BBC MICRO USER

Programmer's handbook

A quick reference guide to essential programming information

Compiled by Ian R. Hirst

BBC Basic Keywords

ABS	absolute value	ERROR=ABS(X2-X1)	
ACS	arc - cosine	ANGLE=ACS(0.4)	ANGLE in radians
ADVAL	analogue to digital converter value	X=ADVAL(Z)	see note (1)
AND	logical AND or bitwise AND	IF X=9 AND Y=0 THEN... A=B AND C	logical bitwise
ASC	ASCII code for 1st char. of string	PRINT ASC("GOOD")	prints 71, ASCII code for "G"
ASN	arc-sine	ANGLE=ASN(0.5)	ANGLE in radians
ATN	arc-tangent	ANGLE=ATN(1.5)	ANGLE in radians
AUTO	auto line number	AUTO 100,5	start line number from 100, increment by 10
BGET#	get a byte from file	X=BGET#3	X becomes next value from file
BPUT#	put a byte to file	BPUT#3,5	sends value 5 to file 3
CALL	call machine code subroutine	CALL &FFF4 CALL 1234,A	call subroutine at FFF4(HEX) pass A in parameter block and call subroutine at 1234(DEC)
CHAIN	load and run program	CHAIN "PROG"	clears all except resident integer variables
CHR$	character string	PRINT CHR$(71)	prints character whose ASCII code is 71 ie "G"
CLEAR	clear variables	CLEAR	all except @% to Z%
CLOSE#	close file	CLOSE# CLOSE#0	close file 3 close all files
CLG	clear graphics screen	CLG	clear graphics area to current graphics background area
CLS	clear text screen	CLS	clear text area (incl.graphics) to current text background colour
COLOUR	select text colours	COLOUR 1 COLOUR (128+1)	set text foreground to red set text background colour to red
COS	cosine	PRINT COS(ANGLE)	ANGLE in radians
COUNT	count characters 'printed'	A=COUNT	counts all characters 'PRINT'ed
DATA	data to be READ	DATA 1,2,3,"A"	
DEF	define	DEF FNVAT (g)=0.15*g DEF PROCINIT	define functions and procedures
DEG	degree	Y=DEG(A)	converts angles in radians to degrees
DELETE	delete lines	DELETE 120,200	delete from line 120 to 200 inclusive
DIM	dimension array	DIM A(10) DIM X 10	create 11 element numeric array reserve 10 bytes pointed to by X
DIV	integer division	PRINT 11DIV2	produces 5 ie INT(11/2)
DRAW	draw	DRAW X,Y	draw from last point to X,Y in current graphics foreground colour
ELSE	else	IF X=0 THEN C=D ELSE...	alternative action for IF
END	end	END	optional program terminator
ENDPROC	end procedure	ENDPROC	
ENVELOPE	sound envelope	ENVELOPE (14 PARAMETERS)	see note(2)
EOF#	end of file	X=EOF#3	X is -1 if end of file 3 has been reached
EOR	exclusive OR	R=X EOR Y	bitwise logical operation
ERL	error line number	X=ERL	gives line number of last error
ERR	error	X=ERROR	gives error number of last error
EVAL	evaluate	X=EVAL(A$)	evaluate function in A$
EXP	exponent	Y=EXP(X)	Y becomes E raised to the power X

EXT#	extent	N=EXT#(3)	determines size of file 3 in bytes (not cassette)
FALSE	false	X=FALSE	numerical value (0) representing false
FN	function	A=FNMEAN(X,Y)	indicates user defined function
FOR	for....next	FOR X=1 TO 5	loop executed at least once
GCOL	graphics colour	GCOL 0,2	see note (3)
GET	get key code	X=GET	no echo to screen, X contains key code
GET$	get string	X$=GET$	no echo to screen, X$ contains key character
GOSUB	go to subroutine	GOSUB 4000	
GOTO	goto line	GOTO 330	
HIMEM	highest memory	HIMEM=HIMEM-40	variable pointing to highest memory used
IF	if	IF X=2 THEN 200	THEN, and ELSE optional
INKEY	input key number	Y=INKEY(5)	Y returns value of any key pressed within 50ms
		Y=INKEY(-66)	tests for A being pressed, -1 : no key
INKEY$	input key string	Y$=INKEY$(20)	waits 200ms, returns $
INPUT	input to computer	INPUT X	accepts numeric, prints ?
		INPUT "TYPE NAME" N$	suppresses ? (no comma)
		INPUT "WHAT AGE",A	prints ?
		INPUT LINE A$	accepts anything in a line
INPUT	input from file	INPUT#3, A,B	inputs from file 3
INSTR	in string	INSTR("HELLO", "L")	returns position of 2nd string in 1st(ie 3)
INT	integer part	X=INT(5.3)	next smallest integer (ie 5)
		X=INT(-5.3)	gives -6
LEFT$	left string	PRINT LEFT$(A$,N)	prints left N characters of A$
LEN	length of string	X=LEN(A$)	returns length of A$
LET	let	LET A=2	LET is optional
LIST	list prog.	LIST 200	print line 200
		LIST 100,300	list from line 100 to 300
		LIST, 300	list upto 300
LISTO	list option	LISTO 0	insert no spaces during LIST
		LISTO 7	format with spaces
LN	natural log	X=LN(A)	
LOAD	load program	LOAD "MYPROG"	clears all variables except @% to Z%
LOCAL		LOCAL A,B,Z$	declares variables local in PROC or FUNCT.
LOG	logarithm	Y=LOG(X)	log to base 10 of x
LOMEM	low memory	PRINT LOMEM	pointer to start of variable memory
MID$	middle string	A$=MID$(Y$,I,J)	A$ returns J characters starting from Ithin Y$
MOD	modulus	X=12 MOD 5	returns remainder after division (ie 2)
MODE	graphics mode	MODE 5	select graphics mode see note (4)
MOVE	move	MOVE X,Y	move graphics cursor to absolute position
NEW	new program	NEW	removes a program
NEXT	FOR...NEXT	NEXT	indicates end of FOR...NEXT loop
		NEXT I	'pops' FOR-NEXT until I is found
NOT	logical NOT	IF NOT (A=6) THEN	logical not
		A= NOT 8	equivalent to unary minus (A=-B)
OLD	old program	OLD	restore program deleted by NEW
ON	on	ON X GOTO 1000,2000	if X=1 goes to 1000, if 2, then 2000
		ON X GOSUB 100,200,300	
		ON ERROR GOTO 1500	errors handled by 1500
		ON ERROR OFF	restores normal error handling
OPENIN	open file for input	X=OPENIN("filename")	opens file called "filename" and allocates channel number
OPENOUT	open file for output	X=OPENOUT("filename")	if file does not exist, it is created
OPT	assembler option	OPT N	N=0 errors suppressed, no listing
			N=1 errors suppressed, listing
			N=2 errors reported, no listing
			N=3 errors reported, listing
OR	logical OR	IF A=6 OR B=2 THEN...	
PAGE	page	PAGE=&5000	variable which points to program start
PI	pi	AREA=2*PI*R	constant=3.14159265
PLOT	plot	PLOT 4,50,50	multi purpose see note(6)

POINT	point	colour=POINT(X,Y)	returns colour of point X,Y(-1 off screen)
POS	position	X=POS	returns position of text cursor
PRINT	print	PRINT A,B	for format see note (7)
PRINT#	print to file	PRINT#3,A	prints A to file 3
PROC	procedure	PROCINIT	indicates user defined procedure
PTR#	pointer	PTR#4=100	sets pointer to non-cassette file
RAD	radian	X=RAD(Y)	converts from degrees to radians
READ	read	READ A$,2	read from DATA statement
REM	remark	REM comment	indicates rest of line is comment
RENUMBER		RENUMBER	renumbers from 10 with increments of 10
		RENUMBER 100,20	renumbers from 100 with increments of 20
		RENUMBER 50	renumbers 50 with increments of 10
REPEAT	repeat...until	REPEAT	repeat loop always performed at least once
REPORT	report error	REPORT	reports last error in words
RESTORE	restore data pointer	RESTORE	sets data-pointer to first DATA line
		RESTORE 2000	sets data-pointer to DATA line 2000
RETURN	return from subroutine	RETURN	indicates end of subroutine
RIGHT$	right string	A$=RIGHT$(X$,3)	selects last 3 char of X$
RND	random	X=RND(N)	N<0 resets generator
			N>0 generates random no >0 and <N
			N=0 reports last number with N=1
		X=RND	generates integer between
			-2147483648 and 2147483647
RUN	run program	RUN	clears all variables except @% to Z%
SAVE	save program	SAVE "FRED"	save on cassette or disc
SGN	sign	X=SGN(Y)	returns sign of Y (ie +1,0, or -1)
SIN	sine	X=SIN(ANGLE)	ANGLE in radians
SOUND	sound	SOUND 1,-15,53,20	see note (8)
SPC	space	PRINT SPC(6);A	generates 6 spaces followed by A
SQR	square root	X=SQR(Y)	negative roots cause error
STEP	step	FOR X=1 TO 7 STEP 2	optional, defaults to 1
STOP	stop	STOP	may be restarted with GOTO...
STR$	string	A$=STR$(X)	converts number to string representation
STRING$	repeat string	A$=STRING$(4,B$)	A$ becomes 4 times B$
TAB	tabulate	PRINT TAB(10);"HELLO"	Tabs to 10th column of correct line
		PRINT TAB(2,10);"HELLO"	moves text cursor to 2,10 & prints HELLO
TAN	tangent	X=TAN(ANGLE)	ANGLE in radians
THEN	IF...THEN...ELSE	IF A=2 THEN 350	OPTIONAL
TIME	time	TIME=((H.*60+MIN)*60+SE)*100	time in 10ms intervals
			may be read or set
TO	FOR...TO...NEXT	FOR X= 1 TO 5	
TOP	top pf program	PRINT TOP	pointer to end of user program
TRACE		TRACE ON	print line numbers during prog execution
		TRACE OFF	turn off trace
		TRACE 6780	report line numbers below 6780
TRUE	logical true	IF RESULT=TRUE	logical true =-1
UNTIL		UNTIL X>10	terminates REPEAT LOOP
USR	user	X=USR(&3000)	calls subroutine and returns value from
			P,Y,X,A registers
VAL	value	A=VAL(B$)	converts number represented by B$ to number
VDU	vdu driver	VDU 28,0,5,39,0	multipurpose statement see note(9)
VPOS	vertical position	V=VPOS	returns vertical position of text cursor
WIDTH	page width	WIDTH 60	sets page width used by computer (0 initially)

FX and OSBYTE Call Summary

	dec	hex	X	Y	Function
*	0	0			Prints operating system version number
	1	1			Reserved for applications programs
	2	2	0		get characters from keyboard and enable RS$ receiver
	2	2	1		get characters from RS423 port
	3	3			Select output devices
	3	3	0		Selects printer and screen
	3	3	1		Selects printer, screen and RS423
	3	3	2		Selects printer
	3	3	3		Selects printer and RS423
	3	3	4		Selects screen
	3	3	5		Selects screen and RS423
	3	3	6		Selects no output
	3	3	7		Selects RS423
*	4	4	0		Cursor editing keys and copy key perform normally
*	4	4	1		Cursor editing keys and copy key return ASCII codes
*	5	5	0		Selects printer sink
*	5	5	1		Selects parallel printer connection
*	5	5	2		Selects serial RS423 printer connection
*	5	5	3		Selects user supplied printer driver
*	6	6			Selects the printer ignore character, e.g.
*	6	6	8		Printer ignores ASCII 8 character
*	7	7	1		RS423 input set to 75 baud
*	7	7	2		RS423 input set to 150 baud
*	7	7	3		RS423 input set to 300 baud
*	7	7	4		RS423 input set to 1200 baud
*	7	7	5		RS423 input set to 2400 baud
*	7	7	6		RS423 input set to 4800 baud
*	7	7	7		RS423 input set to 9600 baud
*	7	7	8		RS423 input set to 19200 baud (This rate not guaranteed)
*	8	8	1		RS423 output set to 75 baud
*	8	8	2		RS423 output set to 150 baud
*	8	8	3		RS423 output set to 300 baud
*	8	8	4		RS423 output set to 1200 baud
*	8	8	5		RS423 output set to 2400 baud
*	8	8	6		RS423 output set to 4800 baud
*	8	8	7		RS423 output set to 9600 baud
*	8	8	8		RS423 output set to 19200 baud (This rate not guaranteed)
*	9	9	n		Flash rate, duration of first colour set to n/50 of a second
*	10	A	n		Flash rate, duration of first colour set to n/50 of a second
*	11	B	n		Repeat key delay set to n/100 of a second
*	11	B	0		Turn off repeat key
*	12	C	0		Reset repeat delay and repeat rate to default values
*	12	C	n		Set repeat rate to n/100 of a second
	13	D	0		Disable output buffer empty event
	13	D	1		Disable input buffer full event
	13	D	2		Disable character entering input buffer event
	13	D	3		Disable ADC conversion complete event
	13	D	4		Disable start of vertical sync. event
	13	D	5		Disable intervval timer crossing zero evvent
	13	D	6		Disable ESCAPE pressed event
	14	E	0		Enable output buffer empty event
	14	E	1		Enable input buffer full event
	14	E	2		Enable Character entering input buffer event
	14	E	3		Enable ADC conversion complete event
	14	E	4		Enable start of vertical sync. event
	14	E	5		Enable interval timer crossing zero event
	14	E	6		Enable ESCAPE pressed event

*	15	F	0		Flushes all internal buffers
*	15	F	1		Flushes the currently selected input buffer
*	16	10	0		Disables all ADC channels
*	16	10	1		Enables ADC channel 1 only
*	16	10	2		Enables ADC channels 1 and 2
*	16	10	3		Enables ADC channels 1,2 and 3
*	16	10	4		Enables all four ADC channels
	17	11	n		Forces start of convert on ADC channel n
	18	12			Resets user defined keys to no longer contain strings
	19	13			Forces machine to wait for next display frame
	20	14	0		Implodes soft character definition
	20	14	1		Explodes soft character definition
	21	15	0		Flushes the keyboard buffer
	21	15	1		Flushes the RS423 serial input buffer
	21	15	2		Flushes the RS423 serial output buffer
	21	15	3		Flushes the printer output buffer
	21	15	4		Flushes SOUND channel 0
	21	15	5		Flushes SOUND channel 1
	21	15	6		Flushes SOUND channel 2
	21	15	7		Flushes SOUND channel 3
	21	15	8		Flushes speech synthesis buffer
*	124	7C			Reset ESCAPE pressed flag
*	125	7D			Sets the ESCAPE pressed flag
*	126	7E			Acknowledge detection of ESCAPE condition

OSBYTE Calls

*	127	7F	n		Returns end of file status of file opened as n, X<>0 if EOF
*	128	80	0		Y returns last ADC channel to complete
*	128	80	n		For n=1 to 4, reads ADC channel n, MSB returned in Y, LSB in X
*	128	80	n	FF	For n=247 to 255, status of buffers returned in X
			247		speech buffer
			248		SOUND channel 3
			249		SOUND channel 2
			250		SOUND channel 1
			251		SOUND channel 0
			252		printer output buffer
			253		RS423 serial output buffer
			254		RS423 serial input buffer
			255		keyboard buffer
*	129	81	n	m	Read key with time limit, limit passed in X and Y(MSB) X returns key, Y=0 if key detected, 1B if ESCAPE, FF if no key If Y>&7F on entry then tests for specific key closure
*	130	82			Read machine higher order address
*	131	83			Read OSHWM, returned in X and Y(MSB)
*	132	84			Read bottem of display RAM address, returned in X and Y(MSB)
*	133	85	n		Read lowest address for mode n, returned in X and Y(MSB)
*	134	86			Return x and y co-ord. of cursor position in X and Y
	135	87			Read character at text cursor position, X returns char. Y returns current display mode
	136	88			Reserved
**	137	89	0		Turn cassette motor off
**	137	89	1		Turn cassette motor on
	138	8A	0	n	Insert character ASCII n into keyboard buffer
	139	8B	n	m	Controls file options, see page 398
*	140	8C	0		Sets default tape speed
*	140	8C	1		Sets tape speed to 300 baud
*	140	8D	2		Sets tape speed to 1200 baud
	141				Reserved
	142				Reserved
	143				Reserved

144	90	n	m	TV control, n sets vertical position, m=0 interlaced, m=1 non interlaced
145	91	0		Get character from keyboard buffer, Yreturns returns character
				C=1 if buffer empty, C=0 if character obtained
145	91	1		Get character from RS423 input buffer, Y returns character
				C=1 if buffer empty, C=0 if character obtained
146	92	X	Y	Read from FRED, X contains offset
147	93	X	Y	Write to FRED, X contains offset
148	94	X	Y	Read from JIM, X contains offset
149	95	X	Y	Write to JIM, X contains offset, character in Y
150	96	X	Y	Read from SHEILA, X contains offset
151	97	X	Y	Write to SHEILA, X contains offset, character in Y
224	E0	0	0	Cancel VDU queue
225	E1	0		Disable function keys (f0 to f9)
225	E1	1		Set function keys to generate strings
225	E1	n		Set base number of function keys to n
226	E2	n		Set base number of SHIFT function key codes
227	E3	n		Set base number of CTRL function key codes
228	E4	n		Set base number of SHIFT/CTRL function key codes
229	E5	0		Set escape key to interrupt a BASIC program
229	E5	1		Set escape key to return ASCII 1B(27)
230	E6	0		Enable flushing of buffers when escape key used
230	E6	1		Disable flushing of buffers when escape key used
231	E7	X		Set mask byte to X when servicing user 6522 IRQ
232	E8	X		Set mask byte to X when servicing 6850 ACIA IRQ

VDU Code Summary

Dec	Hex	CTRL	n	Meaning
0	0	@	0	does nothing
1	1	A	1	send next character to printer only
2	2	B	0	enable printer
3	3	C	0	disable printer
4	4	D	0	write text at text cursor
5	5	E	0	write text at graphics cursor
6	6	F	0	enable VDU drivers
7	7	G	0	make a short beep
8	8	H	0	backspace cursor one character
9	9	I	0	forwardspace cursor one character
10	A	J	0	move cursor down one line
11	B	K	0	move cursor up one line
12	C	L	0	clear text area (CLS)
13	D	M	0	move cursor to start of current line
14	E	N	0	page mode on
15	F	O	0	page mode off
16	10	P	0	clear graphics area (CLG)
17	11	Q	1	define text colour (COLOUR)
			1	colour number
18	12	R	2	define graphics colour
			1	action 0 plot specified colour
				1 OR specified colour with that already there
				2 AND specified colour with that already there
				3 EOR specified colour with that already there
				4 Invert the colour already there
			2	colour number
19	13	S	5	define logical colour
			1	logical colour number
			2	physical colour number
			3 to 5	zero

20	14	T	0 restore default logical colours
21	15	U	0 from keyboard delete current line in program - disable VDU drivers
22	16	V	1 select screen mode (MODE) - does not change HIMEM 1 MODE number
23	17	W	9 user define screen character 1 ascii character code 2 to 9 -contents of each row making up character
24	18	X	8 define graphics window 1 & 2 low and high byte of left x co-ordinate 3 & 4 low and high byte of bottem y co-ordinate 5 & 6 low and high byte of right x co-ordinate 7 & 8 low and high byte of top y co-ordinate
25	19	Y	5 PLOT K, x, y 1 plot action code 2 & 3 low and high byte of x co-ordinate 4 & 5 low and high byte of y co-ordinate
26	1A	Z	0 restore text and graphics default windows
27	1B	[	0 does nothing
28	1C	\	4 define text window 1 left x position (units are character positions) 2 bottem y position 3 right x position 4 top y position
29	1D	]	4 define graphics origin 1 & 2 low and high bytes of x co-ordinate 3 & 4 low and high bytes of y co-ordinate
30	1E	↑	0 home text cursor to top left of screen
31	1F	_	2 move text cursor to x,y position on screen 1 x position (character position units) 2 y position
127		7F	0 backspace and delete (equivalent to using delete key)

Plot Action Codes

0	move relative to last point
1	draw line in current graphics foreground colour
2	draw line relative in logical inverse colour
3	draw line relative in current graphics background colour
4	move to absolute position
5	draw line absolute in current graphics foreground colour
6	draw line absolute in logical inverse colour
7	draw line absolute in current graphics background colour
8-15	as 0-7 but last point omitted when inverting the colour
16-23	as 0-7 but with a dotted line
24-31	as 16-23 but without the last point plotted
32-63	reserved for extension
64-71	as 0-7 but only a single point plotted
72-79	reserved
80-87	as 0-7 but plot and fill triangle (using last two points visited)

Ian R. Hirst