A 32 X 48 DISPLAY EXPANSION KIT FOR THE UK101.

SWANLEY ELECTRONICS, 32 Goldsel Rd., Swanley, Kent BR8 8EZ.

Introduction.

These step by step instructions describe a simple modification to the UK101's video circuitry allowing double the usual number of lines to be displayed. This is achieved by halving the height of all characters and graphics, and providing an extra 1K of VDU memory, giving a total of 2K of VDU ram from 53248 to 55295 dec (D000 to D7FF hex).

The actual modification requires 11 breaks to be made to the PCB tracks and the insertion of 14 wire links between various IC pads. The only components required are two extra 2114 ram chips. As long as the instructions are carefully carried out and all the work double checked no difficulties should be encountered. It is however reccommended that only those with a reasonable degree of proficiency in the use of a soldering iron should attempt this conversion. Those who have built their own UK101's from kits are definetly qualified!

Breaks in the PCB tracks can be made easily using a sharp knife or file. Wire links should be made with insulated solid core connecting wire, or better still wire-wrap wire.

The following convention has been used to make the instructions clearer:-

U55,1 means pin 1 of IC 55.
PT means pcb track.
RVS means the reverse side of the pcb.
CS means component side.

Principles.

The formerly unused C7 output of the binary counting chain (which caused all VDU dot rows to be displayed twice) is now included in the VDU timing chain. C14 is used as a chip select signal for the second 1K block of VDU ram to enable the bottom half of the screen. All characters are now half their original height, allowing twice the number of character rows to be displayed. This involves 4 main steps:-

1. C7 is included in the VDU counting chain and all the row clock outputs are moved one step down the chain, i.e. C7 goes to U41,8 instead of C8; C8 goes to U41,7 instead of C9 etc.

2. "NOT A11" is connected to U20,1 and U20,2 allowing the VDU to occupy 2K of the memory map. It is also necessary to connect U56,2 to +5V for the VDU to be decoded as a 2K block.

3. A10 is connected to U55,11 and C14 to U55,10. This enables the top 1K of the VDU ram.

4. "NOT A10" is connected to U55,14 and "NOT C14" to U55,13. "NOT C14" is derived from C14 using a spare inverter in U18. U55,12 is connected to pins 8 of the 2 new ram chips. This enables the bottom 1k of the VDU ram.

Construction.

1. Connect U18,9 to U30,13.
2. Isolate U20,1 by cutting the PT on the RVS.
3. Connect U20,1 to U20,2.
4. Isolate U30,14 by cutting the PT on the CS.
5. Connect U30,14 to U55,6.
6. Isolate U30,13 by cutting the PT on the RVS.
7. Isolate U55,11 by cutting the PT on the CS.
8. Connect U55,11 to U21,9.
9. Isolate U55,10 by cutting the PT on the RVS.
10. Connect U55,10 to U30,13.
11. Connect U55,14 to U21,8.
12. Connect U55,13 to U18,8.
13. Isolate U56,2 by cutting the PT on the CS.
14. Connect U56,2 to +5V.
15. Connect U60,12 to U41,8.
16. Isolate U60,11 by cutting the PT on the CS.
17. Connect U60,11 to U41,7.
18. Isolate U61,14 by cutting the PT on the RVS.
19. Connect U61,14 to U41,6.
20. Isolate U61,13 by cutting the PT on the CS.
21. Connect U61,13 to U54,10.
22. Isolate U61,11 by cutting the PT on the CS.
23. Isolate U61,12 by cutting the PT on the CS.
24. Connect U61,12 to U54,13.
25. Connect U61,11 to U55,3.
26. Make up two composites each of two 2114L's by "piggy-backing". Solder all pins except pins 8 to the one beneath. i.e. pin 1 to pin 1, pin 2 to pin 2 etc. Lift pin 8 on both of the upper IC's and connect them to U55,12. This completes the modification.
27. To test the screen run the following program:-

```
10 FOR X=53248 TO 55295: POKE X,32:NEXT
```

This should result in a complete screen clear.

Software Modifications.

After the hardware modifications described above you will find that only the top half of the screen is used by Basic and by PRINT statements. You can however use the lower half as well with POKE commands or machine code instructions. This section deals with modifications to the software of the computer which permit Basic and PRINT statements to use all 32 lines of the screen. The most satisfactory method of doing this is to replace the existing monitor rom with a special version of the Cegmon monitor rom which we stock for the enhanced UK101 at a price of £22-50 + 50p post + 15% VAT.

A cheaper alternative is to use the software patch listed on the attached sheet. This patch is best loaded into the UK101 via a basic program using read and data statements. The patch has the advantage that being in ram, the cursor's starting position, the line length and the like may all be user specified, so as to allow for varying degrees of overscan on your system. The UK101 screen handling routine looks at locations 021A and 021B to pick up its output vector to FF69 for the normal screen handling. This has to be changed to the start of the patch at 0222 in order for the patch to take over from the built-in subroutine. This vector is reset to point to FF69 after "break" and so must be reset to point to 0222 after any pressing of the "break" as well as after a cold start. An easy way of doing this is by POKEs in direct mode:- POKE 538,34: POKE 539,02. But note that in direct mode these must be on the same line.

The following notes apply to the software patch:-

1. F9 is the maximum permitted displacement of the cursor position before a carriage return is forced. It is equal to the cursor starting position + number of characters per line - 1.
2. BF here is the end of the line above the cursor line. It is equal to the cursor starting position - number of characters lost in the left overscan.
3. This note is a technicality and should be ignored.
4. C8 here is the cursor start position allowing for eight characters overscan.

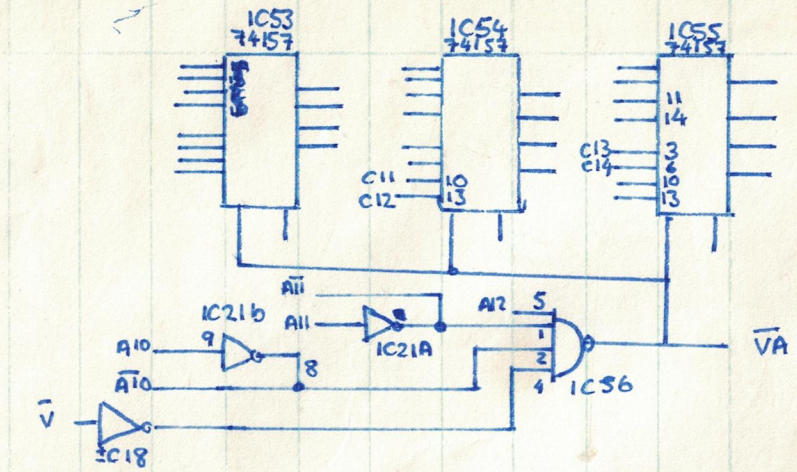
The above 3 addresses may have to be changed to suit your TV or monitor. It is understood in the above that the most significant byte of all these addresses is D7. i.e. the full addresses in the above are D7F9, D7BF and D7C8.

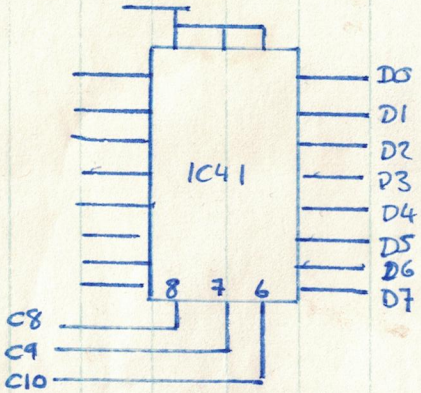
```
0222  8D 02 02  STA 0202   Copy of BF2D routine
0225  48        PHA
0226  8A        TXA
0227  48        PHA
0228  98        TYA
0229  48        PHA        All registers saved on stack
022A  AD 02 02  LDA 0202   Retrieve A from parking space
022D  F0 4C     BEQ 4C     Return if null
022F  AC 06 02  LDY 0206   Start of delay routine:
0232  F0 08     BEQ 08     picks up delay loop counter from 518(10);
0234  A2 40     LDX 40     larger value gives longer delay
0236  CA        DEX
0237  D0 FD     BNE FD
0239  88        DEY
023A  D0 F8     BNE F8     End of delay
023C  C9 0A     CMP 0A     Linefeed?
023E  F0 46     BEQ 46     if yes, jump to 0286 for line-feed

0240  C9 01     CMP 01     Screen clear?
0242  D0 1A     BNE 1A     if no, jump over screen clear routine
0244  A9 20     LDA 20     Screen clear routine
0246  A0 08     LDY 08
0248  A2 00     LDX 00
024A  9D 00 D0  STA D000, X
024D  E8        INX
024E  D0 FA     BNE FA     Load Dn00+X, 20(16) until one 'page' is done
0250  EE 4C 02  INC 024C   Dn00=D(n+1)00 - restart one 'page' down
0253  88        DEY
0254  D0 F4     BNE F4     Do until all eight 'pages' of screen are done
0256  A9 D0     LDA D0
0258  8D 4C 02  STA 024C   Restore routine to start value - D000
025B  4C 7B 02  JMP 027B   and exit
025E  C9 0D     CMP 0D     Carriage return?
0260  D0 06     BNE 06     if no, jump over carriage-return routine
0262  20 D2 02  JSR 02D2   Else do carriage-return
0265  4C 7B 02  JMP 027B   and exit
0268  8D 01 02  STA 0201   If A not above, save it at 0201
026B  20 C8 02  JSR 02C8   and print it
026E  EE 00 02  INC 0200   Increment cursor index
0271  A9 F9     LDA F9     Maximum for cursor index (see Note 1)
0273  CD 00 02  CMP 0200
0276  30 0B     BMI 0B     Do CR/LF if greater than maximum
0278  20 DA 02  JSR 02DA   Else print cursor
027B  68        PLA        retrieve registers
027C  A8        TAY
027D  68        PLA
027E  AA        TAX
027F  68        PLA
0280  4C 6C FF  JMP FF6C   and exit back to BASIC
```

```
0283  20 D5 02  JSR 02D5   Save next char. position - for backspace?
0286  20 C8 02  JSR 02C8
0289  A9 BF     LDA BF     Nominal cursor start - see Note 2
028B  EA        NOP
028C  EA        NOP
028D  8D 02 02  STA 0202
0290  A2 07     LDX 07     Pick up scroll routine from BFF3(16)
0292  BD F3 BF  LDA BF3, X
0295  9D 07 02  STA 0207, X and store at 0207(16) to 020E(16)
0298  CA        DEX
0299  10 F7     BPL F7
029B  A2 D7     LDX D7     X defines the bottom of the main scroll;
029D  A9 40     LDA 40     40(16) here tells the routine to transfer the
029F  8D 08 02  STA 0208   char. 40(16) 'down' to the current Dnnn
02A2  A0 00     LDY 00     and do scroll
02A4  20 07 02  JSR 0207
02A7  D0 FB     BNE FB
02A9  EE 09 02  INC 0209   move down after first four lines done

02AC  EE 0C 02  INC 020C
02AF  EC 09 02  CPX 0209   Down to D7nn yet?
02B2  D0 F0     BNE F0     If not, go back for another 4-line 'page'
02B4  20 07 02  JSR 0207
02B7  CC 02 02  CPY 0202   Do until nominal cursor start - see Note 2
02BA  D0 F8     BNE F8
02BC  A9 20     LDA 20     Clear text entry line
02BE  20 0A 02  JSR 020A
02C1  CE 08 02  DEC 0208
02C4  D0 F8     BNE F8
02C6  F0 AE     BEQ AE     and jump back to exit-to-BASIC
02C8  AE 00 02  LDX 0200   0200 stores current cursor position
02CB  AD 01 02  LDA 0201   0201 stores character to be printed
02CE  9D 00 D7  STA D700, X Place A on screen
02D1  60        RTS
02D2  20 C8 02  JSR 02C8   Place A on screen - see Note 3
02D5  A9 C8     LDA C8     Carriage return - C8 is cursor start - Note 4
02D7  8D 00 02  STA 0200
02DA  AE 00 02  LDX 0200   Enter here to save next character on
02DD  BD 00 D7  LDA D700, X Could be used for backspace?
02E0  8D 01 02  STA 0201
02E3  A9 5F     LDA 5F     5F is cursor - jump back to print cursor
02E5  D0 E7     BNE E7     at next character location on.
```





Link

1	U18,9 U30,13
2	U20,1 (I) U20,2
3	U30,14 (I) U55,6
4	U30,13 (I)
5	U55,11 (I) U21,9
6	U55,10 (I) U30,13
	U5
7	U55,14 U21,8
8	U55,13 U18,8
9	U56,2 (I) +5V
10	U60,12 U41,8
11	U60,11 (I) U41,7
12	U61,14 (I) U41,6
13	U61,13 (I) U54,10
14	U61,11 (I) U55,3
15	U61,12 (I) U54,13