```
 1
 2
 3                ;* ------------------------------------------------------------------------ *
 4                ;* Name   : GAWGETIM
 5                ;* Author : Gerard Wassink
 6                ;* Date   : December 25, 2021
 7                ;* Purpose: Get RTC time on the RC2014 CP/M computer
 8                ;* Versions:
 9                ;*   0.1  : Initial code base, and 1st ASM program on CP/M
10                ;*   0.2  : Code cleanup and optimisation
11                ;*
12                ;* ------------------------------------------------------------------------ *
13                ;
14                ;* ------------------------------------------------------------------------ *
15                ;*             GNU LICENSE CONDITIONS
16                ;* ------------------------------------------------------------------------ *
17                ;* This program is free software; you can redistribute it and/or modify
18                ;* it under the terms of the GNU General Public License as published by
19                ;* the Free Software Foundation; either version 2 of the License, or
20                ;* (at your option) any later version.
21                ;*
22                ;* This program is distributed in the hope that it will be useful,
23                ;* but WITHOUT ANY WARRANTY; without even the implied warranty of
24                ;* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
25                ;* GNU General Public License for more details.
26                ;*
27                ;* You should have received a copy of the GNU General Public License along
28                ;* with this program; if not, write to the Free Software Foundation, Inc.,
29                ;* 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
30                ;*
31                ;* ------------------------------------------------------------------------ *
32                ;*       Copyright (C) December 2021 Gerard Wassink
33                ;* ------------------------------------------------------------------------ *
34                ;
35   0100         START       ORG     0100H
36                ;
37                ; BDOS and BIOS addresses and functions
38                ;
39   0005 =       BDOS        EQU     0005H           ; BDOS address
40   0009 =       PRTSCR      EQU     009H            ; Print $ terminated string function
41                ;
42   0020 =       RTCGTTM     EQU     020H            ; Get HBIOS time
43                ;
44                ;* ------------------------------------------------------------------------ *
45                ;*                                                                MAIN LINE *
46                ;* ------------------------------------------------------------------------ *
47   0100 C5      GAWGETIM    PUSH    B               ; Save
48   0101 D5                  PUSH    D               ;  registers
49   0102 E5                  PUSH    H               ;   on the stack
50                           ;
51                           ; Get time from RTC (Real Time Clock)
52                           ;
53   0103 0620               MVI     B,RTCGTTM       ; BIOS RTCGETTIM function
54   0105 214C01             LXI     H,TIMBUF        ; HL points to buffer for BIOS time
55   0108 CF                 RST     1               ; Call BIOS function 08H (8 times
     specified value)
56                           ;
57                           ; Convert date from BCD to Ascii
58                           ;
59   0109 014C01             LXI     B,BCDDATE       ; BC points to value to convert
60   010C 117601             LXI     D,DSPDATE       ; DE points to receiving buffer
61   010F CD2701             CALL    BCD2ASCII       ; Convert BCD to displayable
62                           ;
63                           ; Convert time from BCD to Ascii
64                           ;
65   0112 014F01             LXI     B,BCDTIME       ; BC points to value to convert
66   0115 118001             LXI     D,DSPTIME       ; DE points to receiving buffer
```

```
 67    0118 CD2701               CALL    BCD2ASCII       ; Convert BCD to displayable
 68                              ;
 69                              ; Print result string
 70                              ;
 71    011B 0E09                 MVI     C,PRTSCR        ; Print string function in reg C
 72    011D 115201               LXI     D,DISPLTIM      ; Data address in DE
 73    0120 CD0500               CALL    BDOS            ; Call BDOS for print string function
 74                              ;
 75                              ; Program end, restore registers and return
 76                              ;
 77    0123 E1                   POP     H               ; Restore registers
 78    0124 D1                   POP     D               ;  from the
 79    0125 C1                   POP     B               ;   stack
 80                              ;
 81    0126 C9                   RET                     ; Go back to caller
 82                              ;
 83                    ;* ------------------------------------------------------------------------ *
 84                    ;*                  Convert BCD values to ascii bytes for display (BCD2ASCII) *
 85                    ;* ------------------------------------------------------------------------ *
 86    0127 =          BCD2ASCII    EQU     $
 87                              ;
 88                              ; Initialize counter
 89                              ;
 90    0127 3E03                 MVI     A,03H           ; load counter for 3 times
 91    0129 324B01               STA     COUNT           ;  and store it
 92                              ;
 93                              ; Loop start
 94                              ;
 95    012C 0A         BCDAGAIN: LDAX    B               ; load BCD value
 96    012D E6F0                 ANI     0F0H            ; Isolate first nibble
 97    012F 0F                   RRC                     ; Rotate
 98    0130 0F                   RRC                     ;  register
 99    0131 0F                   RRC                     ;   A for
100    0132 0F                   RRC                     ;    four times
101    0133 F630                 ORI     030H            ; Make Ascii number
102    0135 12                   STAX    D               ; Store printable value 1st part
103                              ;
104    0136 13                   INX     D               ; Increment DE for next store
105                              ;
106    0137 0A                   LDAX    B               ; load value again
107    0138 E60F                 ANI     00FH            ; Isolate second nibble
108    013A F630                 ORI     030H            ; Make Ascii number
109    013C 12                   STAX    D               ; Store printable value 2nd part
110                              ;
111                              ; Increment pointers to next values
112                              ;
113    013D 03                   INX     B               ; Point to next BCD byte
114    013E 13                   INX     D               ; Point to
115    013F 13                   INX     D               ;   next Ascii value
116                              ;
117                              ; Check for loop end, go around if not
118                              ;
119    0140 3A4B01               LDA     COUNT           ; Get counter value
120    0143 3D                   DCR     A               ; Decrement
121    0144 324B01               STA     COUNT           ; Store counter back
122    0147 C22C01               JNZ     BCDAGAIN        ; Not done, again
123                              ;
124    014A C9                   RET
125                              ;
126                    ;* ------------------------------------------------------------------------ *
127                    ;*                                                              DATA STORAGE *
128                    ;* ------------------------------------------------------------------------ *
129    014B 03         COUNT     DB      03H             ; Counter storage
130                              ;
131    014C =          TIMBUF    EQU     $               ; Reserve room for date / time buffer
132    014C 000000     BCDDATE   DB      00H, 00H, 00H
133    014F 000000     BCDTIME   DB      00H, 00H, 00H
```

```
                        ;
0152 =          DISPLTIM    EQU     $               ; Reserve room for displayable time
0152 4375727265             DB      'Current date and time from the RTC: '
0176 30302D3030DSPDATE      DB      '00-00-00, '    ; YEAR, MONTH,DAY
0180 30303A3030DSPTIME      DB      '00:00:00'      ; HOURS, MINUTES,SECONDS
0188 0D0A                   DB      0DH, 0AH        ; CR/LF
018A 24                     DB      '$'             ; End of string character
                        ;
018B            ENDPROG     END
```