Computer Numerically Controlled Machine

CNC932

FEEDBACK INCORPORATED
620 SPRINGFIELD AVENUE
BERKELEY HEIGHTS, NEW JERSEY 07922
PHONE (201) 464-5181

FEEDBACK INSTRUMENTS LIMITED **CROWBOROUGH** **SUSSEX** **ENGLAND**
Tel: Crowborough (089 26) 3322 **Cables: Feedback Crowbr Telex** **Telex: 95255**

CONTENTS

GENERAL DESCRIPTION AND INSTALLATION — CHAPTER 1

Introduction

The CNC932 Computer Numerically Controlled Machine is one of the Feedback range of practical computer applications. It comprises a working model of a coordinate drilling machine equipped with electronically controlled drive mechanisms. These are intended to allow the machine to be numerically controlled by means of any ordinary microcomputer equipped with a Feedback MICA (or MICA-compatible) interface.

This manual is addressed to the instructor, rather than to the student. This is because each teaching situation will require its own specific preparation. The manual is intended to enable the instructor to operate the equipment and to understand it in the necessary depth, so that he can frame the explanations and exercises appropriate to his particular students.

The CNC932 consists of:

- A model of a CNC drilling machine
- A control unit with integral power supply
- A manual
- A small supply of workpieces for drilling exercises

Specification

X travel (table movement)	100mm
Y travel (table movement)	100mm
Z travel (drill feed)	25mm
Maximum no load drill spindle speed	16,000 rev/min
Maximum drill size	3.2mm
X, Y, Z drive	Stepping motors, 48 steps/revolution
Maximum rate of movement (any axis)	4mm/second
Minimum step size	0.021mm
Power requirements	100, 110, 120V (use 1A fuse) 200, 220, 240V (use 500mA fuse) 50VA 50 to 60 Hz

Milling capability

This machine is not intended for any but light milling, using small cutters on some plastics. An especially suitable material for milling is Toysteel, obtainable from Denford Machine Tools Limited, Birds Royd, Brighouse, HD6 1NM, West Yorkshire.

However it could in principle perform contour or three-dimensional milling if suitable computer programs were developed, subject only to limitations of step size, backlash and available force.

Installation

All power required for the machine is supplied from the control box, which in turn uses the main electricity supply.

Before connecting the CNC932 to the supply line check and if necessary adjust the voltage selector on the back panel (fig 2). At the same time check that the correct fuse is installed (see specification). To do this, press and turn the small knob (A) in the centre of the selector until it springs out with the fuse. The selector may then be turned by means of a coin or suitable tool engaged in the slots (B) which are then visible, until the pointer points to the required voltage. The fuse (C) may be pulled out of the holder knob A. Assemble by reversing the

PRINCIPLES OF OPERATION

CHAPTER 2

Axes

The machine, fig 1, is intended to have a workpiece mounted on the table. This will typically be a sheet of plastic or similar material to be drilled. The table can be moved in either sense along the two axes marked X and Y. The drill motor can be moved up and down (Z axis) and its rotation can be started or stopped. The X, Y and Z motions are all controlled by stepping motors; the drill rotation is by a simple d.c motor.

The stepping motor

(The reader who is fully conversant with stepping motors may proceed to the headed paragraph 'Controlling the stepping motor').

The essential parts of any stepping motor are a rotor of magnetic material, often magnetised, having usually a largish number of poles, and a stator having one or more electromagnets with (usually) a corresponding number of poles. The CNC932 stepping motor has a 24-pole ceramic magnet rotor and two corresponding electromagnets which are mutually displaced by half a pole pitch. (Other stepping motors are different in detail, but much the same principles apply). Fig 3 is a diagram showing the stator and rotor 'unwrapped' into straight lines. At any one time both the stator's electromagnets are energised. At (a) the rotor has aligned itself so that each south pole S on the rotor has aligned itself with the composite north pole N formed by the two electromagnets. Reversing the polarity of one of them (b) causes the pattern of stator magnetism to move half a pole pitch to the right, and the rotor follows it. Reversing the other continues the process.

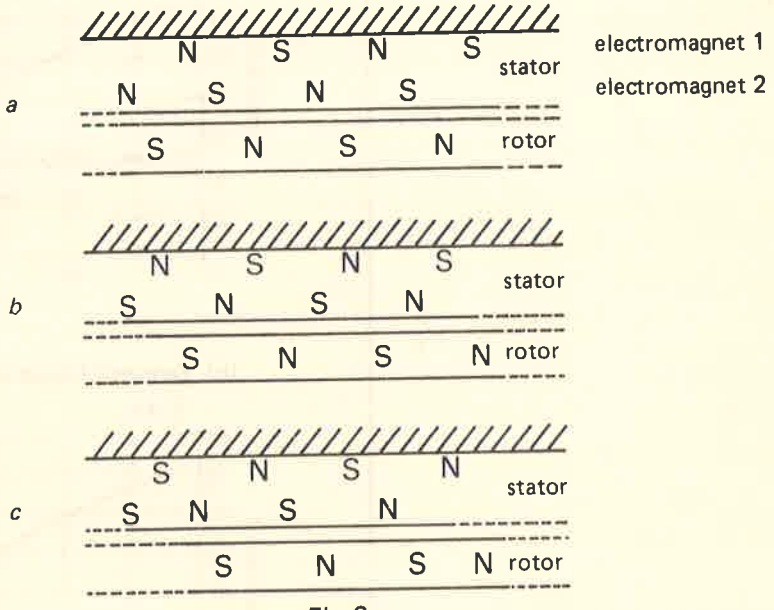

Fig 3

If the sequence of polarities applied to the two electromagnet windings is reversed, then the motion of the motor also will be reversed. The stepping motor can thus be made to rotate through any number of steps in either direction. This feature makes it useful for automatic control systems, especially those which control the position of something, since the stepping motor removes the need for an external position reference, subject to two conditions. The first condition is that the motor must have started from a known position. To meet this requirement some external reference may be needed, but it need only establish *one* position, not all possible positions; the photo detectors on the X and Y axes of the CNC932 do this.

stepping rate and with a load torque which define a point in the normal zone, the start will be successful. If the motor is started and then either the speed or the torque is increased to bring the operating point into the 'pre-stall' zone between the two graph curves, the motor will continue running, but it will not start from rest in the pre-stall zone. The stall zone denotes conditions which will stall the motor, i.e cause it to stop because of excessive load torque. Naturally if a mechanical load is applied to the motor its inertia will during any acceleration require a torque output from the motor, thus tending to limit the available acceleration. Fig 5 shows some of the particular characteristics which can be deduced from fig 4.

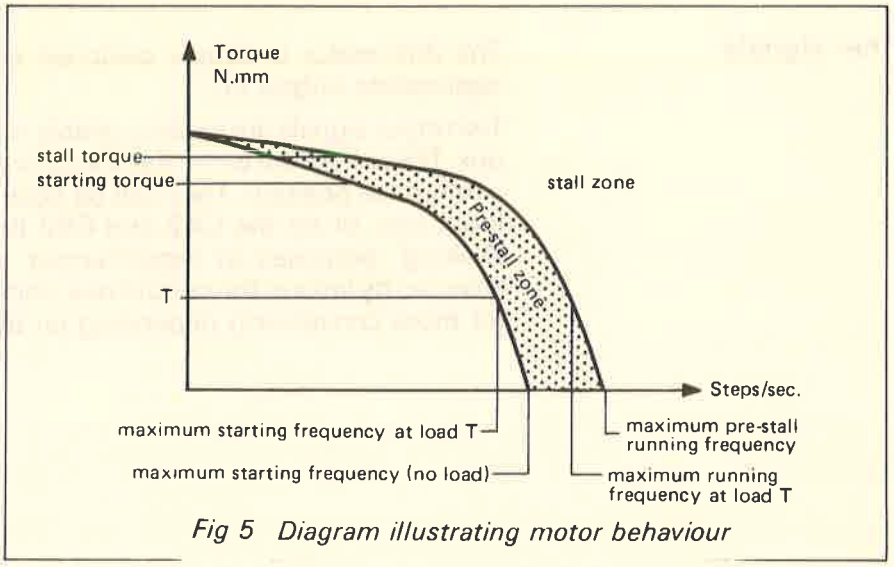

Fig 5 Diagram illustrating motor behaviour

Controlling the stepping motors

Each stepping motor has two pairs of windings, fig 6, which must be energised in the tabulated sequence for clockwise rotation (viewed from the spindle end). The sequence is taken in the reverse order for rotation in the opposite direction. Each step is of 7.5°, so that 48 steps are required to make one revolution. The motors drive leadscrews of 1mm pitch, so that 48 steps drive the appropriate linear motion through 1mm.

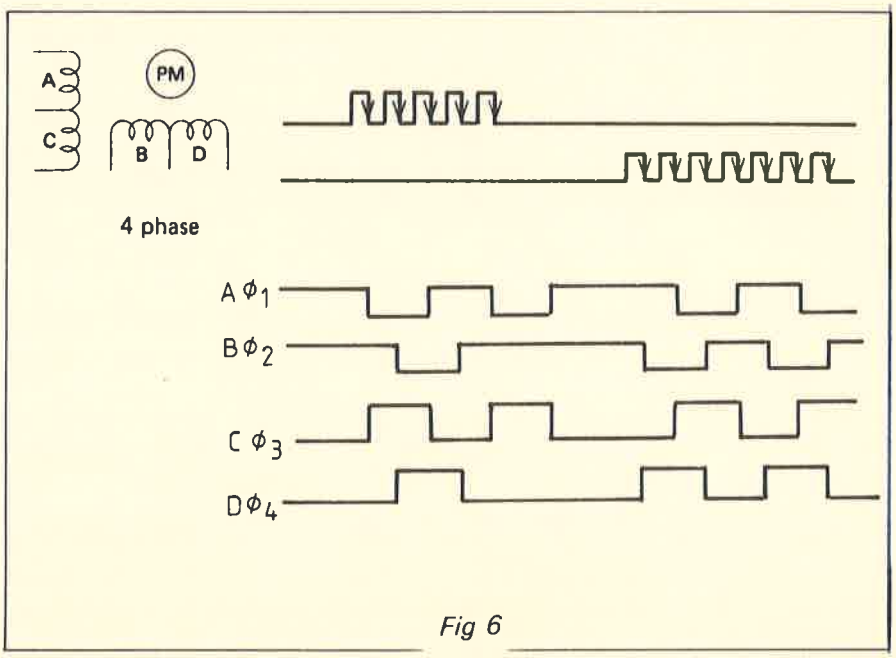

Fig 6

The experienced computer user will doubtless be able to skip much of this chapter. However the step-by-step approach to program building which follows may help to provide a thorough understanding of the equipment, and some hints as to the steps which may be taken (or omitted, as appropriate) in presenting ideas to students. It will also help in diagnosing faults if the following procedure is gone through.

Program development has been carried out on the BBC microcomputer, for which all the various program segments are listed and explained in detail. The final, comprehensive program in this chapter is listed both for the BBC microcomputer and in a tested version for APPLE-II. For other computers the required port initialisation is given below, and comments on their programming requirements more generally are given in Chapter 4.

Initialisation

The first essential program requirement is to make the computer capable of communicating with the control box. The input/output ports of any ordinary computer use devices which can perform many alternative functions. These devices must be instructed as to which particular functions they are required to perform before they can be used. The process is called initialisation.

The following listings show versions of a BASIC subroutine for initialising the ports of various microcomputers. For each complete program that follows, the initialising subroutine for the BBC computer will be shown, but the correct subroutine should be substituted if any other computer is in use.

Important note.

For the BBC, PET Commodore, and APPLE-II computers alternative port addresses are possible. The most probable address has been shown in these listings, but the interface manual should be consulted to check that this is the actual address in use. It is essential that anyone who concerns himself with initialisation is properly informed on this question, since the equipment will simply seem to be faulty if incorrect addressing is used.

```
 10 REM * PORT INITIALISATION FOR BBC MICRO *
100 REM * PROCEDURE INITIALISE PORTS
110 PA=64704: REM * Set Port A base address
120 AC=PA+1:  REM * Control Register A address
130 PB=PA+2:  REM * Port B address
140 BC=PA+3:  REM * Control Register B address
150 ? AC=0:   REM * Enable Data Direction Register A
160 ? BC=0:   REM * Enable Data Direction Register B
170 ? PA=15:  REM * PA, LS 4 bits o/p, MS 4 bits i/p
180 ? PB=255: REM * Port B is OUTPUT port
190 ? AC=4:   REM * Enable Port A
200 ? BC=4:   REM * Enable Port B
210 ? PA=0:   REM * Set Port A to zero
220 ? PB=0:   REM * Set Port B to zero
230 RETURN:   REM * END INITIALISE PORTS
```

Preparation for testing

Initially the machine should be left disconnected from the control box. The control box, interface and computer should be connected as indicated in Chapter 1.

Until initialisation has been completed, leave the control box of the CNC932 switched off.

Switch on the computer and load the appropriate initialisation sub-routine, as indicated previously, in a program which calls it and halts. The following listing applies to the BBC computer.

```
 10 REM * PROGRAM 1 FOR BBC MICRO *
 20 GOSUB 100: REM * Initialise Ports
 30 END
100 REM * PROCEDURE INITIALISE PORTS
110 PA=64704: REM * Set Port A base address
120 AC=PA+1:  REM * Control Register A address
130 PB=PA+2:  REM * Port B address
140 BC=PA+3:  REM * Control Register B address
150 ? AC=0:   REM * Enable Data Direction Register A
160 ? BC=0:   REM * Enable Data Direction Register B
170 ? PA=15:  REM * PA, LS 4 bits o/p, MS 4 bits i/p
180 ? PB=255: REM * Port B is OUTPUT port
190 ? AC=4:   REM * Enable Port A
200 ? BC=4:   REM * Enable Port B
210 ? PA=0:   REM * Set Port A to zero
220 ? PB=0:   REM * Set Port B to zero
230 RETURN:   REM * END INITIALISE PORTS
```

Run the program.

Sending commands to the control box

When the initialisation program has been run, see that the machine is disconnected from the CNC932 control box, then switch on the power to the control box.

This process, or possibly others such as using any listing or recording facilities available, may generate interference which upsets the initialisation. As a precaution now, or whenever communication between machine and computer seems to be lost, run the initialisation program again.

It should now be possible to send commands to the output ports, and the results will be displayed by the indicator lamps on the front panel of the control box. The commands are numbers which are the sum of the values of bits assigned to the required actions in the table which follows.

If proper responses are not obtained, check:

- power supply
- all connections
- accuracy of the initialisation program
- the address settings on the MICA interface (where appropriate)

Connect the machine to the control box. Verify that typing in the same command sequence as before causes the Y drive motor to step round.

The sequence of instructions can be incorporated into a program, for execution after the initialisation procedure. Program 2 would be a direct implementation of this. It is not suggested that this program be run, but it it were run, the instructions would be executed in far too rapid a sequence for the motor to follow; a delay is needed between instructions. This is provided in Program 3 by the subroutine Delay.

```
 10 REM * PROGRAM 2 FOR BBC MICRO *
 20 GOSUB 100: REM * Initialise ports
 30 GOSUB 200: REM * Step motor
 40 END
100 REM * Procedure INITIALISE PORTS
110 PA=64704
120 AC=PA+1: PB=PA+2: BC=PA+3
130 ? AC=0: ? BC=0: ? PA=15: ? PB=255
140 ? AC=4: ? BC=4: ? PA=0: ? PB=0
150 RETURN: REM * End INITIALISE PORTS
200 REM * Procedure STEP MOTOR
210 ? PA=1: REM * Enable Y
220 ? PB=1: REM * +Y LED on
230 ? PB=0: REM * +Y LED off, A-B LEDs on
240 ? PB=1: REM * +Y LED on
250 ? PB=0: REM * +Y LED off, B-C LEDs on
260 ? PB=1: REM * +Y LED on
270 ? PB=0: REM * +Y LED off, C-D LEDs on
280 ? PB=1: REM * +Y LED on
290 ? PB=0: REM * +Y LED off, D-A LEDs on
300 ? PB=1: REM * +Y LED on
310 ? PB=0: REM * +Y LED off, A-B LEDs on
320 ? PA=0: REM * All LEDs off
330 RETURN: REM * End STEP MOTOR
```

```
 10 REM * PROGRAM 3 FOR BBC MICRO *
 20 GOSUB 100: REM * Initialise Ports
 30 GOSUB 200: REM * Step motor
 40 END
100 REM * PROCEDURE INITIALISE PORTS
110 PA=64704
120 AC=PA+1: PB=PA+2: BC=PA+3
130 ? AC=0: ? BC=0: ? PA=15: ? PB=255
140 ? AC=4: ? BC=4: ? PA=0: ? PB=0
150 RETURN: REM * END INITIALISE PORTS
200 REM * PROCEDURE STEP MOTOR
210 ? PA=1: REM * Enable Y
220 ? PB=1: REM * +Y LED on
230 ? PB=0: REM * +Y LED off, A-B LEDs on
235 GOSUB 400: REM * Delay
240 ? PB=1: REM * +Y LED on
250 ? PB=0: REM * +Y LED off, B-C LEDs on
255 GOSUB 400: REM * Delay
260 ? PB=1: REM * +Y LED on
270 ? PB=0: REM * +Y LED off, C-D LEDs on
```

```
 50 PRINT: PRINT
 60 PRINT "   The X, Y & Z axes may be driven from"
 70 PRINT "the computer keyboard.": PRINT
 80 PRINT "   Pressing X, Y or Z will drive in the"
 90 PRINT "+ direction, pressing x, y or z will"
100 PRINT "drive in the - direction."
110 PRINT "   To start DRILL press D, to stop"
120 PRINT " DRILL press d.": PRINT
130 PRINT "   Drive is stepped in 0.166 mm"
140 PRINT "increments. Hold the key down to repeat."
150 PRINT "   To exit program press E or e."
170 REM * VARIABLES:
180 REM * PA, PB = Ports A & B.
190 REM * AC, BC = Control registers A & B.
200 REM * S = Number of steps.
210 REM * N$ = Function input.
220 REM * E = Drive enable.
230 REM * D = Drive output.
240 GOTO 390: REM * MAIN PROGRAM
250 REM * Procedure INITIALISE PORTS *
260 PA=&FCC0: REM * Base address
270 AC=PA+1: PB=PA+2: BC=PA+3
280 ? AC=0: ? BC=0: ? PA=15: ? PB=255
290 ? AC=4: ? BC=4: ? PA=0: ? PB=0
300 RETURN: REM * End INITIALISE PORTS *
310 REM * Procedure OUTPUT *
320 S=8
330 ? PA=E: REM * Enable drive
340 ? PB=D: ? PB=0: REM * Drive pulse
350 S=S-1: IF S>0 THEN GOTO 340
360 ? PA=0: REM * Disable drives
370 RETURN: REM * End OUTPUT
380 REM ** MAIN PROGRAM **
390 GOSUB 260: REM * INITIALISE
400 N$ = INKEY$(0)
410 IF N$ = "X" THEN GOTO 520
420 IF N$ = "x" THEN GOTO 560
430 IF N$ = "Y" THEN GOTO 600
440 IF N$ = "y" THEN GOTO 640
450 IF N$ = "Z" THEN GOTO 680
460 IF N$ = "z" THEN GOTO 720
470 IF N$ = "D" THEN GOTO 760
480 IF N$ = "d" THEN GOTO 790
490 IF N$ = "E" THEN GOTO 820
500 IF N$ = "e" THEN GOTO 820
510 GOTO 400
520 PRINT TAB(17,22) "+X       "
530 E=2: D=4
540 GOSUB 320
550 GOTO 400
560 PRINT TAB(17,22) "-X       "
570 E=2: D=8
580 GOSUB 320
590 GOTO 400
600 PRINT TAB(17,22) "+Y       "
```

If input is via port A, then the input is obtained by any BASIC instruction which contains the expression ? PA for the BBC computer, or PEEK(PA) for most others. (It is assumed here that PA has previously been set equal to the numeric address of port A, as in the initialisation routines given earlier).

Thus the execution of the instruction

```
L = ? PA
```

sets the variable labelled L equal to the sum of the values of those bits of port A which are set to 1 (or true). Usually only one bit value is wanted at a time. This can be selected (on most computers) by using the AND function, as in line 1050 of the following listing. On at least one computer, APPLE-II, it is only possible in BASIC to test individual bits of port A by means of an arithmetic test, for which Chapter 4 gives a suggestion.

```
150 PRINT "   To exit program press E or e.": PRINT
160 PRINT "   To find DATUM press R."
170 REM * VARIABLES:




500 IF N$ = "e" THEN GOTO 820
505 IF N$ = "R" THEN GOSUB 1010
510 GOTO 400




1000 REM * Procedure FIND DATUM
1010 PRINT TAB(17,22) "DATUM    "
1020 REM * X Datum
1030 ? PA=2:          REM * Enable X drive
1040 ? PB=8: ? PB=0: REM * Step -X
1050 L=? PA AND 32:  REM * Get detector output
1060 IF L=0 GOTO 1040: REM * If zero try again
1070 S=10:            REM * If not zero step back 10 steps
1080 ? PB=4: ? PB=0: REM * Step +X
1090 S=S-1: IF S>0 GOTO 1080: REM * 10 steps then find Y
1100 ? PA=1:          REM * Enable Y drive
1110 ? PB=2: ? PB=0: REM * Step -Y
1120 L=? PA AND 16:  REM * Get detector output
1130 IF L=0 GOTO 1110: REM * If zero try again
1140 S=10:            REM * If not zero step back 10 steps
1150 ? PB=1: ? PB=0: REM * Step +Y
1160 S=S-1: IF S>0 GOTO 1150: REM * 10 steps then Return
1170 ? PA=0: RETURN: REM * End FIND DATUM
```

Program 6 BBC continued

```
340 REM * M(N) = Miscellaneous function.
350 REM * MO$ = Mode.
360 REM * MOV = Drill vertical movement.
370 REM * PA = Port A.
380 REM * PB = Port B.
390 REM * PM = Program Mode.
400 REM * SN = Any count of steps within a subroutine.
410 REM * SO = Number of 1mm output steps.
420 REM * SP = Speed.
430 REM * ST = 48 steps for 1mm of travel.
440 REM * UP = Drill feedrate up.
450 REM * X(N) = X dimension (absolute).
460 REM * XA = X ordinate.
470 REM * XB = New X ordinate.
480 REM * XL = XY equality flag.
490 REM * XT = Number of steps from XA to XB.
500 REM * Y(N) = Y dimension (absolute).
510 REM * YA = Y ordinate.
520 REM * YB = New Y ordinate.
530 REM * YT = Number of steps from YA to YB.
540 REM * Z(N) = Z dimension (absolute).
550 REM * ZA = Z ordinate.
560 REM * ZB = New Z ordinate.
570 REM
580 REM ##############################
590 REM
600 REM * Procedure INITIALISE PORTS
610 PA=64704
620 AC=PA+1: PB=PA+2: BC=PA+3
630 ? AC=0: ? BC=0: ? PA=207: ? PB=255
640 ? AC=4: ? BC=4: ? PA=0: ? PB=0
660 RETURN: REM * End INITIALISE PORTS
670 REM ----------------------------
680 REM * Procedure INIT VARIABLES
690 DIM FR(40),G$(100),F$(100),M$(100),X$(100),Y$(100),Z$(100)
700 FR(1)=266:FR(2.5)=105:FR(5)=50:FR(7.5)=31:FR(10)=21
    :FR(15)=12:FR(20)=7
710 FR(25)=5:FR(30)=2:FR(35)=1:FR(40)=0: REM * Feedrate delays
720 AD=0: DC=0: ZA=0: DL1=0: DL2=0: PM=0
730 RETURN: REM * End INIT VARIABLES
740 REM ----------------------------
750 REM * Procedure OUTPUT
760 ? PA=EN:          REM * Enable drive
770 FOR SN=1 TO SO:   REM * Loop for 1mm steps
780 FOR ST=1 TO 48:   REM * 48 steps = 1mm
790 ? PB=DR: ? PB=0: REM * Step once
800 FOR DL=0 TO DL1: NEXT DL: REM * Delay
810 NEXT ST
820 NEXT SN
830 ? PA=0:           REM * Enable to zero
840 RETURN: REM * End OUTPUT
850 REM ----------------------------
860 REM * Procedure DATUM
870 REM * X Datum
880 ? PA=2:           REM * Enable X drive
890 ? PB=8: ? PB=0:   REM * Step -X
900 DX=? PA AND 32:   REM * Get detector output
910 IF DX=0 THEN GOTO 890: REM * If zero try again
```

Program 6 BBC continued

```
1540 FOR DL=0 TO UP1: NEXT DL: REM * Delay
1550 NEXT SN
1560 ? PA=0:            REM * Enable to zero, Drill off
1570 RETURN: REM * End DRILL CYCLE
1580 REM ----------------------------
1590 REM * Procedure DISPLAY1
1600 CLS: PRINT
1610 PRINT "       ********************************"
1620 PRINT "       * C.N.C. DRILLING MACHINE. *"
1630 PRINT "       ********************************": PRINT: PRINT
1640 RETURN: REM * End DISPLAY1
1650 REM ----------------------------
1660 REM * Procedure INPUT 1
1670 GOSUB 1600:        REM * Display 1
1680 PRINT "            SPECIFY DRILL CYCLE.": PRINT
1690 INPUT " MOVEMENT (up & down maximum 25) mm? " MOV: PRINT
1710 IF MOV)25 THEN GOTO 1670
1720 MOV1=MOV*48:       REM * 48 steps/mm
1730 PRINT " FEEDRATE the following feedrates may"
1740 PRINT "be entered, all are in mm/second:"
1750 PRINT "0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0,"
1760 PRINT "2.5, 3.0, 3.5, 4.0. Do not attempt to"
1770 PRINT "enter any other numbers.": PRINT
1780 INPUT "            FEEDRATE DOWN? " DO: PRINT
1790 INPUT "            FEEDRATE UP?   " UP: PRINT
1800 PRINT " Is above data correct (Y or N)? ";
1810 IF GET$="Y" THEN GOTO 1830
1820 GOTO 1670
1830 D=DO*10: DO1=FR(D): REM * Delay for down feed
1840 D=UP*10: UP1=FR(D): REM * Delay for up feed
1850 RETURN: REM * End INPUT 1
1860 REM ----------------------------
1870 REM * Procedure INPUT 2
1880 GOSUB 1600:        REM * Display 1
1890 PRINT "                 * MODE *": PRINT: PRINT
1900 PRINT "    Two modes are available, 'SINGLE"
1910 PRINT "BLOCK' or 'PROGRAM'."
1920 PRINT "    In the SINGLE BLOCK mode each"
1930 PRINT "block of instructions is acted upon"
1940 PRINT "when it is entered."
1950 PRINT "    In the PROGRAM mode a number of"
1960 PRINT "blocks of instructions may be entered,"
1970 PRINT "and run to perform a sequence of"
1980 PRINT "operations.": PRINT
1990 INPUT "Which mode do you require (S or P)? " MO$
2000 BL=1
2010 IF MO$="P" THEN GOTO 2030
2020 IF MO$()"S" THEN GOTO 1990
2030 RETURN: REM * End INPUT 2
2040 REM ----------------------------
2050 REM * Procedure INPUT 3
2060 GOSUB 1600:        REM * Display 1
2070 PRINT "            * SINGLE BLOCK ";BL " *": PRINT: PRINT
2080 BL=BL+1
2090 PRINT "   Enter FUNCTION and XY co-ordinates."
2100 PRINT "   FUNCTION may be:- DATUM"
2110 PRINT "                     DRILL"
```

Program 6 BBC continued

```
2690 Q=GET
2700 CLS: PRINT:PRINT
2710 PRINT "  The X and Y ordinates are initialised"
2720 PRINT "to zero (machine datum), all ordinates"
2730 PRINT "are specified with respect to the"
2740 PRINT "machine datum.": PRINT
2750 PRINT "  The Z ordinate is initialised to 25,"
2760 PRINT "that is fully up."
2770 PRINT "  The initial position of the drill"
2780 PRINT "(Z ordinate), must be set manually"
2790 PRINT "before running a program.": PRINT: PRINT
2800 PRINT "Type Y to continue.";
2810 Q=GET
2820 @%=5: N%=0: N=0
2830 CLS: PRINT TAB(5,1) "N","G","F","M","X","Y","Z": PRINT
2840 IF N)17 THEN N1=N%-17 ELSE N1=1
2850 FOR N=N1 TO N%
2860 PRINT " " N,G$(N),F$(N),M$(N),X$(N),Y$(N),Z$(N)
2870 NEXT N
2880 N%=N%+1: N=N%
2890 GOSUB 2950:       REM * Input 4
2900 IF M$(N)="02" THEN GOTO 2920: REM * End of Program?
2910 IF N<101 THEN GOTO 2830 ELSE GOTO 2830
2920 RETURN: REM * End PENTER
2930 REM ----------------------------
2940 REM * Procedure INPUT 4
2950 PRINT TAB(0,22) "ENTER BLOCK ";N " (Type - for no data)"
2960 PRINT; TAB(0,23) N;
2970 INPUT TAB(5,23)  "G=" G$(N)
2980 INPUT TAB(11,23) "F=" F$(N)
2990 INPUT TAB(17,23) "M=" M$(N)
3000 INPUT TAB(23,23) "X=" X$(N)
3010 INPUT TAB(29,23) "Y=" Y$(N)
3020 INPUT TAB(35,23) "Z=" Z$(N)
3030 PRINT "Have you entered correct data (Y or N)?";
3040 IF GET$="Y" THEN GOTO 3080
3050 PRINT TAB(0,23) "                                        "
3060 PRINT "                                              ";
3070 GOTO 2950:       REM * Try again
3080 RETURN: REM * End INPUT 4
3090 REM ----------------------------
3100 REM * Procedure EDIT
3110 GOSUB 1600:      REM * Display 1
3120 PRINT "              * EDIT *": PRINT
3130 PRINT "     Each program block will be"
3140 PRINT "displayed, if you wish to modify it"
3150 PRINT "type Y in response to the prompt. If"
3160 PRINT "you type N the next block will be"
3170 PRINT "displayed.": PRINT: PRINT
3180 PRINT "Do you wish to proceed (Y or N)?";
3190 IF GET$="Y" THEN GOTO 3210
3200 GOTO 3370:       REM * Return
3210 REM * List & Edit
3220 @%=5: N%=1: N=1
3230 CLS: PRINT: PRINT "                * EDIT *"
3240 PRINT TAB(5,3) "N","G","F","M","X","Y","Z": PRINT
3250 IF N)9 THEN N1=N%-9 ELSE N1=1
```

Program 6 BBC continued

```
3830 GOSUB 1380:        REM * Z Move
3840 RETURN: REM * End POSITION
3850 REM ----------------------------
3860 REM * Procedure HOLD
3870 PRINT "HOLD type G to continue?"
3880 IF GET$()"G" THEN GOTO 3870
3890 RETURN: REM * End HOLD
3900 REM ----------------------------
3910 REM * Procedure FIXED DRILL CYCLE
3920 UP1=0:             REM * Feedrate up
3930 Z=EVAL(Z$(N))
3940 DO1=DL2: REM * Feedrate down
3950 MOV1=Z*48:    REM * Z steps
3960 GOSUB 4000:        REM * Drill
3970 RETURN: REM * End FIXED DRILL CYCLE
3980 REM ----------------------------
3990 REM * Procedure DRILL
4000 IF X$(N)="-" THEN XB=XA ELSE XB=EVAL(X$(N))
4010 IF Y$(N)="-" THEN YB=YA ELSE YB=EVAL(Y$(N))
4020 DL1=0:             REM * XY max feedrate
4030 GOSUB 1130:        REM * XY travel
4040 GOSUB 1230:        REM * XY drive
4050 GOSUB 1470:        REM * Drill cycle
4060 RETURN: REM * End DRILL
4070 REM
4080 REM ##############################
4090 REM
4100 REM  ** START OF MAIN PROGRAM **
4110 REM
4120 GOSUB 610:         REM * Initialise Ports
4130 GOSUB 690:         REM * Initialise Variables
4140 GOSUB 1600:        REM * Display 1
4150 GOSUB 880:         REM * Find Datum
4160 GOSUB 1880:        REM * Input 2, Mode
4170 IF MO$="P" THEN GOTO 4310: REM * Program Mode
4180 REM * SINGLE BLOCK MODE
4190 GOSUB 2060:        REM * Input 3, Single Block
4200 IF FC=5 THEN GOTO 4140:  REM * Restart program
4210 IF FC=1 THEN GOSUB 880: GOTO 4280: REM * Datum
4220 IF FC=2 THEN GOSUB 1670: REM * Input 1, Drill
4240 DL1=0
4250 GOSUB 1130:        REM * XY Travel
4260 GOSUB 1230:        REM * XY Drive
4270 IF FC=2 OR FC=3 THEN GOSUB 1470:  REM * Drill Cycle
4280 GOTO 4190:         REM * Get next block
4290 REM
4300 REM * PROGRAM MODE
       Enter FUNCTION?
4310 PM=0: GOSUB 2340: REM * PMODE
4320 IF PM=4 THEN GOTO 4140: REM * Restart program
4330 IF PM=1 THEN GOSUB 2510: REM * PENTER
4340 IF PM=2 THEN GOSUB 3100: REM * Edit
4350 IF PM=3 THEN GOSUB 3400: REM * Run
4360 GOSUB 880:         REM * Find Datum
4370 GOTO 4310:         REM * Get next
4380 REM   ** END OF MAIN PROGRAM **
4390 REM
```

Program 6 BBC continued

```
4970 X$(50)="61"
4980 X$(51)="59": Y$(51)="12"
4990 Y$(52)="18"
5000 X$(53)="61": Y$(53)="20"
5010 X$(54)="63"
5020 X$(55)="64": Y$(55)="19": Z$(55)="25"
5030 X$(56)="67": Y$(56)="20": Z$(56)="21"
5040 Y$(57)="10"
5050 Y$(58)="15"
5060 X$(59)="68"
5070 X$(60)="73": Y$(60)="20"
5080 X$(61)="68": Y$(61)="15"
5090 X$(62)="73": Y$(62)="10": Z$(62)="25"
5100 M$(63)="02"
5110 RETURN: REM * End DEMO
5120 END: REM * END OF CNC932 PROGRAM *
5130 REM ################################
```

Typical outputs from Program 6 BBC

```
                ********************************
                * C.N.C. DRILLING MACHINE. *
                ********************************
                        * MODE *
     Two modes are available, 'SINGLE
BLOCK' or 'PROGRAM'.
     In the SINGLE BLOCK mode each
block of instructions is acted upon
when it is entered.
     In the PROGRAM mode a number of
blocks of instructions may be entered,
and run to perform a sequence of
operations.
Which mode do you require (S or P)? S
                ********************************
                * C.N.C. DRILLING MACHINE. *
                ********************************
                     * SINGLE BLOCK 1 *
   Enter FUNCTION and XY co-ordinates.
   FUNCTION may be:- DATUM
                     DRILL
                     POSITION
                     EXIT
   X & Y ordinates must not exceed 95.
        Enter FUNCTION?   DRILL
        Enter X ordinate? 20
        Enter Y ordinate? 30
   Do you wish to proceed (Y or N)?
                ********************************
                * C.N.C. DRILLING MACHINE. *
                ********************************
                   SPECIFY DRILL CYCLE.
 MOVEMENT (up & down maximum 25) mm? 15
 FEEDRATE the following feedrates may
be entered, all are in mm/second:
0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0,
```

Typical outputs from Program 6 (BBC) continued

```
    X & Y ordinates must not exceed 95.
         Enter FUNCTION?   EXIT
    Do you wish to proceed (Y or N)?
       ********************************
       * C.N.C. DRILLING MACHINE. *
       ********************************
       ********************************
       * C.N.C. DRILLING MACHINE. *
       ********************************
                  * MODE *
     Two modes are available, 'SINGLE
BLOCK' or 'PROGRAM'.
     In the SINGLE BLOCK mode each
block of instructions is acted upon
when it is entered.
     In the PROGRAM mode a number of
blocks of instructions may be entered,
and run to perform a sequence of
operations.
Which mode do you require (S or P)? P
       ********************************
       * C.N.C. DRILLING MACHINE. *
       ********************************
             * PROGRAM MODE *
      You may Enter a CNC program of up
to 100 blocks, Edit, or Run it. You can
not Run or Edit a program until it has
been Entered.
      The last instruction in a program
must be:- M=02 (End of program).
Type ENTER, EDIT, RUN, DEMO, or EXIT? ENTER
       ********************************
       * C.N.C. DRILLING MACHINE. *
       ********************************
You may enter the following functions:-
 G 00  Positioning; Point-to-Point.
 G 05  Hold.
 G 80  Cancel Fixed Drill Cycle.
 G 81  Fixed Drill Cycle.
 M 00  Program Stop.
 M 02  End of Program.
 M 03  Spindle (Drill) On.
 M 05  Spindle Off.
 F     Feedrate may be entered in the
       following units ONLY, units are
       in mm/second:-
       0.1   0.25  0.5   0.75
       1.0   1.5   2.0   2.5
       3.0   3.5   4.0
Type Y to continue.
  The X and Y ordinates are initialised
to zero (machine datum), all ordinates
are specified with respect to the
machine datum.
  The Z ordinate is initialised to 25,
that is fully up.
  The initial position of the drill
```

Typical outputs from Program 6 (BBC) continued

```
        ********************************
        * C.N.C. DRILLING MACHINE. *
        ********************************
              * PROGRAM MODE *
     You may Enter a CNC program of up
to 100 blocks, Edit, or Run it. You can
not Run or Edit a program until it has
been Entered.
     The last instruction in a program
must be:- M=02 (End of program).
Type ENTER, EDIT, RUN, DEMO, or EXIT? EDIT
        ********************************
        * C.N.C. DRILLING MACHINE. *
        ********************************
                  * EDIT *
     Each program block will be
displayed, if you wish to modify it
type Y in response to the prompt. If
you type N the next block will be
displayed.
Do you wish to proceed (Y or N)?
                  * EDIT *
N     G     F     M     X     Y     Z
      1     81    -     -     20    20    15
Do you wish to change Block 1 (Y or N),
type E to leave EDIT?ENTER BLOCK 1 (Type - for no data)
1G=81
F=3
M=-
X=20
Y=20
Z=15
Have you entered correct data (Y or N)?
                  * EDIT *
N     G     F     M     X     Y     Z
      1     81    3     -     20    20    15
      2     -     -     -     -     30    -
Do you wish to change Block 2 (Y or N),
type E to leave EDIT?
                  * EDIT *
N     G     F     M     X     Y     Z
      1     81    3     -     20    20    15
      2     -     -     -     -     30    -
      3     -     -     -     30    -     -
Do you wish to change Block 3 (Y or N),
type E to leave EDIT?
        ********************************
        * C.N.C. DRILLING MACHINE. *
        ********************************
              * PROGRAM MODE *
     You may Enter a CNC program of up
to 100 blocks, Edit, or Run it. You can
not Run or Edit a program until it has
been Entered.
     The last instruction in a program
must be:- M=02 (End of program).
Type ENTER, EDIT, RUN, DEMO, or EXIT? RUN
```

FEEDBACK

Please note that the following corrections apply to the CNC932 - Volume 1 instruction manual.

Refer to the Apple II program listing on pages 43 - 51 of the manual, the following five lines should be changed to read:

```
15ØØ FOR DL = Ø TO DO1 : NEXT DL

154Ø FOR DL = Ø TO UP1 : NEXT DL

347Ø F = VAL(F$(N)): D = F*1Ø: DP = FR(D)

392Ø UP1 = Ø

394Ø DO1 = DP
```

PROGRAM 6 (APPLE-II)

```
10  REM  ********************************
20  REM  * PROGRAM 6 FOR CNC MACHINE *
30  REM  *           CNC 932          *
40  REM  ********************************
50  REM
60  REM  * FOR APPLE II COMPUTER
70  REM  * COPYRIGHT (C) 1983
80  REM  * FEEDBACK INSTRUMENTS LTD.
90  REM  * COLIN WILLIAMS, JULY 1983
100  REM  * VER. 1
110  REM
120  GOTO 4120: REM  * START OF MAIN PROGRAM
130  REM  ##############################
590  REM
600  REM  * PROCEDURE INITIALISE PORTS
610 PA = 49280:AC = PA + 1
620 PB = PA + 2:BC = PA + 3
630  POKE AC,0: POKE BC,0: POKE PA,207: POKE PB,255
640  POKE AC,4: POKE BC,4: POKE PA,0: POKE PB,0
660  RETURN : REM  * END INITIALISE PORTS
670  REM  ------------------------------
680  REM  * PROCEDURE INIT VARIABLES
690  DIM FR(40),G$(100),F$(100),M$(100),X$(100),Y$(100),Z$(100)
700 FR(1) = 266:FR(2.5) = 105:FR(5) = 50:FR(7.5) = 31:FR(10)
    = 21:FR(15) = 12
710 FR(20) = 7:FR(25) = 5:FR(30) = 2:FR(35) = 1:FR(40) = 0
720 AD = 0:DC = 0:DP = 0:D1 = 0:PM = 0:ZA = 0
730  RETURN : REM  * END INIT VARIABLES
740  REM  ------------------------------
750  REM  * PROCEDURE OUTPUT
760  POKE PA,EN
770  FOR SN = 1 TO SO
780  FOR ST = 1 TO 48
790  POKE PB,DR: POKE PB,0
800  FOR DL = 0 TO DN: NEXT DL
810  NEXT ST
820  NEXT SN
830  POKE PA,0
840  RETURN : REM  * END OUTPUT
850  REM  ------------------------------
860  REM  * PROCEDURE DATUM
870  REM  * X DATUM
880  POKE PA,2
890  POKE PB,8: POKE PB,0
900 DX = ( PEEK (PA) ) 31)
910  IF DX = 0 THEN  GOTO 890
920 SN = 20
930  POKE PB,4: POKE PB,0
940 SN = SN - 1: IF SN ) 0 THEN  GOTO 930
950  POKE PA,1
960  POKE PB,2: POKE PB,0
970 DY = ( PEEK (PA) ) 15)
980  IF DY = 0 THEN  GOTO 960
990 SN = 20
1000  POKE PB,1: POKE PB,0
1010 SN = SN - 1: IF SN ) 0 THEN  GOTO 1000
1020  POKE PA,0:XA = 0:YA = 0
```

Program 6 Apple-II continued

```
1610  PRINT "       ********************************"
1620  PRINT "       * C.N.C. DRILLING MACHINE. *"
1630  PRINT "       ********************************": PRINT
1640  RETURN : REM  * END DISPLAY1
1650  REM  ------------------------------
1660  REM  * PROCEDURE INPUT 1
1670  GOSUB 1600
1680  PRINT "           SPECIFY DRILL CYCLE.": PRINT
1690  INPUT " MOVEMENT (UP & DOWN MAXIMUM 25) MM? ";MV: PRINT
1710  IF MV ) 25 THEN  GOTO 1670
1720 ME = MV * 48
1730  PRINT " FEEDRATE, THE FOLLOWING FEEDRATES MAY"
1740  PRINT "BE ENTERED, ALL ARE IN MM/SECOND:"
1750  PRINT " 0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0,"
1760  PRINT " 2.5, 3.0, 3.5, 4.0. DO NOT ATTEMPT TO"
1770  PRINT " ENTER ANY OTHER NUMBERS.": PRINT
1780  INPUT "           FEEDRATE DOWN? ";DO: PRINT
1790  INPUT "           FEEDRATE UP?   ";UP: PRINT
1800  PRINT " IS ABOVE DATA CORRECT (Y OR N)?"
1810  GET A$: IF A$ = "Y" THEN  GOTO 1830
1820  GOTO 1670
1830 D = DO * 10:DO1 = FR(D)
1840 D = UP * 10:UP1 = FR(D)
1850  RETURN : REM  * END INPUT 1
1860  REM  ------------------------------
1870  REM  * PROCEDURE INPUT 2
1880  GOSUB 1600
1890  PRINT "                * MODE *": PRINT : PRINT
1900  PRINT "    TWO MODES ARE AVAILABLE 'SINGLE"
1910  PRINT "BLOCK' OR 'PROGRAM'."
1920  PRINT "    IN THE 'SINGLE BLOCK' MODE EACH"
1930  PRINT "BLOCK OF INSTRUCTIONS IS ACTED UPON"
1940  PRINT "WHEN IT IS ENTERED."
1950  PRINT "    IN THE 'PROGRAM' MODE A NUMBER OF"
1960  PRINT "BLOCKS OF INSTRUCTIONS MAY BE ENTERED,"
1970  PRINT "AND RUN TO PERFORM A SEQUENCE OF"
1980  PRINT "OPERATIONS.": PRINT
1990  INPUT "WHICH MODE DO YOU REQUIRE (S OR P)? ";MO$
2000 BL = 1
2010  IF MO$ = "P" THEN  GOTO 2030
2020  IF MO$ (  ) "S" THEN  GOTO 1990
2030  RETURN : REM  * END INPUT 2
2040  REM  ------------------------------
2050  REM  * PROCEDURE INPUT 3
2060  GOSUB 1600
2070  PRINT "          * SINGLE BLOCK "BL" *": PRINT : PRINT
2080 BL = BL + 1
2090  PRINT "  ENTER FUNCTION AND XY CO-ORDINATES."
2100  PRINT "  FUNCTION MAY BE:- DATUM"
2110  PRINT "                    DRILL"
2120  IF BL ) 2 THEN  PRINT "                          REPEAT (DRILL)"
2130  PRINT "                    POSITION"
2140  PRINT "                    EXIT"
2150  PRINT "  X & Y ORDINATES MUST NOT EXCEED 95.": PRINT
2160  INPUT "       ENTER FUNCTION?   ";FC$
2170 FC = 0
2180  IF FC$ = "DATUM" THEN FC = 1: GOTO 2280
```

Program 6 Apple-II continued

```
2750  PRINT "  THE Z ORDINATE IS INITIALISED IN THE"
2760  PRINT "PROGRAM TO 25, THAT IS FULLY RAISED."
2770  PRINT "  THE INITIAL POSITION OF THE DRILL"
2780  PRINT "(Z ORDINATE), MUST BE SET MANUALLY"
2790  PRINT "BEFORE RUNNING A PROGRAM.": PRINT : PRINT
2800  PRINT "TYPE Y TO CONTINUE.";
2810  GET A$
2820 N% = 0:N = 0
2830  HOME : PRINT "   N    G    F    M    X    Y    Z": PRINT
2840  IF N ) 17 THEN N1 = N% - 17: GOTO 2850
2845 N1 = 1
2850  FOR N = 1 TO N%
2860  PRINT  TAB( 4)N TAB( 9)G$(N) TAB( 14)F$(N) TAB( 19)M$(N)
      TAB( 24)X$(N) TAB( 29)Y$(N) TAB( 34)Z$(N)
2870  NEXT N
2880 N% = N% + 1:N = N%
2890  GOSUB 2950
2900  IF M$(N) = "02" THEN  GOTO 2920
2910  IF N ( 101 THEN  GOTO 2830
2920  RETURN : REM  * END PENTER
2930  REM  ------------------------------
2940  REM  * PROCEDURE INPUT 4
2950  VTAB 22: PRINT "ENTER BLOCK "N" (TYPE - FOR NO DATA)"
2960  PRINT N
2970  VTAB 23: HTAB 6: INPUT "G=";G$(N)
2980  VTAB 23: HTAB 12: INPUT "F=";F$(N)
2990  VTAB 23: HTAB 18: INPUT "M=";M$(N)
3000  VTAB 23: HTAB 24: INPUT "X=";X$(N)
3010  VTAB 23: HTAB 30: INPUT "Y=";Y$(N)
3020  VTAB 23: HTAB 36: INPUT "Z=";Z$(N)
3030  PRINT "HAVE YOU ENTERED CORRECT DATA (Y OR N)?"
3040  GET A$: IF A$ = "Y" THEN  GOTO 3080
3050  VTAB 23: PRINT "                                        "
3060  PRINT "                                        "
3070  GOTO 2950
3080  RETURN : REM  * END INPUT 4
3090  REM  ------------------------------
3100  REM  * PROCEDURE EDIT
3110  GOSUB 1600
3120  PRINT "                * EDIT *": PRINT
3130  PRINT "     EACH PROGRAM BLOCK WILL BE"
3140  PRINT "DISPLAYED, IF YOU WISH TO MODIFY IT"
3150  PRINT "TYPE Y IN RESPONSE TO THE PROMPT.  IF"
3160  PRINT "YOU TYPE N THE NEXT BLOCK WILL BE"
3170  PRINT "DISPLAYED.": PRINT : PRINT
3180  PRINT "DO YOU WISH TO PROCEED (Y OR N)?";
3190  GET A$: IF A$ = "Y" THEN  GOTO 3210
3200  GOTO 3370
3210  REM  * LIST & EDIT
3220 N% = 1:N = 1
3230  HOME : PRINT : PRINT "                * EDIT *"
3240  PRINT : PRINT "   N    G    F    M    X    Y    Z": PRINT
3250  IF N ) 9 THEN N1 = N% - 9: GOTO 3260
3255 N1 = 1
3260  FOR N = N1 TO N%
3270  PRINT  TAB( 4)N TAB( 9)G$(N) TAB( 14)F$(N) TAB( 19)M$(N)
      TAB( 24)X$(N) TAB( 29)Y$(N) TAB( 34)Z$(N)
```

Program 6 Apple-II continued

```
3810  GOSUB 1130
3820  GOSUB 1230
3830  GOSUB 1380
3840  RETURN : REM  * END POSITION
3850  REM  ------------------------------
3860  REM  * PROCEDURE HOLD
3870  PRINT "'HOLD' TYPE G TO CONTINUE?";
3880  GET A$: IF A$ (  ) "G" THEN  GOTO 3870
3890  RETURN : REM  * END HOLD
3900  REM  ------------------------------
3910  REM  * PROCEDURE FIXED DRILL CYCLE
3920 U1 = 0
3930 Z =  VAL (Z$(N))
3940 D1 = DP
3950 ME = Z * 48
3960  GOSUB 4000
3970  RETURN : REM  * END FIXED DRILL CYCLE
3980  REM  ------------------------------
3990  REM  * PROCEDURE DRILL
4000  IF X$(N) = "-" THEN XB = XA: GOTO 4010
4005 XB =  VAL (X$(N))
4010  IF Y$(N) = "-" THEN YB = YA: GOTO 4020
4015 YB =  VAL (Y$(N))
4020 DN = 0
4030  GOSUB 1130
4040  GOSUB 1230
4050  GOSUB 1470
4060  RETURN : REM  * END DRILL
4070  REM
4080  REM  ##############################
4090  REM
4100  REM   ** START OF MAIN PROGRAM **
4120  GOSUB 610: REM  * INITIALISE PORTS
4130  GOSUB 690: REM  * INITIALISE VARIABLES
4140  GOSUB 1600: REM  * DISPLAY 1
4150  GOSUB 880: REM  * FIND DATUM
4160  GOSUB 1880: REM  * INPUT 2, MODE
4170  IF MO$ = "P" THEN  GOTO 4310
4180  REM  * SINGLE BLOCK MODE
4190  GOSUB 2060: REM  * INPUT 3, SINGLE BLOCK
4200  IF FC = 5 THEN  GOTO 4140: REM  * RESTART
4210  IF FC = 1 THEN  GOSUB 880: GOTO 4280: REM  * DATUM
4220  IF FC = 2 THEN  GOSUB 1670: REM  * INPUT 1
4240 DN = 0
4250  GOSUB 1130: REM  * XY TRAVEL
4260  GOSUB 1230: REM  * XY DRIVE
4270  IF FC = 2 OR FC = 3 THEN  GOSUB 1470: REM  * DRILL CYCLE
4280  GOTO 4190: REM  * GET NEXT BLOCK
4290  REM
4300  REM  * PROGRAM MODE
4310 PM = 0: GOSUB 2340: REM  * PMODE
4320  IF PM = 4 THEN  GOTO 4140: REM  * RESTART
4330  IF PM = 1 THEN  GOSUB 2510: REM  * PENTER
4340  IF PM = 2 THEN  GOSUB 3100: REM  * EDIT
4350  IF PM = 3 THEN  GOSUB 3400: REM  * RUN
4360  GOSUB 880: REM  * FIND DATUM
4370  GOTO 4310: REM  * GET NEXT
4380  REM    ** END OF MAIN PROGRAM **
```

Program 6 Apple-II continued

```
4970 X$(50) = "71"
4980 X$(51) = "69":Y$(51) = "22"
4990 Y$(52) = "28"
5000 X$(53) = "71":Y$(53) = "30"
5010 X$(54) = "73"
5020 X$(55) = "74":Y$(55) = "29":Z$(55) = "25"
5030 X$(56) = "77":Y$(56) = "30":Z$(56) = "20"
5040 Y$(57) = "20"
5050 Y$(58) = "25"
5060 X$(59) = "78"
5070 X$(60) = "83":Y$(60) = "30"
5080 X$(61) = "78":Y$(61) = "25"
5090 X$(62) = "83":Y$(62) = "20":Z$(62) = "25"
5100 M$(63) = "02"
5110  RETURN : REM  * END DEMONSTRATION
5120  END : REM  * END OF CNC932 PROGRAM *
5130  REM        ############################
```

PROGRAM DIFFERENCES FOR VARIOUS COMPUTERS — CHAPTER 4

First the features will be described which are common to the BASIC language of most computers. Exceptions and peculiarities will then be described for individual computers.

Initialisation

Each computer requires a different procedure to initialise the ports. These differences are accommodated by the initialisation subroutines in Chapter 2. Further information is given in the manual for the interface of each computer (or 'Interfacing MICA Modules').

Variable names

Different versions of BASIC impose different restrictions on the format of the name of a variable. The names used in the APPLE listing of Program 6 should be acceptable in most cases.

Input/Output Instructions

In the versions of BASIC used by most computers, input is obtained by using the expression PEEK(PA) in exactly the same way as the BBC computer uses the expression ? PA.

Similarly output is sent to the ports using the instruction

```
POKE Port-address, expression
```

to replace the BBC computer instruction

```
? port-address, = expression.
```

Thus for instance line 760 of Program 6 would become

```
760 POKE PA, EN : REM * enable drive
```

Selecting bits

The detailed operation of different versions of BASIC in relation to logical statements varies. However in most cases the expression (V AND32)will have a non-zero value if bit 5 of variable V is set, and 0 otherwise. V is likely to be replaced by PEEK(PA) in this application, corresponding to ? PA in lines 900, 910 and 970, 980 of the BBC Program 6 (Chapter 3).

Programming APPLE-II

On the APPLE-II the AND instruction does not select bits in this way, so that the relevant input bits must be tested by arithmetic. This is facilitated if the state of the high-order non-input bits is known. For this reason bits 6 and 7 of port A are initialised for output in the initialisation routine of Chapter 3. Every byte output to port A can then reset bits 6 and 7 to zero. The following tests can then be used for the required input bits.

When bit 5 (X reference detector) signals '1', the expression (PEEK(PA) > 31) will be true (i.e a numeric value of 1).

Thus Program 6, lines 900 to 970 replaced by the following:

```
900 DX = (PEEK (PA) > 31)
910 IF DX = 0 THEN GOTO 890
    ... (920 to 960 unchanged)
970 DY = (PEEK(PA) > 15)
    ... (980 onwards unchanged)
```

If bit 5 of port A is set then line 910 transfers control to line 890 of the program. Otherwise bit 5 is clear and operation continues in sequence. It is assumed that nothing has altered the clear state of bit 5 by the time line 970 is reached.

Programming the ABC80

The BASIC language of the ABC80 does not allow two-letter names for variables, but does allow names comprising one letter followed by one digit. Variable names must be amended accordingly.

Machine

The machine is designed to require a minimum of maintenance and if lubricated and kept free from abrasive dust should give a long life with minimum need for adjustment.

The lead screws and guides should be oiled with a light machine oil. This should be done regularly but using very little oil; too much oil attracts dust.

There are no routine adjustments. If the thrust bearings become worn, the slack can be taken up by moving the appropriate flexible coupling, which may be freed by the use of a 2mm hexagon key.

The stepping motors and the drill head motor are not repairable; if faulty they will have to be replaced. Spare parts can be obtained from Feedback Instruments Ltd., who can also undertake repairs on the machine if required.

Control Box

No routine maintenance is required for the control box apart from a periodic check that the main supply cable is in good order. The panel may be cleaned with a moist soft cloth, but chemical cleaners and free water must be avoided.

WARNING

> Dangerous voltages are present inside the control box. Before attempting to remove any part of its cover, disconnect from the electricity supply. Avoid contact with any part of the supply circuit during operation with any part of the cover removed.

The links which connect the X and Y reference input signals to port A or to the CA2 and CB2 input lines are accessible by removing the right-hand (viewed from the front) end cover. They are wire links to be soldered in the required position as indicated on the circuit board.

The control box should be extremely reliable. However if it does break down, repairs should be attempted only by a qualified and skilled engineer. A circuit diagram is provided, fig. 9. All components should be widely available, with the possible exception of the integrated circuits which drive the stepping motors. Parts may be obtained from Feedback Instruments Limited.

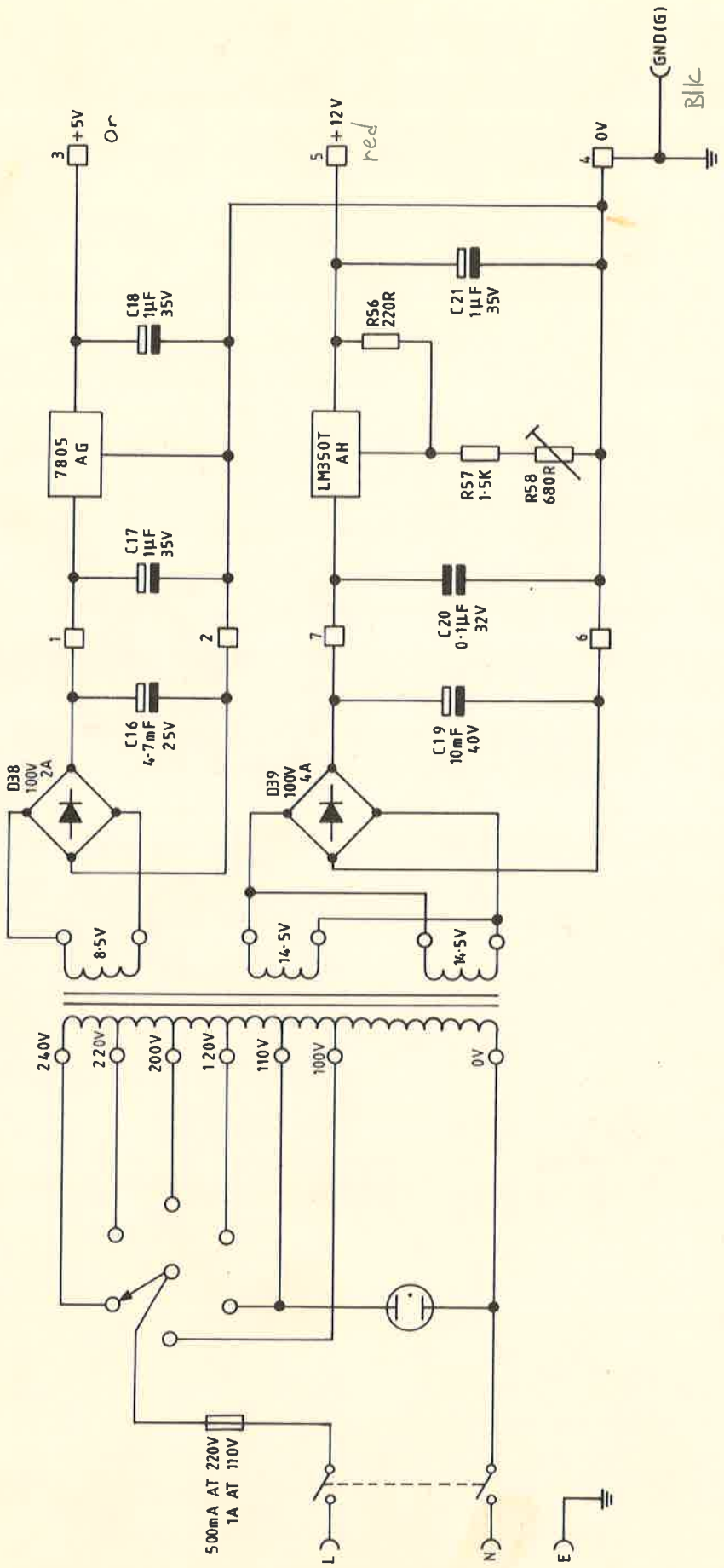

Fig 9. Circuit diagram of Power Supply (3-932-10391 Iss.1)