Support Group Application Note
Number: 028
Issue: 1
Author:

ISSUE II BASIC

Applicable Hardware :
BBC B

Related Application Notes:

Support Group
Acorn Computers Limited
Acorn House
Vision Park
Histon
Cambridge CB4 4AE

The new version of BASIC's differences with the old version are as follows:

1 ELSE no longer leaves a byte on the hardware stack in ON. . . GOTO/GOSUB.

2 INSTR no longer leaves the shorter string of INSTR ("AB", "ABC") on the Software stack.

3 The lexical analyser is now called correctly from EVAL so that EVAL("TIME") works.

4 ABS can take the absolute value of integers without bit 31 set without returning a string (ie PRINT -ABS 1 works).

5 ASC" : " can be used in the assembler without confusing it.

6 A new statement OSCLI has been introduced, shortest abbreviation OS., token value &FF. It takes a string expression and gives it to the operating system, eg OSCLI "KEY "+STR$Z+" This is key "STR$Z. It has no unique errors of its own, just the normal "Type mismatch" error.

7 The value of the token for OPENIN has changed to &BE and now just opens files for input only using the &40 open. A new keyboard openup (open for update) has been introduced with the old taken value &AD which opens files for update using the &CO open. this results in old programs automatically changing to OPENUP when they are LOADed into the new interpreter.

8 The next bit in OPT (OPTs 4 to 7) controls whether code generated by the assembler will be put at the program counter P% or at the code original O%. If it is set code will be put at O% and both O% and P% incremented, if unset only P% will be used. Current implementation results in all OPTs greater than 3 causing code to be put at O%.

9 The LN and LOG functions are completely recoded to make them more accurate and to avoid the LN(2E-39) bug.

10 The fix routine avoids the INT1E38 bug.

11 The SIN/COS functions are completely recoded to make them more accurate.

12 The binary to decimal string conversion routines, used for PRINT and STR$, have been changed to allow the use of 10 figures of precision on printing. The initial value of @% is now &0000090A to give the same results on startup. Note that @%=10 now gives the internal default of 10 figures. The changes allow the maximum positive integer 214783647 to be printed out (and indeed get 2^32 right). STR$ when not controlled by @% uses the new 10 figure default which will result in it giving different answers to before eg 7.7 (a recurring binary fraction) will be converted to 7.699999999.

13 Four new operations are available in the assembler. These are EQUB, EQUW, EQUD, EQUS. They have no errors of their own apart from the Type Mismatch error. They take a single argument and put its value into the assembly code. EQUS puts all the characters of a string into the code WITHOUT a carriage return like $P%; if you need a carriage return just add it to the string. EQUS may be used together with FN to implement macros as shown in this example:

```
DEF FNOSBYTE  (A,X,Y)
IF A>127  [OPT Z:LDY£Y:1]
[OPT Z:LDX£X:LDA£A:JSR&FFF4:]
=""

[OPT Z
EQUS FNOSBYTE(&99,33,44)
EQUS FNOSBYTE(2,1,0)
]
```

14 The startup REPORT is the way to tell the new version of BASIC from the old one, the new one has (c) 1982.

15 The bug associated with ON ERROR GOTO 9999 has been removed.

16 A MODE change now resets COUNT.

17 BASIC will now only execute if A contains 1 on entry at 8000. There is no entry at 8003 any more.

18 BASIC is of type 60; this means that it contains its own tube relocation address after all the standard sideways ROM items. Of course, with the general release this will be 00008000 (the default for no relocation) but this gives more trouble free production of relocated versions to run at 0000B000.

19 The version number is 01.

20 In INPUT ';' is introduced and functions as the ',' does.

21 Fatal errors have been introduced. Errors whose ERR is zero causes an ON ERROR OFF effect while they are being processed.

22 STOP has been redefined as a fatal error, this causes the STOP at line 0 message to be corrected to STOP.

23 The "No room" error is a fatal error.

24 The standard error handling procedures do not use stack any more. The situation of running out of all free space no longer causes error messages to be printed out following by a "No room" message.

25 The allocation of space for strings has been made slightly more efficient, it can cope with REPEAT A$=A$+"*": UNTIL LEN A$ = 255 and only allocate 255 bytes. It still has problems if B$ is being done alternately with A$.

26 A new error 45 "Missing £" arrives if PTR, EOF, BGET, BPUT, EXT have a missing £.

27 DIM P% -2 gives Bad DIM.