Key points

Roger Gann unlocks some of the secrets of your keyboard.

Boldly going where last month's column feared to tread, I'm now going to show you how you can pull some interesting stunts with your keyboard. Some of these tricks entail a modicum of programming but if you managed to customise your prompts using ESC commands and good old ANSI.SYS having read last month's column, then this exercise should pose no real problems for you.

I'll start by using ANSI.SYS to remap keys on your keyboard. Not all the keys are used under DOS and the technique I describe below enables you to assign really useful commands to what would otherwise be dead keys.

Customising your keyboard

Using ANSI.SYS' keyboard remapping facility it's easy to assign DOS commands to function keys. Note that you have to add a /X switch to the ANSI.SYS line in CONFIG.SYS in order to activate this feature:

```
DEVICE=C:\DOS\ANSI.SYS /X
```

and don't forget to reboot to make this change effective.

Right, having done that, here's how to map your F10 key to display a directory listing sorted by name and in wide format. Add this line to AUTOEXEC.BAT:

```
ECHO ←[0;68;"DIR /O:N /W↙
 /P";13p
```

(↙ Code string continues)

The general sequence for keyboard reassignment is

```
ESC[CODE;STRING; p
```

— CODE is the keycode for a particular key and STRING is either the ASCII code for a single character or a literal string contained in quotation marks.

Here's what my little remapping instructions mean. The left-pointing arrow in the command is the Escape character. The '0;68 is the keycode for F10 — you have to look them up in the DOS help — and the DIR command to be attached to F10 is contained within the quotes.

Note also the addition of a '13' before the final, lowercase p. This is the ASCII equivalent of a <CR> and saves you the bother of having to hit the Return key after the F10 key.

Don't know how to input that crazy ESC character on-screen? Well those with short memories won't remember that I gave the low-down on this mysterious trick a month ago. Using MS-DOS EDIT, you can generate the elusive ESC character by pressing CTRL-P and then hitting the ESC key [Fig 1]. On the screen you'll see it represented as a left-pointing arrow.

Die-hard EDLIN users have to employ a different trick. In EDLIN, you press CTRL-V and then the left bracket [. However, this is potentially confusing because the good old left bracket already features heavily in ANSI escape codes and it can look like you've got too many brackets in your command string.

It's easy to assign DOS commands to function keys

I do not think it is actually possible to generate an ESC character from the command prompt and it can only be created in text files. For this reason I do not believe you can reassign keys on-the-fly using, for instance, the TYPE command. As we saw in last month's column it is, however, possible to use the PROMPT command as a workaround, which will enable you to summon the ESC character from the command line using one of its metacharacters; $E. So try this at the DOS prompt:

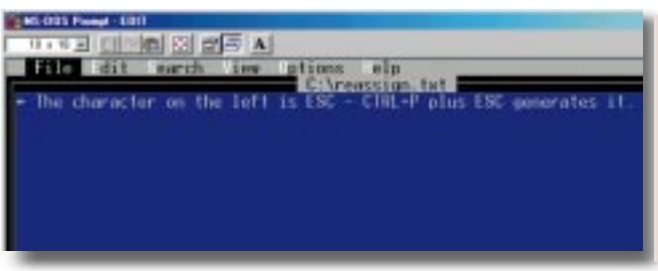
```
PROMPT $E[0;59;"DIR /O /P$_"p
```

Note that the 'p' at the end of the line must be lowercase. This will, incidentally, destroy your existing prompt (i.e. C:\>) so restore the normal command prompt by typing PROMPT PG.

You will now have a simple, sorted directory every time you hit the F1 key. Check out ANSI.SYS' online help for other key scan codes. Also, be careful not to use function keys that DOS uses such as F3, F8 or F9. If you've turned off command echoing with ECHO OFF, you must turn it back on by placing the command ECHO ON before these PROMPT statements in AUTOEXEC.BAT, otherwise they won't work.

One of the problems with ANSI.SYS is that once key reassignments are made, they are difficult to remove. Suppose, for example, that you have assigned the double quote to the single quote, eliminating the need to use the Shift key. If you later want to restore the single quote key to its original meaning, your only recourse is to make an additional assignment, defining the single quote back to itself.

Since ANSI.SYS does not remove the old reassignment from the buffer, you have that much less room for new key definitions. Note also that when



◀FIG 1 GETTING THE ESC CHARACTER INTO A BATCH FILE IS LESS EASY THAN IT MIGHT AT FIRST APPEAR. YOU HAVE TO HIT CTRL+P AND THEN THE ESC KEY

you load a major DOS application it will most likely initialise the keyboard back to its default state, meaning that when you quit that app, it will probably have zapped your clever key reassignments.

Keyboard acceleration

While we're on the subject of keyboards, did you know that you can accelerate your keyboard? Ever since MS-DOS 4.0, the MODE command has been able to reprogram most keyboards [Fig 2] to increase their typematic rate (the rate at which a character repeats when a key is held down).

MODE also lets you control the keyboard's delay interval, the length of time between when a key is pressed and typematic repeat begins. This command:

```
MODE CON: RATE=32 DELAY=2
```

maximises the typematic rate while leaving the delay set to its 0.5-sec default.

The speed freaks amongst you might be more interested in

```
MODE CON: RATE=32 DELAY=1
```

which maximises the typematic rate and minimises the delay interval, producing the fastest possible typematic operation. Experiment with different keyboard settings until you find one you like, then add it to your AUTOEXEC.BAT file.

Controlling NumLock

Ever since the BIOS of the IBM PS/2 defaulted to Num Lock=On, many users have wanted a way to disable Num Lock at bootup. While most modern BIOS setup screens let you specify Num Lock's default condition at bootup, many older ones do not. Microsoft solved the problem in DOS 6.0. If you're running it, you can disable Num Lock by adding the following line to your CONFIG.SYS file:

```
NUMLOCK=OFF
```

If you haven't installed DOS 6.n, here's how to create a tiny COM file to switch NumLock off.

First, fire up EDIT and enter these commands:

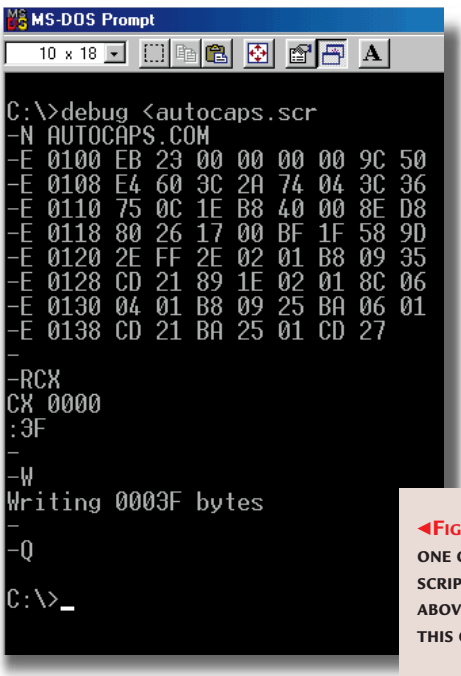
```
N NUMOFF.COM
E 0100 B8 40 00 8E D8 80 26 17
E 0108 00 DF C3

RCX
0B

W

Q
```

Note the three extra returns between the last four lines. Save it as NUMLOCK.SCR. Then type this at the DOS prompt:

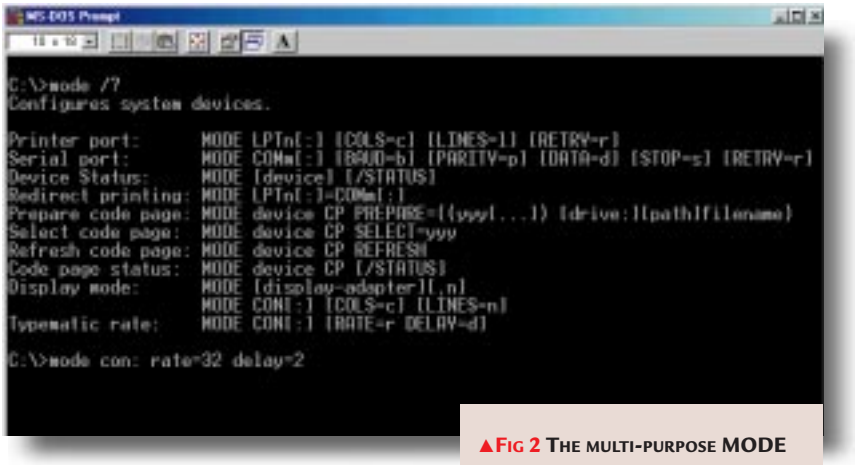
```
DEBUG <NUMLOCK.SCR <CR>
```

What we have done is to create a little script file which we are redirecting into DEBUG. There is nothing to stop you simply loading DEBUG and manually typing each instruction, but simply redirecting a text file into DEBUG does exactly the same thing and is more convenient, particularly when you have more than a few lines to input.

This script file will automatically create the minuscule NUMOFF.COM, which you should find in the same directory as DEBUG. Add a line, which calls NUMOFF to your AUTOEXEC.BAT startup file so that it executes each time you boot up, and that pesky NUMLOCK LED is quenched.



◄FIG 3 WHEN YOU RUN ONE OF THE DEBUG SCRIPTS I HAVE LISTED ABOVE, YOU SHOULD SEE THIS ON THE SCREEN



▲FIG 2 THE MULTI-PURPOSE MODE COMMAND ALLOWS YOU TO CONFIGURE PLENTY OF OTHER SETTINGS, TOO. ENTER MODE/? TO GET BRIEF HELP

Toggle CapsLock

You can perform a similar trick with the CapsLock key. Unlike a typewriter, pressing Shift while CapsLock is on does not turn it off. AUTOCAPS.COM [Fig 3] lets your keyboard work just like a typewriter's.

☛ This is the script file:

```
N AUTOCAPS.COM
E 0100 EB 23 00 00 00 00 9C 50
E 0108 E4 60 3C 2A 74 04 3C 36
E 0110 75 0C 1E B8 40 00 8E D8
E 0118 80 26 17 00 BF 1F 58 9D
E 0120 2E FF 2E 02 01 B8 09 35
E 0128 CD 21 89 1E 02 01 8C 06
E 0130 04 01 B8 09 25 BA 06 01
E 0138 CD 21 BA 25 01 CD 27

RCX
3F

W

Q
```

Once again, you will notice the absence of an extra carriage return between the RCX and 3F. Create AUTOCAPS.COM by typing the following line:

```
DEBUG <AUTOCAPS.SCR <CR>
```

☛ **Next month,** I will have more Prompt fun and games for you.

PCW CONTACTS

Roger Gann can be contacted via the PCW editorial office (address, p14) or email 16bit@pcw.co.uk