
RCSL No: 52-AA1019

Edition: December, 1980

Author: Bo Bagger Laursen

Title:

General Information for
RC3503 Standard Autoload

<u>CONTENTS</u>	<u>PAGE</u>
-----------------	-------------

1. SHORT DESCRIPTION	1
2. DESCRIPTION	2
3. SELECTION OF INPUT DEVICE	4
3.1 Switch Input	4
4. DATA FORMAT	6
5. STORING OF INPUT	7
6. AUTOLOAD FROM TES201/202	8
7. AUTOLOAD FROM FPA100	10
8. PHYSICAL DESCRIPTION	13

APPENDICES:

A. REFERENCES	15
B. PROGRAM LIST	16

1. SHORT DESCRIPTION

1.

To place information in the RC3503 memory without relying on a program already in memory, the operator must load eight words at a time manually from a Debug Console (consult [2]); the information loaded in this manner is usually a bootstrap loader to take care of the total program loading.

By using the RC3503 AUTOLOAD function this program loading is done automatically simply by pushing the AUTO button.

The standard RC3503 AUTOLOAD contains program loading from the following input devices:

- Paper Tape Reader (RC500 or RC2500)
- Asynchronous Devices (110 bps., i.e. TTY)
- Front End Processor Adaptor, FPA100 (RC35-221)
- TES201/202 Image Load Module (RC0117/RC3514)

Optional program loading from other input devices can be delivered.

For further information about RC3503 refer to: RCSL: 52-AA1013, RC3503 General Purpose Controller, General Information [1].

If the input device is the RC500 Paper Tape Reader, the operator pushes the READ-button at this time. (Previously the Reader must be reset by pushing the clear button).

If the input device is an FPA100, the operator initiates an auto-load from the RC8000 connected to the FPA100.

Switch 0- 2: Device type.

SWITCH			DEVICE TYPE
0	1	2	
0	0	0	LAM116 channel 0-7
0	0	1	LAM116 channel 8-15
0	1	0	TES201/202
0	1	1	RC500/RC2500
1	0	0	Not used
1	0	1	Not used
1	1	0	FPA100
1	1	1	Not used

Switch 3- 7: MODULE NUMBER.

Switch 8-15: PROGRAM INDEX.

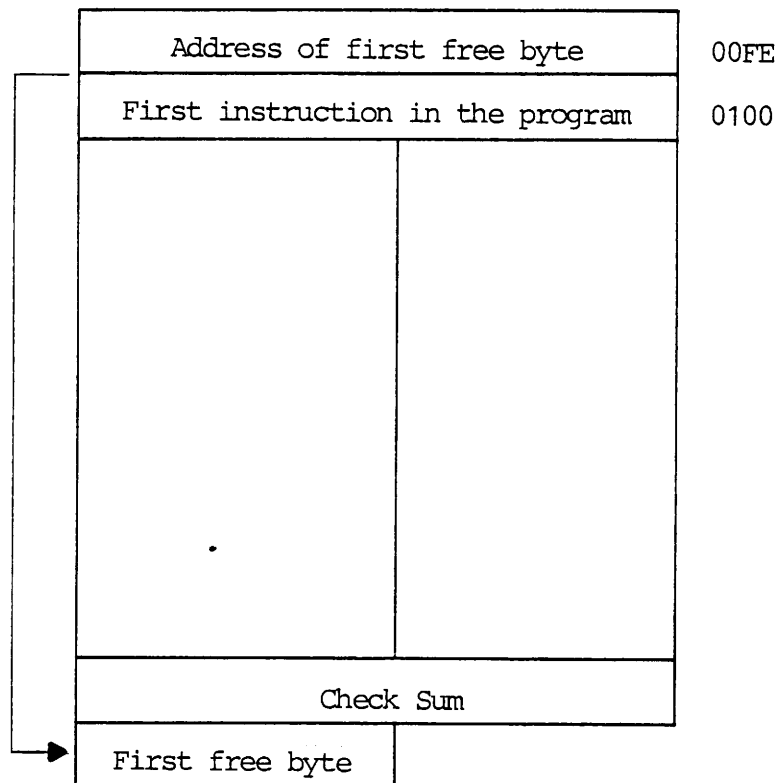
When the input device is TES201/202, switch 3-7 is the module number of the TES201/202.

Switch 8-15 concatenated with 8 binary zeroes is the start address in the module of the image, selected for copying to RAM memory.

5. STORING OF INPUT

5.

The explanation is connected to the figure below, showing a part of the RC3503 memory.



The unpacked information bytes (see section 4) are stored from address 00FE and onward.

The first stored word contains the address of the first byte following the whole program. The last stored word contains a checksum, which is an addition of all bytes, considered as positive numbers. The sum is formed modulo 2^{16} . If a checksum error is found, the program will stop in B9 (FFFC), else continue in cell 0100 in the stored program, if the GO/NOGO switch is false.

If all checksums are OK, the autoloader program jumps to 0100.

Note that the field 'Address of first free byte' is an address in the RAM memory after movement. This address is used as a stop criterion when loading. So the image EPROM may be placed in any position. The only demand is that the image can be addressed by the switches (Module#. (Program index < 8 + 0)).

The following string of RC8000 file processor commands will:

- A. Assemble an RC3503 program textfile into a binary RC3503 program
- B. Prepare a bootload file for the RC3503
- C. Load the RC3503 with the RC3503 bootload program

Provided the necessary programs are present or accessible in RC8000 computer.

```
RC3503BIN = SLAM4 RC3503TEXT
RC3503BOOT = PUNCH16 MODE.BOOT FPABOOTBIN RC3503BIN
MAIN3500 = AUTOLOAD RC3503BOOT
```

Explanation

RC3503BIN is the name of a temporary file used to hold the binary version of an RC3503 program as output from the RC3503 assembler.

SLAM4 is the name of the file containing the RC3503 assembler running on an RC8000 computer. The RC8000 version of the RC3503 assembler is a normal RC8000 ALGOL program, the textfile is usually called slamtext.

RC3503TEXT is the name of a file containing an RC3503 program text.

RC3503BOOT is the name of a temporary file containing a bootload version of the RC3500 program.

PUNCH16 is the name of a file containing a program capable of converting a binary version of an RC3503 program into a loadable version.

8. PHYSICAL DESCRIPTION

8.

The RC3503 autoloader program is placed in one EPROM containing 2K x 8 bits. The EPROM is labelled ROA596 and is placed in position 11 on MEM204.

The standard autoloader, including program loading from RC500, LAM116, FPA100, and TES201/202 uses 472 words, so there are 40 words free for optional use.

Before putting an optional driver into the standard autoloader the following facts may be useful:

Device No.

Allocate a free number from the table in chapter 3.

Program Code.

Write the program as running from address 009C and onward.

Start Address.

Note the address of the first word of the driver. This address is placed in the table B1 according to the selected device No. (see program list, Appendix B).

Return Address.

If the driver loads according to the standard autoloader format, this address describes the point in the driver to which the program should return after processing a character in the standard autoloader program. This address is placed in the table B11 according to the selected device No. (see program list, Appendix B).

A. REFERENCES

A.

- [1] RC3503 General Purpose Controller
General Information
RCSL: 52-AA1013

- [2] RC3503 Reference Manual
RCSL: 42-i1540

- [3] General Information for FPA100
RCSL: 52-AA566

- [4] RC35-103/104
Low Speed Asynchronous Multiplexer for RC3500
General Information
RCSL: 52-AA069

- [5] Programming Information about the
Available Interfaces to RC3500
RCSL: 52-AA030, Chapter 6.0 (PTR)

- [6] RC3514, TES202
System Generation Guide
RCSL: 52-AA1020

- [7] General Information for
TES201/202 Eprom Modules
RCSL: 52-AA1021

- [8] RC3503 Autoload
Source Paper Tape
RCSL: 52-AA1024

- [9] RC3503 Autoload
Binary Paper Tape
RCSL: 52-AA1025

```

72 156 000234 009c          17
72 156 000234 009c    b0=b0+k-b4,k=b4
73 156 000234 009c
73 156 000234 009c    b.e10
74 156 000234 009c    w.          ; asynchrone char-multiplexer
75 156 000234 009c
75 156 000234 009c    a0=b0,a1=b0
76 156 000234 009c        rsw w2 1 0
77 158 000236 009e        lsr w2 9
78 160 000240 00a0        and w2 1 15<1
79 162 000242 00a2    e5:  ldr w1 1 x2+13<5+3<11
80 166 000246 00a6        wcc w1 0
81 168 000250 00a8        ldr w1 1 -1
82 170 000252 00aa        wcc w2 x2 0
83 172 000254 00ac        rwc w2 x2 0
84 174 000256 00ae        sub w1 1 x2 1
85 178 000262 00b2        bsz w1 1 31,
86 182 000266 00b6        add w0 1 e5-e4
87 186 000272 00ba    e4:
88 186 000272 00ba    e.
89 186 000272 00ba

```

```

89 186 000272 00ba
89 186 000272 00ba    b0=b0+k-b4,k=b4
90 156 000234 009c
90 156 000234 009c    b.          ;ptr autoloader
91 156 000234 009c    w.
92 156 000234 009c
92 156 000234 009c    a3=b0
93 156 000234 009c        rsc w3 x3 0
94 158 000236 009e        wcc w3 0
95 160 000240 00a0        rwc w3 x3 0
96 162 000242 00a2        ldr w0 1
97 164 000244 00a4    e.
97 164 000244 00a4
98 164 000244 00a4

```

```

177 216 000330 00d8
178 216 000330 00d8 e4: xsr x3 d12 ; word := getepromword
179 218 000332 00da bsw w1 x3 d10 , b50+2 ; if word = total then goto start
180 222 000336 00de ldr aw0 1 b9 ; goto checksum error
181 224 000340 00e0
181 224 000340 00e0 e5: ; procedure getepromword
182 224 000340 00e0 ldr w1 x3 d2 ; w1 := from<8 + to
183 226 000342 00e2 ldr w2 x3 d4 ; w2 := fromaddress
184 228 000344 00e4 mov move
185 230 000346 00e6 ldr w1 x3 d0 ; w1 := word
186 232 000350 00e8 add w2 1 2 ; fromaddress := fromaddress + 2
187 234 000352 00ea str w2 x3 d4 ;
188 236 000354 00ec xsr x3 d12 ; return
189 238 000356 00ee sub aw0 1 e6-e5 ;
190 240 000360 00f0 e6: ;
191 240 000360 00f0 e0: ;
192 240 000360 00f0 0 ; word
193 242 000362 00f2 0 ; fromprommodule<8 + toamodule
194 244 000364 00f4 0 + 2 ; frompromaddress
195 246 000366 00f6 b50 ; toamaddress
196 248 000370 00f8 0 ; subtotal
197 250 000372 00fa 0 ; total
198 252 000374 00fc e5 ; procedure getepromword
199 254 000376 00fe e. ;
200 254 000376 00fe
200 254 000376 00fe
200 254 000376 00fe
200 254 000376 00fe
200 254 000376 00fe ;*

```

```

201 254 000376 00fe
201 254 000376 00fe b0=b0+k-b4
202 254 000376 00fe 0, r. (-c4-b1-b0)/2 ;
203 270 000416 010e k= -b1
204 65428 177624 ff94
204 65428 177624 ff94 b1=k ; startaddresses:
205 65428 177624 ff94 ; c ccm mmx xxd ddd ddd: switches
206 65428 177624 ff94 ; c: devicecode, m: modification
207 65428 177624 ff94 ; d: devicenumber
208 65428 177624 ff94
208 65428 177624 ff94 c4+a0 ; 0x xxx xxd ddd ddd: lam multiplexer
209 65430 177626 ff96 c4+a1 ; 0 1x xxx xxd ddd ddd: lam multiplexer
210 65432 177630 ff98 c4+a2 ; 0 0x xxx xxd ddd ddd: imageload
211 65434 177632 ff9a c4+a3 ; 0 11x xxx xxd ddd ddd: paper tape reader
212 65436 177634 ff9c 0
213 65438 177636 ff9e 0
214 65440 177640 ffa0 c4+a6 ; 1 10x xxx xxd ddd ddd: fpa autoloader
215 65442 177642 ffa2 0
216 65444 177644 ffa4 ; return addresses:
217 65444 177644 ffa4 b11:
218 65444 177644 ffa4 b4
219 65446 177646 ffa6 b4
220 65448 177650 ffa8 0
221 65450 177652 ffaa b4
222 65452 177654 ffac 0
223 65454 177656 ffae 0
224 65456 177660 ffb0 0
225 65458 177662 ffb2 0
226 65460 177664 ffb4 ;*

```

```

names:      e5
e 0         0         0 0000
e 1         2         2 0002
e 2         38        46 0026
e 3         40        50 0028
e 4         44        54 002c
e 5         48        60 0030
31 50 000062 0032 000376 00fe
33 52 000064 0034      2 begin store char
names:      e10
34 52 000064 0034 040010 4008
35 54 000066 0036 161050 e228
36 56 000070 0038 077000 7e30
37 58 000072 003a 163054 e62c
38 60 000074 003c 175112 fa4a
38 62 000076 003e 000177 007f
39 64 000100 0040 175121 fa51
39 66 000102 0042 000100 0040
39 68 000104 0044 000134 005c
40 70 000106 0046 174600 f980
40 72 000110 0048 177644 ffa4
41 74 000112 004a 175121 fa51
41 76 000114 004c 000177 007f
41 78 000116 004e 000106 0046
42 80 000120 0050 111041 9221
42 82 000122 0052 000106 0046
43 84 000124 0054 131020 b210
43 86 000126 0056 000146 0066
44 88 000130 0058 161064 e234
45 90 000132 005a 035061 3a31
46 92 000134 005c 012064 1434
47 94 000136 005e 176104 fc44

```

```

47 96 000140 0060 000100 0040
48 98 000142 0062 174600 f980
48 100 000144 0064 177644 ffa4
49 102 000146 0066 051017 520f
50 104 000150 0068 175010 fa08
50 106 000152 006a 000061 0031
51 108 000154 006c 012062 1432
52 110 000156 006e 171447 f327
53 112 000160 0070 176021 fc11
53 114 000162 0072 000376 00fe
53 116 000164 0074 000212 008a
54 118 000166 0076 022062 2432
55 120 000170 0078 175010 fa08
55 122 000172 007a 177702 ffc2
56 124 000174 007c 044174 487c
57 126 000176 007e 147040 ce20
58 128 000200 0080 143600 e780
59 130 000202 0082 021102 2242
60 132 000204 0084 015403 1b03
61 134 000206 0086 174600 f980
61 136 000210 0088 177644 ffa4
62 138 000212 008a 001576 037e
63 140 000214 008c 105400 8b00
63 142 000216 008e 000222 0092
64 144 000220 0090 000174 007c
65 146 000222 0092 175006 fa06
65 148 000224 0094 177704 ffc4
66 150 000226 0096 131002 b202
66 152 000230 0098 000400 0100
67 154 000232 009a 044002 4802
68 156 000234 009c      2 end store char

```

```

116 190 000276 00be 173500 f740
117 192 000300 00c0 001000 0200
118 194 000302 00c2 141700 c3c0
119 196 000304 00c4 001002 0202
120 198 000306 00c6 144040 c820
121 200 000310 00c8 142440 c520
122 202 000312 00ca 175161 fa71
122 204 000314 00cc 000020 0010
122 206 000316 00ce 000234 009c
123 208 000320 00d0 011104 1244
124 210 000322 00d2 175121 fa51
124 212 000324 00d4 000373 00fb
124 214 000326 00d6 000346 00e6
125 216 000330 00d8 175161 fa71
125 218 000332 00da 000371 00f9
125 220 000334 00dc 000234 009c
126 222 000336 00de 175100 fa40
126 224 000340 00e0 000400 0100
127 226 000342 00e2 174100 f840
127 228 000344 00e4 000240 00a0
128 230 000346 00e6 047020 4e10
130 232 000350 00e8 171446 f326
131 234 000352 00ea 025600 2b80
132 236 000354 00ec 041600 4380
133 238 000356 00ee 015600 1b80
134 240 000360 00f0 172761 f5f1
134 242 000362 00f2 000350 00e8
135 244 000364 00f4 105000 8a00
135 246 000366 00f6 000400 0100
136 248 000370 00f8 174100 f840
136 250 000372 00fa 000234 009c
137 252 000374 00fc 2 end

```

```

tps autoloop bbl 00.12.16

```

```

names: e4
e 1 156 234 009c
e 2 160 240 00a0
e 3 232 350 00e8
e 4 230 346 00e6
139 156 000234 009c 2 begin (image)load bbl 00.12.17
names: d12 e6
148 156 000234 009c 075000 7a00
149 158 000236 009e 177100 fe40
149 160 000240 00a0 000360 00f0
150 162 000242 00a2 035604 3084
151 164 000244 00a4 161050 e228
152 166 000246 00a6 051037 521f
153 168 000250 00a8 035602 3b82
154 170 000252 00aa 024614 298c
157 172 000254 00ac 176166 fc76
157 174 000256 00ae 007777 0ff7
157 176 000260 00b0 000276 00be
158 178 000262 00b2 045610 4b88
159 180 000264 00b4 105000 8a00
159 182 000266 00b6 000272 00ba
160 184 000270 00b8 000174 007c
161 186 000272 00ba 015610 1b88
162 188 000274 00bc 044024 4814
165 190 000276 00be 041610 4388
166 192 000300 00c0 015610 1b88
167 194 000302 00c2 001600 0380
168 196 000304 00c4 041612 438a
169 198 000306 00c6 015612 1b8a
170 200 000310 00c8 001600 0380

```

```

223 65454 177656 ffae 000000 0000
224 65456 177660 ffb0 000000 0000
225 65458 177662 ffb2 000000 0000
228 65460 177664 ffb4      2 begin
names:      e4
229 65460 177664 ffb4 001000 0200
230 65462 177666 ffb6 021102 2242
232 65464 177670 ffb8 147640 cfa0
233 65466 177672 ffb8 041077 423f
234 65468 177674 ffb8 111000 9200
234 65470 177676 ffb8 000234 009c
235 65472 177700 ffc0 044010 4808
236 65474 177702 ffc2 000000 0000
237 65476 177704 ffc4 177777 ffff
238 65478 177706 ffc6 176100 fc40
238 65480 177710 ffc8 177000 fe00
239 65482 177712 ffc8 003000 0600
240 65484 177714 ffc8 171440 f320
241 65486 177716 ffc8 171744 f3e4
242 65488 177720 ffd0 177161 fe71
242 65490 177722 ffd2 000234 009c
242 65492 177724 ffd4 177714 ffc8
243 65494 177726 ffd6 075000 7a00
244 65496 177730 ffd8 021104 2244
245 65498 177732 ffd8 161054 e22c
246 65500 177734 ffd8 051016 520e
247 65502 177736 ffd8 175161 fa71
247 65504 177740 ffe0 000004 0004
247 65506 177742 ffe2 177750 ffe0
248 65508 177744 ffe4 002037 041f
249 65510 177746 ffe6 036105 3c45
250 65512 177750 ffe8 176200 fc80
250 65514 177752 ffea 177624 ff94
251 65516 177754 ffec 171440 f320
252 65518 177756 ffec 171744 f3e4
253 65520 177760 ffe0 177121 fe51
253 65522 177762 ffe2 000376 00fe
253 65524 177764 ffe4 000052 002a
254 65526 177766 ffe6 000154 006c
255 65528 177770 ffe8 001000 0200
256 65530 177772 ffea 147640 cfa0
257 65532 177774 ffec 064174 687c
259 65534 177776 ffe8 000106 0046
260      0 000000 0000      2 end

```

move the program to the botto

move the program to the botto

```

names:      e4
e 0 -70 177672 ffb8
e 1 -58 177706 ffc6
e 2 -52 177714 ffc8
e 3 -20 177754 ffec
e 4 -24 177750 ffe8
261 0 000000 0000      1 end      autoloan
names:      a20 b50 c5
a 0 156 234 009c
a 1 156 234 009c
a 2 290 442 0122
a 3 186 272 00ba
a 6 194 302 00c2
a 10 54 66 0036
a 11 56 70 0038
b 0 388 604 0184
b 1 -108 177624 ff94
b 2 20 24 0014
b 3 42 52 002a
b 4 156 234 009c
b 5 50 62 0032
b 6 49 61 0031
b 7 -8 177770 fff8
b 8 -76 177664 ffb4
b 9 -4 177774 fffc
b 10 -62 177702 ffc2
b 11 -92 177644 ffa4
b 12 -60 177704 ffc4
b 50 254 376 00fe
c 4 -512 177000 fe00
261 0 000000 0000 777777 ffff

```

```

261 0 000000 0000      prog end, a: 92, b: 10, c: 35
errorcount: 0, swapcount: 15

```

```

33 b. e10 ; store char
68 e.
e0 def 46
   use 39
e1 def 40
   use 41      42
e2 def 41
   use 34
e3 def 49
   use 43
e4 def 39
   use 34      47
e5 def 62
   use 53
e6 def 34
   use 46
e7 def 57
   use 56
e8 def 59
   use 56
e9 def 65
   use 63

```

```

73 b. e10 ; asynchrone char-multiplexer
88 e.
e4 def 87
   use 86
e5 def 79
   use 86

```

```

90 b. iptr autoloan
97 e.

```

```

99 b. e4 ; fpa autoloan bbl 80.12.16
137 e.
e1 def 102
   use 108      122      125      136
e2 def 103
   use 127
e3 def 129
   use 134
e4 def 128
   use 124

```

```

139 b. d12, e6 ; imageload bbl 80.12.17
199 e.
d0 def 141
   use 142      167      170      185
d2 def 142
   use 143      153      182
d4 def 143
   use 144      150      183      187
d6 def 144
   use 145      171      173
d8 def 145
   use 146      158      161      165      166
d10 def 146
   use 147      168      169      179
d12 def 147
   use 154      178      188
e0 def 191
   use 149
e1 def 154
   use 162      175
e2 def 161
   use 159
e3 def 164
   use 157      162
e4 def 178
   use 174      175
e5 def 181
   use 189      198
e6 def 190
   use 189

```

RETURN LETTER

General Information for
Title: RC3503 Standard Autoload

RCSL No.: 52-AA1019

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