SHARP mz-700 SERIES

BASIC EXPANDER

SOLO SOFTWARE

SEARCH and FIND	LVAR and LVAR$

IF…THEN…ELSE	WHILE..WEND

REPEAT…UNTIL	OLD

SOUND	QUIET	BEEP

AND	OR	NOT

A MUST FOR THE PROGRAMMER WHO NEEDS A MORE STRUCTURED LANGUAGE THAN STANDARD 'BASIC'

BASIC EXPANDER

SHARP mz-700

BASIC EXPANDER

A very handy addition to standard Sharp BASIC, the expander adds the FIND command allowing you to locate an item in a program and another useful function adds a LVAR and LVAR$ which gives a list of the variables currently in use. Added commands include IF…THEN…ELSE and WHILE…

WEND, AND OR NOT and REPEAT…UNTIL. The useful additions of TONE, QUIET and BEEP will also be appreciated by games programmers. In addition we thought it advisable to build in the OLD command which will reverse accidental NEW commands. The Basic Expander is a must for the home programmer who requires a more structured language than standard BASIC.

SOLO SOFTWARE LTD · 95B BLACKPOLE TRADING ESTATE WEST WORCESTER WR3 8TJ · TELEPHONE (0905) 58351

SHARP MZ-700 SERIES

SOLO SOFTWARE

SOLO BASIC EXPANDER MANUAL

This expander enhances the standard Sharp S-BASIC for the MZ-700 to give a superbly structured BASIC with many additional commands and functions.

LOADING

If this is the first time of loading your BASIC EXPANDER, the routine to follow is explained below:

1. Load S-BASIC as usual.
2. When loaded, rewind and remove.
3. Type BOOT followed by [CR].
4. Now type L and load the BASIC EXPANDER.
5. When this has loaded the message will be displayed TYPE 'R' TO ENTER BASIC. But before you do this, WE ADVISE YOU TO MAKE A COPY FIRST.
6. Type M004F and the display will show :004F 64
7. Overtype the '64' with FF and press [CR].
8. Now press SHIFT and BREAK together.
9. Type T0000 6BFE 2000 followed by [CR].
10. Now type
    S2000 8BFE 8BDF:BASIC+SOLO BX followed by [CR].
11. Take a blank cassette (minimum length C15) and place in deck.
12. Press RECORD and PLAY together. Recording will take around six minutes.

You now have a copy of S-BASIC and the SOLO BASIC EXPANDER on one cassette which you will load in future instead of the standard Sharp S-BASIC.

When the recording is complete, type G8BEE [CR] to re-enter BASIC.

You may now place the BASIC EXPANDER cassette in a safe place along with the Sharp S-BASIC cassette for use in the future in case your combined BASIC + EXPANDER tape ever wears out or is corrupted.

COMMANDS

IF ... THEN ... ELSE

A simple but useful addition to Sharp BASIC saving much time and memory space. Two opposing instructions can now be placed on the same line.

For example:

    10 FOR T=1 TO 20
    20 X=INT (2*RND(1))
    30 PRINT X; " IS WRITTEN AS ";
    30 IF X=0 THEN PRINT "NIL" ELSE PRINT "ONE"
    40 END

The only restriction to this command is that the IF, THEN and ELSE must always be contained within the same programme line.

REPEAT ... UNTIL

This is rather like a FOR ... NEXT loop but is conditional.

The loop starts with the word REPEAT and ends with UNTIL followed by the condition that must apply before the loop is exited.

```
10 A=10
20 REPEAT
30 A=A+5
40 PRINT "THE VALUE OF A IS";A
50 UNTIL A=50
60 PRINT "AND THE LOOP IS
       COMPLETED"
70 END
```

WHILE ... WEND

The difference between this and REPEAT ... UNTIL is that the condition is stated at the beginning of the loop and therefore the loop will be skipped entirely if the condition does not apply.

```
10 A=10
20 WHILE A>0
30 A=A-1
40 PRINT "THE VALUE OF A IS";A
50 WEND
60 PRINT "AND THE LOOP IS
       COMPLETED"
70 END
```

SOUND

This command which is always followed by a number, generates a variable pitch tone invaluable to games writers.

The number following the command must be within the range 257 to 65535. The lower the number, the higher the note will be.

Let's use the sound of a man falling off a cliff as an example:

```
10 FOR N=1000 TO 5000
20 SOUND N
30 NEXT
```

Now type RUN [CR] and listen.

QUIET

You will find that your computer is still playing the final sound and it may be getting on your nerves so add this line and RUN it again.

```
40 QUIET
```

Ahhh — peace, perfect peace.

BEEP

This is a simpler way of typing USR(62). The length of the BEEP will depend on the TEMPO setting (between 0 for short and 7 for long).

AND, OR, NOT

The AND and the OR statements in Sharp BASIC are tokenised (* means AND and the + symbol means OR).

When using * and + the arguments must be bracketed as shown below.

IF (A>7) * (B=5) THEN GOTO 1000

With the AND and OR the brackets are not required.

NOT is used as a test of a statement being true or false. If false, the NOT gives a value of -1. If true, the value is 0.

For example:

```
10 A=10
20 FOR T = 1 TO 10
30 A=A-1
40 X=NOT A=5
50 PRINT A; " = 5 IS";
60 IF X=0 THEN PRINT " TRUE"
   ELSE PRINT " FALSE"
70 NEXT
80 END
```

FUNCTIONS

LVAR

If you are ever writing a long programme and you want to use a variable, it is essential that you avoid using one that is already employed. This simple command quickly gives a list of all variables in use, including string variables and dimensioned arrays with their sizes.

OLD

Ever typed NEW by accident and then suddenly realised that you forgot to save first? If you have, then you will realise the benefit in this command that restores the program in its entirety.

FIND

This is quite simply the most useful function for any programmer. The number of times we have all spent peering blearily at the listing flashing past our eye, seeking that illusive line we want to amend . . . well no more.

The FIND works in three separate ways.

1. FIND n , text   This finds commands, variables, etc.
2. FIND n " text   This finds words within PRINT statements.
3. FIND n GOTO n   This finds GOTO's (even if they are omitted).

In the above three examples, the n after FIND stands for the line number from which the search is to commence (and in the third example, the GOTO line number also).

The word 'text' is the item being searched for.

We suggest you load in a simple programme and try it out. It's great.

If the item for which you are searching is not in the listing the message NOT FOUND will be displayed.

Just a couple of warnings about FIND.

Occasionally, the line found will not contain what you were looking for. This is because many of the Sharp 'keywords' like PRINT are tokenised and have the same numeric value as some lowercase and graphic characters. Simply continue the FIND routine using a slightly higher start number after FIND than the line number which was wrongly displayed.

If the item searched for is not typed in exactly as per the original entry it will not be found.

ABBREVIATIONS

REPEAT	REP.
UNTIL	UN.
WHILE	WH.
WEND	WE.
ELSE	EL.
SOUND	SO.
QUIET	Q.
BEEP	BE.
LVAR	LV.
FIND	FI.