TRS8BIT'S

SELECTED ARTICLES SERIES

THE BASIC FAMILY PEDIGREE

BY

JIM ADAMS

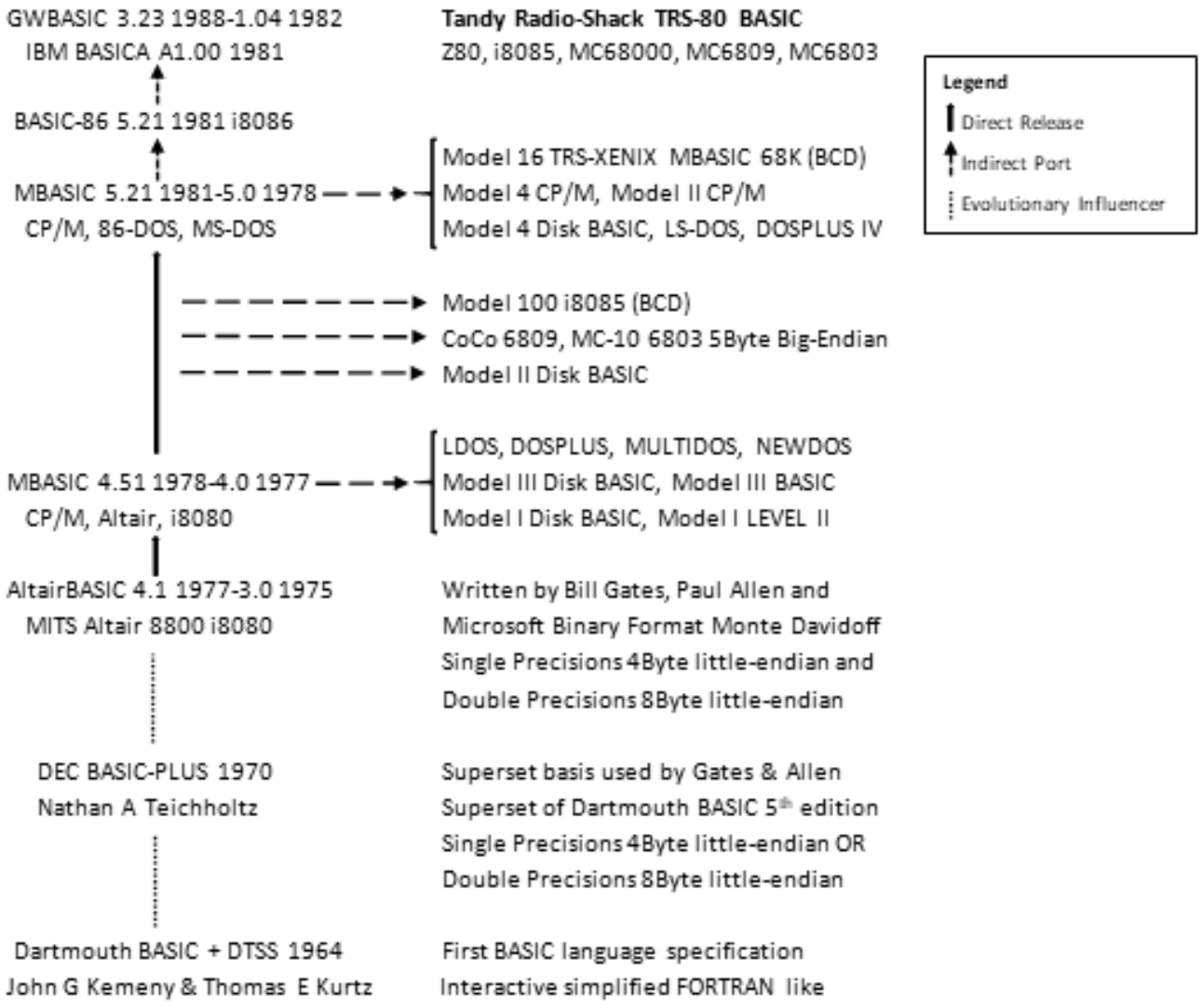
The BASIC Family Pedigree

Jim Adams

Most of us have used BASIC on one or more TRS-80 microcomputer and noticed how each TRS-80 BASIC dialect is very similar yet different from the other. Have you ever wondered about their common origin? Some might have read about *how* the TRS-80 BASIC interpreters came about, while few others already read about its history, and hopefully, after reading this article, few more will become curious to enjoy reading the historical tales. A quick search online yields numerous sites, some with factual events and many with entertaining antidotal stories of how BASIC, TRS-80, TRS-80 BASIC have started.

So I will not rehash the all the history here, instead I will focus on how the BASIC language, in particular the commands, statements, functions and operators, have evolved over time while highlighting few historical events from 1963 to 1990 that greatly influenced the TRS-80 BASIC.

Microsoft BASIC Interpreters Pedigree (Family Ancestry)



To this end, I started searching for firsthand sources, such as biographies, books, articles, and interviews by the original authors or key technologists such as Bill Gates, Paul Allen, Monte Davidoff, Ed Roberts, Nate Teichholtz (since 1984 legally known as Nathan Brookwood), Thomas Kurtz and John Kemeny. I also dived through numerous BASIC reference manuals. Finally, when in doubt about a feature or a topic and the manuals were vague, sometimes intentionally as BASIC documentation was intended for ease of learning not technical completeness, I programmatically validated them using real TRS-80s (I/III/4/4P, II/12/16/16B/6000, 100, CoCo1/2/3), and emulators (trs80gp, Xroar, VirtualT, AltairZ80, Z80Pack, SIMH, pcjs.org, PCE-ibmpc, DOSBox). *[I have done my best to ensure the information is correct, but I am fully aware of my limited knowledge, so please forgive any unintentional errors].*

Now, together we will explore the dawn of the BASIC language and its evolution that led to the most popular family of BASIC interpreters, the TRS-80 BASIC.

Step 0: 1963 BASIC, the Teaching Computer Language

BASIC is an acronym for Beginner's All-purpose Symbolic Instruction Code. It was envisioned, created and designed by two Dartmouth professors of Mathematics, Prof. John G Kemeny and Prof. Thomas E Kurtz. Prof Kemeny has sought to create an easy to learn computer language so computer illiterate undergraduate/high school students can quickly grasp it and utilize the computer to pursue their academic endeavors, ie math calculations, statistics problems, humanity correlations, etc. After two prior attempts with DARSIMCO in 1956 and DOPE in 1962, they garnered the lessons learned and applied them in 1963 to create what became the specifications for the BASIC language. BASIC is simple by design, influenced by FORTRAN and ALGOL but without the unnecessary constrains and complexities.

Step 1: 1964 Dartmouth BASIC

Dartmouth Time Sharing System DTSS ***Commands (14)***

BYE, CAT, HELLO, LIST, LIST-xxxx, NEW, OLD, RENAME, RUN, SAVE, SCRATCH, STOP, SYSTEM, UNSAVE
variations +2: GOODBYE, CATALOG
introduced the READY prompt and the 6-character filenames

Statements (15)

DATA, DEF FN*a*, DIM, END, FOR-TO-STEP, GOSUB, GOTO, IF-THEN, LET, NEXT, PRINT, READ, REM, RETURN, STOP
variations +2: GO SUB, GO TO

Functions (11)

ABS(), ATN(), COS(), EXP(), FN*a*(), INT(), LOG(), RND(), SIN(), SQR(), TAN()

Operators (14)

( ), ^, unary + -, * /, + -, < <= = <> >= >
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (1, 1)

variables default to real
arrays default to real()

The first BASIC program ran successfully on 5/1/1964 at 4am at Dartmouth College, hence validating the BASIC language specification, the interactive Dartmouth BASIC language "interpreter" and the Dartmouth Time-Sharing System operating environment. The very first successfully executed two-line program was:

```
10 PRINT 2+2
20 END
```

In February of 1964, Dartmouth installed a General Electric GE-225 20bit 16K word mainframe and connected to a DN-30 18bit 16K word front-end processor; by June of 1964, GE upgraded it to a GE-235 20bit 16K word mainframe that had three times faster core memory.

Dartmouth BASIC first edition, May 1964, was designed for simplicity, intended to be self-taught by students quickly. The DTSS immediate direct commands were entered at the READY prompt without line numbers. The BASIC language statements and functions were entered as program lines each proceeded with a line number from 1 to 99999. The BASIC program was executed via the DTSS RUN command and it will execute from the lowest line number until it encounters a line with the END statement. Variables were numeric only. A variable name referenced either a simple scalar, a single dimension array or a two dimension array. May not use the same name as a scalar and an array, eg either A or A() were valid; but both names may not have appeared in the same program.

In addition to infamous READY prompt and all the commands, statements, functions and operators, Dartmouth BASIC first edition also gave us the following familiar BASIC language features: variables defaulted to real floating-point 6-digit significant, arrays defaulted to 0 to 10 subscripts, and PRINT supported the field separators: comma and semicolon. New editions incrementally added more features.

Step 2: 1970 DEC BASIC-PLUS

DEC PDP-11 RSTS-11 ***Commands (22)***

ASSIGN, BYE, CAT, COMPILE, CONT, DEASSIGN, DELETE *start-end*, HELLO, KEY, LENGTH, LIST, LIST *start-end*, LISTNH *start-end*, NEW, OLD, RENAME, REPLACE, RUN, RUNNW, SAVE, TAPE, UNSAVE
variations +2: GOODBYE, CATALOG

Statements (50)

!, CHAIN, CHANGE, CLOSE, DATA, DEF FN*a*, DIM, END, FIELD, FNEND, FOR-STEP-UNTIL, FOR-STEP-WHILE, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT LINE, INPUT LINE#, INPUT#, KILL, LET, LSET, MAT, MAT INPUT, MAT PRINT, MAT READ, NAME, NEXT, ON ERROR GOTO, ON GOSUB, ON GOTO, OPEN, PRINT, PRINT USING, PRINT#, PRINT# USING, PUT, RANDOM, READ, REM, RESTORE, RESUME, RETURN, RSET, SLEEP, STOP, UNLOCK, WAIT
variations +3: GO SUB, GO TO, RANDOMIZE
modifiers +5: FOR, IF, UNLESS, UNTIL, WHILE

Functions (40)

ABS(), ASCII(), ATN(), CHR$(), COS(), CVT%$(), CVT$%(), CVT$F(), CVTF$(), DATE$(), EXF(), EXP(), FIX(), FN*a*(), INSTR(), INT(), INV(), LEFT(), LEN(), LOG(), LOG10(), MAGTAPE(), MID(), NUM$(), POS(), RAD$(), RIGHT(), RND(), SGN(), SIN(), SPACE$(), SQR(), SWAP(), TAB(), TAN(), TIME(), TIME$(), TRN(), VAL(), XLATE()
system values +8: DET, e, ERL, ERR, NUM, NUM2, PI, RECOUNT

Operators (22)

( ), ^, unary + -, * /, + -, < <= = <> >= > ==, NOT, AND, OR XOR, EQV, IMP
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (3, 3)

variables default to real, integers %, string $
arrays default to real(), integer arrays %(), string arrays $()

During the mid and late 1960's the Dartmouth DTSS and BASIC spread across academia, helped with the fact Dartmouth was giving away the source code for free. Dartmouth allowed high schools remote access to DTSS and BASIC, true to form to Kemeny's and Kurtz's vision to allow students to quickly learn a simple computer language, BASIC, and begin utilizing computers for many applications.

Commercializing on the rapid popularity of BASIC, Hewlett Packard (HP) in 1969 introduced the HP 2000A 16bit mini-computer for 16 concurrent users/students with Time-Shared BASIC (modeled after Dartmouth BASIC fourth edition 1968), targeting the midsize business segment and the educational market.

In response to HP 2000A Time-Shared BASIC success which tokenized BASIC statements, Digital Equipment Corporation (DEC) in 1970, introduced the DEC PDP-11 16bit mini-computer for 16 concurrent users/students with the RSTS-11 operating system and the DEC BASIC-PLUS interpreter designed by Nathan A Teichholtz. To distinguish its product from the competition, DEC BASIC-PLUS specifications added few enhancements to Dartmouth BASIC 5th edition 1969-1970. DEC also ported DEC BASIC-PLUS to its PDP-10 36bit mainframe product line, used by many universities that also offered remote access for high schools.

DEC BASIC-PLUS included more familiar BASIC language features: keywords do not require spaces as delimiters, immediate commands may be entered as a program line with line number (eg 1000 RUN "PGM2"), BASIC statements may be entered as immediate commands without line numbers (eg PRINT 2+2), string variables with $, integer variables with %, variable name plus the data type form a unique variable (eg A, A$, A%, A(), A$(), A$() are all different names and reference different data variables), all variables are initialized to 0 or "", ERR for error number, and ERL for error line number. Later editions incrementally added more features.

A Monumental Step

DEC BASIC-PLUS is a significant stepping stone towards the origin of the TRS-80 BASIC. Already we can see the foundation of the majority of TRS-80 BASIC syntax and features embodied in DEC BASIC-PLUS; and indeed, your patience for reading through the origin of the BASIC language is about to pay off high dividends.

By the fall of 1969, far away from Hanover, New Hampshire (Dartmouth BASIC) and Maynard, Massachusetts (DEC BASIC-PLUS), but at Seattle, Washington (Lakeside private school, grades 5-12), is where the 8th grader William H Gates and the 10th grader Paul G Allen met while they taught themselves programming in DEC BASIC on a Teletype Model 33 ASR remotely connected. By 1970, they had access to a DEC PDP-10 mainframe running TOPS-10 time sharing system with DEC BASIC-PLUS. They mastered DEC BASIC-PLUS so well, that by the time Paul graduated high school in 1971 and Bill graduated in 1973, they both have automated Lakeside's class scheduling using DEC BASIC-PLUS.

Step 3: 1975 MITS Altair 8800 BASIC

ALTAIR EXTENDED BASIC 3.2 12K December 1975

Statements/Commands (43)

? , CLEAR, CLOAD, CLOAD?, CONT, CSAVE, DATA, DEF FN*a*, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERASE, FOR-TO-STEP, GOSUB, GOTO, IF-THEN-ELSE, INPUT, LET, LIST, NEW, NEXT, NULL, ON GOSUB, ON GOTO, POKE, PRINT, PRINT USING, READ, REM, RESTORE, RETURN, RUN, STOP, SWAP, TROFF, TRON, WAIT

variations due to optional space delimiters, eg GOTO, GO TO, DEFDBL, DEF DBL, etc

Functions (30)

ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), EXP(), FN*a*(), FRE(), INP(), INT(), LEFT$(), LEN(), LOG(), MID$(), PEEK(), POS(), RIGHT$(), RND(), SGN(), SIN(), SPC(), SQR(), STR$(), TAB(), TAN(), USR*n*(), VAL()

Operators (20)

( ), ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR

ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $

arrays default to single precision real () or !(), double precision real arrays #(), integer arrays %(), string arrays $()

ALTAIR EXTENDED BASIC 4.1 16K July 1977

Statements/Commands (65)

? , ' , AUTO, CLEAR, CLOAD, CLOAR?, CLOAD*, CLOSE, CONT, CSAVE, CSAVE*, DATA, DEF FN*a*, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERASE, ERROR, FOR-TO-STEP, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, LET, LINE INPUT, LINE INPUT#, LIST, LLIST, LPRINT, LPRINT USING, MID$()=, NEW, NEXT, NULL, ON ERROR, ON GOSUB, ON GOTO, OPEN, OUT, POKE, PRINT, PRINT USING, PRINT#, PRINT# USING, READ, REM, RENUM, RESTORE, RESUME, RETURN, RUN, STOP, SWAP, SYSTEM, TROFF, TRON, WAIT, WIDTH

variations due to optional space delimiters, eg GOTO, GO TO, DEFDBL, DEF DBL, etc

Functions (45)

& , &H, &O, ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), EOF(), ERL, ERR, EXP(), FIX(), FN*a*(), FRE(), HEX$(), INKEY$, INP(), INSTR(), INT(), LEFT$(), LEN(), LOG(), LPOS(), MID$(), OCT$(), PEEK(), POS(), RIGHT$(), RND(), SGN(), SIN(), SPACE$(), SPC(), SQR(), STR$(), STRING$(), TAB(), TAN(), USR*n*(), VAL(), VARPTR()

Operators (23)

( ), ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR, XOR, EQV, IMP

ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $

arrays default to single precision real () or !(), double precision real arrays #(), integer arrays %(), string arrays $()

[For ALTAIR DISK EXTENDED BASIC language details, please see the Microsoft BASIC-80 4.51 18K 1977]

With the advent of the Intel 8080 8bit microprocessor in 1974, the late Dr. Ed Roberts, MD, quickly designed the Micro Instrumentation and Telemetry Systems (MITS) Altair 8800 microcomputer, originally as a full kit, then offered as a fully assembled units. The watershed moment was featuring the MITS Altair 8800 on the cover of the January 1975 issue of Radio Electronics magazine, causing thousands of back orders by February 1975. The Intel 8080 was Intel's *first* 8bit general purpose MPU after their *initial* i8008 8bit bit-slicing MPU designed for Datapoint 2200 programmable terminal.

William H Gates, while at Harvard as an undergraduate student, discussed the MITS Altair 8800 with Paul G Allen and they embarked on creating a BASIC interpreter for it. Hence, creating the very first BASIC interpreter for the microcomputer industry. Paul developed an Intel 8080 emulator that ran on Harvard's PDP-10, while Bill and Paul proceeded to create Altair BASIC, drawing from their own experiences with DEC BASIC-PLUS on PDP-10 and PDP-8.

They enlisted Monte Davidoff, another Harvard student to create the floating point math package, which later became known as the Microsoft Binary Format used by all the Microsoft language products from 1975 to 1987 when they switched executively to the IEEE 754-1985 standard. *[Enough reminiscing of my other journey researching the MBF origin, DEC PDP floating point influence, accuracy, limits, etc.]*

Interestingly enough, when Paul Allen demonstrated their BASIC interpreter to Ed Roberts for the first time on a real MITS Altair 8800, the first BASIC statement he executed was PRINT 2+2, the very same first test statement of the Dartmouth BASIC.

MITS Altair 8800 BASIC 2.0 4K 1975 paper tape, became Micro-Soft's (as it was originally known) inaugural product and the first microcomputer BASIC language. New versions 3.0, 3.2, 4.0, 4.1... and releases 8K, EXTENDED BASIC 16K, DISK BASIC 18K added more of the familiar TRS80 BASIC features.

Step 4: 1977 Microsoft BASIC

Microsoft non-disk EXTENDED BASIC-80 4.51 16K 1977

[For non-disk EXTENDED BASIC language details, please see the ALTAIR EXTENDED BASIC 4.1 16K 1977]

Microsoft BASIC-80 4.51 18K 1977 for CP/M

Statements/Commands (77)

?, ', AUTO, CLEAR, CLOAD, CLOAR?, CLOAD*, CLOSE, CONT, CSAVE, CSAVE*, DATA, DEF FN*a*, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, DSKO$, EDIT, END, ERASE, ERROR, FIELD, FILES, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, KILL, LET, LINE INPUT, LINE INPUT#, LIST, LLIST, LOAD, LPRINT, LPRINT USING, LSET, MERGE, MID$()=, NAME, NEW, NEXT, NULL, ON ERROR, ON GOSUB, ON GOTO, OPEN, OUT, POKE, PRINT, PRINT USING, PRINT#, PRINT# USING, PUT, READ, REM, RENUM, RESTORE, RESUME, RETURN, RSET, RUN, SAVE, STOP, SWAP, SYSTEM, TROFF, TRON, WAIT, WIDTH
variations due to optional space delimiters, eg GOTO, GO TO, DEFDBL, DEF DBL, etc

Functions (56)

&, &H, &O, ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), CVD(), CVI(), CVS(), DSKF(), DSKI$(), EOF(), ERL, ERR, ERR(), EXP(), FIX(), FN*a*(), FRE(), HEX$(), INKEY$, INP(), INSTR(),

INT(), LEFT$(), LEN(), LOC(), LOF(), LOG(), LPOS(), MID$(), MKD$(), MKI$(), MKS$(), OCT$(), PEEK(), POS(), RIGHT$(), RND(), SGN(), SIN(), SPACE$(), SPC(), SQR(), STR$(), STRING$(), TAB(), TAN(), USR*n*(), VAL(), VARPTR()

Operators (23)

( ), ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR, XOR, EQV, IMP
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $
arrays default to single precision real () or !(), double precision real arrays #(), integer arrays %(), string arrays $()

Altair EXTENDED BASIC 4.1 and DISK BASIC 4.1 became the basis for Microsoft BASIC-80 for CP/M, or MBASIC, which became the de facto standard for most 8bit microcomputers. In addition to the original Intel 8080, MBASIC 4.1 1977 to 4.51 1977 were ported to most of the popular 8bit MPUs: 8085, Z80, 6800, 6809, 6502; and were customized for the TRS-80, Apple, PET, ATARI, OSI-500, and many more home computers.

MBASIC 4.x and Altair EXTENDED BASIC 4.x/3.2 all support four data types for scalar variables and four data types for multi-dimension arrays, namely single precision floating points (default), double precision floating points, integers and strings. Variable names are upto 2 character significant. Thus allowing 3,848 unique scalar variables and 3,848 unique arrays. These are theoretical maximum since the memory required for the variables storage (Variable List Table) and the assignment statements (Program Statement Table) will exceed the 64KB maximum memory accessible by 8bit MPUs.

[Here is how I derived the maximum variables, please feel free to correct my misunderstanding]
2char significant (1st char A-Z, 2nd char A-Z or 0-9)
26 A-Z + (26 A-Z * (((26 A-Z) + (10 0-9)) ^1)) = 962 unique names
962 * 4 data types = 3,848 unique scalar variables, and 962 * 4 data types = 3,848 unique arrays

The very first TRS-80 Model I LEVEL II BASIC ROM 12K, announced in 1977 but began shipping in March 1978, was indeed a licensed port of the Microsoft EXTENDED BASIC 16K 4.46 1977 reduced to 12K to meet Tandy's memory map requirements.

Microsoft BASIC-80 5.21 24K 1981 for CP/M

Statements/Commands (84)

?, ', AUTO, CALL, CHAIN, CLEAR, CLOAD, CLOSE, COMMON, CONT, CSAVE, DATA, DEF FN*a*, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERASE, ERROR, FIELD, FILES, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, KILL, LET, LINE INPUT, LINE INPUT#, LIST, LLIST, LOAD, LPRINT, LPRINT USING, LSET, MERGE, MID$()=, NAME, NEW, NEXT, NULL, ON ERROR, ON GOSUB, ON GOTO, OPEN, OPTION BASE, OUT, POKE, PRINT, PRINT USING, PRINT#, PRINT# USING, PUT, RANDOMIZE, READ, REM, RENUM, RESET, RESTORE, RESUME, RETURN, RSET, RUN, SAVE, STOP, SWAP, SYSTEM, TROFF, TRON, WAIT, WEND, WHILE, WIDTH, WIDTH LPRINT, WRITE, WRITE#
requires at least a single space as a keyword delimiter

Functions (54)

&, &H, &O, ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), CVD(), CVI(), CVS(), EOF(), ERL, ERR, EXP(), FIX(), FN*a*(), FRE(), HEX$(), INKEY$, INP(), INPUT$(), INSTR(), INT(), LEFT$(), LEN(), LOC(), LOF(), LOG(), LPOS(), MID$(), MKD$(), MKI$(), MKS$(), OCT$(), PEEK(), POS(), RIGHT$(), RND(), SGN(), SIN(), SPACE$(), SPC(), SQR(), STR$(), STRING$(), TAB(), TAN(), USR*n*(), VAL(), VARPTR()

Operators (23)

( ), ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR, XOR, EQV, IMP
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $
arrays default to single precision real () or !(), double precision real arrays #(), integer arrays %(), string arrays $()

MBASIC 5.x, similar to MBASIC 4.x also supports four data types for scalar variables and four data types for multi-dimension arrays, but variables names are upto 40 character significant. Thus allowing 1.5E+63 unique scalar variables and 1.5E+63 unique arrays. These are theoretical maximum since the memory required for the variables exceeds the 64KB maximum memory.

[Here is how I derived the maximum variables, please feel free to correct my misunderstanding]
2char significant (1st char A-Z, 2nd to 40th char A-Z, 0-9 or .)
26 A-Z + (26 A-Z * (((26 A-Z) + (10 0-9) + (1 .)) ^39)) = 3.75E+62 unique names
3.75E+62 * 4 data types = 1.5E+63 unique scalar variables, and
3.75E+62 * 4 data types = 1.5E+63 unique arrays

The Microsoft BASIC-80 for CP/M 1977-1981 *(aka MBASIC)* became the de facto BASIC interpreter standard and its ubiquitous availability on most 8bit microcomputers, fueled Microsoft's growth.

Step 5: 1976 Tiny BASIC

Palo Alto Tiny BASIC 1.77K 1976

Commands (3)

LET *L.*, NEW *N.*, RUN *R.*

Statements (11)

FOR-TO-STEP *F. S.*, GOSUB *GOS.*, GOTO *G.*, IF, INPUT *IN.*, LET *implied*, NEXT *N.*, PRINT *P.*, REM, RETURN *R.*, STOP *S.*
variations are due to the optional *abbreviation* of statements, commands and functions

Functions (3)

ABS() *A.()*, RND() *R.()*, SIZE *S.*

Operators (16)

( ), unary + -, * /, + -, < <= = # >= >, * *AND*, + *OR*
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (1, 1)

variables default to integer
a single array defaults to integer ()

Dennis Allison from Stanford SLAC, and encouraged by Bob Alberecht of the Homebrew Computer Club, began a series of articles to describe and revise the minimum BASIC, starting with the *"BUILD YOUR OWN BASIC"* in the *PEOPLE'S COMPUTER COMPANY Newsletter*, volume 3, number 4, May 1975. Dennis' do it yourself kit for BASIC specifications listed the minimum as: < 4K of memory, integer only math, variables are A-Z, RND() and seven statements (GOSUB, GOTO, IF, INPUT, LET, PRINT, RETURN). This eventually grew into *Dr. Dobb's Journal of Tiny BASIC Calisthenics & Orthodontia*, first issue in January 1976. Dr. Dobb's larger journal format began printing source code of Tiny BASIC interpreters for the intel 8008 and intel 8080.

In the May 1976 issue of *Dr. Dobb's Journal*, volume 1, issue 5, it featured the annotated source code listing of the *"Palo Alto Tiny BASIC"* by Dr. Li Chen Wang for the i8080. This public domain version of Tiny BASIC (aka PATB) met Dennis' specifications and more than doubled the number of statements, double the functions, added a single array @(), all while squeezing it into 1.77KB. Li Chen also documented hocks to expand the parser's accepted statements and functions. PATB influence is found in many later BASIC dialects, most notably, the TRS-80 LEVEL I BASIC and the Sharp PC-1211 S'-BASIC (TRS-80 PC-1).

Step 6: 1977 TRS-80 BASIC

TRS-80 LEVEL I BASIC 4K 1977

Statements/Commands (29)

CLOAD, CLS, CONT *C.*, CSAVE, DATA *D.*, END *E.*, FOR-TO-STEP *F. S.*, GOSUB *GOS.*, GOTO *G.*, IF-THEN *T.*, INPUT *IN.*, LET *implied*, LIST *L.*, NEW *N.*, NEXT *N.*, ON GOSUB, ON GOTO, PRINT *P.*, PRINT AT *P.A.*, READ *REA.*, REM, RESET *R.*, RESTORE *REST.*, RETURN *RET.*, RUN *R.*, SET *S.*, STOP *ST.*

Model III LEVEL I 4K 1980 added: LLIST, LPRINT

variations are due to the optional *abbreviation* of statements, commands and functions

Functions (6)

ABS() *A.()*, INT() *I.()*, MEM *M.*, POINT() *P.()*, RND() *R.()*, TAB *T.()*

Operators (16)

( ), unary + -, * /, + -, < <= = <> >= >, * *AND*, + *OR*

ordered from highest to lowest precedence, commas separate each precedence group

Data Types (2, 1)

variables default to single precision real, string $
arrays default to single precision real ()

TRS-80 LEVEL I BASIC 4K for the Model I was Tandy's port of Li-Chen Wang's public domain Palo Alto Tiny BASIC (*PATB*) 1.77K, published in the May 1976 issue of *Dr. Dobb's Journal*. Tandy's Steven W Leininger added floating point math; support for the Model I hardware I/O (keyboard, video, and cassette); and added more BASIC features (15 statements, 3 functions, 1 string variable) while squeezing it all into a 4K ROM as he ported Tiny BASIC in i8080 to the TRS-80. This was the only release of TRS-80 BASIC that was

not developed by Microsoft and does not conform to the Microsoft's BASIC syntax (lacks EDIT, PRINT USING, MBF floating point format).

LEVEL I BASIC supports two data types for scalar variables and one data type single-dimensions array. 1 character significant variable names and the only variables allowed are A-Z, A$, B$, A().

26 real floating point scalar variables A-Z
2 string scalar variables A$, B$
1 real floating point array A()

TRS-80 Model III LEVEL I 4K 1980 was updated by Tandy and adds two statements, the LLIST, LPRINT.

TRS-80 Pocket Computer 1 (PC-1) 1980, was a rebranded Sharp PC-1211 1980, and had a similar BASIC which was also derived from Li-Chen Wang's PATB but independently developed by Sharp.

TRS-80 LEVEL II BASIC 12K March 1978

Statements/Commands (56)

?, ', *comma*, AUTO, CLEAR, CLOAD, CLOAR?, CLS, CONT, CSAVE, DATA, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERROR, FOR-TO-STEP, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, LET, LIST, LLIST, LPRINT, LPRINT USING, NEW, NEXT, ON ERROR, ON GOSUB, ON GOTO, OUT, POKE, PRINT, PRINT USING, PRINT#, PRINT# USING, PRINT@, RANDOM, READ, REM, RESET, RESTORE, RESUME, RETURN, RUN, SET, STOP, SYSTEM, TROFF, TRON
variations due to optional space delimiters, eg GOTO, GO TO, DEFDBL, DEF DBL, etc

Functions (36)

ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), ERL, ERR, EXP(), FIX(), FRE(), INKEY$, INP(), INT(), LEFT$(), LEN(), LOG(), MEM, MID$(), PEEK(), POINT(), POS(), RIGHT$(), RND(), SGN(), SIN(), SQR(), STR$(), STRING$(), TAB(), TAN(), USR(), VAL(), VARPTR()

Operators (18)

( ), ^, unary + -, * /, + -, < <= = <> >= >, NOT, AND, OR
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $
arrays default to single precision real () or !(), double precision real arrays #(), integer arrays %(), string arrays $()

As Microsoft BASIC-80 spread across the sea of CP/M microcomputers becoming the de facto standard. Consequently, Tandy licensed the Model I LEVEL II BASIC from Microsoft, and proprietary hardware hooks and details furbished by Tandy's own developers. Tandy announced the planned TRS-80 LEVEL II in late 1977 while it shipped the initial orders of the Model I with LEVEL I ROM. LEVEL II BASIC ROM is mostly Microsoft Extended BASIC-80 16K 4.46 reduced to fit into 12K only, per Tandy's memory map specifications.

The TRS-80 LEVEL II BASIC ROM began shipping in March 1978 as an upgrade to existing Model LEVEL I 4K or 16K; upgrade catalog # 26-1120 for $120 including installation. It was also available as new Model I LEVEL II 4K, 26-1054 $689.95 or Model I LEVEL II 16K, 26-1056 $988.95. The final LEVEL II ROM version was 1.3 in 1979.

In 1980, as the Model III was being designed, Tandy internally derived the Model III BASIC ROM 14K 1980 from the Model I LEVEL II BASIC ROM 12K to avoid paying loyalty to Microsoft for the TRS-80 Model III; and to distinguish it as a separate independent product, the Model III BASIC was never officially labeled LEVEL II BASIC but the manuals referred to it as the Model III BASIC. The Model III BASIC ROM was 14K with 2K additional routines to support the lowercase, built in real clock timer and added the TIME$ function.

Microsoft also offered LEVEL III BASIC in 1979, developed by Bill Gates, for the TRS80 Model I LEVEL II. It added functionality removed to reduce Microsoft EXTENDED BASIC-80 16K to LEVEL II BASIC 12K ROM. Similar to the non-disk enhancements included in TRSDOS 2.3 DISK BASIC 2.2 but limited to cassette I/O. It may be loaded from TRSDOS but only supports loading and saving cassette files. It added 14 statements: DEF FN*a*, DEF USR*n*, GET, INPUT#LEN, LINE, LINE INPUT, LINE INPUT#LEN, LOAD, LSET, MID$()=, NAME, PRINT#-3, PUT, SAVE; and 7 functions: &, &H, &O, FN*a*(), INSTR(), TIME$, USR*n*().

George Phillips published TWOHAF in *TRS8BIT* volume 17 issue 4 December 2023, updating Gil Spencer's LEVEL 2.5 that was originally published in the *80 Micro* July 1981 issue 19. George's version adds 3 statements: DEF FN*a*, DEF USR*n*, MID$()=; and 10 functions: &, CVD(), CVI(), CVS(), FN*a*(), INSTR(), MKD(), MKI(), MKS(), USR*n*().

All the LEVEL II BASICs support variable names upto 2 character significant and 4 data types; theoretically allowing 3,848 unique scalar variables and 3,848 unique arrays *[see Microsoft BASIC-80 4.51 1977 for details].*

TRS-80 DISK BASIC 2.2 16K 1978 for TRSDOS 2.3

Statements/Commands (88)

?, ', *comma*, AUTO, CLEAR, CLOAD, CLOAD?, CLOSE, CLS, CMD"A", CMD"B", CMD"C", CMD"D", CMD"E", CMD"I", CMD"J", CMD"L", CMD"O", CMD"P", CMD"R", CMD"S", CMD"T", CMD"X", CMD"Z", CONT, CSAVE, DATA, DEF FN*a*, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERROR, FIELD, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, KILL, LET, LINE INPUT, LINE INPUT#, LIST, LLIST, LOAD, LPRINT, LPRINT USING, LSET, MERGE, MID$()=, NAME, NEW, NEXT, ON ERROR, ON GOSUB, ON GOTO, OPEN, OUT, POKE, PRINT, PRINT USING, PRINT#, PRINT# USING, PRINT@, PUT, RANDOM, READ, REM, RESET, RESTORE, RESUME, RETURN, RSET, RUN, SAVE, SET, STOP, SYSTEM, TROFF, TRON

variations due to optional space delimiters, eg GOTO, GO TO, DEFDBL, DEF DBL, etc

Functions (51)

&, &H, &O, ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), CVD(), CVI(), CVS(), EOF(), ERL, ERR, EXP(), FIX(), FN*a*(), FRE(), INKEY$, INP(), INSTR(), INT(), LEFT$(), LEN(), LOC(), LOF(), LOG(), MEM, MID$(), MKD$(), MKI$(), MKS$(), PEEK(), POINT(), POS(), RIGHT$(), RND(), SGN(), SIN(), SQR(), STR$(), STRING$(), TAB(), TAN(), TIME$, USR*n*(), VAL(), VARPTR()

Operators (18)

( ), ^, unary + -, * /, + -, < <= = <> >= >, NOT, AND, OR

ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $

arrays default to single precision real () or !(), double precision real arrays #(), integer arrays %(), string arrays $()

Tandy also licensed the Model I TRSDOS DISK BASIC (4K addon to the Model I LEVEL II BASIC ROM 12K) from Microsoft with hardware hooks and TRSDOS details added or supplied by Tandy's engineers. TRS-80 DISK BASIC is based mostly on the Microsoft BASIC-80 4.46 for CP/M and the 8080 code base.

In 1980, while designing the TRS-80 Model III, Tandy internally derived the Model III Disk BASIC 1980 (5K addon to Model III BASIC ROM 14K) from the Model I DISK BASIC to avoid paying loyalty to Microsoft.

All BASIC for the TRS-80 Model I, the Model III, and the Model 4 in a Model III mode support variable names upto 2 character significant and 4 data types; theoretically allowing 3,848 unique scalar variables and 3,848 unique arrays *[see Microsoft BASIC-80 4.51 1977 for details]*.

All Disk BASIC for the TRS-80 Model I, the Model III, and the Model 4 in a Model III mode, eg {TRSDOS 2.3, TRSDOS 1.3, MODELA/III, LDOS 5.3.1H, MULTIDOS 5.10, DOSPLUS 3.5, NEWDOS/80 2.5, NEWDOS/90 3.2a, VTOS 4.0 ...}, are a modified port of the Microsoft BASIC-80 4.46 code base.

Step 7: 1983 TRS-80 Model 4 Disk BASIC

TRS-80 Model 4 Disk BASIC 1.1.1 24K 1985 for TRSDOS 6.2.1AR

Statements/Commands (83)

?, ', AUTO, CALL, CHAIN, CLEAR, CLOSE, CLS, COMMON, CONT, DATA, DEF FN*a*, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERASE, ERROR, FIELD, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, KILL, LET, LINE INPUT, LINE INPUT#, LIST, LLIST, LOAD, LPRINT, LPRINT USING, LSET, MERGE, MID$()=, NAME, NEW, NEXT, ON ERROR, ON GOSUB, ON GOTO, OPEN, OPTION BASE, OUT, POKE, PRINT, PRINT USING, PRINT#, PRINT# USING, PRINT@, PRINT@(), PUT, RANDOM, READ, REM, RENUM, RESTORE, RESUME, RETURN, RSET, RUN, SAVE, SOUND, STOP, SWAP, SYSTEM, SYSTEM"*fff*", TROFF, TRON, WAIT, WEND, WHILE, WIDTH, WRITE, WRITE#

requires at least a single space as a keyword delimiter

Functions (59)

&, &H, &O, ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), CVD(), CVI(), CVS(), DATE$, EOF(), ERL, ERR, ERRS$, EXP(), FIX(), FN*a*(), FRE(), HEX$(), INKEY$, INP(), INPUT$(), INSTR(), INT(), LEFT$(), LEN(), LOC(), LOF(), LOG(), LPOS(), MEM, MID$(), MKD$(), MKI$(), MKS$(), OCT$(), PEEK(), POS(), RIGHT$(), RND(), ROW(), SGN(), SIN(), SPACE$(), SPC(), SQR(), STR$(), STRING$(), TAB(), TAN(), TIME$, USR*n*(), VAL(), VARPTR()

Operators (23)

( ), ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR, XOR, EQV, IMP

ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $

arrays default to single precision real () or !(), double precision real arrays #(), integer arrays %(), string arrays $()

The Model 4 was introduced on April 26, 1983 and Tandy licensed a new Disk BASIC from Microsoft. It is a port of a subset of the Microsoft BASIC-80 5.21 1981 for CP/M that was modified for the TRS-80

RSDOS 6 and the Model 4 hardware. In addition, the Model 4 could operate in a Model III mode that also included the unmodified Model III BASIC 14K 1980.

All Model 4 Disk BASICs require space delimiters for keywords, have four data types and support variable names upto 40 character significant; theoretically allowing 1.5E+63 unique scalar variables and 1.5E+63 unique arrays *[see Microsoft BASIC-80 5.21 1981 for details]*.

All Disk BASICs for the TRS-80 Model 4 in bank-switching no ROM mode, eg {TRSDOS 6.2.1AR, LS-DOS 6.3.1H, DOSPLUS IV 1.0, DRI CP/M Plus 1.0, MM CP/M 2.2 2.32M ...}, are a modified port of the Microsoft BASIC-80 5.21 code base.

TRS-80 Model 4 Disk BASIC 1.1.2 24K 1990 for LS-DOS 6.3.1H

In June 1988, Misosys Inc acquired Logical Systems Inc, the original author of TRSDOS 6 and in 1990, MSI released the final update to the TRSDOS 6.2.1AR as the LS-DOS 6.3.1H with the final update of the TRS-80 Model 4 Disk BASIC 1.1.2 27K, which was mostly bug fixes.

TRS-80 Model 4 Microsoft BASIC-80 5.21 24K 1981 for CP/M
Digital Research Inc CP/M Plus BIOS 1.0 1984
or Montezuma Micro CP/M 2.2 BIOS 2.32 1984

[see Microsoft BASIC-80 5.21 1981 for details]

Step 8: 1984 Other TRS-80 BASICs

TRS-80 Model II Disk BASIC 1.2 18K 1983 for TRSDOS-II 4.3

Statements/Commands (70)

? , ' , AUTO, CLEAR, CLOSE, CLS, CONT, DATA, DEF FN*a*, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERASE, ERROR, FIELD, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, KILL, LET, LINE INPUT, LINE INPUT#, LIST, LLIST, LOAD, LPRINT, LPRINT USING, LSET, MERGE, MID$()=, NAME, NEW, NEXT, ON ERROR, ON GOSUB, ON GOTO, OPEN, PRINT, PRINT USING, PRINT#, PRINT# USING, PRINT@, PRINT@(), PUT, RANDOM, READ, REM, RENUM, RESTORE, RESUME, RETURN, RSET, RUN, SAVE, STOP, SWAP, SYSTEM, SYSTEM"*fff*", TROFF, TRON
variations due to optional space delimiters, eg GOTO, GO TO, DEFDBL, DEF DBL, etc

Functions (56)

&, &H, &O, ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), CVD(), CVI(), CVS(), DATE$, EOF(), ERL, ERR, ERR$, EXP(), FIX(), FN*a*(), FRE(), HEX$(), INKEY$, INPUT$(), INSTR(), INT(), LEFT$(), LEN(), LOC(), LOF(), LOG(), MEM, MID$(), MKD$(), MKI$(), MKS$(), OCT$(), POS(), RIGHT$(), RND(), ROW(), SGN(), SIN(), SPACE$(), SPC(), SQR(), STR$(), STRING$(), TAB(), TAN(), TIME$, USR*n*(), VAL(), VARPTR()

Operators (23)

( ), ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR, XOR, EQV, IMP
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $
arrays default to single precision real () or ![], double precision real arrays #[], integer arrays %[], string arrays $[]
[see Microsoft BASIC-80 4.51 1977 for variable names rules, maximum variables and arrays]

Licensed from Microsoft and released with the original Model II TRSDOS in 1979, based on the Microsoft BASIC-80 4.46 1977 and few BASIC-80 5.1 1978. Few interesting facts: lacks POKE, INP[], OUT, PEEK[] per Tandy's request to discourage hacking; defaulted to 0 file buffers while most other BASICs defaulted to 3 file buffers; and the Model II had a cassette port built in but Disk BASIC supports disk I/O exclusively.

Disk BASIC 1.2 1981 initially included with TRSDOS 2.0a and continued to be included, unchanged, upto Disk BASIC 1.2 1983 included with TRSDOS-II 4.3 1983.

TRS-80 Model II Microsoft BASIC-80 5.21 24K 1981 for CP/M
Digital Research Inc CP/M Plus BIOS 1.1 1984
or Pickles & Trout CP/M 2.2 BIOS 2.2mH 1985

[see Microsoft BASIC-80 5.21 1981 for details]

TRS-80 Model 16B/16 TRS-XENIX MBASIC 1.2.0 51K 1984

Statements/Commands (89)

?, ', APPEND, AUTO, BREAK, CALL, CHAIN, CLEAR, CLOSE, CLS, COMMON, CONT, DATA, DATE$=, DEF FN*a*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERASE, ERROR, FIELD, FILES, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, KILL, LET, LINE INPUT, LINE INPUT#, LIST, LLIST, LOAD, LOCK, LPRINT, LPRINT USING, LSET, MERGE, MID$()=, NAME, NEW, NEXT, ON BREAK, ON ERROR, ON GOSUB, ON GOTO, OPEN, OPTION BASE, OUTPUT, POKE, PRINT, PRINT USING, PRINT#, PRINT# USING, PUT, RANDOM, RANDOMIZE, READ, REM, RENUM, RESTORE, RESUME, RETURN, RSET, RUN, SAVE, SHELL, STOP, SWAP, SYSTEM, TIME=, TROFF, TRON, UNLOCK, WAIT, WEND, WHILE, WIDTH, WIDTH LPRINT, WRITE, WRITE#
requires at least a single space as a keyword delimiter

Functions (56)

&, &H, &O, ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), CVD(), CVI(), CVS(), DATE$, EOF(), ERL, ERR, EXP(), FIX(), FN*a*(), FRE(), HEX$(), INKEY$, INPUT$(), INSTR(), INT(), LEFT$(), LEN(), LOC(), LOF(), LOG(), LPOS(), MEM, MID$(), MKD$(), MKI$(), MKS$(), OCT$(), PEEK(), POS(), RIGHT$(), RND(), SGN(), SHELL(), SIN(), SPACE$(), SPC(), SQR(), STR$(), STRING$(), TAB(), TAN(), TIME$, VAL(), VARPTR()

Operators (23)

( ), ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR, XOR, EQV, IMP
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $

arrays default to single precision real () or ![], double precision real arrays #[], integer arrays %[], string arrays $[]

[see Microsoft BASIC-80 5.21 1981 for variable names rules, maximum variables and arrays]

Licensed from Microsoft, a 16bit Motorola 68000 port of the Microsoft BASIC-80 5.21 1981 with added support for the TRS-XENIX and a big-endian adaption of Paul Allen's BCD floating point format *[same as the Model 100 BCD floating points format]*.

TRS-80 Model 100 BASIC 1.1 21K 1983

Statements/Commands (70)

? , ' , AUTO, CLEAR, CLOSE, CLS, CONT, DATA, DEF FN*a*, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, EDIT, END, ERASE, ERROR, FIELD, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, KILL, LET, LINE INPUT, LINE INPUT#, LIST, LLIST, LOAD, LPRINT, LPRINT USING, LSET, MERGE, MID$[]=, NAME, NEW, NEXT, ON ERROR, ON GOSUB, ON GOTO, OPEN, PRINT, PRINT USING, PRINT#, PRINT# USING, PRINT@, PRINT@[], PUT, RANDOM, READ, REM, RENUM, RESTORE, RESUME, RETURN, RSET, RUN, SAVE, STOP, SWAP, SYSTEM, SYSTEM"*fff*", TROFF, TRON

variations due to optional space delimiters, eg GOTO, GO TO, DEFDBL, DEF DBL, etc

Functions (56)

&, &H, &O, ABS[], ASC[], ATN[], CDBL[], CHR$[], CINT[], COS[], CSNG[], CVD[], CVI[], CVS[], DATE$, EOF[], ERL, ERR, ERR$, EXP[], FIX[], FN*a*[], FRE[], HEX$[], INKEY$, INPUT$[], INSTR[], INT[], LEFT$[], LEN[], LOC[], LOF[], LOG[], MEM, MID$[], MKD$[], MKI$[], MKS$[], OCT$[], POS[], RIGHT$[], RND[], ROW[], SGN[], SIN[], SPACE$[], SPC[], SQR[], STR$[], STRING$[], TAB[], TAN[], TIME$, USR*n*[], VAL[], VARPTR[]

Operators (23)

[ ], ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR, XOR, EQV, IMP

ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $

arrays default to single precision real () or ![], double precision real arrays #[], integer arrays %[], string arrays $[]

[see Microsoft BASIC-80 4.51 1977 for variable names rules, maximum variables and arrays]

Licensed from Microsoft, a subset of the Microsoft EXTENDED BASIC 4.51 16K 1977 with added support for the Intel 8085. It earned the honor of being the last product primarily developed by Bill Gates, while Paul Allen implemented the Binary Coded Decimal floating point format *[per Tandy's request, to reduce rounding errors, and improve the financial calculations accuracy]*. The Model 100 Disk Video Interface included Disk BASIC 1.1 25K 1984 which added 3 statements: DSKO$, LFILES, WIDTH; and 3 functions: DSKI$[], LOC[], LOF[].

TRS-80 CoCo Disk Extended Color BASIC 1.1 24K 1983

Statements/Commands (94)

? , ' , AUDIO, BACKUP, CIRCLE, CLEAR, CLOAD, CLOADM, CLOSE, CLS, COLOR, CONT, COPY, CSAVE, CSAVEM, DATA, DEF FN*a*, DEFUSR*n*, DEL, DIM, DIR, DLOAD, DOS, DRAW, DRIVE, DSKI$, DSKINI, DSKO$, EDIT, END, EXEC, FIELD, FILES, FOR-TO-STEP, GET, GOSUB, GOTO, IF-THEN-ELSE,

INPUT, INPUT#, KILL, LET, LINE, LINE INPUT, LIST, LLIST, LOAD, LOADM, LSET, MERGE, MID$()=, MOTOR, NEW, NEXT, ON GOSUB, ON GOTO, OPEN, PAINT, PCLEAR, PCLS, PCOPY, PLAY, PMODE, POKE, PRESET, PRINT, PRINT USING, PRINT#, PRINT# USING, PRINT@, PSET, PUT, READ, REM, RENAME, RENUM, RESET, RESTORE, RETURN, RSET, RUN, SAVE, SAVEM, SCREEN, SET, SKIPF, SOUND, STOP, TIMER=, TROFF, TRON, UNLOAD, VERIFY, WRITE
variations due to optional space delimiters, eg GOTO, GO TO, DEFFN*a*, DEF FN*a*, etc

Functions (44)

&, &H, &O, ABS(), ASC(), ATN(), CHR$(), COS(), CVN(), EOF(), EXP(), FIX(), FN*a*(), FREE(), HEX$(), INKEY$, INSTR(), INT(), JOYSTK(), LEFT$(), LEN(), LOC(), LOF(), LOG(),MEM, MID$(), MKN$(), PEEK(), POINT(), POS(), PPOINT(), RIGHT$(), RND(), SGN(), SIN(), SQR(), STR$(), STRING$(), TAB(), TAN(), TIMER, USR*n*(), VAL(), VARPTR()

Operators (18)

( ), ^, unary + -, * /, + -, < <= = <> >= >, NOT, AND, OR
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (2, 2)

variables default to single precision real, string $
arrays default to single precision real (), string arrays $()

The TRS-80 Color Computer was announced on July 31st, 1980 and shipped during October 1980. It featured the Motorola MC6809E and was available with multiple BASICs licensed from Microsoft: the Color BASIC 1.0 8K 1980, the Extended Color BASIC 1.0 1980 and the Disk Extended Color BASIC 1.0 1981. The combined ECB+CB 16K 1980 were an MC6800/MC6809 port of a subset of the Microsoft EXTENDED BASIC-80 4.51 16K 1977 with added support for the Color Computer graphics and sounds. The optional disk controller cartridge included the DISK EXTENDED COLOR BASIC 1.0 8K 1981 (*aka RSDOS*); combined with ECB+CB 16K would be a subset of the Microsoft BASIC-80 4.51 18K 1977.

Supports two data types for scalar variables and two data types for multi-dimension arrays, namely single precision floating points (default) and strings. Variable names are upto 2 character significant. Thus allowing 1,924 unique scalar variables and 1,924 unique arrays. Floating point reals default to a single precision 5byte in big-endian format.

[Here is how I derived the maximum variables, please feel free to correct my misunderstanding]
2char significant (1st char A-Z, 2nd char A-Z or 0-9)
26 A-Z + (26 A-Z * (((26 A-Z) + (10 0-9)) ^1)) = 962 unique names
962 * 2 data types = 1,924 unique scalar variables, and 962 * 2 data types = 1,924 unique arrays

The Color Computer 3 was introduced in 1986 and included the Super Extended Color BASIC 2.0 1986. To avoid licensing a new BASIC from Microsoft, the CoCo3 included the unmodified EXTENDED COLOR BASIC 16K (ECB 1.1 8K + CB 1.2 8K) and a new 16K ROM by Microware Systems Corp. While booting, the Microware 16K ROM copies the original ECB 1.1 8K + CB 1.2 8K to RAM and updates them to SECB 2.0 with an additional 8K in RAM, for a total of 24K in RAM. In other words, the 32K ROM (ECB 1.1 8K+ CB 1.2 8K + SECB 2.0 16K) are loaded, updated and reduced to 24K RAM. Similarly, if the RSDOS 1.1 floppy controller is plugged in, it is loaded into RAM and updated for a combined Super Disk Extended Color BASIC 2.1 32K 1986 in RAM (or SDECB 2.0 32K 1986 for the earlier DECB 1.0 1981).

SDECB 2.1/SDECB2.0/SECB 2.0 *all* add 22 statements: ATTR, HBUFF, HCIRCLE, HCLS, HCOLOR, HDRAW, HGET, HLINE, HPAINT, HPRINT, HPUT, HRESET, HSCREEN, HSET, HSTAT, LOCATE, LPOKE, ON BRK, ON ERR, PALETTE, PALETTE RGB, WIDTH; and 5 functions: BUTTON(), ERLIN, ERNO, HPOINT(), LPEEK().

TRS-80 MC-10 MicroColor BASIC 1.0 8K 1983

Statements/Commands (37)

? , CLEAR, CLOAD, CLOAD*, CLS, CONT, CSAVE, CSAVE*, DATA, DIM, END, FOR-TO-STEP, GOSUB, GOTO, IF-THEN, INPUT, LET, LIST, LLIST, LPRINT, NEW, NEXT, ON GOSUB, ON GOTO, POKE, PRINT, PRINT@, READ, REM, RESET, RESTORE, RETURN, RUN, SET, SKIPF, SOUND, STOP
variations are the same token, eg GOTO or GO TO becomes GOTO; ? or PRINT becomes PRINT

Functions (23)

ABS(), ASC(), CHR$(), COS(), EXP(), INKEY$, INT(), LEFT$(), LEN(), LOG(), MEM, MID$(), PEEK(), POINT(), RIGHT$(), RND(), SGN(), SIN(), SQR(), STR$(), TAB(), TAN(), VAL()

Operators (18)

( ), ^, unary + -, * /, + -, < <= = <> >= >, NOT, AND, OR
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (2, 2)

variables default to single precision real, string $
arrays default to single precision real (), string arrays $()
[see CoCo DECB 1.1 24K 1983 for variable names rules, maximum variables and arrays]

The TRS-80 MicroColor Computer MC-10 was released in 1983 as a response to ~ $100 microcomputer market, eg Sinclair Z81, and it included the MicroColor BASIC 8K 1983. Licensed from Microsoft as a Motorola MC6800/MC6803 port of a subset of the Microsoft EXTENDED BASIC-80 4.51 16K 1977. It is a functional subset of the Color Computer COLOR BASIC 1.1 1981 combined with few features from the EXTENDED COLOR BASIC 1.0 1980.

Included additional undocumented 2 statements: CLOADM, EXEC; and 1 function: VARPTR().

Step 9: 1988 Beyond the TRS-80 BASICs

Microsoft GW-BASIC 3.23 79K 1988

Statements/Commands (128)

? , ', AUTO, BEEP, BLOAD, BSAVE, CALL, CHAIN, CHDIR, CIRCLE, CLEAR, CLOSE, CLS, COLOR, COM, COMMON, CONT, DATA, DATE$=, DEF FN*a*, DEF SEG, DEF USR*n*, DEFDBL, DEFINT, DEFSNG, DEFSTR, DELETE, DIM, DRAW, EDIT, END, ENVIRON, ERASE, ERROR, FIELD, FILES, FOR-TO-STEP, GET, GET#, GOSUB, GOTO, IF-THEN-ELSE, INPUT, INPUT#, IOCTL, KEY, KILL, LET, LINE, LINE INPUT, LINE INPUT#, LIST, LLIST, LOAD, LOAD"CAS1:, LOCATE, LOCK, LPRINT, LPRINT USING, LSET, MERGE, MID$=, MKDIR, MOTOR, NAME, NEW, NEXT, ON COM, ON ERROR, ON GOSUB, ON GOTO, ON KEY, ON PEN, ON PLAY, ON STRIG, ON TIMER, OPEN, OPEN "COM*n*", OPTION BASE, OUT, PAINT, PALETTE, PALLETE USING, PCOPY, PEN, PLAY, POKE, PRESET, PRINT, PRINT USING, PRINT#, PRINT# USING, PSET, PUT, PUT#, RANDOMIZE, READ, REM, RENUM, RESET, RESTORE, RESUME, RETURN, RMDIR, RSET, RUN, SAVE, SAVE"CAS1:, SCREEN, SHELL, SOUND, STOP, STRIG,

SWAP, SYSTEM, TIME$=, TROFF, TRON, UNLOCK, VIEW, VIEW PRINT, WAIT, WEND, WHILE, WIDTH, WINDOW, WRITE, WRITE#
requires at least a single space as a keyword delimiter

Functions (71)

&, &H, &O, ABS(), ASC(), ATN(), CDBL(), CHR$(), CINT(), COS(), CSNG(), CSRLIN, CVD(), CVI(), CVS(), DATE$, ENVIRON$(), EOF(), ERDEV, ERDEV$, ERL, ERR, EXP(), EXTERR(), FIX(), FN*a*(), FRE(), HEX$(), INKEY$, INP(), INPUT$(), INSTR(), INT(), IOCTL$(), LEFT$(), LEN(), LOC(), LOF(), LOG(), LPOS(), MID$(), MKD$(), MKI$(), MKS$(), OCT$(), PEEK(), PEN(), PLAY(), PMAP(), POINT(), POS(), RIGHT$(), RND(), SCREEN(), SGN(), SIN(), SPACE$(), SPC(), SQR(), STICK(), STR$(), STRIG(), STRING$(), TAB(), TAN(), TIME$, TIMER, USR*n*(), VAL(), VARPTR(), VARPTR$()

Operators (23)

( ), ^, unary + -, * /, \, MOD, + -, < <= = <> >= >, NOT, AND, OR, XOR, EQV, IMP
ordered from highest to lowest precedence, commas separate each precedence group

Data Types (4, 4)

variables default to single precision real or !, double precision real #, integers %, string $
arrays default to single precision real () or !(), double precision real arrays #(), integer arrays %(), string arrays $()
[see Microsoft BASIC-80 5.21 1981 for variable names rules, maximum variables and arrays]

On August 12th, 1981 the IBM PC was released with built-in Cassette BASIC ROM, IBM PC-DOS Disk BASIC and Advanced BASIC. All three BASIC interpreters were licensed from Microsoft and they were the ultimate growth of the BASIC-80 5.21 1981 ported to the Intel 8086/8088 MPU with the added support for the IBM PC graphics and sounds. So when the OEMs licensed Microsoft first MS-DOS 1.25 in 1982, they could choose to bundle GW-BASIC 1.04 which was functionally equivalent to IBM PC Advanced BASIC without depending on the IBM PC ROM.

Obviously GW-BASIC does not execute on the 8bit TRS-80, but due to their shared ancestry, they have numerous common features with the exception of those relating to the IBM PC graphics and sounds. So I am including the GW-BASIC details for comparison purpose and for the completeness of the TRS-80 Family Pedigree.

Wrapping it up …

Microsoft Disk Extended BASIC-80 5.21 23.75K 1981

Model 4 TRSDOS 6.x, LS-DOS 6.x, DOSPLUS IV
CP/M 2.2, CP/M Plus 3.0
TRS-XENIX MBASIC (BCD)

Microsoft Disk Extended BASIC-80 4.46 18K 1977

LEVEL II, Model III/I TRSDOS Disk BASIC; Model 4 in Model III Mode *all* DOSs
Model II TRSDOS-II/TRSDOS Disk BASIC; Model 100 BASIC/Disk BASIC (BCD)
CoCo CB, ECB, DECB, SDECB; MC10 MCB

Steve Leininger's LEVEL I 4K 1977

Floating point + *Li Chen Wang's Palo Alto TINY BASIC* published in *Dr Dobbs*, May 1976

Wish all of you, *happy* computing.

BASIC Interpreters Feature Summary Table

BASIC Interpreters	Size	Version	Year	Commands, Statements	Functions	Operators	Comments
Dartmouth	10K	1964	1964	29	11	14	GE225+DN30, GE235+DN30 [a]
BASIC-PLUS	16K	1970	1970	72	48	22	DEC PDP-11, DEC PDP-10 [b,c]
Altair BASIC	12K	3.2	1975	43	30	20	Micro-Soft 1st product, v2.0 [d]
Altair BASIC	16K	4.1	1977	65	45	23	Altair EXTENDED BASIC [e]
Altair DISK	18K	4.1	1977	77	56	23	Altair DISK BASIC
MBASIC	18K	4.51	1977	77	56	23	CP/M BASIC-80 4.1 1977
MBASIC	24K	5.21	1981	84	54	23	CP/M BASIC-80 5.0 1978
Tiny BASIC	2K	1.0	1976	14	3	16	Li-Chen Wang's PATB
TRS80 LEVEL I	4K	1977	1977	27	6	16	Basis for Model III LEVEL I [f,g]
LEVEL II	12K	1.3	1978	56	36	18	Basis for the Model III BASIC [h]
DISK BASIC	16K	2.2	1978	88	51	18	Model I TRSDOS 2.3
M4 Disk BASIC	24K	1.1.1	1985	83	59	23	Model 4 TRSDOS 6.2.1AR
M2 BASIC	18K	1.2	1983	70	56	23	Model II TRSDOS-II 4.3
XENIX MBASIC	51K	1.2.0	1984	89	56	23	Model II TRS-XENIX 1.3.5
M100 BASIC	17K	1.1	1983	87	45	23	Model 100 and 102
CoCo DECB	24K	1.1	1983	94	44	18	Color Computer 1 and 2 [i]
MC10 MCB	8K	1.0	1983	37	23	18	MicroColor MC-10 [i]
GW-BASIC	79K	3.23	1988	128	71	23	1st v1.04 1982, MS-DOS 1.25

Additional Table Comments

The BASIC feature tallies are not absolute and may on the surface appear contradictory to other source. However, the issue is a simple one, how one chooses to enumerate the language features, eg list each command, statement, function and operator. For example: PRINT and ? count them as one statement vs count them as two statements, similarly & &O &H count them as one function vs three functions, I choose the later since it accounts for each symbol accepted by the BASIC parser.

a. Dartmouth BASIC first release (May 4th, 1964) was < 4K 20bit Words, ie 4Kx20bit/8bit < 10K Bytes
b. DEC BASIC-PLUS for PDP-11 running RSTS-11 was < 8K 16bit Words, ie 8Kx16bit/8bit < 16K Bytes
c. DEC BASIC-PLUS for PDP-10 running TOPS-10 was 36bit Words, I don't know its size requirement
d. ALTAIR EXTENDED BASIC 3.2 12K December 1975 Reference Manual
e. ALTAIR EXTENDED BASIC 4.1 16K July 1977 Reference Manual
f. Model I and Model III LEVEL I were derived by Tandy from Li-Chen Wang's Tiny BASIC 2K
g. Model III LEVEL I BASIC ROM 4K 1980 adds printer support, 2 statements: LLIST and LPRINT
h. Model III BASIC ROM was 14K 1980, 2K adds lowercase + real clock support, 1 function: TIME$
i. CoCo3 had SDECB 2.1 1986, derived from DECB 1.1 1983, modified by Microware Systems Corps. Adds 22 statements: ATTR, HBUFF, HCIRCLE, HCLS, HCOLOR, HDRAW, HGET, HLINE, HPAINT, HPRINT, HPUT, HRESET, HSCREEN, HSET, HSTAT, LOCATE, LPOKE, ON BRK, ON ERR, PALETTE, PALETTE RGB, WIDTH; and 5 functions: BUTTON(), ERLIN, ERNO, HPOINT(), LPEEK().
j. Undocumented adds 2 statements: CLOADM, EXEC; and 1 function: VARPTR()

Dedication

To Matthew, our son: A young technological mind that always yearned for innovation, never timid from learning new ideas, whose generous soul fueled his willingness to share all what he had with anyone in need. The loving memory of his short lived yet full life, lives with us every day. Giving me the courage to share what I learned with others so that we all can help one another in our common journey.

Acknowledgments

Vernon B Hester, a true legend of the original TRS-80 era, for sharing his remarkable suite of TRS-80 software (ESOTERIC 3, MULTIDOS 6, MULTIDOS 5, ZEUS, ASTROCAL, VFU, ...), his plethora of BASIC (ESOBASIC, CBASIC, BASICH, CBASICH, SUPERBASIC, BOSS BASIC, ...) and for generously sharing his insight and clarifying MBASIC 5.x basis for Model 4 while MBASIC 4.46 the basis for all Model I/III.

George and *Peter Phillips*, the amazing twins who gifted us their emulator, the *trs80gp*, which I used extensively to validate some of the BASIC dialects of every TRS-80 model (LEVEL I, LEVEL II, DOSs, CPM, XENIX, DECB); George's LEVEL 2.5 BASIC article/program; and for their kindness to actually read and respond to emails with in depth details.

Alan Wallace for generously sharing his BASIC knowledge, especially for his trs80gp Facebook Group posting on October 21, 2023 discussing the "IF cond, GOSUB line number" and his kind emails that followed. He ignited my interest to uncover *what are the various valid forms of IF-THEN-ELSE*, leading to *what is the origin of BASIC*.

Dusty Miller for editing and publishing the *TRS8BIT* newsletter, the great medium to quest our thirst for TRS-80 8bit and 16bit microcomputers and the longing for information covering the simpler computing platforms unbounded by layers of virtualization while encapsulated in myriad of abstractions.

Further Readings (ordered by the BASIC timeline steps)

The Kiewit Computation Center and the Dartmouth Time-Sharing System. Dartmouth College. 1971.

BASIC version 2 manual (PDF). Dartmouth College. October 1964.

Time Shared BASIC Manual (PDF). Hewlett Packard. August 1969.

BASIC-PLUS Manual (PDF). DEC. 1972.

Altair BASIC Reference Manual (PDF). MITS. 1975.

BASIC-80 CP/M Reference Manual (PDF). Microsoft. 1979.

ECMA-55 Minimal BASIC Standard (PDF). ECMA. January 1978.

January 1975 issue of *Popular Electronics* was published on November 29, 1974.

Thomas Kurtz Interview with Dartmouth College, July 2, 2002.

Nathan Brookwood (aka Nathan Teichholtz) Interview with DEC Connection, July 12, 2022.

Ed Roberts Interview with *Historically Brewed* magazine, 1996.

Bill Gates Interview with "National Museum of American History", 1993

Allen, Paul. "The Idea Man". Penguin. 2011.

Wallace, James. "Hard Drive: Bill Gates and the Making of the Microsoft Empire". Wiley. 1992.

TRS-80 BASIC Cross Dialects Tables

Jim Adams v1.0 May 2024

In the micro computing world, BASIC was almost ubiquitously present but with many unique twists. It is like "English is spoken here" but each locality has its own idiomatic expressions. Moving between dialects or even between different TRS-80 models always requires finding out what BASIC functions are missing, or how few BASIC statement are spelled differently. I have found a simple table, which lists which BASIC elements are available in a given dialect, to be very helpful. At a quick glance, I can tell if the same feature appears in which dialect. Of course, there is no substitute to reading the manuals to lookup the exact *syntax* (mechanics) and *semantics* (meaning) of a given BASIC feature.

Please allow me to share with you a collection of cross referenced tables of the TRS-80 BASIC dialects along selected few evolutionary BASIC interpreters that are greatly similar to the TRS-80 BASIC.

Cross Reference Tables of BASIC Dialects

- 226 Commands and Statements Table (alphabetical order)
- 115 Functions Table (alphabetical order)
- 33 Operators Table (precedence order)
- 17 Data Types Table (bit size order)
- 26 Special Characters Table (ASCII order)
- 18 BASIC Dialects Summary Table (family tree order)

Legend:

Y	All versions
3	Model III only
1	Model I only
D	Model 100 with Disk Video Interface only [DSKO$, LFILES, WIDTH, DSKI$(), LOC(), LOF()]
U	MC-10 Undocumented but available [CLOADM, EXEC, VARPTR()]
P	IBM PC 5150 1981 and PCjr 4860 1984 only; both supported cassette I/O
C	Dartmouth CARDBASIC batch compiler with: MAT, MAT PRINT, MAT READ
+	DEC BASIC-PLUS had one real format, either 32bit or 64bit configured by Sys Admin
$	Dartmouth, PATB supported string constants only, no string data variables

18 BASIC Dialects Cross-Reference Table

Commands and Statements (226)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	
STATEMENTS part 1 of 7																			Comments
! *exclamation*																Y			Same as REM in DEC BASIC-PLUS
' *apostrophe*	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y					Same as REM
, *comma*		Y	Y	Y															Same as THEN
? *question*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				Same as PRINT
APPEND							Y												
ASSIGN																Y			
AUDIO										Y									
AUTO	Y	Y	Y	Y			Y	Y				Y	Y	Y					
BACKUP										Y									
BEEP									Y			Y							
BLOAD												Y							
BREAK							Y												
BSAVE												Y							
BYE																Y	Y		
CALL	Y						Y		Y			Y	Y						
CAT																Y	Y		
CHAIN	Y						Y					Y	Y			Y			
CHANGE																Y			
CHDIR												Y							
CIRCLE										Y		Y							
CLEAR	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				
CLOAD		Y	Y	Y	Y				Y	Y	Y		Y	Y	Y				
CLOAD?		Y	Y	Y					Y					Y	Y				
CLOAD*											Y			Y					
CLOADM									Y	Y	U								Undocumented MC-10
CLOSE	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y		Y			
CLS	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y							
CMD"A"		Y																	
CMD"B"		Y																	
CMD"C"		Y																	
CMD"D"		Y	Y																
CMD"E"		Y																	
CMD"I"		Y	Y																

18 BASIC Dialects Cross-Reference Table

Commands and Statements (226)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	
STATEMENTS part 2 of 7																			Comments
CMD"J"		Y																	
CMD"L"		Y																	
CMD"O"		Y																	
CMD"P"		Y																	
CMD"R"		Y	Y																
CMD"S"		Y	Y																
CMD"T"		Y	Y																
CMD"X"		Y																	
CMD"Z"		Y																	
COLOR										Y		Y							
COM									Y			Y							
COMMON	Y						Y					Y	Y						
COMPILE																Y			
CONT	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			LEVEL I abbr: C.
COPY										Y									
CSAVE		Y	Y	Y	Y				Y	Y	Y		Y	Y	Y				
CSAVE*											Y			Y					
CSAVEM									Y	Y									
DATA	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I abbr: D.
DATE$=							Y		Y			Y							
DAY$=									Y										
DEASSIGN																Y			
DEF FN*a*	Y	Y	Y				Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	
DEF SEG												Y							
DEF USR*n*	Y	Y	Y					Y		Y		Y	Y	Y	Y				
DEFDBL	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y				
DEFINT	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y				
DEFSNG	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y				
DEFSTR	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y				
DEL										Y									
DELETE	Y	Y	Y	Y			Y	Y				Y	Y	Y	Y	Y			
DIM	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
DIR										Y									

18 BASIC Dialects Cross-Reference Table

Commands and Statements (226)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	
STATEMENTS part 3 of 7																			Comments
DLOAD										Y									
DOS										Y									
DRAW										Y		Y							
DRIVE										Y									
DSKI$										Y									
DSKINI										Y									
DSKO$									D	Y				Y					Added with Disk Video Interface
EDIT	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y	Y				
END	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I abbr: E.
ENVIRON												Y							
ERASE	Y						Y	Y				Y	Y	Y	Y				
ERROR	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y					
EXEC										Y	U								Undocumented MC-10
FIELD	Y	Y	Y				Y	Y		Y		Y	Y	Y		Y			
FILES							Y		Y	Y		Y	Y	Y					
FNEND																Y			
FOR-STEP-UNTIL																Y			
FOR-STEP-WHILE																Y			
FOR-TO-STEP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I, PATB abbr: F. TO S.
GET	Y	Y	Y					Y		Y		Y	Y	Y		Y			
GET#												Y							
GOSUB	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I, PATB abbr: GOS.
GOTO	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I, PATB abbr: G.
HELLO																Y	Y		
IF						Y													
IF-THEN					Y						Y						Y	Y	LEVEL I abbr: IF T.
IF-THEN-ELSE	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y	Y	Y			
INPUT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	LEVEL I, PATB abbr: IN.
INPUT LINE																Y			
INPUT LINE#																Y			
INPUT#	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y		Y			
INPUT$									Y										
IOCTL												Y							

18 BASIC Dialects Cross-Reference Table

Commands and Statements (226)

Model 100									DECB 1.1 CoCo2/1										
TRSDOS 2.0b M2									MCB 1.0 MC10										
TRS-XENIX MBASIC M16									GW-BASIC 3.23 PC										
Palo Alto Tiny BASIC									MBASIC 5.21 CP/M										
LEVEL I M3/M1									MBASIC 4.51 CP/M										
LEVEL II M3/M1									Altair EXTENDED BASIC 3.2										
TRSDOS 2.3 M1									DEC BASIC-PLUS										
TRSDOS 1.3 M3									Dartmouth BASIC DTSS										
TRSDOS 6.2.1AR M4									ANSI X3.60-1978 Minimum										
STATEMENTS part 4 of 7																			Comments
IPL								Y											
KEY								Y			Y				Y				
KILL	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y		Y			
LCOPY								Y											
LENGTH															Y				
LET	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LET keyword maybe *omitted*
LFILES								D											Added with Disk Video Interface
LINE								Y	Y		Y								
LINE INPUT	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y					
LINE INPUT#	Y	Y	Y				Y	Y	Y			Y	Y	Y					
LIST	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		LEVEL I, PATB abbr: L.
LISTNH																Y			
LLIST	Y	Y	Y	Y	3		Y	Y	Y	Y	Y	Y	Y	Y					Model III only, not Model I
LOAD	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y					
LOAD"CAS1:												P							IBM PC 5150 and PCjr 4860 only
LOADM								Y	Y										
LOCATE												Y							
LOCK							Y					Y							
LPRINT	Y	Y	Y	Y	3		Y	Y	Y		Y	Y	Y	Y					Model III only, not Model I
LPRINT USING	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y					
LSET	Y	Y	Y				Y	Y		Y		Y	Y	Y		Y			
MAT																Y	C		CARDBASIC batch compiler
MAT INPUT																Y			
MAT PRINT																Y	C		CARDBASIC batch compiler
MAT READ																Y	C		CARDBASIC batch compiler
MAXFILES=								Y											
MDM								Y											
MENU								Y											
MERGE	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y					
MID$()=	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y					
MKDIR												Y							
MOTOR								Y	Y		P								IBM PC 5150 and PCjr 4860 only
NAME	Y	Y	Y				Y	Y	Y			Y	Y	Y		Y			

18 BASIC Dialects Cross-Reference Table
Commands and Statements (226)

	Model 100								DECB 1.1 CoCo2/1									
	TRSDOS 2.0b M2								MCB 1.0 MC10									
	TRS-XENIX MBASIC M16								GW-BASIC 3.23 PC									
	Palo Alto Tiny BASIC								MBASIC 5.21 CP/M									
	LEVEL I M3/M1								MBASIC 4.51 CP/M									
	LEVEL II M3/M1								Altair EXTENDED BASIC 3.2									
	TRSDOS 2.3 M1								DEC BASIC-PLUS									
	TRSDOS 1.3 M3								Dartmouth BASIC DTSS									
	TRSDOS 6.2.1AR M4								ANSI X3.60-1978 Minimum									
STATEMENTS part 5 of 7																		Comments
NEW	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I, PATB abbr: N.
NEXT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I, PATB abbr: N.
NULL													Y	Y	Y			
OLD																Y	Y	
ON BREAK							Y											
ON COM									Y			Y						
ON ERROR	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y		Y		
ON GOSUB	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
ON GOTO	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y
ON KEY									Y			Y						
ON MDM									Y									
ON PEN												Y						
ON PLAY												Y						
ON STRIG												Y						
ON TIME$									Y									
ON TIMER												Y						
OPEN	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y		Y		
OPEN"COM*n*"												Y						
OPTION BASE	Y						Y					Y	Y					Y
OUT	Y	Y	Y	Y					Y			Y	Y	Y				
OUTPUT							Y											
PAINT										Y		Y						
PALETTE												Y						
PALETTE USING												Y						
PCLEAR										Y								
PCLS										Y								
PCOPY										Y		Y						
PEN												Y						
PLAY										Y		Y						
PMODE										Y								
POKE	Y	Y	Y	Y			Y		Y	Y	Y	Y	Y	Y	Y			
POWER									Y									
PRESET									Y	Y		Y						

18 BASIC Dialects Cross-Reference Table
Commands and Statements (226)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	
STATEMENTS part 6 of 7																			Comments
PRINT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I, PATB abbr: P.
PRINT AT					Y														LEVEL I abbr: P.A.
PRINT USING	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y	Y	Y			
PRINT#	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y		Y			
PRINT# USING	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y		Y			
PRINT@	Y	Y	Y	Y				Y	Y	Y	Y								
PRINT@()	Y							Y											
PSET									Y	Y		Y							
PUT	Y	Y	Y				Y	Y		Y		Y	Y	Y		Y			
PUT#												Y							
RANDOM	Y	Y	Y	Y			Y	Y								Y			
RANDOMIZE							Y					Y	Y					Y	
READ	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I abbr: REA.
REM	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	PATB stats it accepts REM*ARK*
RENAME										Y						Y	Y		
RENUM	Y						Y	Y		Y		Y	Y	Y					
REPLACE																Y			
RESET		Y	Y	Y	Y					Y	Y	Y	Y						LEVEL I abbr: R.
RESTORE	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	LEVEL I abbr: REST.
RESUME	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y		Y			
RETURN	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I abbr: RET. PATB: R.
RMDIR												Y							
RSET	Y	Y	Y				Y	Y		Y		Y	Y	Y		Y			
RUN	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		LEVEL I, PATB abbr: R.
RUNM									Y										
RUNNW																Y			
SAVE	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y		Y	Y		
SAVE"CAS1:												P							IBM PC 5150 and PCjr 4860 only
SAVEM									Y	Y									
SCRATCH																	Y		
SCREEN									Y	Y		Y							
SET		Y	Y	Y	Y					Y	Y								LEVEL I abbr: S.
SHELL							Y					Y							

18 BASIC Dialects Cross-Reference Table
Commands and Statements (226)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	
STATEMENTS part 7 of 7																			Comments
SKIPF										Y	Y								
SLEEP																Y			
SOUND	Y								Y	Y	Y	Y							
STOP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I abbr: ST. PATB: S.
STRIG												Y							
SWAP	Y						Y	Y				Y	Y	Y	Y				
SYSTEM	Y	Y	Y	Y			Y	Y				Y	Y	Y			Y		
SYSTEM"*fff*"	Y							Y											*fff* is a TRSDOS 6.x command
TAPE																Y			
TIME$=									Y			Y							
TIME=							Y												
TIMER=										Y									
TROFF	Y	Y	Y	Y			Y	Y		Y		Y	Y	Y	Y				
TRON	Y	Y	Y	Y			Y	Y		Y		Y	Y	Y	Y				
UNLOAD										Y									
UNLOCK							Y					Y				Y			
UNSAVE																Y	Y		
VERIFY										Y									
VIEW												Y							
VIEW PRINT												Y							
WAIT	Y						Y					Y	Y	Y	Y	Y			
WEND	Y						Y					Y	Y						
WHILE	Y						Y					Y	Y						
WIDTH	Y						Y		D			Y	Y	Y					Added with Disk Video Interface
WIDTH LPRINT							Y						Y						
WINDOW												Y							
WRITE	Y						Y			Y		Y	Y						
WRITE#	Y						Y					Y	Y						

18 BASIC Dialects Cross-Reference Table

Functions (115)

	Model 100									DECB 1.1 CoCo2/1									
		TRSDOS 2.0b M2									MCB 1.0 MC10								
			TRS-XENIX MBASIC M16									GW-BASIC 3.23 PC							
				Palo Alto Tiny BASIC									MBASIC 5.21 CP/M						
					LEVEL I M3/M1									MBASIC 4.51 CP/M						
						LEVEL II M3/M1									Altair EXTENDED BASIC 3.2					
							TRSDOS 2.3 M1									DEC BASIC-PLUS				
								TRSDOS 1.3 M3									Dartmouth BASIC DTSS			
									TRSDOS 6.2.1AR M4									ANSI X3.60-1978 Minimum		
FUNCTIONS part 1 of 4																			Comments
&	Y	Y	Y				Y	Y		Y		Y	Y	Y					&H or &O depending on the dialect
&H	Y	Y	Y				Y	Y		Y		Y	Y	Y					
&O	Y	Y	Y				Y	Y		Y		Y	Y	Y					
ABS()	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I, PATB abbr: A.()
ASC()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				
ASCII()															Y				
ATN()	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	
CDBL()	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y				
CHR$()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			
CINT()	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y				
COS()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
CSNG()	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y				
CSRLIN									Y			Y							
CVD()	Y	Y	Y				Y	Y				Y	Y	Y					
CVI()	Y	Y	Y				Y	Y				Y	Y	Y					
CVN()										Y									
CVS()	Y	Y	Y				Y	Y				Y	Y	Y					
CVT%$()															Y				
CVT$%()															Y				
CVT$F()															Y				
CVTF$()															Y				
DATE$	Y						Y	Y	Y			Y							
DATE$()															Y				
DAY$									Y										
DET															Y				
DSKF()														Y					
DSKI$()									D					Y					Added with Disk Video Interface
e															Y				
ENVIRON$()												Y							
EOF()	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y					
ERDEV												Y							
ERDEV$												Y							
ERL	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y		Y			

18 BASIC Dialects Cross-Reference Table

Functions (115)

	Model 100						DECB 1.1 CoCo2/1										
	TRSDOS 2.0b M2							MCB 1.0 MC10									
	TRS-XENIX MBASIC M16								GW-BASIC 3.23 PC								
	Palo Alto Tiny BASIC									MBASIC 5.21 CP/M							
	LEVEL I M3/M1										MBASIC 4.51 CP/M						
	LEVEL II M3/M1											Altair EXTENDED BASIC 3.2					
	TRSDOS 2.3 M1												DEC BASIC-PLUS				
	TRSDOS 1.3 M3													Dartmouth BASIC DTSS			
	TRSDOS 6.2.1AR M4														ANSI X3.60-1978 Minimum		
FUNCTIONS part 2 of 4																	Comments
ERR	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y		Y			
ERR()														Y					
ERRS$	Y							Y											
EXF()															Y				
EXP()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
EXTERR()												Y							
FIX()	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y		Y			
FN*a*()	Y	Y	Y				Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	
FRE()	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y				
FREE()										Y									
HEX$()	Y							Y		Y		Y	Y	Y					
HIMEM									Y										
INKEY$	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y					
INP()	Y	Y	Y	Y					Y			Y	Y	Y	Y				
INPUT$()	Y						Y	Y	Y			Y	Y						
INSTR()	Y	Y	Y				Y	Y	Y	Y		Y	Y	Y		Y			
INT()	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I abbr: I.()
INV()																Y			
IOCTL$()												Y							
JOYSTK()										Y									
LEFT()																Y			
LEFT$()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				
LEN()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			
LOC()	Y	Y					Y	Y	D	Y		Y	Y	Y					Added with Disk Video Interface
LOF()	Y	Y	Y				Y	Y	D	Y		Y	Y	Y					Added with Disk Video Interface
LOG()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
LOG10()																Y			
LPOS()	Y						Y		Y			Y	Y	Y					
MAGTAPE()																Y			
MAXFILES									Y										
MAXRAM									Y										
MEM	Y	Y	Y	Y	Y		Y	Y		Y	Y								LEVEL I abbr: M.
MID()																Y			

18 BASIC Dialects Cross-Reference Table

Functions (115)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	
FUNCTIONS part 3 of 4																			Comments
MID$()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				
MKD$()	Y	Y	Y				Y	Y				Y	Y	Y					
MKI$()	Y	Y	Y				Y	Y				Y	Y	Y					
MKN$()										Y									
MKS$()	Y	Y	Y				Y	Y				Y	Y	Y					
NUM																Y			
NUM2																Y			
NUM$()																Y			
OCT$()	Y						Y	Y				Y	Y	Y					
PEEK()	Y	Y	Y	Y			Y		Y	Y	Y	Y	Y	Y	Y				
PEN()												Y							
PI																Y			
PLAY()												Y							
PMAP												Y							
POINT()		Y	Y	Y	Y					Y	Y	Y							LEVEL I abbr: P.()
POS()	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y	Y	Y			
PPOINT()										Y									
RAD$()																Y			
RECOUNT																Y			
RIGHT()																Y			
RIGHT$()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				
RND()	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	LEVEL I, PATB abbr: R.()
ROW()	Y						Y	Y											
SCREEN()												Y							
SGN()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	
SHELL()							Y												
SIN()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
SIZE()						Y													
SPACE$()	Y						Y	Y	Y			Y	Y	Y		Y			
SPC()	Y						Y	Y				Y	Y	Y	Y				
SQR()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
STICK()												Y							
STR$()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				

18 BASIC Dialects Cross-Reference Table

Functions (115)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	
FUNCTIONS part 4 of 4																			Comments
STRIG()												Y							
STRING$()	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y					
SWAP%()																Y			
TAB()	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			LEVEL I abbr: T.()
TAN()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
TIME()																Y			
TIME$	Y	Y	Y	3			Y	Y	Y			Y							Model III only, not Model I
TIME$()																Y			
TIMER										Y		Y							
TRN()																Y			
USR()				Y															
USR*n*()	Y	Y	Y					Y		Y		Y	Y	Y	Y				
VAL()	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			
VARPTR()	Y	Y	Y	Y			Y	Y	Y	Y	U	Y	Y	Y					Undocumented MC-10
VALPTR$()												Y							
XLATE()																Y			

The table entries list the intrinsic functions that are builtin the BASIC interpreters.
FN*a*() allows up to 26 user defined BASIC functions, FN*A*() to FN*Z*(), written in the BASIC language.
USR*n*() allows up to 10 user defined functions, USR*0*() to USR*9*(), written in machine language and preloaded into memory (preferably in reserved space so they are not overwritten by BASIC).

18 BASIC Dialects Cross-Reference Table

Operators (33)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	Microsoft Precedence Groups
OPERATORS																			
(*parenthesis*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	1 highest precedence group
) *parenthesis*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	1
functions	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	2 intrinsic functions(), FN*a*(), &
^ *exponential*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	3
** *exponential*																Y			3 same as ^
- *unary negative*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	4
+ *unary positive*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	4
* *multiplication*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	5
/ *division*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	5
\ *integer division*	Y						Y	Y	Y			Y	Y	Y	Y				6 same as (INT(x/y))
MOD *modulus*	Y						Y	Y	Y			Y	Y	Y	Y				7 same as (*x* - *y* * INT(*x*/*y*))
+ *addition*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	8
- *subtraction*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	8
+ *string concatenation*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			9
= *equality*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
== *equality rounded 6digit*																Y			10
<> *inequality*	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
>< *inequality*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				10
# *inequality*						Y													10
< *relational less than*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
<= *less than or equal*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
=< *less than or equal*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				10
>= *greater than or equal*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
=> *greater than or equal*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				10
> *relational greater than*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
NOT *logical complement*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			11
AND *logical conjunction*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			12
* *logical conjunction*					Y	Y													12
OR *logical disjunction*	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			13
+ *logical disjunction*					Y	Y													13
XOR *logical exclusive OR*	Y						Y	Y	Y			Y	Y	Y		Y			14 ((*x* AND NOT *y*) OR (NOT *x* AND *y*))
EQV *logical equivalence*	Y						Y	Y	Y			Y	Y	Y		Y			15 same as (NOT (*x* XOR *y*))
IMP *logical implication*	Y						Y	Y	Y			Y	Y	Y		Y			16 same as (NOT *x* OR *y*)

18 BASIC Dialects Cross-Reference Table
Operators (33)

Precedence and Groups
Operators fall into four categories, in high-to-low precedence order:

Operators Category high-to-low order	Precedence Group in the Operators Table
Numerical	Groups 1-8
String	Group 9
Relational	Group 10
Logical	Groups 11-16

Operators are evaluated according to their *relative* precedence; starting with the highest precedence first then to the relatively lower precedence in the BASIC expression.

Operators within the same precedence group are evaluated from the left to the right as they appear in the BASIC expression.

Precedence group numbers *illustrate* the relative order of evaluation; however, there are no such numbers are assigned or utilized by BASIC.

Microsoft's precedence order and groups are listed in the table as they are the most ubiquitously implemented family of BASIC interpreters. Other BASICs use differing precedence order and groups; please read the documentations for the details of their expression evaluation precedence.

Derived Operators
Many earlier BASIC interpreters did not implement these five operators: \, MOD, XOR, EQV, IMP; however they may be derived using other intrinsic operators.

Operator	Description	Derived Expression with the Same Results
*x**y*	Integer Division	(INT(*x*/*y*))
x MOD *y*	Modulus *remainder*	(*x* - *y* * INT(*x*/*y*)) same as (*x* - (*y* * INT(*x*/*y*))), for positive *x*, *y*
x XOR *y*	Exclusive Disjunction	((*x* AND NOT *y*) OR (NOT *x* AND *y*))
x EQV *y*	Equivalence	(NOT ((*x* AND NOT *y*) OR (NOT *x* AND *y*))) same as (NOT (*x* XOR *y*))
x IMP *y*	Implication	(NOT *x* OR *y*) same as ((NOT *x*) OR *y*)

18 BASIC Dialects Cross-Reference Table

Data Types and Defaults (17)

	Model 100	TRSDOS 2.0b M2	TRS-XENIX MBASIC M16	Palo Alto Tiny BASIC	LEVEL I M3/M1	LEVEL II M3/M1	TRSDOS 2.3 M1	TRSDOS 1.3 M3	TRSDOS 6.2.1AR M4	DECB 1.1 CoCo2/1	MCB 1.0 MC10	GW-BASIC 3.23 PC	MBASIC 5.21 CP/M	MBASIC 4.51 CP/M	Altair EXTENDED BASIC 3.2	DEC BASIC-PLUS	Dartmouth BASIC DTSS	ANSI X3.60-1978 Minimum	Comments
DATA TYPES																			
Integer 16bit	Y	Y	Y	Y		Y	Y	Y	Y			Y	Y	Y	Y	Y			% 2bytes two's complement
Single Precision 32bit	Y	Y	Y	Y	Y		Y	Y	Y			Y	Y	Y	Y	+		Y	! 4bytes floating point real
Single Precision 40bit										Y	Y						Y		5bytes floating point real
Double Precision 64bit	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y	+			# 8bytes floating point real
String	Y	Y	Y	Y	Y	$	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	$	Y	$ length 0bytes up to 255bytes
Integer Array	Y	Y	Y	Y		Y	Y	Y	Y			Y	Y	Y	Y	Y			% 2bytes two's complement
Single Precision Array	Y	Y	Y	Y	Y		Y	Y	Y			Y	Y	Y	Y	+		Y	! 4bytes floating point real
Single Precision Array										Y	Y						Y		5bytes floating point real
Double Precision Array	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y	Y	+			# 8bytes floating point real
String Array	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			$ length 0bytes up to 255bytes
1 dimension Arrays	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Single-dimensional arrays, vectors
2 dimensions Arrays	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Two-dimensional arrays, matrices
Up to 255 dimensions	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				Multi-dimensional arrays
Default Integer 16bit						Y													PATB only variables data type
Default Single 32bit	Y	Y	Y	Y	Y			Y				Y	Y	Y	Y	+		Y	Assumed without any specifiers
Default Single 40bit										Y	Y						Y		The only numeric data type
Default Double 64bit							Y		Y							+			Assumed without any specifiers

ANSI X3.60-1978 did not specify the 32bit single precision, but implied it by specifying: 6 significant digits, with a minimum valid range from 1E-38 to 1E+38.

+ DEC BASIC-PLUS 1970, for the PDP-11 RSTS-11, supported *one* floating point-format only, either the 32bit single precision *or* the 64bit double precision, selectable at *sysgen* to build the MONITOR.

$ Dartmouth BASIC 1964, Palo Alto Tiny BASIC 1976 had string constants only, used by PRINT and INPUT for display messages. No string data variables were supported in these initial releases.

18 BASIC Dialects Cross-Reference Table

Special Characters (26)

Model 100									DECB 1.1 CoCo2/1									
TRSDOS 2.0b M2									MCB 1.0 MC10									
TRS-XENIX MBASIC M16									GW-BASIC 3.23 PC									
Palo Alto Tiny BASIC									MBASIC 5.21 CP/M									
LEVEL I M3/M1									MBASIC 4.51 CP/M									
LEVEL II M3/M1									Altair EXTENDED BASIC 3.2									
TRSDOS 2.3 M1									DEC BASIC-PLUS									
TRSDOS 1.3 M3									Dartmouth BASIC DTSS									
TRSDOS 6.2.1AR M4									ANSI X3.60-1978 Minimum									
SPECIAL CHARACTERS									Comments									
Space optional		Y	Y	Y	Y	Y		Y	Y	Y	Y			Y	Y	Y	Y		Space(s) and/or Tab(s) delimiters
Space required	Y						Y					Y	Y					Y	Space(s) and/or Tab(s) delimiters
!	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y	Y	Y			Single Precision, DEC Comments
"	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	String constants
#	Y	Y	Y	Y		Y	Y	Y	Y	Y		Y	Y	Y	Y	Y			Double, PRINT USING, device#
$	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	String variables, string functions
%	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y	Y	Y	Y			Integer, PRINT USING
&	Y	Y	Y				Y	Y		Y		Y	Y	Y					16bit constants: Hex, Octal
'	Y	Y	Y	Y		Y	Y	Y	Y	Y		Y	Y	Y		Y			Comments; PATB, DEC String
(	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Precedence, function()
)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Precedence, function()
*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Multiplication, PATB AND
+	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Positive, addition, concatenation
,	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Delimiter, PRINT field
-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Negative, subtraction
.	Y	Y			Y	Y	Y	Y		Y		Y	Y	Y					EDIT, LIST; LEVEL I, PATB abbrev
/	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Division
:	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			Statements separator
;	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	PRINT field, INPUT
<	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Relational less than
=	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Assignment, relational equal
>	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Relational greater than
?	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y				PRINT
@	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y								PRINT @, PATB integer array @()
\	Y						Y	Y	Y			Y	Y	Y	Y				Integer division
^	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Exponential

Symbols are listed in ASCII order

space(s) or tab(s) are optional in most BASIC interpreters {as with FORTRAN66 and FORTRAN77}; but they are required delimiters in MBASIC 5.x, TRSDOS 6.x, XENIX, GW-BASIC, ANSI X3.60-1978

. All floating point BASICs accept . as a decimal point; but the table entry refers to either the line number option for AUTO, DELETE, EDIT, LIST, RENUM; or LEVEL I and PATB . for abbreviations

The 18 BASICs in this article, used 24 of the 32 total 7bit ASCII printable special symbols; 8 unused are [ ] _ ` { | } ~

18 BASIC Dialects Cross-Reference Table

Features Summary

Family Tree Step	BASIC Interpreter	Size	Version	Year	Maximum Bytes Free	Commands Statements	Functions	Operators
1	Dartmouth BASIC+DTSS [a]	10K	1964	1964	*TBD*	29	11	14
2	DEC BASIC-PLUS+RSTS-11 [b]	16K	1970	1970	*TBD*	72	48	23
3	Altair EXTENDED BASIC	12K	3.2	1975	55306	44	30	21
4	MBASIC (MS BASIC-80 CP/M)	18K	4.51	1977	37369	77	56	24
4	MBASIC (MS BASIC-80 CP/M)	24K	5.21	1981	30776	84	54	24
5	Palo Alto Tiny BASIC (PATB)	2K	1.0	1976	*TBD*	14	3	16
6	TRS-80 LEVEL I [c]	4K	1977	1977	15871	27	6	16
6	TRS-80 LEVEL II [d]	12K	1.3	1978	48338	56	36	19
6	TRSDOS 2.3 DISK BASIC	16K	2.2	1978	38288	78	50	19
6	TRSDOS 1.3 DISK BASIC	19K	1.3	1980	38202	88	51	19
7	Model 4 Disk BASIC	24K	1.1.1	1985	29924	83	59	24
8	Model II Disk BASIC	18K	1.2	1983	33608	70	56	24
8	TRS-XENIX MBASIC [e]	51K	1.2.0	1984	31341	89	56	24
8	Model 100 BASIC	17K	1.1	1983	29382	87	45	24
8	CoCo Disk ECB [f]	24K	1.1	1983	22823	94	44	19
8	MC-10 MCB [g]	8K	1.0	1983	19526	37	23	19
9	GW-BASIC (MS-DOS)	79K	3.23	1988	60300	128	71	24
n/a	ANSI X3.60-1978 [h]	*n/a*	*n/a*	1978	*n/a*	20	12	14

a. Dartmouth BASIC first release (May 4th, 1964) was < 4K 20bit Words, ie 4Kx20bit/8bit < 10K Bytes
b. DEC BASIC-PLUS for PDP-11 running RSTS-11 was < 8K 16bit Words, ie 8Kx16bit/8bit < 16K Bytes, DEC BASIC-PLUS for PDP-10 running TOPS-10 was 36bit Words, I don't know its size requirement
c. Model I LEVEL I were derived by Tandy from Li-Chen Wang's Palo Alto Tiny BASIC 1.77K, Model III LEVEL I BASIC ROM 4K 1980 added printer support, 2 statements: LLIST and LPRINT
d. Model III BASIC ROM was 14K 1980, 2K added lowercase + real clock support, 1 function: TIME$
e. XENIX MBASIC-68K for the Model 16 TRS-XENIX 1.3.5 also ran on Tandy 6000 XENIX 3.2.0 and the unreleased 3.3.0. A Tandy 6000 with the maximum theoretical RAM of 7168KB, mbasic –m6000000 yields 199437 Bytes free.
f. CoCo3 had SDECB 2.1 1986, derived from DECB 1.1 1983, modified by Microware Systems Corps. Adds 22 statements: ATTR, HBUFF, HCIRCLE, HCLS, HCOLOR, HDRAW, HGET, HLINE, HPAINT, HPRINT, HPUT, HRESET, HSCREEN, HSET, HSTAT, LOCATE, LPOKE, ON BRK, ON ERR, PALETTE, PALETTE RGB, WIDTH; and 5 functions: BUTTON(), ERLIN, ERNO, HPOINT(), LPEEK().
g. MC-10 undocumented added 2 statements: CLOADM, EXEC; and 1 function: VARPTR()
h. ANSI X3.60-1978 (ECMA-55-1978) Standard specified the Minimum BASIC language as a common denominator; but it was mostly ignored. In all practicality, MBASIC was the *de Facto* standard.
i. Family Tree Step numbers correspond to the relatively significant evolutionary BASIC's mile stones as detailed in the "BASIC Family History" article in the March 2024 issue of *TRS8BIT Newsletter*.

18 BASIC Dialects Cross-Reference Table
Wrapping Up

Numbers, more or less…
The BASIC features and various table tallies are not absolute and on the surface may appear contradictory to other sources. However, the issue is a simple one, how one chooses to enumerate the language features, that is, how to list each command, statement, function and operator. For example: PRINT and ? how should they be counted, as one statement *or* counted as two statements? Similarly, & &O &H count them as one function *or* three functions; I choose the later approach since it accounts for each symbol accepted by the BASIC parser.

Astute readers will notice the blending of some of the BASIC interpreters with their operating environment when counting the "Commands & Statements" category; this is because in the early days of computing, to the average user, *BASIC was the environment*. That is, a university operated the Dartmouth Time-Sharing System with Dartmouth BASIC; a school district purchased an HP 2000A with the Time-Shared BASIC; or a hobbyist purchased a TRS-80 with LEVEL II BASIC; primarily to program in BASIC. So the apparent blending of all the commands and statements was accepted as part of the BASIC programming. Obviously, that quickly changed to support multiple languages, then the separation between the environment and the languages became clearer to the average user.

Syntax and Semantics Warning
These tables provide a relative overview of the 18 BASIC interpreters, but a statement, function, operator or special symbol code that appears across multiple BASICs, is more often than not, will have different meaning (semantics) and different format (syntax) in each BASIC. So, use the tables for the initial lookup, but may I recommend following through with reading the manuals for the correct meaning and usage, that is, *the devil is in the details.*

For a detailed history, motivation and advantage of many of the dialects referenced here, please see "The BASIC Family Tree" published in the March 2024 issue of *TRS8BIT Newsletter*.

This has been a fun and rewarding exercise and I hope you find the tables useful. *Happy* computing.

Dedication

To Mary, my lifelong soulmate who teaches me something new every day; a beautiful, generous, loving wife, an amazing MasterChef and a creative Master Baker. I will forever be indebted to your love.
To James, our son, for his accomplishments, career, motivation fueled by the willingness to take on any challenge, while always thinking outside the box. Proud fully, I thank you for all of your achievements.
To Andrew, our son, for applying his academic astuteness to his career, while retaining a delightful sense of humor that is enjoyed by all of his friends. Proud fully, I thank you for all of your achievements.

Acknowledgments

Dusty Miller for editing and publishing the *TRS8BIT* newsletter, *the* great medium to quest our thirst for TRS-80 8bit and 16bit microcomputers with especial gratitude for his encouragement and support with which the articles would not have been possible.

Further Readings

Cross Dialects

TRS-80 BASIC Quick Reference Handbook 2nd Edition 62-2314. William Barden Jr. 1985.
Legendary author, masterful TRS80 BASICs cross reference, has +20 negligible misinformation.

BASIC Conversion Handbook 62-2088. David A Brain. 1981.
Microsoft Disk BASIC conversion across TRS-80, PET, Apple II.

BASIC Handbook, 2nd Edition. David Alvin Lien. 1984.
Great breadth of numerous BASIC dialects; syntactical examples limited to most common cases.

BASIC Book (PDF). Harry Helms. 1983.

TRS-80 Color Computer Quick Reference Guide 26-3194. Tandy. 1982.

The BASIC Family Tree (PDF). *TRS8BIT Newsletter*. March 2024.

Tandy Radio Shack

LEVEL 1 BASIC User's Manual 26-2101. David Alvin Lien. 1978.
Tandy's Don French instructor at San Diego State University; CompuSoft Publishing

LEVEL II BASIC Reference Manual 26-2102. Tandy. 1979.

TRSDOS 2.3 Disk BASIC Reference Manual 26-2104. Tandy. 1979.
Randy V Cook TRSDOS; Microsoft Disk BASIC

Getting Started with TRS-80 BASIC 26-2107. George Stewart. 1983.

Operation and BASIC Language Reference Manual 2e 26-2112. Tandy. 1981.

TRSDOS 1.3 Disk System Owner's Manual 26-2111. Tandy. 1980.
Tandy TRSDOS; Tandy Disk BASIC

TRSDOS 6.2 Disk System Owner's Manual 26-2117. Tandy. 1983.
Logical Systems Inc TRSDOS; Microsoft Disk BASIC

BASIC Evolutionary Steps

BASIC version 2 manual (PDF). Dartmouth College. October 1964.

Time Shared BASIC Manual (PDF). Hewlett Packard. August 1969.

BASIC-PLUS Manual (PDF). DEC. 1972.

Altair BASIC Reference Manual (PDF). MITS. 1975.

BASIC-80 CP/M Reference Manual (PDF). Microsoft. 1979.

ECMA-55 Minimal BASIC Standard (PDF). ECMA. January 1978.

TRS-80 BASIC: What are the Floating Point Numbers Valid Ranges?

Jim Adams v1.0 September 2024

A reasonable and sensible question to ask, and the answer should be easy to find, all it takes is to dig up a manual and look it up. After all, most TRS-80 BASIC manuals list the valid BASIC line numbers as 0 to 65529, a string variable may be up to 255 characters and the range of valid integers as -32768 to 32767. So surely the ranges of valid single precision floating point numbers and of valid double precision floating point numbers must be just as easy to find.

Indeed, a quick look at *TR-80 LEVEL II BASIC Reference Manual*, first edition 1978, shows the ranges of valid single and valid double precision floating point numbers on page A/16. Great, we found the answer and we are done, right? *Well*, not long after you read the answer in the manual, you will realize there are at least few typos *{the manual lists "Single Precision -1.701411E+38 to +1.701411E+38 inclusive"; "Double Precision -1.701411834544556E+38 to +1.701411834544556E+38 inclusive"}*.

No problem, after all, that was the first edition and in the rush of things, mistakes happened. Let's check the *second edition*, second printing of 1979; and page A-17 does correct the Single Precision's typos but not the double precision's typos *{Single Precision -1.701411E±38 to +1.701411E±38 inclusive; Double Precision -1.701411834544556E+38 to +1.701411834544556E+38 inclusive}*. Third time's the charm, let's try the *Model III Operation and BASIC Language Reference Manual*, second edition, February 1981, and on page 220, all typos have been corrected, Single Precision range is listed as **E±38** and Double Precision range is listed as **D±38**. Hooray, now we are done, right? *Well, not really...*

To begin with, anyone who knows about two's complement, will recognize ±38 is erroneous, since the value 0 reduces the + range by one, i.e. -38 to +37 or -39 to +38. Apparently, more reference manuals are needed and I searched as many manuals as I could get my hands on, starting with the Altair BASIC 2.0 1975 through GW-BASIC 3.23 1988, as they are decedents of the same Microsoft family of BASICs.

And there lies the problem, each manual either repeats the inaccuracies or lists different ranges altogether. *Houston, we have a problem.* I cannot find the ranges that show the **largest** and **smallest** numbers showing **all the significant digits** for both the single and the double precision numbers. The manuals, at best, show subsets of the valid ranges; or worse, have erroneous ranges *{see last section}*.

Please keep in mind this dilemma when representing the infinite real numbers with most digital computers: how to squeeze a two pound *bologna or veggies* into a one pound sandwich? A) *real numbers* are infinite, ie given an integer n, there exists n+1 or –n there exists –n-1; and infinite fractions between any two integers, ie given m<n, m/n there exists m/n+1 or –m/n there exists –m/n+1. B) Floating point numbers are a limited finite subset approximation of real numbers; with finite set of non-contiguous digitized binary fractions, ie unrepresented gaps exist between two digitized fractions.

Now that we understand the problem, let's quickly jump to the ranges I calculated after lots more digging and a little bit of programming.

Floating Point Ranges

{Those who are smarter than I, please correct my calculations and share them with the rest of us, I love to learn the correct ranges; and those who already knew the ranges, I should have asked you first!}

GW-BASIC and TRSDOS 6.x BASIC allow the entire MBF format ranges via the BASIC assignment statement. All earlier BASICs [TRS-80, CP/M MBASIC 5.21 through Altair BASIC 3.0] assignment statements accept smaller subsets of the entire valid MBF ranges. However, POKE, VALPTR() or machine language routines can store the values directly into memory and BASIC will evaluate them correctly.

For example:
MBF internally, GW-BASIC and TRSDOS 6 Disk BASIC assignment statements allow:
Normalized real 8digit: -1.7014117E+38 to -2.9387359E-39, 2.9387359E-39 to 1.7014117E+38
CP/M MBASIC 5.21 through Altair EXTENDED BASIC 3.2 assignment statements allow:
BASIC Assignment: -1.7014109E+38 to -1.0000047E-38, 1.0000047E-38 to 1.7014109E+38
Yet, A!=1.0:POKE VALPTR(A!)+3,1 allows storing 2.9387359E-39

Microsoft Binary Format Valid Ranges

Z80, i8080, i8088
zero all zero's: **0**
4byte normalized real 8digit as *integer*: **-16,777,216** to **16,777,216**
normalized numbers with implicit "0.1", effectively 2^24=16777216
PRINTed as 7digit: -1.677722E+07 to 1.677722E+07
4byte normalized real 8digit: **-1.7014117E+38** to **-2.9387359E-39**,
2.9387359E-39 to **1.7014117E+38**
PRINTed as 7digit: -1.701412E+38 to -2.938736E-39, 2.938736E-39 to 1.701412E+38

8byte real 17digit as integer: **-72,057,594,037,927,936** to **72,057,594,037,927,936**
normalized numbers with implicit "0.1", effectively 2^56=72057594037927936
PRINTed as 16digit: -7.205759403792794D+16 to 7.205759403792794D+16
8byte normal real 17digit: **-1.701411834604692D+38** to **-2.938735877055719D-39**,
2.938735877055719D-39 to **1.701411834604692D+38**
PRINTed as 16digit: -1.701411834604692D+38 to -2.938735877055719D-39,
2.938735877055719D-39 to 1.701411834604692D+38

MC6809, MC6803, MC6800, MOS6502
zero all zero's: **0**
5byte normalized real 10digit as integer: **-4,294,967,296** to **4,294,967,296**
normalized numbers with implicit "0.1", effectively 2^32=4294967296
PRINTed as 8digit: -4.2949673E+09 to 4.2949673E+09
5byte normalized real 11digit: **-1.7014118345E+38** to **-2.9387358771E-39**,
2.9387358771E-39 to **1.7014118345E+38**
PRINTed as 9digit: -1.70141183E+38 to -2.93873588E-39,
2.93873588E-39 to 1.70141183E+38

Integer mantissa values represented by MBF are 100% accurate up to 2^24 and 2^56; or 2^32. All other numbers, integers or fractions beyond the significant digits, are close approximations of the true values, due to the "binary fraction intervals" verses the infinite real number line.

BASIC Interpreters Valid Floating Point Ranges



MBF Using 4Byte and 8Byte Little-Endian
GW-BASIC 3.23 through 1.04

4byte, 8digit -1.7014117E+38 PRINTed as 7digit -1.701412E+38	-16777216 -1.67722E+07	-2.9387359E-39 -2.938736E-39	2.9387359E-39 2.938736E-39	16777216 1.67722E+07	1.7014117E+38 1.701412E+38
8byte, 17digit -1.7014118346046923D+38 PRINTed as 16digit -1.701411834604692D+38	-72057594037927936 7.205759403792794D+16	-2.9387358770557188D-39 -2.938735877055719D-39	2.9387358770557188D-39 2.938735877055719D-39	72057594037927936 7.205759403792794D+16	1.7014118346046923D+38 1.701411834604692D+38

Model 4 TRSDOS 6.x Disk BASIC, LS-DOS 6.x Disk BASIC, DOSPLUS 4 Disk BASIC

4byte, 8digit -1.7014117E+38 PRINTed as 6digit -1.70141E+38	-16777216 -1.6772E+07	-2.9387359E-39 -2.93874E-39	2.9387359E-39 2.93874E-39	16777216 1.6772E+07	1.7014117E+38 1.70141E+38
8byte, 17digit -1.7014118346046922D+38 PRINTed as 16digit -1.701411834604692D+38	-72057594037927936 7.205759403792794D+16	-2.9387358770557177D-39 -2.938735877055719D-39	2.9387358770557177D-39 2.938735877055719D-39	72057594037927936 7.205759403792794D+16	1.7014118346046922D+38 1.701411834604692D+38

Model III/II/I TRSDOS Disk BASIC, *All DOSs Disk BASIC*, Model III BASIC, Model I LEVEL II, MBASIC 5.x

4byte, 8digit -1.7014117E+38 PRINTed as 6digit -1.70141E+38	-16777216 -1.6772E+07	-1.0000048E-38 -1.00001E-38	1.0000048E-38 1.00001E-38	16777216 1.6772E+07	1.7014117E+38 1.70141E+38
8byte, 17digit -1.7014118346046922D+38 PRINTed as 16digit -1.701411834604692D+38	-72057594037927936 7.205759403792794D+16	-1.0000000000000005D-38 -1.0000000000000001D-38	1.0000000000000005D-38 1.0000000000000001D-38	72057594037927936 7.205759403792794D+16	1.7014118346046922D+38 1.701411834604692D+38

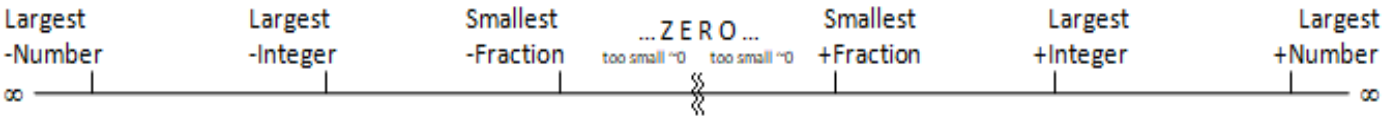
Model III/I LEVEL I

4byte, 7digit -1.701411E+38 PRINTed as 6digit -1.70141E+38	-16777216 -1.6772E+07	-2.938741E-39 -2.93873E-39	2.938741E-39 2.93873E-39	16777216 1.6772E+07	1.701411E+38 1.70141E+38

MBASIC 4.x through ALTAIR BASIC 3.0

4byte, 8digit -1.7014109E+38 PRINTed as 6digit -1.70141E+38	-16777216 -1.6772E+07	-1.0000048E-38 -1.00001E-38	1.0000048E-38 1.00001E-38	16777216 1.6772E+07	1.7014109E+38 1.70141E+38
8byte, 17digit -1.7014118346046922D+38 PRINTed as 16digit -1.701411834604692D+38	-72057594037927936 7.205759403792794D+16	-1.0000000000000005D-38 -1.0000000000000001D-38	1.0000000000000005D-38 1.0000000000000001D-38	72057594037927936 7.205759403792794D+16	1.7014118346046922D+38 1.701411834604692D+38

BASIC Interpreters Valid Floating Point Ranges (Continued)



MBF Using 5Byte Big-Endian
CoCo 3/2/1 SDECB through CB, MC-10 MCB

5byte, 11digit -1.7014118345E+38 PRINTed as 9digit -1.70141183E+38	-4294967296 -4.2949673E+09	-2.9387358771E-39 -2.93873588E-39	2.938735771E-39 2.93873588E-39	4294967296 4.2949673E+09	1.7014118345E+38 1.70141183E+38

MBF Using 8Byte and 4Byte BCD Representation
XENIX MBASIC, Model 200/102/100 DVI, Model 200/102/100 BASIC

8byte, 14digit - 9.9999999999999D+62	-9.9999999999999D+13	-9.9999999999999D-64	9.9999999999999D-64	9.9999999999999D+13	9.9999999999999D+62
4byte, 6digit -9.99999E+62	-9.99999E+5	-9.99999E-64	9.99999E-64	9.99999E+5	9.99999E+62

Additional MBF Comments

a. Floating Points valid ranges are presented in number line order, from negative to zero to positive
b. Values shown as allowed by the BASIC interpreters assignment statement
c. Default precision is listed first, ie top row
d. MBF developed by Monte Davidoff as a contractor for Bill Gates and Paul Allen. Binary representation inspired by DEC PDP-11 BASIC-PLUS [1.4E-39 to 1.7E38, 4Byte or 8Byte big-endian].
e. LEVEL I Floating Point reals developed by Steven W Leininger, in i8080+Z80. Microsoft already contracted for LEVEL II; so in MBF compatible format but coded by Leininger.

MBF Internal Representation

For the sake of brevity, we will focus on the floating point values only. However, here is a glimpse of the internal binary representation 1975-1988 till Microsoft/Intel switched to the IEEE 754-1985 standard: the byte order of the Mantissa, Sign bit, Exponent, and the Exponent's bias. Fortunately, plentiful resources online discuss the details of the Microsoft Binary Format internal representation.

Format

4Byte Single	e7..e0 EXP	s0f22..f16 MSB	f15..f8 2SB	f7..f0 LSB	
8Byte Double	e7..e0 EXP	s0f54..f48 MSB	f47..f40 2SB	f39..f32 3SB	
	f31..f24 4SB	f23..f16 5SB	f15..f8 6SB	f7..f0 LSB	
5Byte	e7..e0 EXP	s0f30..f24 MSB	f23..f16 2SB	f15..f8 3SB	f7..f0 LSB

Value

v = 0 if EXP = 0
v = (-1)^s0 × mantissa × 2^(EXP - 128)
otherwise, v is a binary fraction mantissa, v = 0.1 f54, f30, or f22 f21 f20 ... f0

Little-Endian Order

Memory Low..High 4Byte single = LSBlsb 2SB s0MSB EXP
Memory Low..High 8Byte double = LSBlsb 6SB 5SB 4SB 3SB 2SB s0MSB EXP

Big-Endian Order

Memory Low..High 5Byte = EXP s0MSB 2SB 3SB LSBlsb

MBF-DUMP/BAS

As the old saying goes, if you can't find it, write it; so I started writing a *line or two* of BASIC to verify my understanding of the internal binary formats and the valid ranges. It does not find the valid ranges, I *pre-calculated* the valid ranges and the program stores the pre-calculated values into single precision and double precision variables, then displayed both the PRINT results as well as the raw memory contents. With trial and error, tweaking my calculations and repeated runs, the largest and smallest values were validated.

So here is the listing for MBF-DUMP/BAS in TRS-80 BASIC LEVEL II.

```
1 REM By J A, 25 Jan 2024, BASIC LEVEL II 1978 .. Rev 1.3 1981, Altair 1977..75
10 REM MICROSOFT BINARY FORMAT (MBF) FLOATING-POINT 4BYTE 8BYTE LITTLE-ENDIAN
20 A!=16777215:A%=VARPTR(A!):CLS:'LARGEST SINGLE INTEGER = (2^24) -1
30 REM VARIABLE TYPE/LENGTH AND VARIABLE NAME1(A-Z) AND NAME2(A-Z,0-9)
40 PRINT"MICROSOFT BINARY FORMAT (MBF) FLOATING-POINT":PRINT"4BYTE AND 8BYTE LITTLE-
ENDIAN":PRINT:PRINT "SINGLE TYPE= "; PEEK(A%-3); "   ";
41 'TRSDOS 6.2-> CLS:PRINT "SINGLE TYPE= "; PEEK(A%-4); "   ";
50 PRINT "NAME= "; CHR$(PEEK(A%-1)); CHR$(PEEK(A%-2)); "   "; PEEK(A%-1); PEEK(A%-2); "    ";
51 'TRSDOS 6.2-> PRINT "NAME= "; CHR$(PEEK(A%-3)); CHR$(PEEK(A%-2)); "   "; PEEK(A%-3); PEEK(A%-2); "    ";
60 REM BASIC VALUE AND VARPTR() ADDRESS (POINTS TO EXP BYTE)
70 PRINT "VARPTR= ";A%
80 PRINT "LARGEST SINGLE INTEGER=";A!:GOSUB 1000
90 POKE A%+2,255:PRINT "LARGEST NEGATIVE SINGLE INTEGER= ";A!:GOSUB 1000
100 INPUT "PRESS ANY KEY FOR SINGLE PRECISION";A$
110 POKE A%+2,127:'RESET SIGN BIT FOR POSITIVE MANTISSA
120 POKE A%+3,255:PRINT "LARGEST SINGLE NUMBER= ";A!:GOSUB 1000
130 POKE A%+2,255:PRINT "LARGEST NEGATIVE SINGLE NUMBER= ";A!:GOSUB 1000
140 POKE A%+2,127:'RESET SIGN BIT FOR POSITIVE MANTISSA
150 POKE A%+3,1:PRINT "LARGEST SINGLE FRACTION= ";A!:GOSUB 1000
160 POKE A%+2,255:PRINT "LARGEST NEGATIVE SINGLE FRACTION= ";A!:GOSUB 1000
170 A!=1.0:POKE A%+3,1:PRINT "SMALLEST SINGLE FRACTION= ";A!:GOSUB 1000
180 POKE A%+2,128:PRINT "SMALLEST NEGATIVE SINGLE FRACTION= ";A!:GOSUB 1000
190 INPUT "PRESS ANY KEY TO CONTINUE";A$:CLS
200 A!=0.0:PRINT "ZERO= ";A!:GOSUB 1000
210 A!=1.0:PRINT "ONE= ";A!:GOSUB 1000
220 A!=-1.0:PRINT "-ONE= ";A!:GOSUB 1000
230 A!=1234567:PRINT "VALUE= ";A!:GOSUB 1000
240 A!=1.234567:PRINT "VALUE= ";A!:GOSUB 1000
250 INPUT "PRESS ANY KEY FOR DOUBLE PRECISION";A$:CLS

310 REM MBF DOUBLE
320 A#=72057594037927935:A%=VARPTR(A#):'LARGEST DOUBLE INTEGER = (2^56) -1
330 REM VARIABLE TYPE/LENGTH AND VARIABLE NAME1(A-Z) AND NAME2(A-Z,0-9)
340 PRINT "DOUBLE TYPE= "; PEEK(A%-3); "   ";
341 'TRSDOS 6.2-> PRINT "DOUBLE TYPE= "; PEEK(A%-4); "   ";
350 PRINT "NAME= "; CHR$(PEEK(A%-1)); CHR$(PEEK(A%-2)); "   "; PEEK(A%-1); PEEK(A%-2); "   ";
351 'TRSDOS 6.2-> PRINT "NAME= "; CHR$(PEEK(A%-3)); CHR$(PEEK(A%-2)), PEEK(A%-3); PEEK(A%-2),
360 REM BASIC VALUE AND VARPTR() ADDRESS (POINTS TO EXP BYTE)
370 PRINT "VARPTR= "; A%
380 PRINT "LARGEST DOUBLE INTEGER= ";A#:GOSUB 2000
390 POKE A%+6,255:PRINT "LARGEST NEGATIVE DOUBLE INTEGER= ";A#:GOSUB 2000
```

```
400 INPUT "PRESS ANY KEY TO CONTINUE";A$:CLS
410 POKE A%+6,127:'RESET SIGN BIT FOR POSITIVE MANTISSA
420 POKE A%+7,255:PRINT "LARGEST DOUBLE NUMBER= ";A#:GOSUB 2000
430 POKE A%+6,255:PRINT "LARGEST NEGATIVE DOUBLE NUMBER= ";A#:GOSUB 2000
440 POKE A%+6,127:'RESET SIGN BIT FOR POSITIVE MANTISSA
450 POKE A%+7,1:PRINT "LARGEST DOUBLE FRACTION= ";A#:GOSUB 2000
460 POKE A%+6,255:PRINT "LARGEST NEGATIVE DOUBLE FRACTION= ";A#:GOSUB 2000
470 A#=1.0:POKE A%+7,1:PRINT "SMALLEST DOUBLE FRACTION= ";A#:GOSUB 2000
480 POKE A%+6,128:PRINT "SMALLEST NEGATIVE DOUBLE FRACTION= ";A#:GOSUB 2000
490 INPUT "PRESS ANY KEY TO CONTINUE";A$:CLS
500 A#=0.0:PRINT "ZERO=";A#:GOSUB 2000
510 A#=1.0:PRINT "ONE= ";A#:GOSUB 2000
520 A#=-1.0:PRINT "-ONE=";A#:GOSUB 2000
530 A#=1234567890123456:PRINT "VALUE=";A#:GOSUB 2000
540 A#=1.234567890123456:PRINT "VALUE=";A#:GOSUB 2000
550 INPUT "PRESS ANY KEY TO END";A$:CLS
560 END:'DONE

1000 REM SINGLE PRECISION 4BYTE LITTLE-ENDIAN MEMORY CONTENT
1010 PRINT "MEMORY= ";
1020 FOR I%=0 TO 3
1030 PRINT PEEK(A%+I%);
1040 NEXT I%
1050 PRINT
1060 RETURN

2000 REM DOUBLE PRECISION 8BYTE LITTLE-ENDIAN MEMORY CONTENT
2010 PRINT "MEMORY= ";
2020 FOR I%=0 TO 7
2030 PRINT PEEK(A%+I%);
2040 NEXT I%
2050 PRINT
2060 RETURN
```

MBF-DUMP/BAS, with few minor changes, is easily adaptable for the Model 4, CP/M MBASIC, GW-BASIC, etc. I am confident many inquisitive readers can eloquently enhance my simple brute force program for both speed and portability.

Please see the comments lines for changes to adapt it to different BASICs, they are along the lines:

TRSDOS 6.2 DISK BASIC:	40 PEEK(A%-4), 50 PEEK(A%-3); PEEK(A%-2)
MSDOS GWBASIC 3.23:	same as TRSDOS 6.2 DISK BASIC
CP/M MBASIC 5.28:	same as GWBASIC 3.23 but remove all CLS
ALTAIR BASIC 4.1-3.0:	same as LEVEL II but remove all CLS

BASIC Manuals Floating Point Ranges Discrepancies

None of the manuals accurately list the full range, including largest and the smallest values specified using all the significant digits.

GWBASIC 1991 3.20-2.11 1984, *Microsoft GW-BASIC User's Guide*, July 1987, page 50
- 3.0 x 10^-39 to 1.7 x 10^38
- Single precisions are stored 7 digits, only 6 accurate
- Double precisions are stored 17 digits, only 16 accurate

GWBASIC 1984 2.11-1.04 1982, *MINDSET Microsoft GW-BASIC Reference Manual*, 1984, page 2-6
- 10^-38 to 10^+38
- Single precisions are stored 7 digits, only 6 accurate
- Double precisions are stored 17 digits, only 16 accurate

IBM BASIC 1984 3e 3.0-1.10 2e, *IBM BASIC Manual v1.10 2e*, May 1982, page 3-10
- 2.9E-39 to 1.7E+38
- Single precisions are stored 7 digits, only 6 accurate
- Double precisions are stored 17 digits, only 16 accurate

MBASIC 5.0 1979/BASCOM 5.3 1981, *Microsoft BASIC-80 Reference Manual 5.0*, 1979, page 1-5
- 10^-38 to 10^+38
- Single precisions are stored 7 digits, only 6 accurate
- Double precisions are stored 17 digits, only 16 accurate

TRS-80 Model III/4 Operation and BASIC Reference Manual 2e, February 1983, page 220
TRS-80 Model III Operation and BASIC Reference Manual 2e, February 1981, page 220
- Single Precisions -1.701411E+-38 to 1.701411E+-38
- Double Precisions -1.701411834544556D+-38 to 1.701411834544556D+-38

TRS-80 Model I LEVEL II BASIC Reference Manual 2e, 1979, page A-17
- Single Precisions -1.701411E+-38 to 1.701411E+-38
- Double Precisions -1.701411834544556D+38 to 1.701411834544556D+38

TRS-80 Model I LEVEL II BASIC Reference Manual, 1978, page A/16
- Single Precisions -1.701411E+38 to 1.701411E+38
- Double Precisions -1.701411834544556E+38 to 1.701411834544556E+38

TRS-80 Model I User's Manual for LEVEL I, 1978, page 38, page 69, page 186
- Single precisions are stored 8 digits, only 5 to 6 accurate
- accepts scientific notation n.nE+-nn

MITS Altair BASIC Reference Manual
4.1, July 1977, page 11, page 13: 7 digits and 17 digits; 2.9387E-38 to 1.70141E+38
4.0 January 1977, page 11, page 13: 7 digits and 17 digits; 2.9387E-38 to 1.70141E+38
EXTENDED 16K 3.2, December 1975, page 81: 16 digits; 10^-38 to 10^+38
3.0, August 1975, page 6: 7 digits but only 6 accurate; 2.93875 x 10^-39 to 1.70141 x 10^+38

I hope you have found the journey fun and useful. *Happy* computing.

Dedication

To all the TRS-80 enthusiasts out there, who have been sharing their knowhow with new generation of Trash80 users. 8bit computing is timeless and provides complete hands-on to the bare metal, *putting you in full control of entire computer and best of all, in direct control of the CPU. I wish all of you* Happy Computing!

Acknowledgments

I am indebted to those who are kind enough to share their knowledge and their love for 8bit computing, they are too many to thank by name but please allow to thank the few whom I annoyed the most with my questions yet received their abundance support:

Vernon B Hester for his plethora of software culminating with Esoteric/MultiDOS, and his generous insight into BASIC, BASICG, Z80 and any/all topics of technology. *George and Peter Phillips* for their truly amazing trs80gp emulator. *Ira Goldklang* for reviving the TRS-80 and all things TRS-80, especially the Model I, III, 4, 4P, 4D. *Peter Cetinski* for his preservation of the Model II family of computers (M2/M16/M12/M16B/6000). *Chuck(G)* on vcfed.org forums for his many CP/M insights and early computers. *Dusty Miller* for editing and publishing the *TRS8BIT newsletter*, with special gratitude for his encouragement and support with which all my articles would not have been possible. And many, many more enthusiasts. I thank *all* of you for your help over the decades, and sincerely wish you all the best.

Further Readings

ROM Routines Documented. Jack Decker. 1984.
MBF Single and Double precision representation: pages 23-25. Excellent ROM details book.

More about Arithmetic Operations (PDF). Hubert Howe. *COMPUTRONICS* March 1980.
MBF representation, BCD representation: pages 545-548.

Fast BASIC – Beyond TRS-80 BASIC USR EDTASM. George Gratzer. 1982.
Variable Table: pages 52-55; MBF Single and Double precision details: pages 227-229.

Microsoft BASIC Decoded and Other Mysteries. James Farvour. 1981.
MBF Floating Point Accumulators WRA1+WRA2: page 12; VLT, VARPTR(): pages 44-46.

LEVEL 1 Disassembly. Andy Vaught Mesa. 2018.
4Byte variables, 5Byte FPACC, https://github.com/andyv/level1/blob/master/memory.txt

Machine Language Programming for the "8008". Nat Wadsworth. SCELBI 1975.
Identical format to MBF and LEVEL I: page 5-13; Excellent Arithmetic: pages 5-1 through 5-45.

Microsoft KB35826 (PDF). Microsoft. 2006.

MBF Format, https://en.wikipedia.org/wiki/Microsoft_Binary_Format

William Kahan, IEEE 754, http://www.cs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF

LLL BASIC Floating Point Math Package. David Mead. *Interface Age* February 1977.
Public Domain floating point implementation: pages 104-108, 120.

Michael Maples, LLL Floating Point, https://www.retrotechnology.com/herbs_stuff/float.html

Monte Davidoff, https://www.theregister.com/2001/05/11/microsoft_altair_basic_legend_talks/



SUPPORTED BY YOU AND PRODUCED AT WWW.TRS-80.ORG.UK
THE HOME OF THE TRS80BIT NEWSLETTER

WHERE 8 BITS AND 16K STILL RULE OK