl
273     pop dph
274     inc dptr
275     clr a
276     movc a, @a+dptr ; separator
```

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

```
329     subb a, #30h
330
331     acall saveValue2
332
333     push dph
334     push dpl
335     mov dptr, #RTCxs
336     movx @dptr, a
337
338     mov a, r1
339     clr c
340     subb a, #60
341     jc secondsValidationCorrect
342     acall secondsValidationIncorrect
343 secondsValidationCorrect:
344     pop dpl
345     pop dph
346     ret
347 daysValidationIncorrect:
348     mov dptr, #RTCdx
349     movx @dptr, #0
350     mov dptr, #RTCxd
351     movx @dptr, #1
352     // jezeli poprawiamy miesiac to zapiszmy jego nowo wersje czyli
353     // styczen
354     mov r0, a
355     mov a, #01
356     mov r2, a
357     mov a, r0
358     ret
359 monthsValidationIncorrect:
360     mov dptr, #RTCnx
361     movx @dptr, #0
362     mov dptr, #RTCxn
363     movx @dptr, #1
364     mov r0, a
365     mov a, r1
366     mov r3, a
367     mov a, r0
368     ret
369
370 dayMonthValidation:
371     ; pod r2 kryje sie zapis dni
372     ; pod r3 kryje sie zapis miesiaca
373     mov a, r3
374     clr c
375     subb a, #01
376     jz month31
377     mov a, r3
378     clr c
379     subb a, #02
```



```
380     jz month28
381     mov a, r3
382     clr c
383     subb a, #03
384     jz month31
385     mov a, r3
386     clr c
387     subb a, #04
388     jz month30
389     mov a, r3
390     clr c
391     subb a, #05
392     jz month31
393     mov a, r3
394     clr c
395     subb a, #06
396     jz month30
397     mov a, r3
398     clr c
399     subb a, #07
400     jz month31
401     mov a, r3
402     clr c
403     subb a, #08
404     jz month31
405     mov a, r3
406     clr c
407     subb a, #09
408     jz month30
409     mov a, r3
410     clr c
411     subb a, #10
412     jz month31
413     mov a, r3
414     clr c
415     subb a, #11
416     jz month30
417     mov a, r3
418     clr c
419     subb a, #12
420     jz month31
421     ret
422
423 month31:
424     mov a, r3
425     clr c
426     subb a, #32
427     jz acall monthsValidationIncorrect
428     jmp noCheck
429 month30:
430     mov a, r3
431     clr c
```

```
432     subb a, #31
433     jz acall monthsValidationIncorrect
434     jmp noCheck
435 ret
436 month28:
437     mov a, r3
438     clr c
439     subb a, #29
440     jz acall monthsValidationIncorrect
441     jmp noCheck
442 ret
443
444 saveDays:
445     mov r0, a
446     mov a, r1
447     mov r2, a
448     mov a, r0
449 ret
450
451 saveMonth:
452     mov r0, a
453     mov a, r1
454     mov r3, a
455     mov a, r0
456 ret
457
458 // inicjalizacja daty
459 data_start: clr c
460     clr a
461     mov dptr, #Dzien
462     movc a, @a+dptr ; dziesiatki dni
463     subb a, #30h
464
465     acall saveValue1
466
467     push dph
468     push dpl
469     mov dptr, #RTCdx
470     movx @dptr, a
471     pop dpl
472     pop dph
473     inc dptr
474     clr a
475     movc a, @a+dptr ; jednosci dni
476     subb a, #30h
477
478     acall saveValue2
479
480     acall saveDays
481
482     push dph
483     push dpl
```

```
484     mov dptr, #RTCxd
485     movx @dptr, a
486
487     mov a, r1
488     clr c
489     subb a, #32
490     jc daysValidationCorrect
491     acall daysValidationIncorrect
492 daysValidationCorrect:
493
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 ; dziesiątki miesiąca
502     subb a, #30h
503
504     acall saveValue1
505
506     push dph
507     push dpl
508     mov dptr, #RTCnx
509     movx @dptr, a
510     pop dpl
511     pop dph
512     inc dptr
513     clr a
514     movc a, @a+dptr ; jedności miesiąca
515     subb a, #30h
516
517     acall saveValue2
518
519     acall saveMonth
520
521     push dph
522     push dpl
523     mov dptr, #RTCxn
524     movx @dptr, a
525
526     mov a, r1
527     clr c
528     subb a, #13
529     jc monthsValidationCorrect
530     acall monthsValidationIncorrect
531     jmp noCheck
532 monthsValidationCorrect:
533     acall dayMonthValidation
534 noCheck:
535     pop dpl
```

```
536     pop dph
537     inc dptr
538     clr a
539     movc a, @a+dptr ; separator
540     inc dptr
541     clr a
542     movc a, @a+dptr ; cyfra tysiecy roku
543     inc dptr
544     clr a
545     movc a, @a+dptr ; cyfra setek roku
546     inc dptr
547     clr a
548     movc a, @a+dptr ; dziesiatki roku
549     subb a, #30h
550     push dph
551     push dpl
552     mov dptr, #RTCyx
553     movx @dptr, a
554     pop dpl
555     pop dph
556     inc dptr
557     clr a
558     movc a, @a+dptr ; jednosci roku
559     subb a, #30h
560     push dph
561     push dpl
562     mov dptr, #RTCxy
563     movx @dptr, a
564     pop dpl
565     pop dph
566     inc dptr
567     clr a
568     movc a, @a+dptr ; separator
569     inc dptr
570     clr a
571     movc a, @a+dptr ; dzien tygodnia
572     subb a, #30h
573     push dph
574     push dpl
575     mov dptr, #RTCdw
576     movx @dptr, a
577     pop dpl
578     pop dph
579     ret
580
581
582     ; program glówny
583 start:  init_LCD
584
585     acall czas_start
586     acall data_start
587
```

```

588
589 czas_plynie:    acall disp_time
590                acall disp_date
591                sjmp  czas_plynie
592                NOP
593                NOP
594                NOP
595                JMP $
596 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 Dzień: 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 wywołania macra – bajt
    sterujacy
39     LOCAL loop ; LOCAL oznacza ze etykieta loop moze sie
        powtorzyc 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 wywołania
        –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 wywołaniem
    macra ma byc w A
50 LCDcharWR MACRO
51     LOCAL tutu ; LOCAL oznacza ze etykieta tutu moze sie
        powtorzyc 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 ; wywołanie macra LCDcntrlWR –
        inicjalizacja LCD
65     LCDcntrlWR #CLEAR ; wywołanie macra LCDcntrlWR –
        czyszczenie LCD
66     LCDcntrlWR #LCDON ; wywołanie 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,#RTCyh ; 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
```