

The Key to the IBM



The IBM's BASICA package has a number of features to make life easier on the lowly programmer. Here's a utility to make one of them work for you.

by Steve Rimmer

Microsoft disk BASIC has evolved through a number of strata of development and it's gotten pretty well refined over the years. The CP/M based implementation would seem to have all of the bells and whistles one could want in such a thing. However, the real programming revelation is in BASICA, the version which comes with MS DOS for the IBM PC.

BASICA has more doo dahs in it that can be counted on both hands of an Arc-turian Megatarsal. It's powerful beyond all measure and description, but, more to the point, it has features which make programming in it more convenient than BASIC ever thought it could be.

One of the really fine things happening in BASICA is the ability to program the function keys of the PC. These ten keys can each be made to emit any string you fancy with a simple command. Thereafter, you can, for example, hit key one to RUN, key two to LIST and so on.

The only drawback to this is that you have to load the keys each time you boot BASIC if you don't fancy the default definitions... which aren't all that useful for most general programming.

The KEYWORD program shown here should largely nullify this hassle, however.

When run, it loads a set of your own personal definitions in from a disk file, zaps them into BASICA's function key string buffer and then cheerfully kills itself to make way for your next program. At the same time, it has facility to allow you to edit the current set of definitions and replace the existing disk set with a new one.

Multi-functional

The program is fairly simple. All articles with programs in them say this to play with your head and lull you into a false sense of security, but it's actually true in this case. KEYWORD consists of a simple editor and some disk routines. It can be entered and run in minutes.

The only really vital aspect of it is that it erases itself as soon as it has run, so do be sure to SAVE it *prior* to the inaugural flight.

When you run KEYWORD it gets the contents of the file FUNCTION.KEY from the disk. If there is no file by this name there it creates one and assumes that all the definitions are blank.

In the main editing mode of KEYWORD one selects a key to define or alter by hitting the appropriate function. Unfortunately, these things return the strings of their definitions even if there is a program running.

This creates an interesting problem as, for example, if key one holds the string "LIST" and key two holds "LOAD" both will return the character "L" when hit.

To get around this, the program redefines the keys so that they hold the numbers from zero to nine.

If you hit a key to edit the cursor will leap up to the corresponding line on the screen. The system will then look for input. A line is terminated by the ESC key rather than a carriage return as, in many cases, one will want a carriage return imbedded in the string programmed into the key. If you hit a carriage return the editor will treat it as any other printable character... it uses character one hundred and twenty eight, a small "C" with the tail of a newt dangling below it... to represent carriage returns.

Keyed Up

The KEYWORD program can make developing software a lot quicker. It obviates a lot of typing and other unpleasant manual stuff and makes the function keys work for you.

There are some standard definitions you might want to adopt. For example, keys one and two are usually set to RUN and

The Key to the IBM

```

0  ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, 460 RETURN
20 , , , , , , , , , , , , , , , , , , , 470 REM input a line
30 , KEYWORD FOR THE IBM PC , 480 PRINT SPACE$(M);
40 , Copyright (c) 1984 Steve Rimmer , 490 LOCATE (P-1+(F/2)),AX,1
50 , , , , , , , , , , , , , , , , , , , 500 A$=""
60 ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, 510 WHILE C$(<)ESC$
70 , 520 C$=INPUT$(1)
80 W=20 : FLAG=0 530 IF C$=DEL$ AND LEN(A$)>0 THEN
90 PASS$="ABCDEFGHIJKLMNPOQRSTUVWXYZ 1234567 A$=LEFT$(A$,LEN(A$)-1) : PRINT RUB$:
890!@#%&'()*-+;:?.>.,' + CHR$(34) 540 IF INSTR(PASS$,C$) <> 0 AND LEN(A$)
100 A1=10 : A2=40 : P=10 : DIM F$(10) <W THEN A$=A$+C$ : PRINT C$:
110 FLE$="FUNCTION.KEY" 550 IF C$=CR$ THEN A$=A$+C$ : PRINT C$:
120 GOSUB 420 560 WEND
130 CR$=CHR$(13) : ESC$=CHR$(27) : CS$=CHR$(12B) : BS$=CHR$(29) 570 IF A$="" THEN 590
140 RUB$=BS$+" "+BS$ : DEL$=CHR$(8) 580 F$(F)=A$
150 CLS 590 RETURN
160 PRINT TAB(30) "FUNCTION KEY EDITOR" 600 REM load function keys with matrix
170 GOSUB 660 610 FOR X=1 TO 10
180 LOCATE P,1,0 620 IF INSTR(F$(X),CS$)<>0 THEN MID$(F$(X),
190 PRINT TAB(A1)"f1 "F$(1)TAB(A2)"f2 "F$(2) 630 KEY X,F$(X)
200 PRINT TAB(A1)"f3 "F$(3)TAB(A2)"f4 "F$(4) 640 NEXT X
210 PRINT TAB(A1)"f5 "F$(5)TAB(A2)"f6 "F$(6) 650 RETURN
220 PRINT TAB(A1)"f7 "F$(7)TAB(A2)"f8 "F$(8) 660 REM get functions from disk
230 PRINT TAB(A1)"f9 "F$(9)TAB(A2)"f10 "F$(10) 670 ON ERROR GOTO 840
240 LOCATE P+10,A1,1 680 OPEN "I",#1,FLE$
250 PRINT "Enter key to change, RETURN 690 INPUT #1,F$(0)
to quit. "; 700 IF F$(0)<>"KEYWORD" THEN CLOSE : GOTO 760
260 C$=INPUT$(1) 710 FOR X=1 TO 10
270 C=ASC(C$) 720 LINE INPUT #1,F$(X)
280 IF C=13 THEN 370 730 NEXT X
290 FLAG=1 740 CLOSE
300 IF C<48 OR C>57 THEN 260 750 RETURN
310 F=C-47 760 REM put functions on disk
320 IF INT(F/2)*2=F THEN AX=A2+3 ELSE AX=A1+3 770 OPEN "O",#1,FLE$
330 LOCATE (P-1+(F/2)),AX,1 780 F$(0)="KEYWORD"
340 GOSUB 470 790 FOR X=0 TO 10
350 LOCATE 1,1,0 800 PRINT #1,F$(X)
360 GOTO 180 810 NEXT X
370 REM get ready to get lost 820 CLOSE
380 IF FLAG = 1 THEN GOSUB 760 830 RETURN
390 GOSUB 600 840 REM file not found traoper
400 CLS : PRINT STRING$(24,10); 850 CLOSE
410 NEW 860 OPEN "O",#1,FLE$
420 REM redefine keys 870 PRINT #1,"KEYWORD"
430 FOR X=0 TO 9 880 CLOSE
440 KEY X+1,RIGHT$(STR$(X),LEN(STR$(X))-1) 890 RESUME 750
450 NEXT X

```

LIST. Three and four can be LOAD " and SAVE ". Nine and ten are KEY ON and KEY OFF in my system to control the menu on the bottom line of the screen. You can also define some useful defaults, like CLS or SCREEN 0, depending on the sort of pro-

gram you're working on.

I also have a key defined as RUN 'KEYWORD', but it's a good idea to have that one in there without a carriage return so you can hit ESCape if you stab the thing by mistake.

CNI