

```

002      *
003      *
004      *
005      * =====
006      *** UPDATES in BASIC version V1.1 ***
007      * =====
008      *
009      * The differences in BASIC V1.1 from V1.0 are given.
010      *
011      * ]      : Indicates a total change of instruction
012      * !      : Indicates only a corrected jumpaddress
013      *           due to a move of the original routine.
014      * No mark: Instructions only moved to other
015      *           memory addresses (Mostly to obtain
016      *           space for new routines).
017      *
018      *
019      *****
020      * INTEGER COMPARE *
021      *****
022      *
023      * The temporary storage register E (sign byte) is
024      * cleared to avoid problems when using the combined
025      * FPT/INT exit LC27. This is done to remove a bug in
026      * comparing 2 relational numeric operations. The
027      * unitary operator '?' is now recognised correctly.
028      *
029      *           ORG      :C0B4
030  C0B4 C3C8D1      JMP      :D1CB      ]
031      *
032      *           ORG      :D1CB
033  D1CB 7B          MOV      A,E      ] Get signbyte in A
034  D1C9 1E00        MVI      E,:00      ] Clear E
035  D1CB F28CC0      JP        :C08C      ] Compare if both nrs same
036      *                                     sign
037  D1CE C3A4C0      JMP      :C0A4      ] Else: abort
038      *
039      *****
040      * RESET *
041      *****
042      *
043      * - Now switches off the volume of sound channel 2
044      *   during initialisation. This seems useless,
045      *   because in C76C all sound is already switched
046      *   off.
047      * - The header is changed to 'BASIC V1.1'
048      *
049      *           ORG      :C789
050  C789 32E401      STA      :01E4      ] Volume SCB2 = 0
051  C78C 328F02      STA      :028F
052  C78F 117502      LXI      D,:0275
053  C792 218F02      LXI      H,:028F
054  C795 CD7CDE      CALL     :DE7C
055  C798 FB          EI
056  C799 CF          RST      1
057  C79A 15          DATA   :15
058      *
059      *           ORG      :C7DD
060  C7DD 31          DATA   :31      ] Header changed to V1.1
061      *
062      *
063      *

```

```

064 *****
065 * START FROM SCRATCH *
066 *****
067 *
068 * Added is the reset of all keyboard pointers at
069 * various moments:
070 * - When restarting the Basic interpreter.
071 * - After command execution. Then also the output
072 * direction is reset to screen (to avoid a not
073 * useable keyboard).
074 * - Idem before executing 'break'.
075 * Resetting the keyboard pointers clears the key
076 * input buffer in order to avoid keybounce.
077 *
078                    ORG     :C827
079 C827 CD63D5        CALL    :D563        ] Keyb. pntrs to default
080 C82A 210000        LXI     H, :0000
081 C82D 220001        SHLD   :0100
082 C830 220401        SHLD   :0104
083 C833 221301        SHLD   :0113
084 C836 7C            MOV    A,H
085 C837 322201        STA    :0122
086 *
087                    ORG     :C897
088 C897 21C402        LXI     H, :02C4        ]
089 C89A 7E            MOV    A,M            ] Get break flag
090 C89B B7            ORA    A
091 C89C CA7FCB        JZ     :C87F
092 C89F CD63D5        CALL   :D563        ] Keyb.pntrs to default
093 C8A2 AF            XRA    A            ]
094 C8A3 323101        STA    :0131        ] Output to screen
095 *
096                    ORG     :C8AC
097 C8AC CA9FCB        JZ     :C89F        ] Keyb.pntrs to default,
098                                                            output to screen
099 *
100 *****
101 * EMERGENCY STOP ROUTINE *
102 *****
103 *
104 * No change in execution, only using a new entry on
105 * 'Run NEW'.
106 *
107                    ORG     :CA25
108 CA25 CDBODE        CALL   :DEB0        Run NEW with default heap
109 *
110 *****
111 * RUN NEW *
112 *****
113 *
114 * - An additional entry is made for run NEW with
115 * returning the heapsize to its default value.
116 * In BASIC V1.0, this entry was available on
117 * #CEC6.
118 * - All jumps/calls to RNEW with other heap sizes
119 * are updated.
120 *
121                    ORG     :DEB0
122 DEB0 210001        LXI     H, :0100        ]
123 DEB3 229D02        SHLD   :029D        ] Set heap to default size
124 DEB6 00            NOP
125 DEB7 00            NOP                    Into RNEW

```

```

126          *
127          ORG      :C778
128 C778 CDB8DE CALL   :DEB8      !
129          *
130          ORG      :CF02
131 CF02 B8DE   DBL    :DEB8      !
132          *
133          ORG      :D2A9
134 D2A9 CDB8DE CALL   :DEB8      !
135          *
136          *****
137          * RUN SAVE *
138          *****
139          *
140          * Instructions only moved.
141          *
142          ORG      :D257
143 D257 3E30   MVI    A,:30
144 D259 CDC502 CALL   :02C5
145 D25C E1     POP    H
146 D25D D1     POP    D
147 D25E CDC802 CALL   :02CB
148 D261 D1     POP    D
149 D262 C3D8D7 JMP    :D7DB
150          *
151          *****
152          * RUN LOAD *
153          *****
154          *
155          * Instructions only moved.
156          *
157          ORG      :D274
158 D274 CD23CB CALL   :CB23
159 D277 CD91E7 CALL   :E791
160 D27A C5     PUSH   B
161          *
162          ORG      :CF16
163 CF16 74D2   DBL    :D274      !
164          *
165          *****
166          * RUN CHECK *
167          *****
168          *
169          * CALL :D7E6 is replaced by the contents of D7E6.
170          * No change in execution.
171          *
172          ORG      :D2C9
173 D2C9 CDCE02 CALL   :02CE
174 D2CC FE33   CPI    :33
175 D2CE D2EBD2 JNC    :D2EB
176 D2D1 0C     INR    C
177 D2D2 3E00   MVI    A,:00      ] Old contents
178 D2D4 CDD702 CALL   :02D7      ] D7E6
179 D2D7 00     NOP
180          *
181          *****
182          * PRINT MESSAGE ON NEW LINE *
183          *****
184          *
185          * New routine. It moves the cursor to a new line
186          * before a message is printed. Used for printing
          * 'STOPPED IN LINE ...' and 'END PROGRAM'.

```

```
188                               *
189                                ORG     :D7E6
190 D7E6 CD55DD                   CALL   :DD55         ] Cursor to column 0
191 D7E9 C3FFDA                   JMP     :DAFF         ] Print message
192
193                               * RSTOP:
194
195                                ORG     :DF03
196 DF03 CDE6D7                   CALL   :D7E6         ]
197
198                               * REND:
199
200                                ORG     :DF0C
201 DF0C CDE6D7                   CALL   :D7E6         ]
202
203                               *
204                               *****
205                               * FAST PRINTING *
206                               *****
207                               *
208                               * Instruction only moved.
209                               *
210                                ORG     :D7EC
211 D7EC E5                         PUSH   H
212                               *
213                                ORG     :D3CC
214 D3CC C4ECD7                   CNZ     :D7EC         !
215                               *
216                                ORG     :D78D
217 D78D C3ECD7                   JMP     :D7EC         !
218                               *
219                               *****
220                               * READ BLOCK FROM TAPE *
221                               *****
222                               *
223                               * Now cassette motors are switched off directly
224                               * after reading is finished.
225                               * Especially in LOADA, switching off the cassette
226                               * motors after re-arranging all data was very
227                               * annoying.
228                               *
229                                ORG     :D89A
230 D89A C365D2                   JMP     :D265         ]
231                               *
232                                ORG     :D265
233 D265 D2B3D2                   JNC     :D2B3         ] Evt run loading error
234 D268 C3D402                   JMP     :02D4         ] Stop cassette motors
235                               *
236                               *****
237                               * INPUT TEXT LINE *
238                               *****
239                               *
240                               * Added: Output back to screen if break pressed.
241                               * This avoids a not-useable keyboard on return.
242                               *
243                                ORG     :DD2D
244 DD2D DA36CD                   JC      :CD36         ] Jump if break pressed
245                               *
246                                ORG     :CD36
247 CD36 AF                       XRA     A             ] A=0
248 CD37 C35AD5                   JMP     :D55A         ]
249 CD3A FF                       DATA    :FF           ]
250 CD3B FF                       DATA    :FF           ]
```

```

250          *
251          ORG      :D55A
252 D55A 323101    STA      :0131      ] Output to screen
253 D55D C349DD    JMP      :DD49      ] Ignore line if break
254                                     pressed
255          *
256          *****
257          * MULTIPLY HL * A *
258          *****
259          *
260          * Routine only moved.
261          *
262          ORG      :DE9C
263 DE9C DAA9DE    JC      :DEA9      !
264 DE9F B7        ORA      A
265 DEA0 CAACDE    JZ      :DEAC      !
266 DEA3 EB        XCHG
267 DEA4 29        DAD      H
268 DEA5 EB        XCHG
269 DEA6 D296DE    JNC     :DE96
270 DEA9 D1        POP      D
271 DEAA F1        POP      PSW
272 DEAB C9        RET
273 DEAC D1        POP      D
274 DEAD F1        POP      PSW
275 DEAE 3F        CMC
276 DEAF C9        RET
277          *
278          *****
279          * RUN RUN *
280          *****
281          *
282          * The sequence of the instructions is modified.
283          * This enables RUN <linenr> to be used without
284          * destroying the contents of heap and symtab.
285          *
286          ORG      :DF9E
287 DF9E CD23CB    CALL     :CB23      Empty heap and symtab
288 DFA1 2A9F02    LHLD    :029F      Get start textbuf
289 DFA4 44        MOV     B,H          ) in BC
290 DFA5 4D        MOV     C,L          )
291 DFA6 CD01E4    CALL     :E401      Run RESTORE
292 DFA9 210000    LXI     H,:0000
293 DFAC 221501    SHLD   :0115      Reset step/trace flag
294 DFAF AF        XRA     A
295 DFBO 322601    STA     :0126      No suspended program
296 DFB3 3100F9    LXI     SP,:F900    Reset SP
297
298          * RUN <linenr> :
299
300          ORG      :DFBD
301 DFBD C3A4DF    JMP     :DFA4      ! Process RUN
302          *
303          *****
304          * STORE DATA (READ, INPUT) *
305          *****
306          *
307          * Handle string inputs (no change):
308          *
309          ORG      :E385
310 E385 CCB8E4    CZ      :E4B8      ] Only JZ :E3A2 replaced
311 E388 C4B4E4    CNZ     :E4B4      ]           by its contents

```

```

312
313 * Error exit changed: Now checks if end of input
314 * is reached. EFEPT is only updated if not running
315 * inputs (anymore).
316
317 ORG :E39C
318 E39C CA67E3 JZ :E367 ] Read next data if no error
319 E39F 3A1701 LDA :0117 If error: Get 'run-input'
320 flag
321 E3A2 CDC3E3 CALL :E3C3 ]
322 E3A5 C30BDA JMP :DA0B ] Run 'SYNTAX ERROR'
323 *
324 ORG :E3C3
325 E3C3 A7 ANA A ] Set flags
326 E3C4 C0 RNZ ] Abort if running inputs
327 E3C5 2A3201 LHLD :0132 Get EFEPT
328 E3C8 11FCFF LXI D,:FFFC
329 E3CB 19 DAD D -4
330 E3CC 220001 SHLD :0100 Set start current line
331 E3CF C9 RET
332 *
333 *****
334 * PREPARE GETTING INPUTS *
335 *****
336 *
337 * The keyboard pointers are set to their default
338 * values before inputs are asked. This avoids
339 * keybounce.
340 *
341 ORG :E447
342 E447 221D01 SHLD :011D
343 E44A CD63D5 CALL :D563 ] Init keyb ptrs
344 E44D 3EFF MVI A,:FF
345 E44F 321701 STA :0117
346 E452 C3D0E3 JMP :E3D0
347 *
348 *****
349 * cont. of 0E401 *
350 *****
351 *
352 * Instruction only moved.
353 *
354 E455 322301 STA :0123
355 E458 C9 RET
356 E459 FF DATA :FF
357 *
358 ORG :E409
359 E409 C355E4 JMP :E455 !
360 *
361 *****
362 * RUN CLEAR *
363 *****
364 *
365 * Routine is completely modified.
366 * Max. useable heap space in V1.0 was #7FFF-4.
367 * Now it is #7FFF.
368 * Doesnot set up anymore a complete new heap, but
369 * just empties heap and symboltable entries and
370 * shifts the program to after the new heap.
371 *
372 ORG :E6B5
373 E6B5 CDF8E6 CALL :E6FB ] Get reqd space in HL

```

```

374 E6B8 E5          PUSH  H          ] Preserve it
375 E6B9 7C          MOV   A,H         ] Get hi byte in A
376 E6BA 2B          DCX  H           ]
377 E6BB 2B          DCX  H           ]
378 E6BC 2B          DCX  H           ]
379 E6BD 2B          DCX  H           ] Reqd space -4
380 E6BE B4          ORA  H           ]
381 E6BF FA15DA      JM   :DA15       ] Run 'NUMBER OUT OF RANGE'
382                                     error if < 4 or > 32K.
383 E6C2 D1          POP   D          ] Get reqd space in DE
384 E6C3 2A9D02      LHLD :029D       ] Get old heapsize
385 E6C6 EB          XCHG              ]
386 E6C7 229D02      SHLD :029D       ] Store new heapsize
387 E6CA C314D2      JMP  :D214       ]
388 E6CD FF          DATA :FF       ]
389
390
391 D214 CD1ADE      CALL :DE1A       ] Calc difference old/new
392 D217 E5          PUSH H           ] Preserve result
393 D218 CD23CB      CALL :CB23       ] Empty heap and symtab,
394                                     move program to after heap
395 D21B 2A0001      LHLD :0100       ] Get pntr to current line
396 D21E 7C          MOV   A,H         ]
397 D21F B5          ORA  L           ]
398 D220 D1          POP   D          ] Get difference
399 D221 C8          RZ              ] Abort if direct cmd
400 D222 19          DAD  D           ] Add difference to pntr
401                                     current line
402 D223 C3BBCE      JMP  :CEBB       ]
403 D226 FF          DATA :FF       ]
404
405
406 CEBC 220001      SHLD :0100       ] Update pntr
407 CEBE CD01E4      CALL :E401       ] Run RESTORE
408 CEC1 D5          PUSH D          ] Preserve difference
409 CEC2 C37FD8      JMP  :D87F       ]
410
411
412 D87F E1          POP   H          ] Get difference
413 D880 09          DAD  B           ] Add textpntr
414 D881 44          MOV  B,H         ] New textpntr in BC
415 D882 4D          MOV  C,L         ]
416 D883 B7          ORA  A           ] No special action
417 D884 C9          RET              ]
418 D885 FF          DATA :FF       ]
419
420
421
422
423
424
425
426
427
428 E9BE CDF0ED      ORG   :E9BE
429                                     CALL  :EDF0       ] Another calculation
430                                     routine is used
431
432
433
434
435

```

\*  
\*\*\*\*\*  
\* RUN A VARIABLE POINTER \*  
\*\*\*\*\*  
\*  
\* This enables the use of DIM (255,255). In V1.0,  
\* the max. dimension was 254.  
\*  
\*  
\* Better entry to keyboard scan routine to

```

436          * avoid keybounce.
437          *
438          ORG      :EB75
439 EB75 AF      XRA      A          ]
440 EB76 32B902 STA      :02B9      ] Clear breakflag
441 EB79 CDBED6 CALL     :D6BE      ] Run GETC
442          *
443          ORG      :EA04
444 EA04 756B    DBL      :6B75      !
445          *
446          *****
447          * RUN TAB *
448          *****
449          *
450          * The old routine worked only for DOUTC=0. Now
451          * DOUTC is not checked anymore, but the Z-flag
452          * set by CE60 is evaluated. If Z=1 (all O.K.),
453          * the TAB-instruction is executed. If Z=0, then
454          * the number of tab's is not correct. Then only
455          * one space is printed.
456          *
457          ORG      :EAA5
458 EAA5 1F      RAR          ] Dummy to preserve Z-flag
459          *
460          *****
461          * RUN INT *
462          *****
463          *
464          * Running INT on a number with value 0 gave as
465          * result: -1. Now this failure is corrected.
466          *
467          ORG      :EB90
468 EB90 212901 LXI      H, :0129      ]
469 EB93 E7      RST      4          ] Copy MACC to WORKE
470 EB94 0F      DATA   :0F          ]
471 EB95 C1      POP      B          ]
472 EB96 E7      RST      4          ] MACC = INT (MACC)
473 EB97 1E      DATA   :1E          ] for FPT number
474 EB98 B7      ORA      A          ] Set flags on exp byte
475 EB99 F0      RP          ] Ready if nr positive
476 EB9A C3C5CE JMP      :CEC5      ]
477          *
478          ORG      :CEC5
479 CEC5 CD0CC0 CALL     :C00C      ] FPT compare nrs in MACC
480          ] and in WORKE
481 CEC8 21F1D0 LXI      H, :D0F1      ] Addr FPT (-1)
482 CECB C8      RZ          ] Ready if nr = 0
483 CECC E7      RST      4          ] Add (-1) to MACC if nr <0
484 CECD 00      DATA   :00          ]
485 CECE C9      RET          ]
486          *
487          *****
488          * part of RUN TALK *
489          *****
490          *
491          * A bug is removed, caused by XCHG in #EC71. The
492          * return from #EC78 (JMP CD67) could not be executed
493          * correctly, because of a wrong address stored in
494          * the HL-registers. This problem existed only for
495          * the 'TALK'-codes #0C (wait) and #0D (ML call).
496          *
497          * Entry: DE: Wait time or address ML-routine.

```



```

498          *
499          ORG   :EC71
500 EC71 CCCCCA      CZ   :DACC      ] If to be waited
501 EC74 C462CD      CNZ  :CD62      ] Goto MLP address
502 EC77 C367CD      JMP  :CD67      Handle next code
503 EC7A FF          DATA :FF
504
505          * If wait:
506          * As old routine, but HL replaced by DE.
507
508          ORG   :DACC
509 DACC 1B          DCX  D           ]
510 DACD 7A          MOV  A,D        ]
511 DACE B3          ORA  E           ]
512 DACF C2CCDA      JNZ  :DACC      ] Wait for DE=0
513 DAD2 C9          RET             ]
514 DAD3 FF          DATA :FF      ]
515
516          * To enable ML call:
517
518          ORG   :CD62
519 CD62 D5          PUSH D         ] Address MLP on stack
520 CD63 C9          RET             ] Go to it
521
522          *
523          *****
524          * TEST A FPT VARIABLE *
525          *****
526          *
527          * The old routine tested all bytes of the FPT nr.
528          * Now only the exponent byte and the hibernate of the
529          * mantissa is tested. In this way, very small FPT
530          * numbers are considered to be zero.
531          *
532          ORG   :EC91
533 EC91 CA98EC      JZ   :EC98      ] Abort if A and B are 0
534 EC94 7B          MOV  A,E        ] Exp byte in A
535 EC95 F601        ORI  :01        ] Clear CY-flag
536 EC97 00          NOP             ]
537
538          *
539          * =====
540          ***** ROM BANK 1 *****
541          * =====
542          *
543          *****
544          * FPT EXP *
545          *****
546          *
547          * Test for overflow is modified. Overflow occurs
548          * for e^X when -45 < X < 43.6.
549          * This is checked now before exponent routine is
550          * entered.
551          *
552          ORG   :E688
553 E688 CAC9EF      JZ   :EFC9      ]
554
555          *
556          ORG   :EFC9
557 EFC9 7A          MOV  A,D        ] Get lobyte MACC
558 EFCA E6C0        ANI  :C0        ] Check for max. value
559 EFCC C28BE6      JNZ  :E688      ] Jump if nr too big
560 EFCE C394E6      JMP  :E694      ] If D.K.

```

```

560 *
561 *****
562 * LOADA *
563 *****
564 *
565 * Switching off the cassette motors is now part of
566 * D897. So the cassette motors are switched off
567 * directly after the reading from tape is done.
568 *
569 *          ORG    :EE6B
570 EE6B C332EE *          JMP    :EE32    ] Adapted to new situation
571 *
572 *
573 *          =====
574 ***** ROM BANK 2 *****
575 *          =====
576 *
577 *
578 *****
579 * WINDOW DOWN *
580 *****
581 *
582 * A bug which let the cursor disappear sometimes is
583 * removed.
584 *
585 *          ORG    :ECB6
586 ECB6 2AA900 *          LHLD   :00A9    ] Get offset top of window
587 *                                     from start buffer
588 ECB9 EB *          XCHG   ] in DE
589 ECBA CD6BD2 *          CALL   :D26B    ]
590 ECBD FCABED *          CM     :EDAB    Then cursor down
591 *
592 *          ORG    :D26B
593 D26B 2AAC00 *          LHLD   :00AC    ] Get Y-offset cursor in
594 *                                     document
595 D26E CDF2E6 *          CALL   :E6F2    ] HL=HL-DE
596 D271 2D *          DCR    L      ] -1
597 D272 37 *          STC     ] Set 'window changed' flag
598 D273 C9 *          RET     ]
599 *
600 *
601 *          =====
602 ***** ROM BANK 3 *****
603 *          =====
604 *
605 *
606 *****
607 * ERROR EXIT 'ENCODE INT NR INTO EBUF *
608 *****
609 *
610 *          ORG    :EBB7
611 EBB7 219FE3 *          LXI    H,:E39F  ] (0) Addr routine 'STORE
612 *                                     DATA'
613 EBB8 E5 *          PUSH   H      ] Preserve it as returnaddr
614 EBBB 3A4000 *          LDA    :0040  ] Get POROM
615 EBBE E63F *          ANI    :3F    ] Select bank 0
616 EBC0 F5 *          PUSH   PSW    ]
617 EBC1 C3E6C6 *          JMP    :C6E6  ] Bank return
618 *
619 *
620 *
621 EBC4 *          END

```