

BASIC 2.0 / BASIC 4.0 ROM Routines

Jim Butterfield, Toronto Ont.

The BASIC 4.0 40-character and 80-character machines are the same except for addresses \$E000-\$E7FF. This map shows where various routines lie. The first address is not necessarily the proper entry for the routine. Similarly, many routines require register setup or data preparation before calling.

BASIC 2.0 ROM Routines

Address	Description	Address	Description	Address	Description	Address	Description
C000 - C045	Action addresses for primary keywords	CDEC	Evaluate expr. within ()	D8C8 - D815	Constants	E34C - E38A	Set screen print parameters
C046 - C073	Action addresses for functions	CD2F - CE02	Check parentheses, comma	D8F6	Perform [LOG]	E38B - E395	Prevent 80-char line getting longer
C074 - C091	Hierarchy & action adds for operators	CE03 - CE07	Syntax error exit	D937 - D997	Perform multiplication	E396 - E3B3	Turn 40 char line into 80 char line
C092 - C192	Table of BASIC keywords	E088 - E288	Variable name setup	D998 - D9C2	Unpack memory into accum*2	E3B4 - E3D7	Back into previous line
C193 - C2A9	BASIC messages, mostly error msgs	CE89 - CE7F	Set up function references	D9C3 - D9DF	Test & adjust accumulators	E3D8 - E518	Handle ASCII char for screen output
C2AA - C2D7	Search stack FOR/GOSUB	CEC8 - CE7F	Perform [OR], [AND]	D9E0 - D9ED	Handle overflow and underflow	E519 - E53E	Go to next screen line
C2D8 - C31A	Open up space in memory	CEF8 - CF5F	Perform comparisons	D9EE - DA04	Multiply by 10	E53F - E5B9	Scroll screen
C31B - C327	Test: stack too deep?	CF60 - CF6C	Perform [DIM]	DA05 - DA09	10 in floating binary	E5BA - E61A	Open a line on screen
C328 - C354	Check available memory	CF6D - CF7E	Search for variable	DA0A	Divide by 10	E61B - E62D	Main Interrupt entry
C355	Send canned error message, then:	CF7F - D077	Create new variable	DA13	Perform divide-by	E62E - E6E9	Interrupt: clock, cursor, keyboard
C359 - C3AA	Warm start (ready)	D078 - D088	Setup array pointer	DA1E - DAAD	Perform divide-into	E6EA - E6F7	Output character
C3AB - C441	Handle new BASIC line input	D089 - D08C	32768 in floating binary	DAAE - DAD2	Unpack memory into accum*1	E6F8 - E756	Table: keyboard matrix decoder
C442 - C46E	Rebuild chaining of BASIC lines	D08D - D0AB	Evaluate integer expression	DAD3 - DB07	Pack accum*1 into memory	E75A - E796	MLM sub: output hex digits
C46F - C494	Receive line from keyboard	DA0C - D258	Find or make array	DB08 - DB17	Move accum*2 to *1	E797 - E7A6	MLM sub: swap TMP0 and TMP2
C495 - C52B	Search keywords into BASIC tokens	D259 - D27F	Perform [FRE], and:	DB18 - DB26	Move accum*1 to *2	E7A7 - E7F6	MLM sub: input hex digits
C52C - C55A	Search BASIC for given line number	D260 - D279	Convert fixed-to-floating	DB27 - DB36	Round accum*1	E7F7 - E7FF	File messages
C55B	Perform [NEW], and:	D27A - D27F	Perform [POS]	DB37 - DB44	Get accum*1 sign	F000 - F0B5	File messages
C577 - C5A6	Perform [CLR]	D280 - D28C	Check not Direct	DB45 - DB63	Perform [SGN]	F0B6 - F127	Send 'Talk', 'Listen', IEEE command
C5A7 - C5B4	Reset BASIC execution to start	D28D - D2BA	Perform [DEF]	DB64 - DB66	Perform [ABS]	F128 - F135	Send char to IEEE
C5B5 - C657	Perform [LIST]	D2BB - D2CD	Check Fnx syntax	DB67 - DBA6	Compare accum*1 to memory	F136 - F155	Write Timeout, Device Not Present
C658 - C6FF	Perform [FOR]	D2CE - D33E	Evaluate FNX	DBA7 - DBD7	Floating-to-fixed	F156 - F163	Send canned I/O message
C700 - C72F	Execute BASIC statement	D33F - D34E	Perform [STR\$]	DBD8 - DBFE	Perform [INT]	F164 - F16E	Send 'Listen', secondary address
C730 - C73E	Perform [RESTORE]	D34F - D360	Do string vector	DBFF - DC89	Convert string to floating-point	F16F - F17E	Send normal (deferred) IEEE char
C73F - C76A	Perform [STOP] or [END]	D361 - D3CD	Scan, set up string	DC8A - DCBE	Get new ASCII digit	F17F - F18B	Drop IEEE device
C76B - C784	Perform [CONT]	D3CE - D3FF	Allocate space for string	DCBF - DCCD	Constants	F18C - F1D0	Input byte from IEEE
C785 - C78F	Perform [RUN]	D400 - D516	Garbage collection	DCCE	Print IN, then:	F1D1 - F1E0	GET a byte
C790 - C7AC	Perform [GOSUB]	D517 - D553	Concatenate	DCD9 - DCE8	Print BASIC line *	F1E1 - F231	INPUT a byte
C7AD - C7D9	Perform [GOTO]	D554 - D57C	Store string	DCE9 - DE1C	Convert floating-point to ASCII	F232 - F26D	Output a byte
C7DA	Perform [RETURN], then:	D57D - D5B4	Discard unwanted string	DE1D - DE5D	Constants	F26E	Abort files
C7F3 - C80D	Perform [DATA], skip statement	D5B5 - D5C5	Clean descriptor stack	DE5E	Perform [SOR]	F284 - F28C	Restore default I/O devices
C80E	Scan for next BASIC statement	D5C6 - D5D9	Perform [CHRS]	DE88	Perform power function	F28D - F2A8	Find/setup file data
C811 - C82F	Scan for next BASIC line	D5DA - D605	Perform [LEFT\$]	DEA1 - DEAB	Perform negation	F2A9 - F300	Perform [CLOSE]
C830	C843 - C852	D606 - D610	Perform [RIGHT\$]	DEAC - DED9	Constants	F301 - F30E	Test STOP key
C831 - C852	Perform [REM]: skip line	D611 - D63A	Perform [MID\$]	DEDA - DF2C	Perform [EXP]	F30F - F314	Action STOP key
C853 - C872	Perform [ON]	D63B - D655	Pull string data	DF2D - DF76	Series evaluation	F315 - F31C	Send message if Direct mode
C873 - C8AC	Accept fixed-point number	D656 - D689	Perform [LEN]	DF77 - DF7E	RND constants	F31D - F321	Test if Direct mode
C8AD - C98A	Perform [LET]	D68C - D664	Switch string to numeric	DF7F - DFD7	Perform [RND]	F322 - F3C1	Program load subroutine
C98B - C990	Perform [PRINT*]	D665 - D674	Perform [ASC]	DFF8 - DF7F	Perform [COS]	F3C2 - F409	Perform [LOAD]
C991 - C9A4	Perform [CMD]	D675 - D686	Get byte parameter	DFDF - E027	Perform [SIN]	F40A - F43D	Print Searching, Loading, Verifying
C9A5 - CA1B	Perform [PRINT]	D687 - D6C5	Perform [VAL]	E028 - E053	Perform [TAN]	F43E - F45F	Get Load/Save parameters
CA1C - CA38	Print string from memory	D6C6 - D6D1	Parameters for POKE/WAIT	E054 - E08B	Constants	F460 - F465	Get a byte parameter
CA39 - CA4E	Print single format character	D6D2 - D6E7	Convert floating-to-fixed	E08C - E0BB	Perform [ATN]	F466 - F493	Send filename to IEEE
CA4F - CA7C	Handle bad input data	D6E8 - D706	Perform [PEEK]	E0BC - E0F8	Constants	F494 - F4B6	Find specific tape header
CA7D - CA6A	Perform [GET]	D707 - D70F	Perform [POKE]	E0F9 - E110	CHRG/CHR sub for zero page	F4B7 - F4CD	Perform [VERIFY]
CAA7 - CAC0	Perform [INPUT*]	D710 - D72B	Perform [WAIT]	E111 - E115	Initial RND seed	F4CE - F50D	Get Open/Close parameters
CAC1 - CAF9	Perform [INPUT]	D72C - D732	Add 0.5	E116 - E1B6	BASIC cold start	F50E - F515	Abort if end-of-line
CAF9 - CB06	Prompt and receive input	D733 - D744	Perform subtraction	E1B7 - E1DD	Power up msg. "bytes free"	F516 - F520	Check comma, else Syntax Error
CB07 - CBFB	Perform [READ]	D745 - D76D	Microsoft Joke (WAIT 6502)	E1DE	Init I/O regs and:	F521 - F5A5	Perform [OPEN]
CBFC - CC1F	Canned Input error messages	D76E - D852	Perform addition	E229	Clear screen and:	F5A6 - F5D9	Find any tape header
CC20 - CC78	Perform [NEXT]	D853 - D889	Complement accum*1	E257 - E284	Home cursor on:	F5DA - F63B	Write tape header
CC79 - CC9E	Check type mismatch	D88A - D88E	Overflow exit	E285 - E33E	Input from screen or keyboard	F63C - F655	Get start/end address from header
CC9F	Evaluate expression	D88F - D8C7	Multiply-a-byte	E33F - E34B	Test for quote, test quote flag	F656 - F66B	Set buffer address

BASIC 4.0 ROM Routines

Address	Description	Address	Description	Address	Description	Address	Description
B000 - B065	Action addresses for primary keywords	C086 - C0B5	Perform [OR], [AND]	CCD8 - CCFC	Unpack mem. into accum*1	DB9E - DBD6	Query ARE YOU SURE?
B066 - B093	Action addresses for functions	C0B6 - C11D	Perform comparisons	CCFD - CD31	Pack accum*1 into memory	DBD7 - DBE0	Print BAD DISK
B094 - B0B1	Hierarchy & action adds for operators	C11E - C12A	Perform [DIM]	CC32 - CD41	Move accum*2 to *1	DBE1 - DBF9	Clear DS\$ and ST
B0B2 - B20C	Table of BASIC keywords	C12B - C1BF	Search for variable	CD42 - CD50	Move accum*1 to *2	DBFA - DC67	Assemble disk command string
B20D - B321	BASIC messages, mostly error msgs	C1C0 - C2C7	Create new variable	CD51 - CD60	Round accum*1	DC68 - DE29	Parse BASIC DOS command
B322 - B34F	Search stack for FOR/GOSUB	C2C8 - C2D8	Setup array pointer	CD61 - CD6E	Get accum*1 sign	DE2C - DE48	Get Device number
B350 - B392	Open up space in memory	C2D9 - C2DC	32768 in floating binary	CD6F - CD8D	Perform [SGN]	DE49 - DE86	Get file name
B393 - B39F	Test: stack too deep?	C2DD - C2FB	Evaluate integer expression	CD8E - CD90	Perform [ABS]	DE87 - DE9C	Get small variable parameter
B3A0 - B3CC	Check available memory	C2FC - C4A7	Find or make array	CD91 - CD9D	Compare accum*1 to memory	**Entry points only for E000-E7FF**	
B3CD	Send canned error message, then	C4A8	Perform [FRE], and:	CD1D - CE01	Floating-to-fixed	E000	Register/Screen initialization
B1FF - B1FE	Handle new BASIC line input	C4BC - C4CN	Convert fixed-to-floating	CE02 - CE28	Perform [INT]	E001	Input from keyboard
B1F0 - B4E1	Rebuild chaining of BASIC lines	C4CA - C4CE	Check not Direct	CE29 - CE3B	Convert string to floating-pt	E1A7	Input from screen
B4E2 - B4F8	Receive line from keyboard	C4CF - C4FB	Check not Direct	CE3C - CE3E	Get new ASCII digit	E202	Output character
B4F9 - B5A2	Crunch keywords into BASIC tokens	C4FC - C509	Perform [DEF]	CE3F - CE98	Constants	E203	Output character
B5A3 - B5D1	Search BASIC for given line number	C50A - C51C	Check Fnx syntax	CF78	Print IN, then:	E442	Main Interrupt entry
B5D2	Perform [NEW], and:	C51D - C58D	Evaluate FNX	CF7F - CF92	Print BASIC line *	E455	Interrupt: clock, cursor, keyboard
B5EC - B621	Perform [CLR]	C58E - C59D	Perform [STR\$]	CF93 - D00C	Convert floating-pt to ASCII	E600	Exit from Interrupt
B622 - B62F	Reset BASIC execution to start	C59E - C5A4	Do string vector	D0C7 - D107	Constants	F000 - F0D1	File messages
B630 - B6DD	Perform [LIST]	C5B0 - C61C	Scan, set up string	D108	Perform [SOR]	F0D2	Send 'Talk'
B6DE - B784	Perform [FOR]	C61D - C669	Allocate space for string	D112	Perform power function	F0D5	Send 'Listen'
B785 - B7B6	Execute BASIC statement	C66A - C74E	Garbage collection	D148 - D155	Perform negation	F0D7	Send IEEE command character
B7B7 - B7C5	Perform [RESTORE]	C74F - C78B	Concatenate	D156 - D183	Constants	F109 - F142	Send byte to IEEE
B7C6 - B7ED	Perform [STOP] or [END]	C78C - C7B4	Store string	D184 - D1D6	Perform [EXP]	F143 - F150	Send byte and clear ATN
B7EF - B807	Perform [CONT]	C7B5 - C810	Discard unwanted string	D1D7 - D228	Series evaluation	F151 - F16B	Option: timeout or wait
B808 - B812	Perform [RUN]	C811 - C821	Clean descriptor stack	D229 - D281	Perform [RND]	F16C - F16F	DEVICE NOT PRESENT
B813 - B82F	Perform [GOSUB]	C822 - C835	Perform [CHRS]	D282	Perform [COS]	F170 - F184	Timeout on read, clear control lines
B830 - B85C	Perform [GOTO]	C836 - C861	Perform [LEFT\$]	D289 - D2D1	Perform [SIN]	F185 - F192	Send canned file message
B85D	Perform [RETURN], then:	C862 - C86C	Perform [RIGHT\$]	D2D2 - D2FD	Perform [TAN]	F193 - F19D	Send byte, clear control lines
B883 - B890	Perform [DATA], skip statement	C86D - C881	Pull string data	D2FE - D32B	Constants	F19E - F1AD	Send normal (deferred) IEEE char
B891	Scan for next BASIC statement	C882 - C887	Perform [LEN]	D32C - D358	Perform [ATN]	F1AE - F1B7	Drop IEEE device
B892 - B8B2	Scan for next BASIC line	C888 - C8C1	Switch string to numeric	D35C - D398	Constants	F1C0 - F204	Input byte from IEEE
B8B3	Perform [IF], and perhaps:	C8C1 - C8D0	Perform [ASC]	D399 - D3B5	CHRG/CHR sub for zero page	F205 - F214	GET a byte
B8C6 - B8D5	Perform [REM]: skip line	C8D1 - C8E2	Get byte parameter	D3B6 - D471	BASIC cold start	F215 - F265	INPUT a byte
B8D6 - B8F5	Perform [ON]	C8E3 - C920	Perform [VAL]	D472 - D716	Machine Language Monitor	F266 - F2A1	Output a byte
B8F6 - B92F	Accept fixed-point number	C921 - C92C	Parameters for POKE/WAIT	D717 - D7AB	MLM subroutines	F2A2	Abort files
B930 - B987	Perform [LET]	C92D - C942	Convert floating-to-fixed	D7AC - D802	Perform [RECORD]	F2A6 - F2C0	Restore default I/O devices
B988 - B9A0	Perform [PRINT*]	C943 - C959	Perform [PEEK]	D803 - D837	Disk parameter checks	F2C1 - F2DC	Find/setup file data
B9A1 - BAA1	Perform [CMD]	C95A - C962	Perform [POKE]	D838 - D872	Dummy disk control msgs	F2DD - F334	Perform [CLOSE]
BAA2 - BB1C	Perform [PRINT]	C963 - C97E	Perform [WAIT]	D873 - D919	[CATALOG] or [DIRECTORY]	F335 - F342	Test STOP key
BB1D - BB39	Print string from memory	C97F - C985	Add 0.5	D91A - D92E	Output	F343 - F348	Action STOP key
BB3A - BB4B	Print single format character	C986	Perform subtraction	D92F - D941	Find spare secondary address	F349 - F350	Send message if Direct mode
BB4C - BB79	Handle bad input data	C988 - CA7C	Perform addition	D942 - D976	Perform [APPEND]	F351 - F355	Test if Direct mode
BB7A - BBA3	Perform [GET]	CA7D - CAB3	Complement accum*1	D977 - D990	Perform [APPEND]	F356 - F400	Program load subroutine
BBB4 - BBBD	Perform [INPUT*]	CAB4 - CAB8	Overflow exit	D991 - D991	Get disk status	F401 - F448	Perform [LOAD]
BBBE - BBF4	Perform [INPUT]	CAB9 - CAF1	Multiply-a-byte	D9D2 - DA06	Perform [HEADER]	F449 - F46C	Print SEARCHING
BBF5 - BC01	Prompt and receive input	CAF2 - CB1F	Constants	DA07 - DA30	Perform [CLOSE]	F46D - F47C	Print LOADING or VERIFYING
BC02 - BC06	Perform [READ]	CB20	Perform [LOG]	DA31 - DA84	Set up disk record	F47D - F4A4	Get Load/Save parameters
BC07 - BC18	Canned Input error messages	CB2E - CB81	Perform multiplication	DA85 - DA7D	Perform [COLLECT]	F4A5 - F4D2	Send name to IEEE
BC19 - BC71	Perform [NEXT]	CB82 - CBEC	Unpack mem. into accum*2	DA7E - DA66	Perform [BACKUP]	F4D3 - F4F5	Find specific tape header
BC72 - B097	Check type mismatch	CBED - CC39	Test & adjust accumulators	DA77 - DAC6	Perform [COPY]	F4F6 - F50C	Perform [VERIFY]
B098	Evaluate expression	CC0A - CC17	Handle overflow & underflow	DAC7 - DAD3	Perform [CONCAT]	F50D - F55F	Get Open/Close parameters
B099	Evaluate expr. within parentheses	CC18 - CC2E	Multiply by 10	DAD4 - DB9C	Insert command string values	F560 - F5E4	Perform [OPEN]
B0E9	Check parentheses, comma	CC2F - CC33	10 in floating binary	DBA0 - DB39	Perform [DSAVE]	F5E5 - F618	Find any tape header
B0F0 - BF0B	Syntax error exit	CC34	Divide by 10	DB3A - DB85	Perform [DLOAD]	F619 - F67A	Write tape header
BF0C - C046	Variable name setup	CC3D	Perform divide-by	DB86 - DB98	Perform [SCRATCH]	F67B - F694	Get start/end address from header
C047 - C085	Set up function references	CC45 - C0D7	Perform divide-into	DB99 - DB9D	Check Direct command	F695 - F6AA	Set buffer address