

Panel Copy

A F Wilson

One of the best features of the Memotech MTX series computers is their built in assembler/disassembler, the Panel. Unfortunately, the one thing missing from it is the ability to dump the contents thereof, to a printer. This program remedies that problem.

(1) For SDX Users

Type in listing, then type *GOTO 100*. This will autosave the program to disc as

"*PANEL.COD*". To use the program, just type *USER LOAD "PANEL.COD"*. This will reload the program back into memory and auto-initialise itself in high memory. Therefore when you enter *PANEL*, function keys F5-F8, are now initialised (see later for meanings).

(2) For SDX/FDX CP/M Users

As in (1), except replace the references to *USER* with *DISC*.

(3) For Tape Users

As in (1), except remove the references to *USER*. To load use *LOAD "PANELCOD"*.

You should now have on disc or tape a program called *PANEL.COD* or *PANELCOD*. This program is interrupt driven and is at top of free memory,

therefore it won't be overwritten. However, when the computer is reset, the *PANEL* extension location at £FA9E is reset, and this subroutine will become invisible to the Operating System and *PANEL*. As the program is still in memory at the desired location, £B007, you simply re-initialise it with *RAND USR(45063)*.

The screen is divided up into three key areas. The *disassembly*, the *registers* and the *Hexdump* at the bottom of the screen. This *PANEL* extension allows you to get a hardcopy of the whole *PANEL* screen (F5), the disassembly and registers (F6), the disassembly only (F7) and the *hexdump* only (F8). See figure below for a snapshot of the panel whole screen.

10 CODE

```

4007 JPANEL:LD A,£C3           ; SET PANEL EXTENSION WITH
4009         LD (£FA9E),A     ; JP £B012.
400C         LD HL,£B012     ;
400F         LD (£FA9F),HL   ;
4012 PSTART:LD A,(£FD7D)     ; WHAT WAS THE LAST KEYPRESS.
4015         CP £84          ; CHECK TO SEE IF IT ONE OF
4017         JR Z,F5         ; THE FOLLOWING OPTIONS.
4019         CP £85          ;
401B         JR Z,F6         ;
    
```

```

408A         JR NZ,PSEND     ; IF NOT F7,KEEP READING.
408C         LD A,(COUNT)  ; UPDATE ROW COLUMN COUNTER
408F         AND A          ; AND CHECK TO SEE IF END OF
4090         INC A          ; ROW YET.
4091         LD (£B0B5),A   ;
4094         LD B,A         ;
4095         LD A,(£B0B0)   ;
4098         CP B          ;
4099         JR Z,UPDVDP    ; UPDATE VDP ROW COUNTER.
409B         JR PSEND      ; NOT AT LINE END YET.
409D UPDVDP: PUSH DE       ; MOVE VDP ON BY 40 (SCREEN
409E         LD DE,40      ; WIDTH).
40A1         ADC HL,DE     ;
40A3         POP DE       ;
40A4         PUSH HL      ;
40A5         XOR A        ;
40A6         LD (£B0B5),A  ; ZERO F7 COL COUNTER.
40A9         JR VDPSET    ;
40AB LF:     DB 27,"A",16  ; SET PRINTER SPACING.
40AE         DB 27,"Q"    ; SET COLUMN
40B0 COL:    DS 1         ; WIDTH.
40B1 SCRNST:DS 2         ; £B0B1
40B3 LENPAN:DS 2         ; £B0B3
40B5 COUNT:  DS 1         ; £B0B5
40B6 END:    LD B,13     ; SEND CR.
40B8         CALL £0CE3   ;
40BB         LD B,10     ; SEND LF.
40BD         CALL £0CE3   ;
40C0         RET         ; RETURN TO PANEL EDITOR.
    
```

20 CODE

```

MOVECODE: LD HL,£4007      ; (OR £B007) START OF CODE.
          LD DE,£B007     ; NEW LOCATION IN MEMORY.
          LD BC,186       ; LENGTH OF CODE TO MOVE.
          LDIR           ; MOVE IT
          RET            ;
    
```

30 RETURN

```

100 USER SAVE "PANEL.COD"
110 GOSUB 20
120 RAND USR(45063)
130 NEW
    
```

although if the pulses work in the same way, simply turning the ball around should give the correct directions. Without ac-

ually trying it - I don't have a tracker ball to play with - I can't be absolutely sure how to do it. However, you can but try.

```

10 DIM x(7),y(7),z(7)
20 GOSUB 500
30 REM OS
40 GOSUB 500
50 GOSUB 200
60 GOSUB 400
70 GOSUB 50
80 REM draw=3
90 CLS
100 CALL MOVETO(x(0),y(0),z(0))
110 CALL LINETO(x(1),y(1),z(1))
120 CALL LINETO(x(2),y(2),z(2))
130 CALL LINETO(x(3),y(3),z(3))
140 CALL LINETO(x(4),y(4),z(4))
150 CALL MOVETO(x(5),y(5),z(5))
160 CALL LINETO(x(6),y(6),z(6))
170 CALL LINETO(x(7),y(7),z(7))
180 CALL LINETO(x(4),y(4),z(4))
190 CALL LINETO(x(3),y(3),z(3))
200 FOR pH=0 TO 3
210 CALL MOVETO(x(0),y(0),z(0))
220 CALL LINETO(x(1),y(1),z(1))
230 NEXT
240 CALL MOVETO(x(0),y(0),z(0))
250 CALL LINETO(x(2),y(2),z(2))
260 CALL MOVETO(x(1),y(1),z(1))
270 CALL LINETO(x(3),y(3),z(3))
280 RETURN
285 REM pnt
290 FOR s=0 TO 7
300 a=(x(0)*COS(s))+(y(0)*SIN(s))
310 b=(x(0)*SIN(s))+(y(0)*COS(s))
320 x(0)=a
330 y(0)=b
340 NEXT
350 RETURN
355 REM xrot
360 FOR s=0 TO 7
370 a=(x(0)*COS(s))+(z(0)*SIN(s))
380 b=(x(0)*SIN(s))+(z(0)*COS(s))
390 x(0)=a
400 z(0)=b
410 NEXT
420 RETURN
425 REM yrot
430 FOR s=0 TO 7
440 a=(x(0)*COS(s))+(z(0)*SIN(s))
450 b=(x(0)*SIN(s))+(z(0)*COS(s))
460 x(0)=a
470 z(0)=b
480 NEXT
490 RETURN

```

reset the points to their starting positions, and the value of the angle increased to give the new absolute rotation. This is, however, slower.

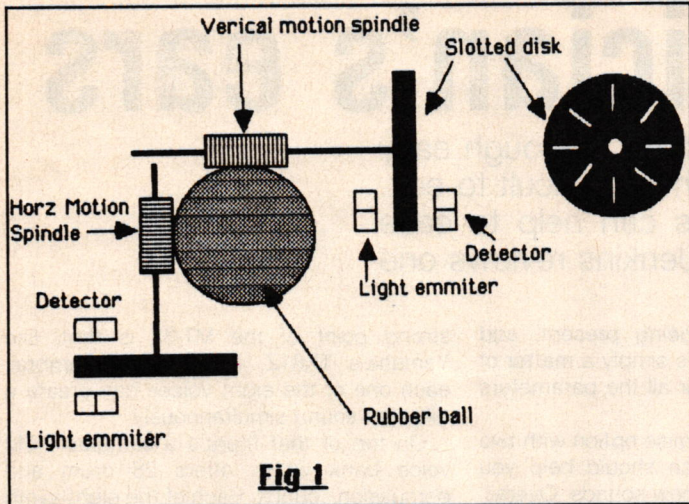


Fig 1

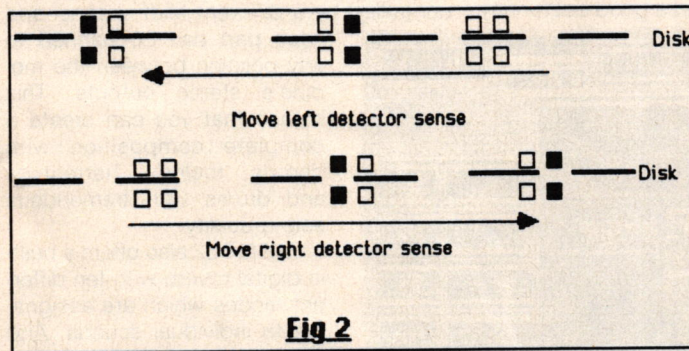


Fig 2

The return of Aquarius

Ian Price, of Peterlee, Co Durham, writes:

Q Help! I have just bought an Aquarius computer and cannot find any hardware for it. Please could you or any of your readers help as I am desperate to gain peripherals for my new computer.

A Well, I can't be of much help as the last time I saw an Aquarius was three years ago. I thought it was dead - maybe not. Perhaps someone can help with this.

Correction to Memotech

Due to solar conditions, the middle portion of the Memotech listing, Panel Copy, was omitted last week. So here it is:

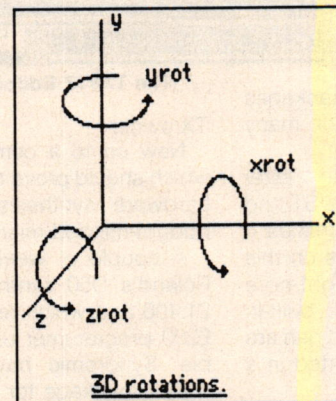
```

4010 CP:ERR
401F JR Z,F7
4021 CP:ERR
4023 JR Z,F8
4025 RET NZ
4026 F5: LD HL,716B
4029 LD DE,960
402C LD A,40
402E JR START
402F LD HL,716B
4032 LD DE,520
4035 LD A,40
4038 JR START
403A F7: LD HL,716B
403D LD DE,520
4040 XOR A
4041 LD (EB0B),A
4044 LD A,26
4046 JR START
4048 F9: LD HL,784B
404B LD DE,240
404E LD A,40
4050 JR START
4052 START: LD (EB0B),HL
4055 LD (EB0B),DE
4059 LD (EB0B),A
405C PUSH HL
405D SEPSON:LD HL,EB0B
4060 LD C,6
4062 POUT: LD B,(HL)
4063 CALL 60C3
4066 INC HL
4067 DEC C
4068 JR NZ,POUT
406A LD DE,(EB0B)
406E VOPSET:POP HL
406F PUSH AF
4070 LD A,L
4071 OUT (2),A
4073 LD A,H
4074 AND 20F
4076 OUT (2),A
4078 POP AF
4079 PSEND: IN A,(1)
407B LD B,A
407C VIA REGISTER B ONLY.
407F DEC DE
4080 AND A
4081 OR E
4082 JR Z,END
4084 LD A,(EB0B)
4087 AND A
4088 CP 26

```

The % symbol after a variable defines an integer which the Spectrum doesn't have, so just enter the variable without the % symbol. All angles are in radians. If you need to scale the drawing a little, simply alter the *100 in the drawing routine.

The important routines are zrot, xrot and yrot as these perform the rotations around the specific axes (see illustration). The variables a and b are needed since the values of x, y, and z are altered by the routine. The new coordinate needs to be evaluated first and then assigned to the array.



3D rotations.

Since each rotation is worked out from the last, the angle (an) need only be set once. The drawback with this is that inaccuracies will eventually creep in that will distort the shape. To get around this, the set-up routine can be used every time to

Is there anything about your computer you don't understand, and which everyone else seems to take for granted? Whatever your problem Peek it to Kenn Garroch and every week he will Poke back as many answers as he can. The address is Peek & Poke, PCW, 12-13 Little Newport Street, London WC2H 7PP.

Spectrum vector graphics

P Dixon, of Grays, Essex, writes:

Q I have been trying to write a program on my Spectrum to rotate wire frame shapes in all three planes. I have managed all right rotating the shape around the x and y axis, but when I came to the XZ and YZ rotations, it all went wrong.

Should I store points as xyz, or use angles from a fixed point? Could you give me the maths for the best method or the name of a suitable book?

A What you are trying to do is known as vector graphics and, as the name implies, the points are stored as vectors. Each point has an X, Y, and Z value, stored in three

arrays X, Y, and Z. When rotating, the point is moved around one of the three axes depending on which rotation you want.

When rotating around the X axis, only Y and Z values are altered, similarly, rotating around the Z axis, only X and Y are changed. There are, therefore, three rotation routines, one for each of the axes. All you need to do is process each point in your object, and then join the dots of the new points using the same sequence every time.

The program demonstrates a rotating cube with a cross on one face. The perspective is as though you were looking straight into it. You will need to convert the program for the Spectrum so note the following:

```

CALL MOVETO(X,Y) is equivalent to
PLOT X,Y
CALL LINETO(X,Y)-----DRAW X,Y

```