

```

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     ;*   0.3  : Return to CP/M using Warm Reboot
12     ;*
13     ;* ----- *
14     ;
15     ;* ----- *
16     ;*           GNU LICENSE CONDITIONS
17     ;* ----- *
18     ;* This program is free software; you can redistribute it and/or modify
19     ;* it under the terms of the GNU General Public License as published by
20     ;* the Free Software Foundation; either version 2 of the License, or
21     ;* (at your option) any later version.
22     ;*
23     ;* This program is distributed in the hope that it will be useful,
24     ;* but WITHOUT ANY WARRANTY; without even the implied warranty of
25     ;* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
26     ;* GNU General Public License for more details.
27     ;*
28     ;* You should have received a copy of the GNU General Public License along
29     ;* with this program; if not, write to the Free Software Foundation, Inc.,
30     ;* 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
31     ;*
32     ;* ----- *
33     ;*           Copyright (C) December 2021 Gerard Wassink
34     ;* ----- *
35     ;
36     0100      START          ORG      0100H
37     ;
38     ; BDOS and BIOS addresses and functions
39     ;
40     0000 =      WARMBOOT      EQU      0000H          ; Warm Reboot
41     0005 =      BDOS          EQU      0005H          ; BDOS address
42     0009 =      PRTSCR        EQU      0009H          ; Print $ terminated string function
43     ;
44     0020 =      RTCGTTM       EQU      0020H          ; Get HBIOS time
45     ;
46     ;* ----- *
47     ;*                                           MAIN LINE *
48     ;* ----- *
49     0100 C5      GAWGETIM      PUSH     B              ; Save
50     0101 D5              PUSH     D              ; registers
51     0102 E5              PUSH     H              ; on the stack
52     ;
53     ; Get time from RTC (Real Time Clock)
54     ;
55     0103 0620      MVI         B,RTCGTTM          ; BIOS RTCGETTIM function
56     0105 214E01     LXI         H,TIMBUF          ; HL points to buffer for BIOS time
57     0108 CF        RST         1              ; Call BIOS function 08H (8 times
specified value)
58     ;
59     ; Convert date from BCD to Ascii
60     ;
61     0109 014E01     LXI         B,BCDDATE         ; BC points to value to convert
62     010C 117801     LXI         D,DSPDATE         ; DE points to receiving buffer
63     010F CD2901     CALL        BCD2ASCII         ; Convert BCD to displayable
64     ;
65     ; Convert time from BCD to Ascii
66     ;

```

```

67 0112 015101 LXI B,BCDTIME ; BC points to value to convert
68 0115 118201 LXI D,DSPTIME ; DE points to receiving buffer
69 0118 CD2901 CALL BCD2ASCII ; Convert BCD to displayable
70 ;
71 ; Print result string
72 ;
73 011B 0E09 MVI C,PRTSCR ; Print string function in reg C
74 011D 115401 LXI D,DISPLTIM ; Data address in DE
75 0120 CD0500 CALL BDOS ; Call BDOS for print string function
76 ;
77 ; Program end, restore registers and return
78 ;
79 0123 E1 POP H ; Restore registers
80 0124 D1 POP D ; from the
81 0125 C1 POP B ; stack
82 ;
83 0126 C30000 JMP WARMBOOT ; Back to CP/M
84 ;
85 ;* -----*
86 ;* Convert BCD values to ascii bytes for display (BCD2ASCII)*
87 ;* -----*
88 0129 = BCD2ASCII EQU $
89 ;
90 ; Initialize counter
91 ;
92 0129 3E03 MVI A,03H ; load counter for 3 times
93 012B 324D01 STA COUNT ; and store it
94 ;
95 ; Loop start
96 ;
97 012E 0A BCDAGAIN: LDAX B ; load BCD value
98 012F E6F0 ANI 0F0H ; Isolate first nibble
99 0131 0F RRC ; Rotate
100 0132 0F RRC ; register
101 0133 0F RRC ; A for
102 0134 0F RRC ; four times
103 0135 F630 ORI 030H ; Make Ascii number
104 0137 12 STAX D ; Store printable value 1st part
105 ;
106 0138 13 INX D ; Increment DE for next store
107 ;
108 0139 0A LDAX B ; load value again
109 013A E60F ANI 00FH ; Isolate second nibble
110 013C F630 ORI 030H ; Make Ascii number
111 013E 12 STAX D ; Store printable value 2nd part
112 ;
113 ; Increment pointers to next values
114 ;
115 013F 03 INX B ; Point to next BCD byte
116 0140 13 INX D ; Point to
117 0141 13 INX D ; next Ascii value
118 ;
119 ; Check for loop end, go around if not
120 ;
121 0142 3A4D01 LDA COUNT ; Get counter value
122 0145 3D DCR A ; Decrement
123 0146 324D01 STA COUNT ; Store counter back
124 0149 C22E01 JNZ BCDAGAIN ; Not done, again
125 ;
126 014C C9 RET
127 ;
128 ;* -----*
129 ;* DATA STORAGE*
130 ;* -----*
131 014D 03 COUNT DB 03H ; Counter storage
132 ;
133 014E = TIMBUF EQU $ ; Reserve room for date / time buffer

```

[illegible]