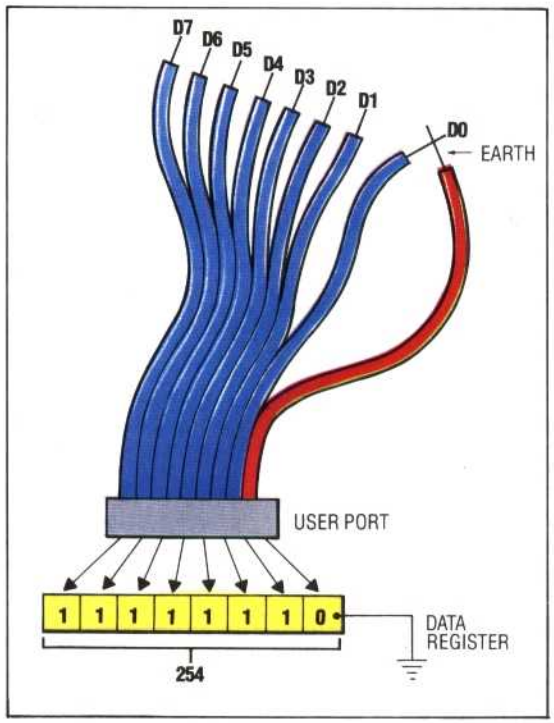
RUNning the program shows the normal contents of the data register to be 255 (shown on screen as HHHHHHHH), which means that all eight lines are high, and all eight cells of DATREG are at voltage level +5. If you now plug a lead into the user port, you can use it to change the voltage levels on the lines, and, therefore, change the numerical contents of DATREG.

We have wired 10 lines to the user port: eight of these are data lines — one line for each bit of DATREG — and the others provide a 'case ground' or 'earth' for the system. Touching a data line to an earth line will pull the data line's voltage level down to zero, thus changing the voltage levels in DATREG. If you earth D0, for example, while the program is running, you will see that DATREG now contains 254 (shown on the screen as HHHHHHHE), meaning that the least significant line is at *earth* voltage (0v), while all the others are *high* (+5v). You might like to try different combinations of lines to assure yourself that you can, by these means, make DATREG contain any number between zero and 255.



WRITING SOFTWARE

The essence of computer decision-making is the testing of numeric values or conditions and branching according to the result obtained in the test. Therefore, it is a simple matter to design software that can monitor physical activity via the user port. By testing the value of the data register and taking the appropriate action, the computer can respond to external changes. A simple example would be to design a program that checks to see if a telephone number has been 'dialled' correctly. Dialling can be done by grounding out certain data lines connected to the user port to produce the correct numeric value in the data register for each digit in the telephone number.

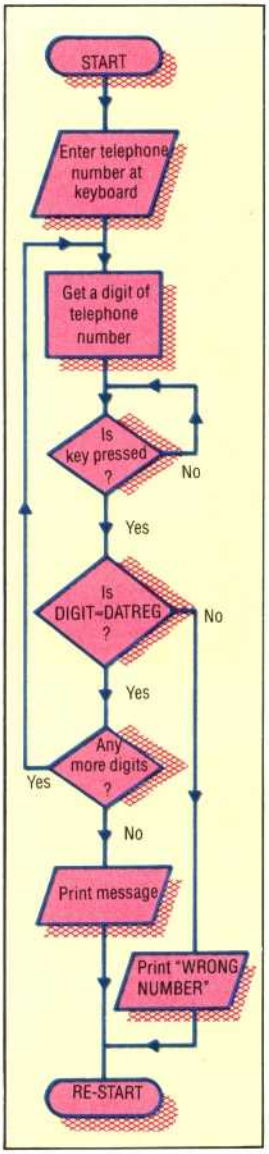
Because of the difficulties involved in simultaneously grounding a number of data lines, the program waits for the user to press a key on the keyboard before analysing the contents of the data register. The logic for the program is shown in the accompanying flowchart and is quite easy to follow.

```
100 REM**********C64**************
101 REM*   TELEPHONE NUMBERS      *
102 REM**********C64**************
120:
130 DATREG=56577:DDR=56579
140 POKE DDR,0:REM = INPUT ONLY
150 :
160 INPUT"TELEPHONE NUMBER";TN$
170 :
180 FOR K=1 TO LEN(TN$)
190 DG$=MID$(TN$,K,1): REM GET DIGIT
200 IF DG$<>" " THEN GOSUB 500
210 NEXT K
250 :
300 PRINT"-------SORRY-------"
310 PRINT" NO ONE IS IN AT ";TN$
320 PRINT" PLEASE CALL LATER"
350 PRINT:PRINT:RUN
400 :
500 REM**CONVERT & CHECK S/R**
550 DG=VAL(DG$)
600 PRINT"SET UP DIGIT ON THE LINES"
650 PRINT"    AND HIT ANY KEY"
700 GET GT$:IF GT$="" THEN 700
750 PE=PEEK(DATREG):IF PE=DG THEN
PRINT DG" OK":RETURN
800 :
850 PRINT"???WRONG NUMBER???"
900 PRINT TN$"<>"LEFT$(TN$,K-1):PE
950 PRINT"??? TRY AGAIN  ???"
999 PRINT:PRINT:RUN
```

On the BBC Micro, make these changes:

```
130 DATREG=&FE60:DDR=&FE62
140 ?DDR=0
700 A=GET
750 PE=?(DATREG):IF PE=DG THEN
PRINT DG" OK":RETURN
```

In this introductory article, we have looked at inputting information from an external source in order to affect the flow of control within programs. In the next instalment, we shall look at output via the user port and the design of a simple computer control system.



Crossed Lines
The Telephone Numbers program accepts as input a telephone number (any format, any length), and checks its digits one-by-one against the contents of the user port memory location. Spaces in the input number are ignored. The program waits for a keypress before comparing an input digit with the user port digit

Program Exercises

Using our telephone number program as an example:

1) Write a program to simulate the action of a burglar alarm.
2) Write a program to count the number of pulses received by the user port within a given time period.
3) Modify your last answer to keep a separate count for each of the eight user port data lines.
4) Write a program to simulate the action of a combination lock.
5) Write a program to change the colour of the computer's screen from the user port.