QVIEW CAPS-LED

```
10  REMark Caps Led Code 81988 Jonathan
Oakley
100 b=RESPR(128)
110 RESTORE 240:check=0
120 FOR i=0 TO [illegible]127
130   READ byte:check=check+byte
140   POKE b+i,byte
150 END FOR i
160 READ checksum
170 IF checksum<>check
180     PRINT #0;'Checksum error - check
[illegible]e typing!'
190 ELSE
200    CALL b
210    PRINT #0;'CAPSLED code installed'
211    INPUT #0;'Enter device to SBYTES to
>';dev$
212    PRINT #0;'Saving
';dev$;'CAPSLED_bin'
215    SBYTES dev$&'CAPSLED_bin',b,128
220 END IF
230 STOP
240 DATA 114,32,116,255,112,24,78,65
250 DATA 74,128,102,26,40,72,65,236
260 DATA 0,8,67,250,0,22,41,73
270 DATA 0,12,112,28,78,65,41,124
280 DATA 76,69,68,37,0,16,74,128
290 DATA 78,117,32,43,0,20,103,8
300 DATA 97,2,96,42,47,0,78,117
310 DATA 74,46,0,51,103,18,83,107
320 DATA 0,2,110,58,8,83,0,0
330 DATA 55,124,0,25,0,2,96,14
340 DATA 66,107,0,2,48,46,0,136
350 DATA 227,88,70,0,22,128,48,19
360 DATA 176,107,0,4,103,24,55,64
370 DATA 0,4,16,60,0,1,63,0
380 DATA 66,167,63,60,12,1,38,79
390 DATA 112,17,78,65,80,143,78,117
400 DATA 7727
```

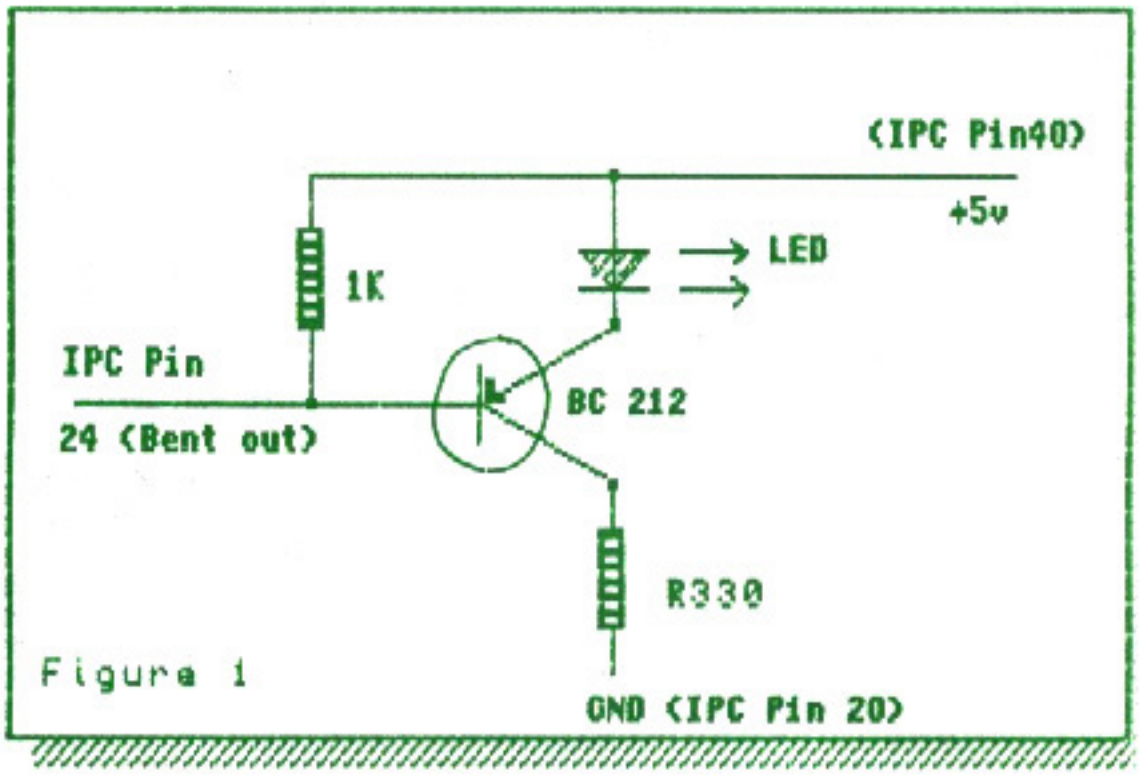

Figure 1

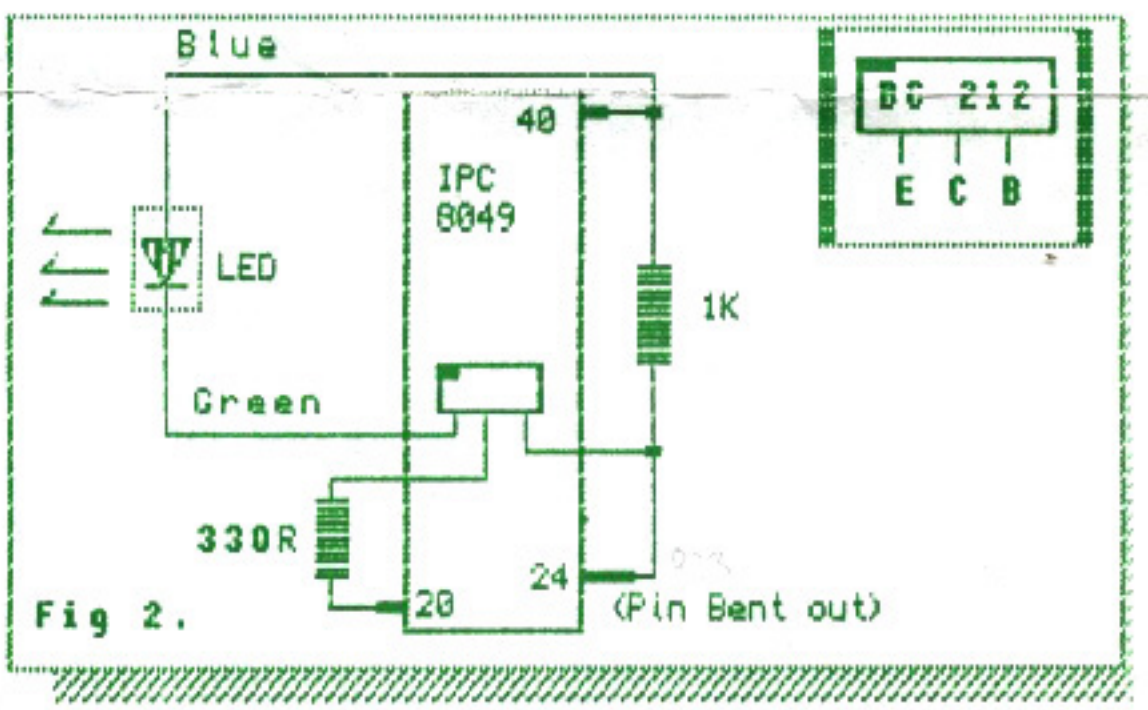

Fig 2.

This code is called on the polling interrupt, and sets the "capslock" LED state according to the state of the capslock and screen freeze system variables. If, however, a user-defined LED status routine is supplied then this is called instead. The user routine may be inserted by searching the polling list until an entry containing the characters "LED%" at an offset of $10 from linkage block start ($08 from the polling link) is found. The user routine address can then be filled in at offset $14. There are two unused long words in the linkage block following this which may be used as required by the user routine. The user routine may unlink itself by clearing the user routine pointer to 0.

CAPSLED KIT FITTING INSTRUCTIONS:

Fitting the CAPSLED modification should not be beyond the ability of anyone who has fitted a keyboard membrane or an internal memory expansion to their QL. The process should be carried out slowly & carefully - there is much to be said for the old adage " more haste less speed!". We estimate it will take approximately an hour.

****

WARNING! Read these instructions thoroughly - incorrect fitting of your CAPSLED kit may damage your IPC 8049 at worst or destroy the BC212 transistor supplied - although we have taken great care in the preparation of these instructions we will not be liable for any direct, indirect or consequential loss or damage which may arise from any error, defect or failure of this kit or instructions.

****

Tools required are : crosshead screwdriver to open the QL casing, low power fine point soldering iron & low melting point solder, hand drill with a small drill bit, small needle file & possibly some form of chip extraction tool (the one supplied with QJUMP's QIMI is ideal) to minimise damage to the 8049 processor when it is removed.

1/ Open the QL casing by removing the screws along the front and back rows of the machine - remember not to undo the screws under the microdrive area otherwise you will end up with free form microdrives!

2/ Remove the keyboard membrane connectors from their sockets by gently easing them out. Undo the screws in the metal backing plate to the keyboard membrane, the membrane can remain stuck to the backing plate. Remove the bubble mat which lies on the keys.

3/ You now have to decide where to site the LED for your capslock indicator - the obvious place is the channel between the Caps Lock key and the F3 function key. Using a 3mm drill bit and a hand drill (NO not the Black & Decker hammer drill !!) make the hole for your LED. This part of the operation is worth taking time to make sure that a tidy hole is made. When the LED is securely in place, solder the GREEN wire to the cathode (the shorter lead) and the BLUE wire to the other lead.

4/ Bend the pins flat against the keyboard and route the wires round the key mouldings up to the channel at the bottom of the keyboard where the other leads run. Check that the wires run freely between the keys without fouling & you can now replace the bubble mat & keyboard membrane with its backing plate.

5/ Gently ease out the 8049 processor - it is the 40 pin IC to the left of MDV1_. If possible place it on a piece of static reducing foam (you should have saved the piece that came with your QIMI shouldn't you!). Keep it the same way round as it came out of the machine so that pin 1 is the top left hand corner. Bend up pin 24 of the IPC - we don't recommend you cut anything off it in case you should want to put it back later if you get bored with your CAPSLED! Solder the 1K resistor (colour coded Brown Black Red) between pins 40 and 24 as in figure two. Whilst it is possible to solder the components to the IPC while in its socket in the QL, we do not recommend this as there is a danger of soldering the chip into the socket!! The usual precautions should be observed when soldering to heat sensitive devices - use a low power soldering iron, low melting point solder & keep contact at the minimum required to melt the solder.

6/ Place the transistor on the middle of the IPC with the flat side facing you and the legs pointing down the IPC. From left to right the legs are then 1>Emitter, 2>Collector, 3>Base. Bend leg 3 to the right and solder it to the wire between the 1K resistor and pin 24 of the IPC. Solder one end of the 330 ohm resistor to pin 20 of the IPC and the other end to leg 2 of the transistor. This a leg of the transistor to be soldered to the GREEN wire from the LED. The BLUE wire should be soldered to the wire between the 1K resistor and pin 40.

7/ Arrange the components so that no short circuits are likely and the components lie close to the top of the IPC. Carefully re-insert the IPC into its socket, pressing it gently but firmly home. You can now ease the keyboard membrane tails back into their sockets and replace the keyboard.

You can now LRUN the following BASIC loader which will install the code into the system & save a binary file out for future use. If your LED now responds to CAPS & CONTROL F5 then all is working correctly - any failure may mean a bad joint recheck for dry joints or shorts. To enable the CAPSLED system in future you should 'LRESPR' or 'LBYTES & CALL' the CAPSLED_bin code in your BOOT file.

```
* Capslock LED code                          v0.01 Nov 1988  J.R.Oakley
*
* This code is called on the polling interrupt, and sets the "capslock" LED
* state according to the state of the capslock and screen freeze system
* variables.  If, however, a user-defined LED status routine is supplied then
* this is called instead.  The user routine may be inserted by searching the
* polling list until an entry containing the characters "LED%" at an offset
* of $10 from linkage block start ($08 from the polling link) is found.  The
* user routine address can then be filled in at offset $14.  There are two
* unused long words in the linkage block following this which may be used as
* required by the user routine.  The user routine may unlink itself by clearing
* the user routine pointer to 0.
*=============================================================================*
*                                                                             *
* User LED routine - called every polling interrupt                           *
*                                                                             *
* Entry:                                                                      *
*       A3              ^ linkage block                                       *
*                                                                             *
* Exit:                                                                       *
*       A3              preserved                                             *
*       LED_CMD(A3).b   0 if LED is to be lit, 1 if extinguished              *
*       All other registers preserved.                                        *
*=============================================================================*
*
        section capsled
*
        include 'flp1_keys_iod'
        include 'flp1_keys_sys'
        include 'flp1_qdos_sms'
*
led_cmd   equ     0
led.bit   equ     0
led_flct  equ     2
led.flct  equ     25
led_ref   equ     4
* iod_pllk        8
* iod_plad        c
led_flag  equ     $10
led.flag  equ     'LED%'
led_code  equ     $14
led_ext1  equ     $18
led_ext2  equ     $1c
*
led.blkl  equ      $20
*
start
        moveq   #led.blkl,d1                ; this much space
        moveq   #myself,d2                  ; for me
        moveq   #sms.achp,d0                ; get it from the heap
        trap    #do.sms2
        tst.l   d0
        bne.s   exit                        ; ...oops
*
        move.l  a0,a4                       ; keep linkage safe
        lea     iod_pllk(a4),a0
        lea     pollint(pc),a1              ; there's a polling routine
        move.l  a1,iod_plad(a4)             ; point to it
        moveq   #sms.lpol,d0
        trap    #do.sms2
*
        move.l  #led.flag,led_flag(a4)      ; flag for external users
*
exit
        tst.l   d0
        rts
*
pollint
        move.l  led_code(a3),d0             ; is there some user code?
```

```
        moveq   £sms.lpol,d0
        trap    £do.sms2
*
        move.l  £led.flag,led_flag(a4)  ; flag for external users
*
exit
        tst.l   d0
        rts
*
pollint
        move.l  led_code(a3),d0         ; is there some user code?
        beq.s   chkfrz                  ; no
        bsr.s   calluser                ; yes, call the code
        bra.s   sendcmd
calluser
        move.l  d0,-(sp)
        rts
*
chkfrz
        tst.b   sys_dfrz(a6)            ; is display frozen?
        beq.s   tstcaps                 ; no
        subq.w  £1,led_flct(a3)         ; time to toggle flash?
        bgt.s   exc1                    ; no
        bchg    £led..bit,led_cmd(a3)   ; yes, flip state
        move.w  £led.flct,led_flct(a3)  ; and reset timer
        bra.s   sendcmd
*
tstcaps
        clr.w   led_flct(a3)            ; flash instantly on screen freeze
        move.w  sys_caps(a6),d0         ; get capslock state
        rol.w   £1,d0                   ; make bottom bit correct
        not.b   d0
        move.b  d0,led_cmd(a3)          ; and store it
*
sendcmd
        move.w  led_cmd(a3),d0          ; get command
        cmp.w   led_ref(a3),d0          ; state changed?
        beq.s   exc1                    ; no, don't do IPC command

        move.w  d0,led_ref(a3)          ; copy this command
        move.b  £1,d0                   ; no reply
        move.w  d0,-(sp)                ; store parameter and reply
        clr.l   -(sp)
        move.w  £$0c01,-(sp)            ; "reduce sensitivity", one parameter
        move.l  sp,a3
        moveq   £sms.hdop,d0
        trap    £do.sms2
        addq.l  £8,sp                   ; pop IPC command
exc1
        rts
*
        end
```