

## CONTENTS

---

0. INTRODUCTION	1
I. GETTING STARTED WITH XBASIC	
1. First steps - Direct & Program mode	3
2. Numbers and Strings	4
3. Variables	5
4. Arrays	6
5. Expressions	6
II. THE SYSTEM EDITOR & SYSTEM COMMANDS	
1. Screen Control Codes	9
2. The Screen Editor	9
3. The Line Editor	10
4. System Commands:	10
MON, NEW, DEL, AUTO, LOAD, SAVE, VERIFY CLEAR, RUN, CHAIN, LIST, HOLD, MGE, RENUM	
5. Chaining and 'Semi-Chaining' Programs	15
III. COMMANDS, STATEMENTS AND FUNCTIONS	
1. Commands/Statements	16
2. Disc Handling Commands:	21
DIR, ERA, REN, LOCK, UNLOCK	
3. Standard Functions	22
4. Standard String Functions	25
5. User-Defined Functions	27
IV. INPUT/OUTPUT FACILITIES	
1. Devices and I/O assignment:	28
PRINT\$, INPUT\$	
2. Direct I/O Port access	30
3. Special Commands affecting I/O:	30
SEP, FMT, IOM, SPEED, NULL, WIDTH, ZONE	
V. XBASIC FILE MANAGEMENT SYSTEM	
1. General	34
2. File Naming Conventions	34
3. The File Descriptor	35
4. Sequential and Random Access Methods	36
5. File-Handling Commands:	37
DRIVE, OPEN, CREATE, CLOSE, APPEND, PRINT\$, INPUT\$, INCH\$	
6. File-Handling Examples	39
VI. ERROR MESSAGES	
1. List of Error Messages	45
2. Error handling within BASIC:	47
ON ERR GOTO/GOSUB, ON EOF GOTO/GOSUB, OFF ERR, OFF EOF, ERR, ERR\$, ERL	
3. Error Message Construction/Extension	48
VII. MACHINE-CODE LINKAGE	
1. MC-code related Commands/Functions:	50
CALL, POKE, PEEK, DOKE, DEEK, PTR, HEX\$	
2. Loading and Saving MC-code Files	52

<b>VIII. COMMAND/FUNCTION EXTENSION</b>	
1. Program storage	53
2. Reserved Word Construction	54
3. The Auxiliary Tables	55
4. Commands and Functions	55
5. How to enter extra Reserved Words	55

#### **APPENDICES**

---

Appx. A.	Index of Reserved Words and Error Messages	58
Appx. B.	The Hardware Configuration, including: MEMORY MAP, SCRATCH-PAD addresses I/O Device assignments, Graphics, incompatibilities with other versions, etc	61
Appx. C.	Useful Subroutines in XBASIC	65
Appx. D.	Examples of Extra Commands and Functions	76
Appx. E.	Translator for Nascom ROM and tape Basic	81

## 0. INTRODUCTION

Nascom Extended Basic (XBASIC) is an interpreter written in Z80 machine code which has been developed by Crystal Research. It is based on experience gained with earlier versions of Xtal BASIC and Nascom ROM BASIC. Extended BASIC is significantly larger than both the earlier Xtal BASIC and Nascom ROM BASIC, but includes many new features, and existing features have been extended.

For those with some experience of machine-code programming, the ability to create user-defined reserved words must be one of the most outstanding features of this BASIC. By writing appropriate sub-routines and by inserting your own defined words in an auxiliary reserved word table, you will be able to expand this interpreter to give the type of BASIC most suited to your own needs. We believe that, for the time being at least (and we have not heard of any equivalent in over two years), this feature is unique to BASIC's from Crystal, and it makes it potentially one of the most powerful BASIC's ever available.

Extended BASIC is designed to allow the incorporation of disc handling commands, as well as handling cassette tape, and the file handling system has been designed with a view to dealing with both. Although we use the terms 'disc' and 'cassette tape' throughout the manual, it is as well to remember that some forms of tape, such as the 'stringy floppy' or 'floppy tape' are theoretically capable of random-access, and may hold separate 'file directories', i.e, to all intents and purposes they behave as disc drives. We therefore include all such devices under the broad term 'disc drives', to distinguish from the sequential-only 'cassette tape' drives.

Nascom Extended BASIC is available in three forms - tape cassette, NAS-DOS and CP/M. The differences involve only the media and the provision of appropriate disc access commands.

### LOADING EXTENDED BASIC ON NASCOM MICROCOMPUTERS (TAPE VERSION)

---

XBASIC will run on any of the Nascom computers, as long as one of the NAS-SYS monitors is being used. It is supplied on tape in CUTS format, to load at 1200 baud.

To load XBASIC, type R and then press the <ENTER> key. Next, press the PLAY button on the cassette recorder. The program should be observed to load block by block until, after about two minutes, loading should be complete. XBASIC occupies the area 1000H to 40FFH (about 12 1/4 K).

To run, type in E1000 and press the <ENTER> key. This is the initialising, or 'COLD', entry to XBASIC. A 'WARM' entry is also allowed from the monitor into XBASIC by typing E1003 <ENTER>. This preserves any current BASIC program and variables. This entry point should not, however, be used unless XBASIC has already been previously entered by a COLD start.

## NOTATION

---

In order to simplify the use and understanding of this manual and, in particular, the command and function descriptions, we have adopted a notation that explains the syntax requirements of each command/function. This consists of a single letter, which may or may not be followed by a number, enclosed thus: < >. If a command/function name has to be followed by an expression, this notation will show the type of expression that is allowed:

- J An expression, which must evaluate to a number in the range 0 to 255. If it is not an integer, the decimal part of the number will be chopped off, the integer part only being used.
- I An expression, which must evaluate to a number in the range -65535 to +65535. Again, only the integer part is actually used. In some cases, the range is restricted to 0 to 65535, or even 0 to 32767 (e.g., Array elements), but mention is made only when those cases apply.
- L A line number, in the range 0 to 65535. This must be a number only, and so may not be given as a variable.
- N Any numeric expression.
- E Any expression, whether numeric or string.
- S Any string expression.
- F A string expression, which must evaluate to give a legal file name (as defined in Chapter V.2).
- U A numeric variable, which may not be an array element.
- V A variable name, which may be of numeric or string type, and may be an array element.
- SV A string variable name, which may not be a string array element.
- X A complete Xtal BASIC statement.

### Examples:

1. In Chapter III.4 we find the LEFT\$ function described thus: LEFT\$(<S>,<J>)

This means that LEFT\$ must have two arguments separated by a comma, and enclosed within parentheses. The first argument must be a legal string expression, and the second argument must be a number in the range 0 to 255 (reasonable, since we cannot have strings longer than 255 characters).

e.g. LEFT\$("NAME "+X\$,7) is legal.

2. In Chapter III.1, the ON..GOTO command is shown:

```
ON <J> GOTO <L1>,<L2>,...,<Ln>
```

This means that ON must be followed by a number in the range 0 to 255 followed by the word GOTO followed by one or more line numbers <L1> to <Ln>. Each of these line numbers (if more than one) must be separated by a comma.

e.g. ON X GOTO 1000,2000,3000 will simply drop to the next line if X is 0 or greater than 3, otherwise a GOTO will be executed to one of lines 1000, 2000 or 3000 according to the value of X being 1, 2 or 3 respectively.