

BROM

Tony Self

Clares Micro Supplies, already well known for their Replica II and Beta-base disc software, have now branched out into the area of ROM software. BROM is on an 8k ROM adding 21 ' commands to the Beeb's repertoire.

The product comes nicely packaged in Clares usual black and gold livery, with a 36 page ring bound manual. The manual is well laid out and entirely up to the standard we have come to expect from the producers of quality ROM software. The ROM is installed in the usual manner in one of your spare ROM sockets. Clares recommend a low priority socket as there should be no conflict of command names, because all of BROM's commands can be prefixed with a "Z". Also BROM includes a command that can turn off any other HOMs which still conflict.

EDIT (<lin>) This is the main feature Clares' have been pushing in their advertising campaign – a full screen program editor – the first to appear in a ROM utility! This claim I feel is not fully justified as Beebugsoft's Toolkit provides slightly better facilities although it* does not have the sophisticated "Wordwise" type scrolling of BROM. However, that apart, it is a very nice editor, allowing you to enter the program at any line number by using the optional parameter. With no parameter the editor is entered at the first line of the program. Once in the editor you easily move around the lines using the cursor keys on their own or in conjunction with the shift or control keys.

A status line at the top of the screen shows whether you are in overwrite or insert mode, this is toggled using the copy key. It is therefore, very easy to add extra* commands to a line or overwrite errors. Also on the status line is a "bytes free" indicator. The author of the program has obviously got the idea from the "characters free" indicator on Wordwise, however I am more interested in the length of the program rather than how many bytes I have got left, especially as the indicator can not possibly take account of variable space required by the program or the settings of PAGE or HIMEM. I did have two more gripes on the

editor. Firstly you were not able to enter extra program lines without leaving the editor and secondly you could not delete forward. However, having mentioned this to Clares, I have now been informed that these two facilities have been implemented.

ERR ON/OFF Like Toolkit, BROM also has the facility to trap errors and enter the program editor automatically at the line in question. This is a very useful utility* when trying to debug a program.

***LCOPY <lin> <lin> <lin>** This routine will copy program lines from one part of the program to another. It's essential to renumber the program after using this command. I fail to see that there should be much use for this command, as any decent programmer should not be repeating the same chunks of code throughout a program, he should be using procedures and functions.*

***LMOVE <lin> <lin> <colin>** This command has a bit more use than the previous one, as it physically moves program lines rather than just coping them. Again a renumber is necessary after using it.

FIND <str> (T) will search the program for any given string and display the program lines containing it. If the optional (T) parameter is added the routine will also be able to look for BASIC keywords.

***CHANGE <str> <st> (T)** enables you to search and replace any string. Again the (T) option allows you to change BASIC commands.

***SCHANG <str> <str> (T)** This is one of the nicest commands on the ROM, and as far as I know is not available on any others. It is a selective search and replace. A real boon when you are only changing strings of two or three characters.

***LIMIT (<lin>) (<lin>)** This command limits the range of line numbers used by the last five commands.

***LVAR (N) (I) (S) (A)** will list the current variables and their values dependant on which of the optional parameters has been used. i.e. numeric, integer, string & arrays.

***FNKEY (<key>)** Similarly, this command will list the contents of your user defined function keys. This aids re-defining the keys using the copy and cursor key.

***ONEKEY** On the surface this command seems a bit of a gimmick. It makes available a single key entry mode similar to the electron. By using the TAB key and pressing another at the same time, a BASIC keyword or command will be input. e.g. Tab A – AUTO, Tab 0 – OLD M. However, this command has far more potential, because it is possible to reprogram these keys yourself. By using *FX238 <addrlo> and *FX239 <addrhi> it is possible for 'ONE-KEY' to utilise your own look up table. Appendix A of the manual explains clearly how to do this and supplies a demo program and a key defining program. This means you have the potential to add a further 29 function keys to the existing ones. There is only one small drawback, your table must fit into one page of memory. i.e. 256 bytes.

CASE (U) (L) (M) This is an interesting idea, which you may find useful if you write programs where you want to restrict the input to either upper or lower case. Alternatively you may have some commercial software which only accepts one case, but it doesn't trap the shift key. Once you have entered *CASE L, for instance, all output to the screen will be in lower case no matter what the setting of the Caps lock/Shift lock. A very handy command for child proofing educational software.

***NORMKEY** returns the keyboard to normal after using either of the last two commands.

***FLUSH** will clear all the variables including the resident integers, except %.

***FLIST** produces a nicely formatted listing of a BASIC program. Each command is displayed on a separate line, which is indented in from the line number. Also BASIC keywords will not be split of the end of a line, being automatically wordwrapped. One other advantage of this command is that it will list programs that have been protected with hidden control codes and also "bad programs" if a LIMIT 0 command is issued first.

***BPCURE** This command will try and recover a "bad program". In all the instances I tried, it worked very quickly and effectively. When it can't interpret a character or it is

a control code a "-" is inserted, so some tidying up is required after using the command. In the same instances Toolkit's *RECOVER did not always work, but when it did it had recovered the program more intelligently. For instance, when I purposely changed an end of line marker (&OD) to (&FF), *BPCURE lost the line number and inserted ~'s, whereas Toolkit recovered the line intact.

***COMPARE <fsp> {<adr>}** will compare a program or section of memory with a file on disc or cassette, reporting a error if a byte doesn't match

***PACK (R) (S)** is used to remove all unwanted spaces and/ or REM statements. The routine advises how much space has been saved.

***PAGE <adr>** enables you to move a BASIC program to a different area of memory and reset the* value of PAGE accordingly. This is a very useful command for disc users, who need to relocate programs below &1900.

***ROMOFF/ROMON (name) (id)** This is one of the more useful routines not found on Toolkit or Caretaker. As the command name implies, it allows you to turn ROMs on and off. Once a ROM is turned off by using the command with either its name or its socket number, it will survive a Ctrl Break! Also included are two *FX commands which will turn all the ROMs off. *FX250,0 <RETURN> will turn off ROMs 15 to 8 and *FX251,0 <RETURN> will turn off ROMs 7 to 0. *FX250,255 and *FX251,255 will turn them back on. This command could be very useful if you have a second processor and you want to turn off those ROMs which react badly with it.

VERDICT

A nicely produced package of well written routines. I will leave it up to you to decide whether this selection meets your requirements better than Toolkit, Caretaker or one of the many other utility ROMs on the market. All I can say is you won't be disappointed by the quality, however at £34.50 better value is available.

Clares Micro Supplies can be contacted on 0606 48511.