I.O.U.G. - First Program Supplement

February ~~1986~~ ~~1985~~
1986

```
1REM  ************************
2REM  * LOAD JOYSTICK ROUTINE *
3PEM  ************************
0 REM * INTODUCTION
1 PEM *   This program allows the machine code routine for the ORIC/IJK
2 REM * Joystick Interface to be loaded into a chosen area of memory.
3 PEM *   Only cursory checks are made on the area chosen therefore care has
4 PEM * to be taken over where the routine is placed.
5 REM *   Obviously, the routine cannot be placed between locations #C000 and
6 PEM * #FFFF which is the ROM area. The routine is 74 bytes (Hex #4A) long,
7 PEM * therefore.if you do not use the ORIC MICRODRIVE then locations #400 to
8 REM * #4B6 may be used for the Start Address of the routine. Otherwise, it
9 REM * may be placed in the area normally reserved for your BASIC program
0 REM * i.e. between #500 and #97B6, depending on the size of the program and
1 REM * its variable space. This can be determined by DEEKing location #A0.
2 REM * e.g. this program, with its REMs, uses up #156F bytes of memory
3 REM * therefore the Start Address must be placed above location #1570.
4 REM * If the 'GRAB' command is used in your program then the highest
5 PEM * location may be extended to #B3B6.
6 REM *   The program starts off by requesting the start address for the
7 REM * routine. Then it requests the location for the first byte of the two
8 REM * bytes required to hold the values returned from the joysticks (left
9 REM * and right).
0 REM * OBTAIN START ADDRESS *
00 CLS
10 PRINT:PRINT"Enter the Start Address for the":PRINT"joystick routine.  "::INP
T SA
20 IF SA<#400 OR SA>#B3B6 THEN ZAP:GOSUB600:GOTO110
30 PRINT:PRINT"Enter the first of the two locations"
40 PRINT"required to hold the left and right":PRINT"joystick values.  "::INPUT
OC
50 IF LOC>#BFFE OR (LOC>=SA AND LOC<SA+#4A) THEN ZAP:GOSUB600:GOTO130
60 REM * LOAD ROUTINE *
70 AD=SA
80 REPEAT
90 :   READ D$:IF D$="END OF CODE" THEN 210
00 :   POKE AD,VAL("#"+D$):AD=AD+1
10 UNTIL D$="END OF CODE"
20 DOKE SA+#2A,LOC:DOKE SA+#35,LOC+1
30 CLS:PRINT:PRINT"The routine is now loaded and ready to":PRINT"use"
40 PRINT:PRINT"The Start Address to be CALLed is:-";SA;"(Hex ";HEX$(SA);")"
50 PRINT:PRINT"The location to be PEEKed for the left":PRINT"joystick is:-";LOC
"(Hex ";
60 PRINT HEX$(LOC);")"
70 PRINT:PRINT"The location to be PEEKed for the":PRINT"right joystick is:-";LO
+1;"
80 PRINT "(Hex ";HEX$(LOC+1);")"
90 PRINT:PRINT"Note: Two variables could be assigned to these values."
00 PRINT"E.g.  J1=PEEK(";HEX$(LOC);"):J2=PEEK(";HEX$(LOC+1);")
10 REM * TEST FOR INTERFACE *
20 CALL SA
30 IF (PEEK(LOC) AND 32)<>0 THEN PRINT:PRINT"The interface is not connected.":G
TO500
40 PRINT:PRINT"The interface is connected."
50 REM * JOYSTICK TEST *
60 PRINT:PRINT"Would you like to test your joysticks?":PRINT"(Y/N)"
70 REPEAT:GET A$:UNTIL A$="Y" OR A$="N"
80 IF A$="N" THEN 500
```

```
390 CLS:PRINT CHR$(17);@4,6;"LEFT JOYSTICK";@21,6;"RIGHT JOYSTICK"
400 PRINT@6,25;"Press ESC to end the test."
410 DIM DIR(8,1):FOR X=1 TO 7 STEP 3:DIR(X,0)=1:DIR(X+1,0)=-1:NEXT
420 FOR Y=3 TO 5:DIR(Y,1)=1:DIR(Y+3,1)=-1:NEXT
430 X=9:Y=1:GOSUB700:X=26:GOSUB700:X=10:Y=126:GOSUB700:X=27:GOSUB700
440 OX=9:OY=15
450 REPEAT
460 :   CALL SA:J1=PEEK(LOC):J2=PEEK(LOC+1)
470 :   J=NOT(J1 OR #E0):X=9:GOSUB800
480 :   J=NOT(J2 OR #E0):X=26:GOSUB800
490 UNTIL KEY$=CHR$(27)
500 REM * SAVE ROUTINE *
510 CLS:PRINT CHR$(17)
520 PRINT"Do you wish to save the Joystick":PRINT"routine? (Y/N)"
530 REPEAT:GET A$:UNTIL A$="Y" OR A$="N"
540 IF A$="N" THEN PING:END
550 PRINT:PRINT"Press RECORD and PLAY on your cassette recorder then press any k
ey on"
560 PRINT"the computer.":GET A$
570 CSAVE"JOYSTICK",A(SA),E(SA+#4A)
580 PRINT:PRINT"The routine has been saved."
590 PING:END
600 REM * Sub-routine Error Message *
610 PRINT:PRINT"You cannot use this area of memory."
620 PRINT"Please choose a different location."
630 RETURN
700 REM * Sub-routine Cardinal Points *
710 FOR X1=0 TO 8:PLOT X+5*DIR(X1,0),15+5*DIR(X1,1),Y:NEXT
720 RETURN
800 REM * Sub-routine Analyse Joystick *
820 IF(J AND 4)=4 THEN SHOOT
830 J=J AND #1B
840 IF J>15 THEN J=J-10 ELSE IF J>7 THEN J=J-5
850 X1=X+5*DIR(J,0):Y1=15+5*DIR(J,1)
870 PLOT OX,OY,1:PLOTX1,Y1,0
880 OX=X1:OY=Y1
890 RETURN
1000 DATA 78,48,AD,0F,03,48,AD,03,03,48,AD,00,03,48,AD,02
1001 DATA 03,48,09,10,8D,02,03,A9,00,8D,00,03,A9,C0,8D,03
1002 DATA 03,A9,7F,8D,0F,03,AD,0F,03,8D,F0,97,A9,BF,8D,0F
1003 DATA 03,AD,0F,03,8D,F1,97,68,8D,02,03,68,8D,00,03,68
1004 DATA 8D,03,03,68,8D,0F,03,68,58,60,"END OF CODE"
```

Allan Whitaker.

Credits in order of appearance are:-

Load Joystick Routine - Allan Whitaker from Staleybridge, Cheshire.
Oric Golf - Chris Ward from Hitchin, Hertfordshire.
The Ball Puzzle - Alan Northcott from Wokingham, Berkshire.
Keyboard article - Steve Brunton from Camberley, Surrey.

Let me know what you think about the supplement and about my plans to bring the next one out on cassette.

ORIC GOLF

Only one key is required to play this game. That key is 'S'. For each of the 18 holes you are first presented with a picture of the fairway. The ball is usually on the fairway but may be in the rough (top and bottom). A moving index points to the choices of distance to the green (far right). Press 'S' to select a distance. The index can now be used to select the direction: U=up; D=down; C=centre. UU,UUU and DD,DDD respectively increase the angle of the given direction. Press 'S' to select the direction.

When the ball is on the green a new picture is presented. A circular index rotates to enable selection of direction with key 'S'. Another index then appears for distance selection, again with key 'S'. When the ball is in the hole the scorecard is displayed before the next hole is played.

The program was written on an Atmos and will need minor changes to run on the Oric-1.

```
10 REM       ****************
20 REM       *** ORIC GOLF **
30 REM       ****************
40 REM
50 REM              CHRIS WARD 25/9/85
60 REM
65 REM    ***************************
70 REM    **1. Main Loop 1080-1160  **
71 REM    **2. Fairway   1180-1690  **
72 REM    **3. Green     1700-2160  **
73 REM    **4. Scorecard 2165-2370  **
75 REM    ***************************
700 TEXT:CLS:PLOT11,13,10:PLOT12,13,"O R I C  G O L F"
710 PLOT11,14,10:PLOT12,14,"O R I C  G O L F"
720 FLOT11,26,"PLEASE PRESS A KEY"
730 GETZ$
740 A=RND(-DEEK(#276))
800 FORZ=#B400+(92*8)TO#B400+(92*8)+7:READQ:POKEZ,Q:NEXTZ
810 DATA 0,0,8,9,10,44,28,8
900 DATA 8,8,8,8,42,20,8,0
910 DATA 0,0,12,30,30,12,0,0
915 DATA 0,14,17,17,17,14,0,0
920 FORZ=#B400+(35*8)TO#B400+(35*8)+7:READQ:POKEZ,Q:NEXTZ
930 FORZ=#B400+(42*8)TO#B400+(42*8)+7:READQ:POKEZ,Q:NEXTZ
940 FORZ=#B400+(43*8)TO#B400+(43*8)+7:READQ:POKEZ,Q:NEXTZ
1000 POKE48035,0
1010 DIM HS(18):POKE #26A,2
1020 DATA 13,13,13,5.13,21,21,5,13
1030 DATA 5,21,13,13,21,5,5,13,13
1040 DIM PAR(18):FORN=1TO18:READPAR(N):NEXTN
1050 DATA 4,4,4,5,4,3,3,5,4
1060 DATA 5,3,4,4,3,5,5,4,4
1070 DIM PP(18):FORN=1TO18:READPP(N):NEXTN
1080 FOR HOLE=1 TO 18
1090 LET SHOT=0
```

1100 LET HI=Ø

```
1110 GOSUB 1180
1120 GOSUB 1700
1130 LET HS(HOLE)=SHOT
1140 LET TT=TT+SHOT
1150 GOSUB 2170
1160 NEXT HOLE
1170 END
1180 REM **** FAIRWAY ****
1190 LET Y=INT(3+(RND(1)*20))
1200 LET X=PAR(HOLE)
1210 DOKE18,48000:PRINT"              ORIC GOLF":CLS
1215 PAPER3:INK0:POKE48003,6
1220 PLOT 5,0,"HOLE":PLOT 10,0,STR$(HOLE)
1230 PLOT 15,0,"PAR":PLOT 20,0,STR$(PP(HOLE))
1240 FOR N=0TO2:PLOT0,N,21:NEXTN
1245 PLOT1,0,7:FORN=1TO2:PLOT1,N,2:NEXTN
1247 FORN=3TO39STEP2:PLOTN,1,"\":NEXTN
1248 FORN=4TO38STEP2:PLOTN,2,"\":NEXTN
1250 FOR N=22TO24:PLOT0,N,21:PLOT1,N,2:NEXTN
1252 FORN=3TO39STEP2:PLOTN,22,"\":PLOTN,24,"\":NEXTN
1254 FORN=4TO38STEP2:PLOTN,23,"\":NEXTN
1260 FORN=25TO26:PLOT1,N,7:NEXTN
1265 FORN=25TO26:PLOT0,N,17:NEXTN
1270 FOR N=13TO16:PLOT35,N,18:NEXTN
1280 PLOT 37,15,"+"
1290 FOR N=5 TO 25 STEP 5
1300 PLOT N,25,STR$(N/5)
1310 PLOT N,26,STR$(10*N)
1320 NEXT N
1330 PLOT X,Y,"*"
1340 LET S=5
1350 REPEAT
1360 PLOT S,25,STR$(S/5)
1370 LET S=S+5
1380 IF S=30 THEN S=5
1390 LET D=S
1400 PLOT S,25," "+CHR$(163)+" "
1410 WAIT 75
1420 LET G$=KEY$
1430 UNTIL G$="S"
1440 PLOT 2,25,"UUU     UU    U    C    D   DD   DDD  "
1450 PLOT 2,26,"                                      "
1460 LET A=3
1470 REPEAT
1480 PLOT A,26," "
1490 LET A=A+5
1500 IF A>35 THEN A=3
1510 PLOT A,26,CHR$(222)
1530 WAIT 75
1540 LET G$=KEY$
1550 UNTIL G$="S"
1560 LET A=((A-3)/5)
1570 LET A=A-3
1580 LET A=A*(PI/9)
1590 LET SHOT=SHOT+1:PING:WAIT 20
1600 PLOT X,Y," "
1610 LET X=X+(D*COS(A)):LET X=INT(X)
1620 IF((X<1)OR(X>39))THEN X=5:Y=5+INT(18*RND(1)):GOTO 1650
```

```
1630 LET Y=Y+(D*SIN(A)):LET Y=INT(Y)
1640 IF ((Y<1)OR(Y>26))THEN Y=5+INT(13*RND(1)):X=5
1650 PLOT X,Y,"*"
1660 IF (((X>34)AND(X<40))AND((Y<17)AND(Y>12))) THEN ZAP :WAIT 15:GOTO 1680
1670 GOTO 1210
1680 IF (X=37 AND Y=15) THEN HI=1
1690 RETURN
1700 REM *** PUTTING GREEN ***
1710 IF HI=1 THEN ZAP:ZAP:GOTO 2160
1720 LET X=22+(RND(1)*150)
1730 LET Y=22+(RND(1)*150)
1735 WAIT50
1740 HIRES:INK0:PAPER2:REM **SHOT**
1750 POKE#26A,2
1760 CURSET X,Y,0:FORN=2TO3:CIRCLEN,1:NEXTN
1770 CURSET 220,100,0:REM ** HOLE **
1780 CIRCLE 8,1:DRAW 0,-25,1:DRAW -3,3,1:DRAW 3,3,1
1790 PRINT:PRINT"PRESS S FOR DIRECTION"
1800 A=0 :CURSET 222,22,0:CIRCLE 15,1
1810 REPEAT
1820 CURSET 222,22,0
1830 DRAW 13*COS(A),13*SIN(A),0
1840 LET A=A+PI/30
1850 CURSET 222,22,0
1860 DRAW 13*COS(A),13*SIN(A),1
1870 LET G$=KEY$
1890 UNTIL G$="S"
1900 PRINT
1910 CLS:PRINT:PRINT"PRESS S FOR STRENGTH"
1920 LET S=5
1930 FOR N=5TO200STEP2:CURSETN,10,1:NEXTN
1940 REPEAT
1950 CURSET S,11,0:DRAW 4,5,0:DRAW -8,0,0:DRAW 4,-5,0
1960 LET S=S+5
1970 IF S=205 THEN S=5
1980 CURSET S,11,0:DRAW 4,5,1:DRAW -8,0,1:DRAW 4,-5,1
1990 WAIT10
2000 LET G$=KEY$
2010 UNTIL G$="S"
2020 LET SP=10
2030 PING:LET SHOT=SHOT+1
2039 LET L=5:H=0
2040 REPEAT
2045 LET H=H+L
2050 CURSET X,Y,0:FORN=2TO3:CIRCLEN,0:NEXTN
2060 LET X=X+L*COS(A)
2070 IF (X<25 OR X>235) THEN X=25:Y=100:GOTO 2110
2080 LET Y=Y+L*SIN(A)
2090 IF(Y<25 OR Y>199) THEN X=25:Y=100:GOTO 2110
2100 CURSET X,Y,0:FORN=2TO3:CIRCLEN,1:NEXTN
2110 LET R=SQR((X-220)^2+(Y-100)^2)
2120 LET L=R/10
2130 UNTIL (R<8 OR H>S)
```

```
2135 IF R<8 THEN ZAP:GOTO 2150
2140 GOTO 1740
2150 REM *** DE BALL IS IN !!!! ***
2160 RETURN
2165 REM **********SCORECARD *******
2170 TEXT:CLS:INK7:PAPER4
2180 POKE#26A,2:POKE 48035,0
2190 LET PC=0
2200 FOR N=1 TO HOLE:LET PC=PC+PP(N):NEXTN
2210 PLOT 13,4,"*************"
2215 PLOT 13,5,"* SCORECARD *"
2220 PLOT 13,6,"*************"
2230 PLOT 3,7,"HOLE":PLOT 12,7,"PAR":PLOT 20,7,"SCORE"
2240 FOR N=1 TO HOLE
2250 PLOT 3,N+7,STR$(N)
2260 PLOT12,N+7,STR$(PP(N))
2270 PLOT20,N+7,STR$(HS(N))
2280 IF PP(N)=HS(N) THEN PLOT 25,N+7,"PAR"
2290 IF PP(N)-HS(N)=1 THEN PLOT 25,N+7,"BIRDIE"
2300 IF PP(N)-HS(N)=2 THEN PLOT 25,N+7,"EAGLE"
2305 IF PP(N)-HS(N)=3 THEN PLOT 25,N+7,"ALBATROSS"
2310 NEXT N
2320 PLOT 5,26,"TOTAL "
2330 PLOT 12,26,STR$(TT)
2340 LET OP=TT-PC
2350 PLOT 19,26,STR$(OP)
2360 WAIT 1000
2370 RETURN
```

EARLY NEWS

Derek Howard from Trowbridge, Wilts. has recently been contacted by W.E. (Software). Their catelogue contains about 140 titles on cassette and about 20 on disk (Oric or Cumana). They also do repairs and upgrades of the Oric 1. Address:-

W.E. Software,
Foley Bank,
Worcester Road,
Great Malvern,
Worcestershire,
WR14 4QW
Tel. (06845) 69059

```
10 REM********************************
20 REM*         BALLS                *
30 REM********************************
40 HIMEM£97FF
50 POKE£26A,74 'VDU,DBL HT,NO CURSOR,CLICK
60 DOKE£26B,£A14 'BLUE BACK, DBL HT
70 DIMLX(20),LY(20),LC(20),RX(20),RY(20),RC(20)
80 TEXT
90 CLS
100 GOSUB1000'INSTRUCTIONS
110 GOSUB2000'INITIAL LAYOUT
120 K$=KEY$:IFK$=""THEN120
130 IFK$=CHR$(8)THENGOSUB400:GOTO120
140 IFK$=CHR$(9)THENGOSUB700:GOTO120
150 IFK$=CHR$(10)THENGOSUB500:GOTO120
160 IFK$=CHR$(11)THENGOSUB600:GOTO120
170 IFK$="N"THEN110
180 IFK$<>"E"THEN120
190 CALL DEEK(£FFFA)
400 C=LC(1)
410 FOR I=1TO19
420 :  LC(I)=LC(I+1)
430 NEXT I
440 LC(20)=C
450 RC(15)=LC(20)
460 RC(20)=LC(15)
470 GOTO800
500 C=LC(20)
510 FOR I=20TO2STEP-1
520 :  LC(I)=LC(I-1)
530 NEXT I
540 LC(1)=C
550 RC(15)=LC(20)
560 RC(20)=LC(15)
570 GOTO800
600 C=RC(1)
610 FOR I=1TO19
620 :  RC(I)=RC(I+1)
630 NEXT I
640 RC(20)=C
650 LC(15)=RC(20)
660 LC(20)=RC(15)
670 GOTO900
700 C=RC(20)
710 FOR I=20TO2STEP-1
720 :  RC(I)=RC(I-1)
730 NEXT I
740 RC(1)=C
750 LC(15)=RC(20)
760 LC(20)=RC(15)
770 GOTO900
800 FOR I=1TO20
810 :  CURSETLX(I),LY(I),3
820 :  FILL12,1,LC(I)
830 NEXT I
840 RETURN
900 FOR I=1TO20
910 :  CURSETRX(I),RY(I),3
920 :  FILL12,1,RC(I)
930 NEXT I
940 RETURN
```

```
1000 PRINT:PRINT"    T H E   B A L L   P U Z Z L E "
1010 PRINT:PRINT:PRINT:PRINT"The puzzle consists of coloured balls"
1020 PRINT:PRINT"in two interwoven rings, either of"
1030 PRINT:PRINT"which can be rolled around. Your task"
1040 PRINT:PRINT"is to restore them to the starting"
1050 PRINT:PRINT"position."
1060 PRINT:PRINT" Use the left and down arrows to move"
1070 PRINT:PRINT"the left ring,up and right arrows for"
1080 PRINT:PRINT"the right ring."
1090 PRINT:PRINT" Press N for a new game,E to end play"
1100 PRINT:PRINT:PRINT:PRINT"      Press any key to start."
1110 REPEAT:K$=KEY$:UNTILK$<>""
1120 PAPER0
1130 RETURN
2000 RESTORE
2010 FORI=1TO20
2020 :  READ LX(I),LY(I),LC(I),RX(I),RY(I),RC(I)
2030 NEXTI
2040 HIRES:PRINTCHR$(17)
2050 FILL200,1,0 'BLACK FOREGROUND
2060 CURSET6,0,3
2070 FILL200,1,151'WHITE BACKGROUND
2080 CURSET12,0,3
2090 FILL200,38,127'SOLID FOREGROUND
2100 CURSET87,99,3
2110 FORI=1TO72
2120 :  CIRCLEI,0
2130 NEXTI
2140 CURSET167,99,3
2150 FORI=45TO72
2160 :  CIRCLEI,0
2170 NEXTI
2180 FORI=1TO20
2190 :  CURSETLX(I),LY(I),3
2200 :  FORJ=1TO6
2210 :     CIRCLEJ,1
2220 :  NEXTJ
2230 :  LY(I)=LY(I)-6
2240 :  CURSETINT((LX(I)+11)/6)*6,LY(I),3
2250 :  FILL12,1,0
2260 :  LX(I)=(INT(LX(I)/6)-2)*6
2270 NEXTI
2280 FORI=1TO20
2290 :  CURSETRX(I),RY(I),3
2300 :  FORJ=1TO6
2310 :     CIRCLEJ,1
2320 :  NEXTJ
2330 :  RY(I)=RY(I)-6
2340 :  CURSETINT((RX(I)+11)/6)*6,RY(I),3
2350 :  FILL12,1,0
2360 :  RX(I)=(INT(RX(I)/6)-2)*6
2370 NEXTI
2380 GOSUB800
2390 GOSUB900
```

```
2400 PRINT:PRINT"    Do you want the balls shuffled?"
2405 PRINT"    Do you want the balls shuffled?";
2410 REPEAT:K$=KEY$:UNTILK$<>""
2420 IFK$="N"THEN 2510
2430 IFK$<>"Y"THEN2410
2440 PRINT:PRINT:PRINT"          How many moves?"
2445 PRINT"          How many moves?"CHR$(136)CHR$(128);
2446 INPUTM
2450 IFM<1THEN2440
2460 FORJ=1TOM
2470 :  PRINT:PRINT:PRINTSPC(18);J:PRINTSPC(18);J;
2480 :  K=RND(1)*4+1
2490 :  ON K GOSUB 400,500,600,700
2500 NEXTJ
2510 PRINT:PRINT:PRINT
2520 RETURN
2600 DATA114,45,0,144,153,2
2610 DATA96,40,0,162,157,2
2620 DATA78,40,0,180,157,2
2630 DATA60,45,0,198,151,2
2640 DATA48,57,0,209,139,2
2650 DATA36,72,0,216,125,2
2660 DATA30,90,0,222,108,2
2670 DATA30,108,0,222,90,2
2680 DATA36,126,0,216,73,2
2690 DATA48,141,0,209,59,2
2700 DATA60,153,3,198,47,4
2710 DATA78,158,3,180,41,4
2720 DATA96,158,3,162,41,4
2730 DATA114,153,3,144,45,4
2740 DATA129,141,3,129,57,4
2750 DATA138,126,3,116,72,4
2760 DATA144,108,3,110,90,4
2770 DATA144,90,3,110,108,4
2780 DATA138,72,3,116,126,4
2790 DATA129,57,4,129,141,3
```

N.B. In lines 40, 50, 60 and 190, enter " # " in place of the " £ " sign.

Alan Northcott.

The software in the Oric ROM only returns one key, regardless of how many are pressed. The exceptions to this are the shift keys, control, and function key (on the Atmos), which are read separately. The reason for this is found in the way that the keyboard hardware is arranged. The keys are arranged on an 8*8 matrix. The GI8912 sound chip is used to select the column of the keyboard being read, and the 6522 is used to select the row of the keyboard.

The keyboard routines in the Oric scan column 7 first, and column 0 last. For each column, the rows are scanned from 7 down to 0. If a key is pressed, then its location is saved, and the next column is scanned. This means that on a single scan of the whole keyboard, a second key press found will overwrite the first. To detect multiple key presses, a separate routine must be written to scan the keyboard directly, by-passing the Oric routines.

In order to do this, an understanding of the hardware involved must be obtained. I shall attempt to present a short description of this.

The GI8912 Programmable Sound Generator

The sound chip has fifteen registers. Registers 0 to 13 deal with producing the noises which form one of the Oric's stronger points. The final register, Hex 0E, is an 8-bit input/output port. This is used to access the key matrix, each bit representing one column of the keyboard matrix. All bits should be at logical 1 except for the bit representing the column being scanned, which should be 0. The Oric ROM routines to set this up can be used. To do this, the accumulator should contain the register number (in this case 0E hex) and the X-register should contain the value to be written to the GI8912. The Atmos routine is at #F590, the Oric-1 routine is at #F353.

The 6522 Versatile Interface Adapter

The 6522 VIA performs a variety of tasks, too many to list in detail here. It contains two data ports, named A and B. Bits 0 to 2 of port B specify which row of the keyboard is to be scanned, and bit 3 indicates whether the key has been pressed or not. The 6522 is memory mapped into the start of page 3 of the Oric's RAM. The data register for port B is register 0, so is to be found at location #300. The number of the row to be scanned must be written to this location. The row number should be 'OR'ed with #08 in order to leave the correct value in the top five bits.

The following page contains a diagram of the keyboard matrix, and shows it's relation to the GI8912 and 6522. There then follows an example program to demonstrate the method of reading the keyboard. The program simply looks for a press of the E key. If this is found, then 'E' is printed to the screen. Pressing Q will leave the program. A routine is used which checks the column held in X against the row held in Y, and sets the Z flag if the key is not pressed. It may not be very imaginative, but serves to demonstrate the methods involved. the The 6522 interrupts should be turned off if the routine is not interrupt driven, in order to avoid any conflict between this and the interrupt routines.

Steve Brunton

```
                                                   row
        8    L    0    /   SFT  RET        =        7   ____   +--------+
        Y    H    G    E         A    S    W        6   ____   |        |
        U    I    O    P   FNT  DEL   ]    [        5   ____   | 3 to 8 |___a0 6522
       SPC   ,    .    ↑   SFT   ←    ↓    →        4   ____   |        |___a1 Port B
        K    9    ;    -              \    '        3   ____   | M'plxr |___a2
        M    6    B    4   CTL   Z    2    C        2   ____   |        |
        J    T    R    F        ESC   Q    D        1   ____   |        |
        7    N    5    V         1    X    3        0   ____   +--------+

column  0    1    2    3    4    5    6    7
        |    |    |    |    |    |    |    |
        +----------------------------------+
        |                                  |
        |             GI8912               |
        |           register 15            |
        |                                  |
        +----------------------------------+
```

```
LOOP    LDX #$03        ; Check if E is pressed.
        LDY #$06        ; Matrix location column 3, row 6
        JSR GETXY       ;
        BEQ TESTQ       ; If E not pressed, go and check Q
        LDX #$45        ; Print an 'E' on the screen.
        JSR $F77C       ; Oric-1 routine is at $F73F
        JMP LOOP        ;
                        ;
TESTQ   LDX #$06        ; If Q is pressed, then leave routine.
        LDY #$01        ; Matrix location column 6, row 1
        JSR GETXY       ;
        BEQ LOOP        ; Not pressed - loop back up
        RTS             ; End
                        ;
GETXY   LDA #$40        ; Routine checks column held in X, row held in Y.
        STA $030E       ; Turn 6522 interrupts off
        TYA             ;
        PHA             ; save row number
        LDA #$FF        ; set all bits to 1
        CLC             ;
        ROL             ; set least significant bit to 0
SETX    DEX             ;
        BMI DONE        ; move the 0 bit along by X bits
        ROL             ;
        BNE SETX        ;
DONE    TAX             ; write column to GI8912
        LDA #$0E        ;
        JSR $F590       ; Oric-1 routine is at $F535
        PLA             ; restore row number
        ORA #$B8        ; OR accumulator with usual value in 6522 port B
        STA $0300       ; send to port B
        LDY #$04        ;
DELAY   DEY             ; wait while keyboard pcb deals with signal
        BNE DELAY       ;
        LDA $0300       ; see if key pressed
        AND #$08        ; If so, bit 3 is set
        PHP             ; save flags
        LDA #$C0        ; restore interrupts
        STA $030E       ;
        PLP             ; restore flags
        RTS             ; return with Z clear if key is pressed
```