```
 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   0000 =       WARMBOOT    EQU     0000H           ; Warm Reboot
40   0005 =       BDOS        EQU     0005H           ; BDOS address
41   0009 =       PRTSCR      EQU     009H            ; Print $ terminated string function
42                ;
43   0020 =       RTCGTTM     EQU     020H            ; Get HBIOS time
44                ;
45                ;* ------------------------------------------------------------------------ *
46                ;*                                                                 MAIN LINE *
47                ;* ------------------------------------------------------------------------ *
48   0100 C5      GAWGETIM    PUSH    B               ; Save
49   0101 D5                  PUSH    D               ;  registers
50   0102 E5                  PUSH    H               ;   on the stack
51                            ;
52                            ; Get time from RTC (Real Time Clock)
53                            ;
54   0103 0620                MVI     B,RTCGTTM       ; BIOS RTCGETTIM function
55   0105 214E01              LXI     H,TIMBUF        ; HL points to buffer for BIOS time
56   0108 CF                  RST     1               ; Call BIOS function 08H (8 times
     specified value)
57                            ;
58                            ; Convert date from BCD to Ascii
59                            ;
60   0109 014E01              LXI     B,BCDDATE       ; BC points to value to convert
61   010C 117801              LXI     D,DSPDATE       ; DE points to receiving buffer
62   010F CD2901              CALL    BCD2ASCII       ; Convert BCD to displayable
63                            ;
64                            ; Convert time from BCD to Ascii
65                            ;
66   0112 015101              LXI     B,BCDTIME       ; BC points to value to convert
```

```
 67    0115 118201               LXI     D,DSPTIME       ; DE points to receiving buffer
 68    0118 CD2901               CALL    BCD2ASCII       ; Convert BCD to displayable
 69                              ;
 70                              ; Print result string
 71                              ;
 72    011B 0E09                 MVI     C,PRTSCR        ; Print string function in reg C
 73    011D 115401               LXI     D,DISPLTIM      ; Data address in DE
 74    0120 CD0500               CALL    BDOS            ; Call BDOS for print string function
 75                              ;
 76                              ; Program end, restore registers and return
 77                              ;
 78    0123 E1                   POP     H               ; Restore registers
 79    0124 D1                   POP     D               ;  from the
 80    0125 C1                   POP     B               ;   stack
 81                              ;
 82    0126 C30000               JMP     WARMBOOT        ; Back to CP/M
 83                              ;
 84                    ;* ------------------------------------------------------------------------ *
 85                    ;*                  Convert BCD values to ascii bytes for display (BCD2ASCII) *
 86                    ;* ------------------------------------------------------------------------ *
 87    0129 =          BCD2ASCII    EQU     $
 88                              ;
 89                              ; Initialize counter
 90                              ;
 91    0129 3E03                 MVI     A,03H           ; load counter for 3 times
 92    012B 324D01               STA     COUNT           ;  and store it
 93                              ;
 94                              ; Loop start
 95                              ;
 96    012E 0A         BCDAGAIN: LDAX    B               ; load BCD value
 97    012F E6F0                 ANI     0F0H            ; Isolate first nibble
 98    0131 0F                   RRC                     ; Rotate
 99    0132 0F                   RRC                     ;  register
100    0133 0F                   RRC                     ;   A for
101    0134 0F                   RRC                     ;    four times
102    0135 F630                 ORI     030H            ; Make Ascii number
103    0137 12                   STAX    D               ; Store printable value 1st part
104                              ;
105    0138 13                   INX     D               ; Increment DE for next store
106                              ;
107    0139 0A                   LDAX    B               ; load value again
108    013A E60F                 ANI     00FH            ; Isolate second nibble
109    013C F630                 ORI     030H            ; Make Ascii number
110    013E 12                   STAX    D               ; Store printable value 2nd part
111                              ;
112                              ; Increment pointers to next values
113                              ;
114    013F 03                   INX     B               ; Point to next BCD byte
115    0140 13                   INX     D               ; Point to
116    0141 13                   INX     D               ;   next Ascii value
117                              ;
118                              ; Check for loop end, go around if not
119                              ;
120    0142 3A4D01               LDA     COUNT           ; Get counter value
121    0145 3D                   DCR     A               ; Decrement
122    0146 324D01               STA     COUNT           ; Store counter back
123    0149 C22E01               JNZ     BCDAGAIN        ; Not done, again
124                              ;
125    014C C9                   RET
126                              ;
127                    ;* ------------------------------------------------------------------------ *
128                    ;*                                                              DATA STORAGE *
129                    ;* ------------------------------------------------------------------------ *
130    014D 03         COUNT     DB      03H             ; Counter storage
131                              ;
132    014E =          TIMBUF    EQU     $               ; Reserve room for date / time buffer
133    014E 000000     BCDDATE   DB      00H, 00H, 00H
```

```
134    0151 000000    BCDTIME     DB      00H, 00H, 00H
135                   ;
136    0154 =         DISPLTIM    EQU     $               ; Reserve room for displayable time
137    0154 4375727265            DB      'Current date and time from the RTC: '
138    0178 30302D3030DSPDATE     DB      '00-00-00, '    ; YEAR, MONTH,DAY
139    0182 30303A3030DSPTIME     DB      '00:00:00'      ; HOURS, MINUTES,SECONDS
140    018A 0D0A                  DB      0DH, 0AH        ; CR/LF
141    018C 24                    DB      '$'             ; End of string character
142                   ;
143    018D           ENDPROG     END
144
```