

Joysticks, circuit and I/O

By R. McFadgen

At \$495 the Dick Smith Wizard represents fairly good value. The main drawbacks seem to be lack of hardware (joysticks mainly), software, and detailed machine/memory information.

Here are three items that may help make this into more the powerful graphics/games machine it should be.

1. Joysticks:

Here is a circuit (figure 1, figure 2) to build your own joysticks. No mechanical details have been given, but I suggest the switches/contacts used should be positioned such that an intermediate angle will cause both adjacent contacts to make. The fire two button is optional as it is only ever used separate to the fire one in the cartridge game, 'Astro Pinball'.

If the fire buttons are too hard to

include they can be omitted and the existing keyboard keys used as below. No attempt has been made to provide for the start button as it is rather difficult and key 6 will do all starts.

Keyboard fire functions:

Player one (left): Fire one = Shift

: Fire two = Ctrl

Player two (right): Fire one = Minus

: Fire two = Right arrow

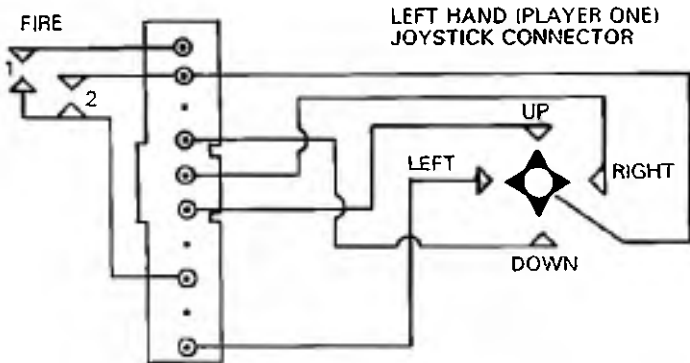


Figure 1

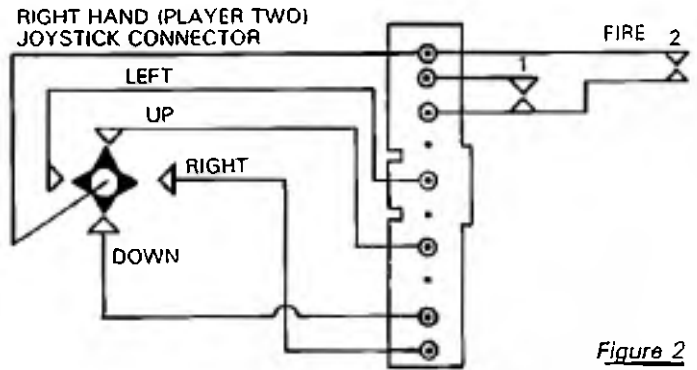


Figure 2

2. Memory investigation program (Inc Typewriter):

The program reprinted will help with software and detail of the machine and memory.

Be careful peeking and poking at location (Dec) 12289 or any location 12289 plus multiples of 256 (e.g. 12289+(256*21)=12801). These areas are visual display processor RAM and can dump the contents of the character generator to the screen. The screen will blank with just this program in memory (peek 12289) but if a graphics intense program such as Hangman has been in and running the results are quite different as the machine retains characters until they are overwritten. To regain normal screen display, reset then printpeek (13313). Some lines may be overwritten with @@@ or the like but they are easily found as Syntax errors when the program is run. Be very careful poking these areas as it can lead to your whole program being overwritten and only power down and reload will work.

Another good area to play with is DEC 4096. This is the sound generator chip and by poking various combinations at it all games sounds can be made. The sound will go on until a machine made sound overrides it, e.g. key entry sound. Trial and error works well.

The program has been with no rationalisation or reduction so it is easier to decode the various sections which have been separated as much as possible for use as is in other programs.

Memory investigator program.

```

10  LOG
20  REM *** A SIMPLE MEMORY ***
30  REM *** INVESTIGATION ***
40  REM *** PROGRAM ***
100 ORIGINATING PROGRAM ENDS YOU TO
110 PRINT
120 PRINT "ORIGINATING PROGRAM ENDS YOU TO"
130 PRINT
140 PRINT "ORIGINATING PROGRAM ENDS YOU TO"

```

```

150 PRINT "CONTENTS OF EITHER THE MEMORY
160 OR THE TO BE PRINTED IF AVAILABLE."
170 PRINT "LOCATION OR B":
180 PRINT:
190 PRINT "INPUT ANY VALUE (0 TO 9999)
200 PRINT "DECIMAL ONLY) TO ANY MEMORY "
210 PRINT "LOCATION IF THE MACHINE WILL "
220 PRINT "ALLOW THAT LOCATION TO CHANGE"
230 PRINT "DOBBE A OR B"
240 REM ** PARTIAL KEYBOARD SCAN ROUTINE **
250 DOB=10:0
260 IF PEEK(18)=129 THEN DOB=200
270 DOB=10:0
280 IF PEEK(18)=144 THEN DOB=000
290 DOB=10:0
300 REM ** SELECT I/O L/P AS HEX OR DEC **
310 REM ** AND D/D TO SCREEN OR PRINTER **
320 CLR
330 PRINT "DOB AT MEMORY CONTENTS **
340 PRINT:
350 PRINT "OPTIONS AVAILABLE ARE: 1 **
360 PRINT "A1 DEC INPUT WITH DEC OUTPUT "
370 PRINT "B1 HEX INPUT WITH DEC OUTPUT "
380 PRINT "C1 DEC INPUT WITH HEX OUTPUT "
390 PRINT "D1 HEX INPUT WITH HEX OUTPUT "
400 PRINT:
410 PRINT:
420 PRINT "DOBBE A TO D"
430 PRINT:
440 PRINT:
450 PRINT:
460 IF PEEK(18)=129 THEN DOB=200
470 IF PEEK(18)=144 THEN DOB=000
480 DOB=10:0
490 DOB=10:0
500 DOB=10:0
510 DOB=10:0
520 DOB=10:0
530 DOB=10:0
540 DOB=10:0
550 DOB=10:0
560 DOB=10:0
570 DOB=10:0
580 DOB=10:0
590 DOB=10:0
600 PRINT "CHOOSE OUTPUT DEVICE"
610 PRINT:
620 PRINT "E FOR SCREEN, F FOR PRINTER"
630 DOB=10:0
640 IF PEEK(18)=130 THEN DOB=70
650 DOB=10:0
660 IF PEEK(18)=141 THEN DOB=90
670 DOB=10:0
680 GO TO 700
690 J=1
700 CLR
1000 REM ** DISPLAY MEMORY **
1010 CLR
1020 PRINT "START ADDRESS IN " : DOB : " **
1030 INPUT A
1040 L=LEN(A$)
1050 IF L=0 THEN DOB=DOB+500
1060 IF L=0 THEN DOB=DOB+110
1070 FOR N=1 TO L
1080 IF PEEK(100+(A$*N,1))=150 THEN DOB=DOB+500
1090 NEXT N
1100 DOB=DOB+100
1110 IF A$=DOB THEN DOB=1010
1120 CLR
1130 PRINT "START ADDRESS (" : DOB : ") = " : A$
1140 IF J=1 THEN PRINT TAB(20); "START ADDRESS
1150 (" : DOB : ") = " : A$
1160 PRINT:
1170 PRINT TAB(5); "CONTENTS (" : DOB : ") "
1180 PRINT:
1190 FOR N=1 TO DOB
1200 DOB=DOB+1
1210 NEXT N
1220 PRINT:
1230 PRINT "CONTINUE"
1240 PRINT "NEW START ADDRESS (PUSH KEY B)"
1250 PRINT "TOTAL RESTART (PUSH KEY C)"
1260 DOB=10:0
1270 IF PEEK(18)=129 THEN DOB=1330
1280 DOB=10:0
1290 IF PEEK(18)=144 THEN DOB=10:0
1300 DOB=10:0
1310 IF PEEK(18)=156 THEN DOB=10:0
1320 GO TO 1260
1330 IF A$=DOB THEN DOB=1010
1340 DOB=10:0
1350 IF L=0 THEN DOB=DOB+500
1360 IF L=0 THEN DOB=DOB+110
1370 GO TO 1040
1400 DOB=PEEK(A)
1410 DOB=DOB+500
1420 DOB=DOB
1430 DOB=PEEK(A+1)
1440 DOB=DOB+500
1450 DOB=DOB
1460 IF J=1 THEN PRINT TAB(22); "P=" : A$
1470 PRINT:
1480 DOB=DOB+500
1490 DOB=DOB+500
1500 CLR
2000 REM ** MEMORY CHANGE ROUTINE **
2010 PRINT "MEMORY CONTENTS CHANGE **
2020 PRINT:
2030 PRINT "INPUT MEMORY LOCATION TO BE "
2040 PRINT "CHANGED (0 TO 65535 DEC ONLY)"
2050 INPUT A
2060 PRINT:
2070 PRINT "UP TO 10 VALUES CAN BE INPUT "
2080 PRINT "HOW MANY VALUES (1 TO 10)?"
2090 INPUT B
2095 PRINT:
2100 PRINT "A DELAY BETWEEN CHANGES TRY "
2110 PRINT "BE LOUD. RANGE 1 TO 10:"
2120 INPUT C
2130 IF C=65535 OR C=10 THEN DOB=DOB
2140 IF C=10000 THEN DOB=DOB

```

VZ200

```

2160 IFD(10RD)10THENC100
2165 CLS
2170 DRV(10)
2180 FORN=1TOP
2185 PRINT
2190 PRINT(OUTPUT VALUE "C:" TO 255):
2200 INPUTV(A)
2205 E=V(A)
2210 IFIV(A)1)2500(VIN)1(OTHER)2190
2220 NEXT
2225 CLS
2230 PRINT"LOADING" "CONTENTS"
2235 PRINT
2240 PRINT"PEEK(M)?" (ORIGINAL)
2250 FORN=1TOP
2260 FORM(VIN)
2270 PRINT"PEEK(M)?" NEW" P(1)?"
2290 FORI=1TO(D=10)
2300 NEXTI
2310 NEXTN
2315 PRINT
2320 PRINT"HOME LOCATION" (PUSH KEY A)
2330 PRINT"NEW LOCATION" (PUSH KEY B)
2340 PRINT"TOTAL ADDRESS" (PUSH KEY C)
2350 POKI(0)
2360 IFPEEK(10)=129THEN2065
2370 POKI(0)
2380 IFPEEK(10)=144THEN2000
2390 POKI(0)
2400 IFPEEK(10)=136THEN400
2410 GOTO2350
5000 REM ## M4 TO D5C ##
5002 REM ## CONVERSION ##
5003 REM ## B=1/D C=0/P ##
5010 B=1
5020 A=0
5030 IFL)ATHENC10)
5040 FORN=1TO10(P-1
5050 C=ASC(M:DO(A, N, 1))
5060 I=C)6ATHENC-C-7
5070 C=C-A8
5080 I=C)150NC10THENC10)
5090 A=A+(C)M)
5100 M=M+16
5110 NEXT
5120 RETURN
6000 REM ## U&C TO M4 ##
6002 REM ## CONVERSION ##
6003 REM ## B=1/D C=0/P ##
6010 REM ## I/P=0 TO 65535 ##
6020 M=4096
6030 C=""
6040 FORK=1TO4
6050 GOTO6100
6060 REM ## I/P=0 "D 255 ##
6070 M=16
6080 C=""

```

```

6090 FORN=1TO2
6100 C=)NT(D/M)
6110 B=D-(C)M)
6120 I=C)7ATHENC-C-9)
6130 I=C)7ATHENC-C-A)
6140 C=C-C)M)C)
6150 M=M+16
6160 NEXT
6170 RETURN
7000 REM # TYPE:TYPEA ROUTINE
7002 REM # TO USE: PRINT#A B
7003 REM # TYPE:TYPEA ROUTINE
7010 B=""
7020 B=""
7030 CLS
7040 PRINT#A:B
7050 P.DT32,2A,05
7060 INPUT#A
7070 P.DT32,2A,05
7080 INPUT#A
7090 PRINT
7100 PRINT#A:B
7110 IFJOY(3))OTHER)000
7120 IFJOY(4))OTHER)100
7130 GOTO7110
7140 LPRINT#A:B
7150 GOTO7030

```

3. New Product: Parallel I/O Module (Y-1606): Having seen these come on the market for \$99 I was most interested in their performance. Dick Smith kindly lent me one for evaluation and this complete article is the result.

It was produced using the Wizzard, the interface, and a U82A Microline printer. As can be seen the trio gave excellent results. Sadly, the printer I used had no graphic capability but the interface appeared able to transfer at least some form of graphics output.

As seems usual with Wizzard products the documentation was lamentable. Only a single diagrammatic sheet came in the

box and it wasn't very good. The only commands for the printer are LLIST and LPRINT.

LLIST: after trying all types of LLIST commands I decided LLIST on its own is the only one that works. This led to many tries to get a single line or group of lines only to be presented with a full listing each time. Apart from that, Oh the joys of hard copy. It is so much easier to work from.

LPRINT: at first glance this appears of little value as the only command left, but after the limitations are realised and compensated for most things seem possible.

This article was made using lines 7000 to 7150 of the program above, and it suited my typing skills to the ground.

The problem is the characters per line allowed by the Wizzard: LPRINT about 50 or 30 as a string. The printer I had allowed 80 or 160 characters per line and half the page went to waste, until I used line 7140 type ways around it. Once I tried this and similar I found the interface invaluable. Given a lot more time I feel I could have it doing all I wanted and more.

The other good sign was the second I/O BUS on the back of the interface which augers well for a disk drive in the near future. (We can but hope.)

All in all the Wizzard is slowly becoming the original promised thing: a full-blown, powerful, games/graphics home micro...