
RCSL No: 52-AA1093

Edition: February 1982

Author: Harald Villemoes

Title:

RC3502 CPUSE

Keywords:

RC3502, Real Time Pascal, CPU-load.

Abstract:

CPUSE is a small REAL TIME PASCAL process, which can be used to measure the CPU load in an RC3502 computer.

This manual describes the function and usage of CPUSE. In addition, it contains examples of usage and a complete process listing.

(14 printed pages)

Copyright © 1982, A/S Regnecentralen af 1979
RC Computer A/S

Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

CPUSE is a small PASCAL80 process which can be used to measure the CPU-load in an RC3502 computer.

The CPU-load, which is output in per cent, is calculated from the number of activations of the dummy process in the microprogram per time unit. The period of measurement is user-selected in the interval 5-60 sec. and the accuracy is approx. 6% using 5 sec. period and approx. 0.5% using 60 sec.

The measurement may be repeated automatically up to 32767 times.

CPUSE is pre-calibrated for an RC3502 CPU with microprogram revision 5, but the user may command the program to recalibrate itself on a CPU with different speed.

CPUSE should be run with high priority in order to obtain reasonable accurate measurement period on a machine with high load.

Rev. 5 program size: 1126 bytes

Rev. 5 stack size: 300 words (default)

App. A: The program text

App. B: Example of use

job h1v2 60ⁿ time 1000000 param mini 25 2
 (
 mode list yes
 bcpuse=SET 20 mini
 score project bcpuse
 bcpuse=pascal18J stack.300 loenvir
 finis
)
 PROCESS cpuse(VAR sv: system^vector);
 CONST
 10
 11
 12
 13 version= coded+date(81,8,10);
 14
 15 (* this process is used to measure the amount of free run-time
 16 in % derived from the number of times the schedule instruc-
 17 tion ceased to find a non-empty activequeue.
 18
 programmed by harald villemoes, tele, rc79 *)
 19
 20 countreq#h3e6; (* scheduler dummy-process msb counter *)
 21 defaultfull= 225; (* initial value of fullcount *)
 22
 23
 24 TYPE
 25
 opbuf= RECORD
 26 !: f, l, n: integer;
 27 !: name: alfa;
 28 !: chars: ARRAY (6+12..6+12+8(r-1)) OF char;
 29
 30 Finn;
 31
 32 VAR
 33
 34 onpool: pool 3 OF onbuf;
 35 timerpool: pool 1;
 36 zi, zo: zone;
 37 kbsem,
 timersem: semaphore;
 38 r: reference;
 39 period, repeats, startcount, endcount, fullcount: integer;
 40
 PROCEDURE getregister(VAR value: integer; index: integer); EXTERNAL;
 41
 FUNCTION msub(a,b: integer): integer; EXTERNAL;
 42
 43
 44
 45
 46

```

47   1 ! BEGIN! (* main program *)
48   2 ! openopzone(z1,sv(operatorsem),ref(kbsem),1,oppool,1,1,1,1);
49   3 ! openopzone(z0,sv(operatorsem),ref(z0,free),2,oppool,2,0,0,0);
50   4 ! outtext(z0,"CPU LOAD'ETE");
51   5 ! outdate(z0,"R = VFRS: #");
52   6 ! outnl(z0); outnl(z0);
53   7 ! alloc(r,timerpool,timersem);
54   8 !
55   9 ! fullcount:= defaultfull;
56  10 !
57  11 ! REPEAT (* main loop *)
58  12 !
59  13 ! REPEAT (* read params *)
60  14 !
61  15 ! opin(z1);
62  16 ! outtext(z0,"period (5-60"); outtext(z0," sec) : #");
63  17 ! outtext(z0);
64  18 ! opwait(z1,oppool); ininteger(z1,period); opin(z1);
65  19 ! outtext(z0,"repeats (-1 "); outtext(z0,"means calibr");
66  20 ! outtext(z0,"ate) : #"); outendl(z0);
67  21 ! opwait(z1,oppool); ininteger(z1,repeats);
68  22 !
69  23 ! UNTIL (period>=5) AND (period<=n1) AND (repeats>=-1);
70  24 !
71  25 ! outtext(z0,"CPUSE starts"); outnl(z0);
72  26 !

```

uselist 81.10.28. 14.29.

pane 3

```
74    27 ! WHILE repeats <> 0 DO
75    28 ! BEGIN (* measurement *)
76    29 !
77    30 !   r^.u3:= period; r^.u4:= 10;
78    31 !   getregister(startcount,countrea);
79    32 !   sendtimer(r); (* delay *)
80    33 !   wait(r, timersem);
81    34 !   getregister(endcount,countrea);
82    35 !
83    36 !   IF repeats < 3 THEN:
84    37 !     BEGIN (* calibration *)
85    38 !       fullcount:= msb(endcount,startcount) * 64 DIV period;
86    39 !       outtext(z0,"count normal"); outnl(z0);
87    40 !       outtext(z0,"count = #"); outnl(z0);
88    41 !       outinteger(z0,fullcount,5); outnl(z0);
89    42 !       repeats:= r;
90    43 !
91    44 !     END (* calibration *)
92    45 !
93    46 !
94    47 !
95    48 !   BEGIN (* normal measurement *)
96    49 !
97    50 !     outtext(z0,"cpu free : #");
98    51 !     outinteger(z0,msb(endcount,startcount) * 64 DIV period
99    52 !     * 10 DIV fullcount,4);
100   53 !     outtext(z0,"%#"); outnl(z0);
101   54 !     repeats:= repeats - 1;
102   55 !
103   56 !
104   57 !
105   58 !
106   59 !
107   60 !   UNTIL b = 7;
108   61 !
109   62 ! END; (* CPUSE *)
110
```

cpuuse list 81.10.28. 14.29.

Date 4

6

a	1*	50	50	50	74	83	49
	29*	35*	49	49	49	49	49
	1*	1*	50	50	50	50	50
b	34*	99	70	88	197	101	101
	4	1*	1*	29*	29*	29*	29*
c	5	107	107	107	107	107	107
d	6	13*	13*	13*	13*	13*	13*
e	7	13*	13*	13*	13*	13*	13*
f	8	77	77	77	77	77	77
g	9	29*	29*	29*	29*	29*	29*
h	10	13*	13*	13*	13*	13*	13*
i	11	29*	29*	29*	29*	29*	29*
j	12	4*	4*	4*	4*	4*	4*
k	13	1*	1*	1*	1*	1*	1*
l	14	70	70	70	70	70	70
m	15	29*	29*	29*	29*	29*	29*
n	16	80	80	80	80	80	80
o	17	13*	13*	13*	13*	13*	13*
p	18	90	90	90	90	90	90
q	19	225	225	225	225	225	225
r	20	300	300	300	300	300	300
s	21	6*	6*	6*	6*	6*	6*
t	22	600	600	600	600	600	600
u	23	1*	1*	1*	1*	1*	1*
v	24	100000	100000	100000	100000	100000	100000
w	25	1*	1*	1*	1*	1*	1*
x	26	96	96	96	96	96	96
y	27	98	98	98	98	98	98
z	28	14*	14*	14*	14*	14*	14*
aa	29	28*	28*	28*	28*	28*	28*
ab	30	54	54	54	54	54	54
ac	31	44*	44*	44*	44*	44*	44*
ad	32	4*	4*	4*	4*	4*	4*
ae	33	5*	5*	5*	5*	5*	5*
af	34	6*	6*	6*	6*	6*	6*
ag	35	81	81	81	81	81	81
ah	36	81	81	81	81	81	81
ai	37	86	86	86	86	86	86
aj	38	98	98	98	98	98	98
ak	39	99	99	99	99	99	99
al	40	86	86	86	86	86	86
am	41	81	81	81	81	81	81
an	42	81	81	81	81	81	81
ao	43	68	68	68	68	68	68
ap	44	40*	40*	40*	40*	40*	40*
aq	45	40*	40*	40*	40*	40*	40*
ar	46	40*	40*	40*	40*	40*	40*
as	47	40*	40*	40*	40*	40*	40*
at	48	40*	40*	40*	40*	40*	40*
au	49	40*	40*	40*	40*	40*	40*
av	50	40*	40*	40*	40*	40*	40*
aw	51	40*	40*	40*	40*	40*	40*
ax	52	40*	40*	40*	40*	40*	40*
ay	53	40*	40*	40*	40*	40*	40*
az	54	40*	40*	40*	40*	40*	40*
ba	55	40*	40*	40*	40*	40*	40*
bc	56	40*	40*	40*	40*	40*	40*
bd	57	40*	40*	40*	40*	40*	40*
be	58	40*	40*	40*	40*	40*	40*
bf	59	40*	40*	40*	40*	40*	40*
bg	60	40*	40*	40*	40*	40*	40*
bh	61	40*	40*	40*	40*	40*	40*
bi	62	40*	40*	40*	40*	40*	40*
bj	63	40*	40*	40*	40*	40*	40*
bk	64	40*	40*	40*	40*	40*	40*
bl	65	40*	40*	40*	40*	40*	40*
bm	66	40*	40*	40*	40*	40*	40*
bn	67	40*	40*	40*	40*	40*	40*
bo	68	40*	40*	40*	40*	40*	40*
bp	69	40*	40*	40*	40*	40*	40*
br	70	40*	40*	40*	40*	40*	40*
bs	71	40*	40*	40*	40*	40*	40*
bu	72	40*	40*	40*	40*	40*	40*
bv	73	40*	40*	40*	40*	40*	40*
bw	74	40*	40*	40*	40*	40*	40*
bx	75	40*	40*	40*	40*	40*	40*
by	76	40*	40*	40*	40*	40*	40*
bz	77	40*	40*	40*	40*	40*	40*
ca	78	40*	40*	40*	40*	40*	40*
cb	79	40*	40*	40*	40*	40*	40*
cc	80	40*	40*	40*	40*	40*	40*
cd	81	40*	40*	40*	40*	40*	40*
ce	82	40*	40*	40*	40*	40*	40*
cf	83	40*	40*	40*	40*	40*	40*
cg	84	40*	40*	40*	40*	40*	40*
ch	85	40*	40*	40*	40*	40*	40*
ci	86	40*	40*	40*	40*	40*	40*
cj	87	40*	40*	40*	40*	40*	40*
ck	88	40*	40*	40*	40*	40*	40*
cl	89	40*	40*	40*	40*	40*	40*
cm	90	40*	40*	40*	40*	40*	40*
cn	91	40*	40*	40*	40*	40*	40*
co	92	40*	40*	40*	40*	40*	40*
cp	93	40*	40*	40*	40*	40*	40*
cr	94	40*	40*	40*	40*	40*	40*
cs	95	40*	40*	40*	40*	40*	40*
cu	96	40*	40*	40*	40*	40*	40*
cv	97	40*	40*	40*	40*	40*	40*
cy	98	40*	40*	40*	40*	40*	40*
cz	99	40*	40*	40*	40*	40*	40*

causalist	81.17.28.	14.29.	♀
job	1*		
busem	37*	49	
list	3*	4*	
ini	1*		
node	3*		
isub	41*	86	98
name	28*		
pbuf	26*	34*	
penopzone	49	50	
operatorsem	49	50	
pin	62	65	
ppool	34*	49	
pwait	65	68	
utdate	52		
utend	64	67	
utinteger	88	98	
utnl	52	72	
uttext	51	63	
ascalau	6*		
period	40*	65	
erm	1*		
ool	34*	35*	
process	9*		
project	5*		
ref	39*	54	
size	49	50	
difference	39*		
repeats	40*	68	
scope	5*		
semaphore	38*		
endtimer	79		
stack	1*		
startcount	6*		
system+vector	40*	78	
time	9*		
lmerpool	1*		
imersem	35*	54	
	38*	54	
	77*	80	

cpuclist 81.10.28. 14.20.
date 6
u4 77= 52
version 13* 52
wait 80
yes 3* 68
zi 36* 49 62 65 65 68
zo 36* 50 50 51 52 63
72 87 87 88 88 100
zone 36*

cpu sublist 81.10.28. 14.29.

9

ALD
ARRAY
ALGIN
CONST
DIV
DO
ELSE
END
FUNCTION
IF
OF
PROCEDURE
RECORD
REPEAT
SET
THEIN
TYPE
UNTIL
VALUE
VAR
WHILE

page 7

B. EXAMPLE OF USE

B.

```

>opsys
lockup sqritest opuse
PROCESS    sqritest      1981.06.12 10.11  1  10000  1.50  100
PROCESS    opuse        1981.06.10 14.44  2  511   1.00  100
prio 2 run opuse
>opuse
CPU LOADMETER - VERS: 1981.06.10
period (5-60 sec) : 5
repeats (-1 means calibrate) : 100
CFVUE starts
>opsys
prio -2 size 5000 run sqritest
>opuse
cpu free :  56 %
cpu free : 101 %
>sqritest
sqritable:
>opuse
cpu free :  58 %
>sqritest
eps:          0.010000
>opuse
cpu free :  5 %
>sqritest
 1:  1.000
>opuse
cpu free :  10 %
>sqritest
 2:  1.00000
               3
               4
>opuse
cpu free :  0 %
>sqritest
               -1
               2
>opsys
stop sqritest
>sqritest
 -0.50000
>opuse
cpu free :  0 %
cpu free :  64 %
cpu free : 101 %
>opsys
stop opuse
>opuse
cpu free :  96 %

```

RETURN LETTER

Title: RC3502 CPUSE

RCSL No.: 52-AA1093

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

Do you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Name: _____

Title: _____

Company: _____

Address: _____

Date: _____

Thank you

..... Fold here

..... Do not tear - Fold here and staple

Affix
postage
here



Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark