6502 Bus Connector

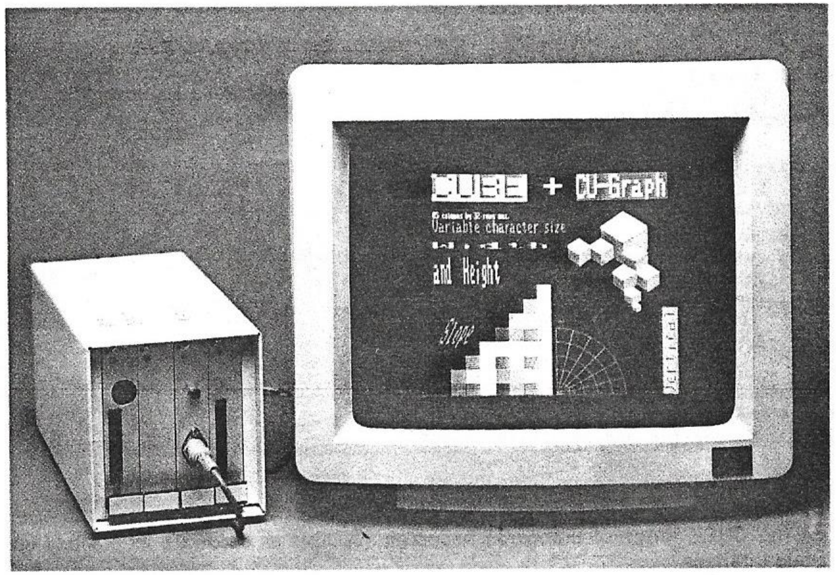
| B |  | A |
|---|---|---|
| +5V | 1 | +5V |
| +12V | 2 | A15 |
| -12V | 3 | A14 |
| -5V | 4 | NWDS |
| +15V | 5 | NRDS |
| -15V | 6 | RESET |
| HPAGE | 7 | A8 |
| A23 | 8 | A7 |
| A22 | 9 | A6 |
| A21 | 10 | A5 |
| A20 | 11 | A4 |
| A19 | 12 | A3 |
| A18 | 13 | A2 |
| A17 | 14 | A1 |
| A16 | 15 | A0 |
| D15 | 16 | D7 |
| D14 | 17 | D6 |
| D13 | 18 | D5 |
| D12 | 19 | D4 |
| D11 | 20 | D3 |
| D10 | 21 | D2 |
| D9 | 22 | D1 |
| D8 | 23 | D0 |
| n/c | 24 | A13 |
| n/c | 25 | A12 |
| n/c | 26 | A11 |
| RDY | 27 | A10 |
| IRQ | 28 | A9 |
| NMI | 29 | phase 2 clock |
| SYNC | 30 | R/W |
| LPAGE | 31 | BLK |
| AGND | 32 | DGND |

**Control Universal Ltd**
Manufacturers of the CUBE Range of
Industrial Microcomputer Systems
Andersons Court
Newnham Road
Cambridge CB3 9EZ
Tel: Cambridge (0223) 358757
Telex 995801 GLOTX-G

# CU-GRAPH PERIPHERAL DRIVER: ISSUES 1 & 2
# AND CUBE HI-RES UNIT

1984

## Notes for Users (provisional)



**Control Universal Ltd**
Manufacturers of the CUBE Range of
Industrial Microcomputer Systems
Andersons Court
Newnham Road
Cambridge CB3 9EZ
Tel: Cambridge (0223) 358757
Telex 995801 GLOTX-G

## CONTENTS

page

## DEMONSTRATION EXAMPLES

In the following examples, it is assumed that the user's system is connected to the graphics terminal via the serial port and that it is appropriately cqnfigured (see 'Baud Rate Select', p.32).

### Command Codes

ASCII codes 0 to 31 are non-printing characters, and these are interpreted by the peripheral driver as command codes. The effects of these codes are the same as those produced by the VDU commands in CU-GRAPH. For example:

```
1. (Hex)      11,3    has the same effect as
        VDU &11,3    namely to change the text colour to yellow

2. (Hex)       C      has the same effect as
        VDU 12       i.e. clear text area

3. Print your name:

   (Hex)  44  41  56  45  0D  0A
          ----------------------
          D   A   V   E   CR  LF
```

4. To draw a white line diagonally across the screen, the following hex bytes must be sent:

```
   (Hex)  19,5,0,4,0,4   corresponding to

   VDU  &19,5,0,4,0,4
```

### Using the HI-RES UNIT with the BBC Micro

When connected to the BBC Micro's serial port, the HI-RES UNIT can be driven directly from BBC BASIC. The BASIC commands convert the high-level statements into the appropriate VDU commands. For example:

```
*FX3,7    Output channel to serial port
PRINT "DAVE"
COLOUR 3
PRINT "DAVE"
MOVE 0,0
DRAW 1024,1024
*FX3
```

This example program produces exactly the same effect as the VDU-coded example above.

NOTE: Cursor control keys are not active in this mode of operation.

2. Baud Rate Select

The baud rate is defined by four inputs on the EuroCUBE front connector. PA0 to PA3 can be wire-wrapped to either Pin 25 or Pin 26 to select the different baud rates.

| | PA3 | PA2 | PA1 | PA0 | baud rate |
|---|---|---|---|---|---|
| binary | 0 | 0 | 0 | 1 | 50 |
| | 0 | 0 | 1 | 0 | 75 |
| | 0 | 0 | 1 | 1 | 109.92 |
| | 0 | 1 | 0 | 0 | 134.58 |
| | 0 | 1 | 0 | 1 | 150 |
| | 0 | 1 | 1 | 0 | 300 |
| | 0 | 1 | 1 | 1 | 600 |
| | 1 | 0 | 0 | 0 | 1200 |
| | 1 | 0 | 0 | 1 | 1800 |
| | 1 | 0 | 1 | 0 | 2400 |
| | 1 | 0 | 1 | 1 | 3600 |
| | 1 | 1 | 0 | 0 | 4800 |
| | 1 | 1 | 0 | 1 | 7200 |
| | 1 | 1 | 1 | 0 | 9600 |
| | 1 | 1 | 1 | 1 | default 9600 baud maximum |

As supplied, Port A is left open circuit - i.e. PA0 to PA3 = 1111 (9600 baud).

The serial data has one start bit, eight data bits, one stop bit, and no parity. The Receiver Ready signal (RxREADY) is a handshaking signal, and should be connected to Clear to Send (CTS) on the driving computer. This ensures that no data is missed at high data rates.

The Peripheral Driver Concept

The EuroCUBE-65 CPU can drive a number of peripherals, thus giving the user a wide choice for his applications. Examples of popular peripherals include CU-GRAPH, a high-resolution graphics display, or Teletext, a simpler text display with chunky graphics, combined with CU-KEY, a QWERTY keyboard. Although this flexibility is desirable from the user's point of view, it poses problems for the designer of the machine operating software. To cater for all the possible hardware combinations, the same number of combinations would have to be provided for in the operating system, which would then require more EPROM space than there is available. This problem can be overcome by standardising the operating system software for the CPU card and separating out the software modules controlling the individual peripherals. These software modules are known as peripheral drivers.

In practice, this means that the EuroCUBE-65 operating system (MOS-B.2) has been standardised for all variants of EuroBEEB/EuroCUBE-65. It occupies the top 4K of the 8K MOS-B.2 EPROM, while the remaining 4K is reserved for the peripheral driver options specified by the user. The following are standard options:

1) Machine Code Monitor

2) CU-GRAPH + CU-KEY 53

3) Teletext + CU-KEY 53

4) VIEWLINE + RACK-PRINT + CU-KEY 25

All versions include drivers for the CUBAN-8 and CUBAN-12 ADC boards, invoked by the ADVAL(x) command in BASIC.

The operating system and the peripheral drivers together provide the interface between the language (in this case BBC BASIC) and the hardware present. The advantage for the user is that he can write his application software in a manner that is largely independent of the hardware.

## CU-GRAPH: Summary of the Main Features

CU-GRAPH is best summarised in a comparison with the BBC micro display.

CU-GRAPH is a video interface for applications requiring high-resolution graphics output. Its highest resolution Mode is similar to Mode 0 of the BBC micro but has the following advantages:

1. All eight colours can be used in CU-GRAPH Mode 0 - only two can be used in BBC Mode 0.

2. None of the 48K of video memory occupies space in the microprocessor memory map, so that over 28K of RAM can be made available to the user in any mode if 32K of RAM is provided by the system.

3. Drawing commands work faster on CU-GRAPH.

CU-GRAPH provides no direct access to the video RAM, so that there is no facility for animation. However, CU-GRAPH is designed as a video display, not as a games display. Its main function is to act as a high-resolution graphics display or as an end-application video card for displaying process control.

CU-GRAPH is easy to use because the BASIC graphics commands are implemented. The user need only write his application programs. The software interface emulates the BBC VDU commands as closely as possible. Although there are minor differences in the commands, the major commands are the same, e.g. MOVE and DRAW. The screen co-ordinates are also similar (1023 x 1023).

The main differences between the BBC micro and CU-GRAPH are:

1. No windows are possible in CU-GRAPH. Scroll is achieved by software.

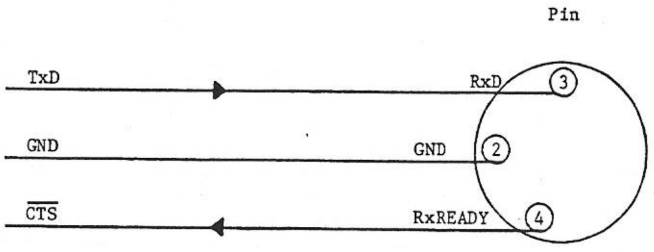
2. CU-GRAPH has a character font of 8 x 6, compared with the BBC Micro's 8 x 8. In Mode 0 there are 512 x 256 pixels on CU-GRAPH and 640 x 256 pixels on the BBC. CU-GRAPH is capable of giving 85 columns x 32 lines, but provides 80 x 32 with EuroBEEB for Mode 0 compatibility on the BBC.

## Serial Connection and Baud Rate Select

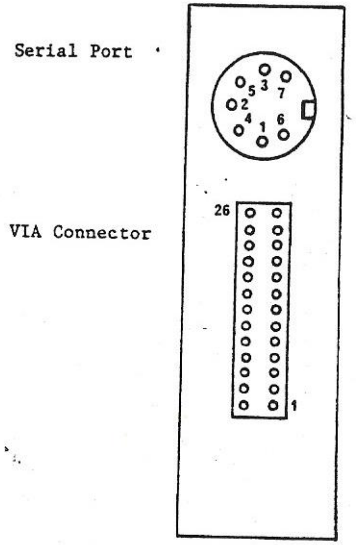
The serial connector supplied is configured for RS-423 (RS-232C compatible) operation. Users requiring RS-422 operation should consult the EuroCUBE-65 User Manual.

1. Serial connection required to run the CUBE HI-RES UNIT (uni-directional communication)

User (e.g. BBC Micro) EuroCUBE-65



NOTE: Pins 6 and 7 on the serial connector are only required for RS-422 communication. Pins 1 and 5 are used for TxD and CTS input respectively.



EuroCUBE-65
front view

VIA Connector

| | | | |
|---|---|---|---|
| GND | 26 | 25 | GND |
| n/c | 24 | 23 | RESET |
| CA1 | 22 | 21 | CA2 |
| PA0 | 20 | 19 | PA1 |
| PA2 | 18 | 17 | PA3 |
| PA4 | 16 | 15 | PA5 |
| PA6 | 14 | 13 | PA7 |
| PB0 | 12 | 11 | PB1 |
| PB2 | 10 | 9 | PB3 |
| PB4 | 8 | 7 | PB5 |
| PB6 | 6 | 5 | PB7 |
| CB1 | 4 | 3 | CB2 |
| +5V | 2 | 1 | +5V |

## CUBE HI-RES UNIT

### INTRODUCTION

The CUBE HI-RES UNIT is a high-resolution graphics display terminal available both in monochrome and in full colour. The monochrome version consists of a 6502 EuroCUBE (CPU) and the CU-GRAPH monochrome card (a graphics VDU interface), both housed in a four-slot mini-rack with built-in power supply. The colour version adds the CU-GRAPH colour extension card with a Centronics parallel printer port. See p. 2 for further details of CU-GRAPH.

The CPU card, EuroCUBE-65, is supplied with 8K CMOS RAM and a version of the MOS-B.2 Operating System incorporating Issue 2 of the CU-GRAPH peripheral driver (see p. 4). Since the HI-RES UNIT is a stand-alone terminal device, BBC BASIC is NOT provided on board. The peripheral driver is able to recognise whether or not BASIC is present and act accordingly.

The CUBE HI-RES UNIT can be used with any RS-423 or RS-232 system, e.g. BBC Micro, EPSON HX-20, etc.

### OPERATION

See p. 3 for a description of the setting-up procedure.

At switch-on the operating system will check for the presence of BBC BASIC. If this is fitted, the unit will act as a standard EuroBEEB with the CU-GRAPH peripheral driver (see p. 3). Otherwise, the following message will be displayed:

```
CU-graph Graphics Terminal

_ (cursor)
```

and the unit will wait for serial data to be transmitted through its serial connector.

## SETTING UP THE SYSTEM: EuroBEEB-3

1. Turn off the power to the equipment.

2. Install the CUBE 'sideways' ROM in the BBC Micro. To do this, you will need to remove the lid of the BBC Micro and the two bolts holding the keyboard in place. The 'sideways' ROM should be plugged into one of the spare ROM sockets underneath the BBC keyboard. The BBC BASIC ROM should occupy the ROM socket on the far right, while the CUBE 'sideways' ROM may be plugged into any of the spare sockets to its left. Refit the keyboard and lid.

3. Check that the correct version of MOS-B.2 with the appropriate peripheral drivers is installed in socket M0 of the EuroBEEB card.

4. Check that the address decoding switches on CU-GRAPH are set to BLOCK D PAGE F (&DF00).

5. If the rack uses a buffered backplane (8",12" and 19" rack), make sure that the EuroBEEB card is in the left-hand slot. The CU-GRAPH card(s) may occupy any of the remaining slots.

6. Plug the 5-pin DIN plug of the serial lead into the 7-pin socket of the EuroBEEB front panel. The domino plug at the other end of the serial lead should be plugged into the RS-423 socket at the back of the BBC micro. Make sure that the cut-out in the metal surround of the domino plug is pointing upwards at the BBC end.

7. Connect the CU-GRAPH video lead to a monitor, using a mono cable and plug if you have CU-GRAPH monochrome, or an RGB cable and plug if you have CU-GRAPH colour.

   NOTE: It is convenient to use a separate monitor connected to CU-GRAPH when developing your application program. We therefore recommend that you use the BBC micro and its monitor to develop programs and then test the programs on a second monitor connected to CU-GRAPH. However, bear in mind that the BBC micro and its monitor are development tools which will not be present in the final application.

8. Connect the BBC micro to a monitor and disk drive(s) in the normal way and switch on the equipment.

If you now type *HELP <RETURN>, you should see

```
EURO 0.1
FLEX 0.1
```

on the screen. Now type *EURO <RETURN> and you should see:

```
Euro-Beeb Terminal File Server Iss.1**
shift f0 returns to BBC computer
shift f9 sends RESET to Euro-Beeb
```

**NOTE
If you have Issue 2 of the *EURO 'sideways' ROM, the message will read:

```
Euro-BEEB Terminal File Server Iss.2
```

If you press <shift f0>, a '*' prompt will appear on the screen. Type in BASIC or B. to return to the BBC Micro. Type <shift f9> to send a RESET to the EuroBEEB.

If you then power up your system rack you should see the message

```
MOSB.2
BASIC
>
```

on the screen. The second monitor attached to CU-GRAPH should now have cleared its screen and positioned the cursor in the top left-hand corner.

If you type *HELP <RETURN> again, you should see on the screen a list of the MOS and peripheral drivers present, together with their issues numbers. In the case of the CU-GRAPH peripheral driver, the message on the screen should be

```
CU-graph
CU-key
MOSB.2
```

If this message fails to appear, you either have the wrong operating system EPROM, or CU-GRAPH is installed at the wrong address.

If the message reads

```
CU-graph.2
etc.
```

you have Issue 2 of the CU-GRAPH PD which became available in September 1984. The new issue enables CU-GRAPH to be used in stand-alone high-resolution video terminal applications without BBC BASIC being present (CUBE HI-RES UNIT) and also includes some enhancements over Issue 1. The features and commands relating to Issue 2 are described in a separate section (see pp.13 ff).

Now type:

```
>PRINT HIMEM <RETURN>
```

If EuroBEEB is fitted with 8K of RAM, HIMEM should equal 8192 (&2000). If 16K of RAM is present, HIMEM should equal 11264 (&2C00). Note that in the latter case the CU-GRAPH peripheral driver has claimed 5K for the scroll buffer.

If 8K of RAM is present, CU-GRAPH will automatically go into PAGE mode (i.e. it will not scroll). If 16K of RAM is present, it will automatically go into SCROLL mode.

Location &300 contains the PAGE start address of the scroll buffer. It is a cold reset variable, i.e. it is set up at power on. If the contents of this location are changed and a MODE command is given, the scroll buffer can be relocated.

```
1160
1170      ====================
1180      DEFPROCblock(C,D,S)
1190      FORcount=1TOS/2 STEP2
1200        GCOL1,C
1210        PLOT 1,0,S
1220        PLOT 0,2,-S
1230        GCOL1,D
1240        PLOT 1,0,S
1250        PLOT 0,2,-S
1260        NEXT
1270      ENDPROC
```

```
 570
 580 MOVE2000,2000
 590 REM Return output to BBC monitor
 600 *FX3,7
 610 END
 620
 630 ===================
 640 DEFPROClogo(H,V,M)
 650 FORC=1TO8
 660   READ X,Y,S:PROCcube(X*M+H,Y*M+V,S*M)
 670   NEXT
 680 ENDPROC
 690 DATA 0,0,24
 700 DATA -24,0,12
 710 DATA -48,0,12
 720 DATA 12,-8,12
 730 DATA 0,-28,12
 740 DATA 24,-28,12
 750 DATA 8,-42,8
 760 DATA 12,-49,4
 770
 780 ===================
 790 DEFPROCcube(X,Y,S)
 800 LOCAL A,Z,H,V
 810 A=1.5
 820 FORZ=Y TO Y+S STEP 4
 830   GCOL0,6:MOVEX,Z:PLOT1,S,S/A
 840   GCOL0,4:MOVEX,Z:PLOT1,-S,S/A
 850   NEXT
 860 GCOL1,7
 870 FORZ=0TO S
 880   H=X-Z:V=Y+S+Z/A
 890   MOVE H,V+4:PLOT1,S,S/A
 900   NEXT
 910 ENDPROC
 920
 930 =================
 940 DEFPROCweb(X,Y,R)
 950 LOCAL S
 960 S=R/5
 970 FORrad=S TO R STEP S
 980   MOVEX,Y+rad
 990   FORang=0 TO 90 STEP 5
1000     DRAW X+rad*SIN(RAD(ang)),Y+rad*COS(RAD(ang))
1010     NEXTang,rad
1020   FORang=0TO90STEP15
1030     MOVE X,Y
1040     PLOT 1,R*SIN(RAD(ang)),R*COS(RAD(ang))
1050     NEXT
1060   ENDPROC
1070
1080   ====================
1090   DEFPROCcolblk(X,Y,S)
1100   FORC=0TO7
1110     MOVE X+C*S,Y+C*S
1120     FORD=C TO7
1130       PROCblock(C,D,S)
1140       NEXTD,C
1150     ENDPROC
```

If the scroll buffer is not required, it can be reclaimed and HIMEM moved to the top of RAM as follows:

```
>?&300=&80          move scroll buffer start to &8000
>*FX254,64          tell OS that top of RAM is &4000
SHIFT f9            reset
>VDU14              select PAGE mode
```

This will move the scroll buffer to &8000, and HIMEM should now equal &4000 (16384). Test this by typing as before

```
>PRINT HIMEM
```

NOTE

In order to work as a graphics terminal, CU-GRAPH uses software scroll which works under interrupt. Issue 1 of the CU-GRAPH peripheral driver causes approximately 10 ms interrupt times which means that other interrupts, e.g. the serial port, cannot operate during this period. The net effect is that characters coming down the serial port are missed. In Issue 2 the CU-GRAPH scroll itself can be interrupted so that the serial port can take priority over scrolling.

OPERATION

*FX3,0 switches off the serial output port of EuroBEEB, so that all of the normal output stream is passed to the CU-GRAPH peripheral driver.

*FX3,7 is the default on EuroBEEB. This command can be used to switch the serial port on and the CU-GRAPH peripheral driver off.

When the command *FX3,0 has been typed in, all the PRINT and PLOT commands are passed to the CU-GRAPH screen.

When you have set up the system, try the following program with CU-GRAPH colour:

```
10 *FX3,0
20 PRINT "CU-GRAPH"
30 MOVE 0,0
40 DRAW 1023,1023
50 GCOL 0,2
60 DRAW 1023,0
70 GCOL 0,4
80 DRAW 0,1023
```

This will draw a green line from the bottom left-hand corner to the top right-hand corner of the screen and a blue line from the top right-hand corner to the bottom right-hand corner and up to the top left-hand corner of the screen. You cannot do this on the BBC micro!

NOTE: If you have CU-GRAPH mono, leave out the GCOL statements (lines 50 and 70).

## EFFECT OF THE VDU COMMANDS

The VDU commands are ASCII characters between 0 and 31 which are passed to the output channel. In the table below, 'Variable' refers to the character(s) passed to the output channel immediately following the VDU command.

| VDU | Variable | Comments | BASIC Word |
|---|---|---|---|
| 0 | | Ignores | |
| 1 | ,P | Send P to printer only | |
| 2 | | Turn printer ON | |
| 3 | | Turn printer OFF | |
| 4 | | Write text at text cursor | |
| 5 | | Write text at graphics cursor | |
| 6 | | Turn video ON | |
| 7 | | Make short bleep | |
| 8 | | Backspace | |
| 9 | | Forward space | |
| 10 | | Linefeed | |
| 11 | | Move up one line | |
| 12 | | Clear screen | CLS |
| 13 | | Move cursor to line start | PRINT' |
| 14 | | Page Mode ON | |
| 15 | | Scroll Mode ON (Page Mode OFF) | |
| 16 | | Clear to graphics background colour | CLG |
| 17 | ,C | Define text colour | COLOUR C |
| 18 | ,O,C | Define graphics colour | GCOL 0,C |
| 19 | ,M,S | Define text size and style | |
| 20 | ,M,S | Define graphic text size and style | |
| 21 | | Turn video OFF | |
| 22 | ,M | Select screen mode | MODE M |
| 23 | ,C,Dt,--------,Db | Programmable character | |
| 24 | | Ignored | |
| 25 | ,P,Xl,Xm,Yl,Ym | Plotting function | PLOT P,X,Y |
| 26 | ,G | Output gate | |
| 27 | | Ignored | |
| 28 | | Cursor ON | |
| 29 | | Cursor OFF | |
| 30 | | Move cursor to top left (home) | |
| 31 | ,X,Y | Tab X,Y | TAB X,Y |
| 32-127 | | Normal ASCII codes | |
| 128 | | Cell block | |
| 129-223 | | Gap block | |
| 224-255 | | User programmable | |

The BBC VDU codes are emulated as closely as possible, and users will find that many of the BBC VDU commands are valid for CU-GRAPH, e.g. VDU1 to VDU13 and others (see BBC User Guide, p. 379ff). However, some of the VDU commands differ considerably, and these differences are described below.

## 5. LOGO PROGRAM

```
 10 REM  ********************************************
 20 REM  *                                          *
 30 REM  *                 LOGO                     *
 40 REM  *                  by                      *
 50 REM  *               DAVE HUNT                  *
 60 REM  *                                          *
 70 REM  ********************************************
 80 REM Send Output to Cu-Graph
 90 *FX3
100 MODE3
110
120 VDU19,&84,0           :REM Change character size
130 VDU31,0,3             :REM Move text cursor to 0,3
140 COLOUR4:COLOUR128+7 :REM Text foreground blue; background white
150 PRINT"CUBE";
160
170 VDU31,33,3            :REM Move text cursor
180 COLOUR7:COLOUR128.    :REM Text foreground white; background nothing
190 PRINT"+"
200
210 VDU19,&11,0           :REM Change character size
220 VDU31,0,5
230 COLOUR7:COLOUR128
240 PRINT"85 columns by 32 rows max."'
250
260 VDU19,&22,0           :REM Change character size again
270 COLOUR1
280 PRINT"Variable character size"
290
300 VDU19,&81,0           :REM Change character size again
310 COLOUR3
320 PRINT"Width "''''
330
340 VDU 19,&24,0          :REM Change character size again
350 PRINT"and  Height"'
360
370 VDU19,&24,1           :REM Change size + sloped characters
380 COLOUR2               :REM Green text foreground
390 PRINT"Slope"
400
410 VDU19,&44,0
420 VDU 31,40,3           :REM Change character size
430 COLOUR7:COLOUR128+4 :REM White text on blue background
440 PRINT"HI-RES UNIT";
450
460 VDU20,&24,2           :REM Vertical characters; applies to
470 MOVE1023,0            :REM Graphics text only
480 GCOL0,5               :REM Graphics foreground magenta
490 PRINT"Vertical";
500
510 PROCcolblk(0,0,60)
520
530 GCOL1,2
540 PROCweb(500,0,300)
550
560 PROClogo(800,560,4)
```

## 4. PRINTER DUMPS

The CU-GRAPH hardware allows screen pixel reads for printer dumps. By writing the command &0F to the EFCIS Controller, location &DF00 will read the eight horizontal pixels given by the EFCIS controller's X,Y registers. The readings are then available in registers &DF10,1,2 as red, green and blue respectively. (NOTE: The least significant bits in the X register are always zero.)

Example program:

```
    10  REM pixel read
    20
    30  *FX3
    40  V=&DF00
    50  MODE4
    60  VDU20,&11,0:MOVE0,0:PRINT"S";:VDU4
    70  FOR Y=28 TO 0 STEP -4
    80  PROCread (0,Y)
    90  PROCpixel (red%)
   100  NEXT
   110  *FX3,7
   120  END
   130
   140  DEF PROCread (x%,y%)
  *150  MOVE x%,y%
   160  ?V=15                        ;command to read
   170  REPEAT UNTIL (?V AND4)            ;finished?
   180  red%=V?16:green%=V?17:blue%=V?18   ;read back pixels
   190  ENDPROC
   200
   210  DEF PROCpixel (x%)
   220  LOCAL c%
   230  FORc%=0 TO 5
   240  IF x% MOD2 COLOUR4 ELSE COLOUR3
   250  VDU128
   260  x%=x% DIV2
   270  NEXT
   280  PRINT
   290  ENDPROC
```

* When using Issue 1 CU-GRAPH peripheral driver, please replace with the following:

```
V?8  = x% DIV256  : V?9 = x%
V?10 = y% DIV256  : V?y%
```

### VDU14

Page Mode ON

This makes the computer wait at the bottom of the page and is used mainly when listing long programs to allow the page to be read. It can also be used for systems with minimum RAM. CU-GRAPH uses a RAM scroll buffer which is copied on to the screen during scrolling; VDU14 disables this scroll buffer. When the computer is waiting at the bottom of the page, the user must press the space bar in order to continue. Page mode is automatically selected if only one 8K RAM is fitted.

### VDU15

Select Scroll Mode

If this is used without scroll RAM, unpredictable characters will be copied on to the screen. If more than 8K of RAM is fitted, the CU-GRAPH peripheral driver will claim 5K of scroll RAM on initialisation. Otherwise page mode will automatically be selected.

### VDU16

Clears screen to current graphics colour background and homes cursor.

### VDU17,C

Text colour - COLOUR C

### VDU18,0,C

Graphics colour - GCOL 0,C

Colour bit definitions (default 7)

| C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
|---|---|---|---|---|---|---|---|
| B/F | 0 | 0 | C/S | I | B | G | R |

= C

| | | |
|---|---|---|
| C7 | 0 = foreground | 1 = background |
| C6 | set to 0 | |
| C5 | set to 0 | |
| C4 | 0 = set pixels | 1 = clear pixels |
| C3 | 0 = invert pixels | |
| C2 | blue | 1 = ON |
| C1 | green | 1 = ON |
| C0 | red | 1 = ON |

Examples

The default for both text and graphics colour foreground is:

00000111 - invert all pixels

The default for text background is:

10010111 - clear to black

The default for graphics background is:

10000000 - do nothing

BGR - define colour

| | C2 | C1 | C0 | |
|---|---|---|---|---|
| | B | G | R | |
| 0 | 0 | 0 | 0 | black |
| 1 | 0 | 0 | 1 | red |
| 2 | 0 | 1 | 0 | green |
| 3 | 0 | 1 | 1 | yellow |
| 4 | 1 | 0 | 0 | blue |
| 5 | 1 | 0 | 1 | magenta |
| 6 | 1 | 1 | 0 | cyan |
| 7 | 1 | 1 | 1 | white |

| C4 | C3 | |
|---|---|---|
| C/S | I | |
| 0 | 0 | Invert |
| 0 | 1 | Set |
| 1 | 0 | Clear |
| 1 | 1 | Clear |

| | | |
|---|---|---|
| Default | - | invert pixels |
| add 8 | - | set pixels |
| add 16 | - | clear pixels (overrides set pixels) |
| add 128 | - | for background |

Users will note that the facility for writing in direct colour is not available, since the pixels cannot be cleared and set in one operation. This problem can easily be resolved by using the invert function.

Consider, for example, how you would write in green on a yellow background. Yellow is made up of red and green pixels. Therefore, to change yellow to green, you need to remove or invert the red pixels. The foreground operation should therefore be 'invert red'.

## 3. OPERATING SYSTEM VARIABLES AND CU-GRAPH MEMORY MAP

Zero-Page usage

&D0 to &E1

VDUV

&226 - changed to point to CU-GRAPH peripheral driver

Video variables

&300 to &32F - text cursor position, graphics cursor position, colours etc.

&C00 to &CFF - user-programmable characters. Eight bytes per characters - total number of characters 32.

CU-GRAPH Card: Register Mapping of Thomson-EFCIS EF9366 Graphic Display Processor

```
V          EQU  &DF00(V)
CMD        EQU  V
STATUS     EQU  V
CTRL1      EQU  V+1
CTRL2      EQU  V+2
XYSIZE     EQU  V+3
DELTAX     EQU  V+5
DELTAY     EQU  V+7
XMSB       EQU  V+8
XLSB       EQU  V+9
YMSB       EQU  V+&0A
YLSB       EQU  V+&0B
LPX        EQU  V+&0C
LPY        EQU  V+&0D
COLOUR     EQU  V+&10      write
RED        EQU  V+&10      read red pixels
GREEN      EQU  V+&11      read green pixels
BLUE       EQU  V+&12      read blue pixels
PPORT      EQU  V+&13
```

OSWORDS (Service Call 08)

OSWORD 9

Read logical colour at point specified.

XY points to

LSB } X co-ordinate
MSB

LSB } Y co-ordinate
MSB

On exit XY+4 gives logical colour (255 if not on screen). Used by BASIC variable POINT.

HELP (Service Call 09)

If the CU-GRAPH card is present, the operating system call HELP will print out the title "Cu-graph" and the issue number. This is a useful feature for testing for the presence of the peripheral driver and whether it is in working order, when taking delivery of a system.

VDU19 and VDU20

Syntax: VDU19,<size>,<style>
VDU20,<size>,<style>

These VDU codes have been redefined and are completely different from the BBC VDU codes. VDU19 is used to select text size and style, while VDU20 is used to select graphic text size and style. Both codes also select the current cursor ON (cf. VDU4 and VDU5 which both select cursors but do not define size and style). When writing text at the graphics cursor, the cursor move codes are ignored.

Size byte

| | |
|---|---|
| MS nibble | 1 to F, 0 multiplies the X axis from 1 to 16 times |
| LS nibble | as for MS, but multiplies the Y axis from 1 to 16 times |

Style

| | |
|---|---|
| 0 | normal character |
| 1 | tilted |
| 2 | vertical (use only with graphics cursor) |
| 3 | tilted and vertical |
| 4-255 | no change |

For example:

```
10 VDU20,128,2
20 MOVE 700,300
30 PRINT "A";
40 VDU4
```

This will print a large 'A' lying on its side.

VDU22

Select screen mode

The CU-GRAPH display has been programmed to emulate the BBC modes. However, with CU-GRAPH all colours are available to all modes. A mode change provides a new row/column specification and character size, clears the screen and homes the cursor.

| Mode | columns x rows |
|---|---|
| 0 | 80 x 32 |
| 1 | 40 x 32 |
| 2 | 20 x 32 |
| 3 | 80 x 25 |
| 4 | 20 x 16 |
| 5 | 10 x 16 |
| 6 | 40 x 25 |
| 7 | 40 x 25 |

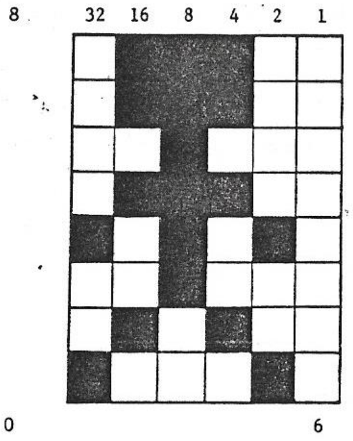
VDU22,M is equivalent to Mode M, but does not change the position of HIMEM. This is particularly useful in the case of EPROMed BASIC programs where both LOMEM and HIMEM have been assigned to another area of RAM. If Mode M were used, it would create the error 'Bad Mode' because of the position of HIMEM. VDU22,M overcomes this problem because it ignores HIMEM.

VDU23

User-programmable characters

VDU23,224-255 is reserved for 32 programmable characters. If you want to use this facility, first draw your character on an 8 x 6 grid (see below). As you can see in the example below, the six horizontal boxes are each assigned a number. Starting with the top line, add up the numbers of all the shaded boxes in a given line, and incorporate the sum of each line in the VDU23 command. The character shown below would thus be programmed in the following way:

```
VDU23,224,28,28,8,28,42,8,20,34
```



```
16 + 8 + 4 = 28
16 + 8 + 4 = 28
           = 8
16 + 8 + 4 = 28
32 + 8 + 2 = 42
           = 8
16 + 4     = 20
32 + 2     = 34
```

Try the following example program:

```
 10 REM exercise
 20 *FX3
 30 VDU23,224,28,28,8,28,42,8,20,34
 40 VDU23,225,28,28,42,28,8,42,20,0
 50
 60 GCOL0,7
 70 GCOL0,128+16+7
 80
 90 VDU20,&08,0
100 CLG
110
120 FORC=0TO20
130 MOVE 500,300
140 VDU224+(C MOD 2)
150 IF INKEY50
160 NEXT
170
180 VDU4
190 *FX3,7
```

You should see the example character 'jumping up and down'.

OSBYTES (Service Call 07)

Summary

| | |
|---|---|
| 5 | printer destination |
| 6 | printer ignore |
| 117 | read VDU status byte |
| 134 | read text cursor position |
| 135 | read character at text cursor |
| 160 | read VDU variable |

*FX5,D

| | |
|---|---|
| D = 0 | dump |
| D = 1 | CU-GRAPH centronics port |
| D = 2 | serial port |
| D = >2 | user printer via user printer vector (UPTV) at &222 |

*FX6,10 - ignore line-feeds

*FX6,0 - allow line-feeds

OSBYTE 117 - read VDU status

| | |
|---|---|
| Bit 0 | printer output enabled by VDU2/3 |
| 1 | not used |
| 2 | page mode (wait at bottom of screen for space key) |
| 3 | not used |
| 4 | cursor ON/OFF |
| 5 | text or graphics cursor |
| 6 | not used |
| 7 | video flag VDU6/21 |

OSBYTE 134

This is called by BASIC to read text cursor position, using variables POS and VPOS.

OSBYTE 135

This is used by screen editing.

OSBYTE 160

Read VDU variable.
Returns &300,X

APPENDIXES: Issues 1 and 2

## 1. VDU VECTOR

The system vector claimed by the CU-GRAPH PD is the VDU vector (VDUV) at &226. When EuroBEEB is switched on, the serial port is the active output channel. This can be switched off with the *FX3 command (see p. 5 ), in which case any unrecognised channels are passed to VDUV.

## 2. SERVICE CALLS

The CU-GRAPH PD claims the following service calls:

| | |
|---|---|
| 03 | Initialise |
| 05 | Interrupt |
| 07 | OSBYTE |
| 08 | OSWORD |
| 09 | HELP |

### Initialise (Service Call 03)

The CU-GRAPH peripheral driver tests for the presence of the CU-GRAPH card. If the card is present, the peripheral driver sets a flag and sets up the default variables. CU-GRAPH requires 5K of RAM to achieve software scroll. The peripheral driver tests the amount of RAM fitted in the system. If only one 8K RAM is fitted, it automatically defaults to page mode. If 16K or more RAM is fitted, it claims 5K from HIMEM down for the scroll buffer. It also claims unrecognised output channel calls. On initialisation, the peripheral driver clears the screen and selects the start-up mode given by the start-up option byte located at &F000. This is Mode 7 as supplied.

### Interrupt (Service Call 05)

The CU-GRAPH scrolling works by copying the scroll buffer on to screen, shifted one line up. Every time a line-feed occurs at the bottom of the screen, software scroll will be initiated if CU-GRAPH is in scroll mode. The scroll feature has been added to CU-GRAPH for the sake of convenience, but is not intended for text processing.

In Issue 1 of the CU-GRAPH peripheral driver, the interrupt overhead was approximately 12 ms. This meant that other interrupts were effectively blocked out for this period of time. For example, if an application required regular serial input data at 9600 baud, this would fail if CU-GRAPH scrolled. This problem has been remedied in Issue 2.

## VDU25

Plotting function

VDU25 requires five parameters:

1. plot type (K)

2. }
X co-ordinate (LS byte first)
3. }

4. }
Y co-ordinate (LS byte first)
5. }

Plot variables to make up a plot value (K):

| | | |
|---|---|---|
| 0 | move graphics cursor | |
| 1 | plot line | set pixels |
| 2 | | invert " |
| 3 | | clear " |

+0 relative co-ordinates
+4 absolute co-ordinates
+8 dot only
+16as 0-15 with last dot missing
+32for a dotted line
+64for a dashed line

Any combination of these numbers can be used. For example, 32+64+4+1 means dot-dashed line, absolute co-ordinates and set pixels.

Try the following program:

```
10 *FX3
20 CLG
30 K=32+64+4+1
40 PLOT K,1023,0
50 VDU4
60 *FX3,7
```

Please note that the first parameter K is very different from the BBC equivalent.

VDU26

Output gate

There is an 'AND' gate on the colour output. By writing to this gate, it is possible to stop a pixel output. This facility can be used to switch rapidly between colour planes. In a black-and-white application it would be possible to have three independent black-and-white pictures and to use the gate to display each picture in turn. In order to achieve this, the three colour weighting resistors R10, R11 and R12 (see CU-GRAPH circuit diagram) would need to be equal, i.e. R11 and R12 would need to be changed to 1K. R10 is red intensity, R11 is green intensity and R12 is blue intensity.

The default is VDU26,7. Please note that this command only comes into immediate effect in conjunction with a screen access.

VDU28, VDU29

Cursor ON, cursor OFF

This facility exists on the BBC Micro with VDU23,0 (see p.77 of the User Guide for the BBC Micro) but is less convenient to use.

VDU26

Control RGB output gate

There is an 'AND' gate on the colour output. By writing to this gate, it is possible to stop a pixel output. This facility can be used to switch rapidly between colour planes. In a black-and-white application it would be possible to have three independent black-and-white pictures and to use the gate to display each picture in turn. In order to achieve this, the three colour weighting resistors R10, R11 and R12 (see CU-GRAPH circuit diagram) would need to be equal, i.e. R11 and R12 would need to be changed to 1K. R10 is red intensity, R11 is green intensity and R12 is blue intensity.

The default is VDU26,7.

In Issue 1 of the CU-GRAPH peripheral driver this command was only effective in conjunction with a screen access. Now it has an immediate effect.

EXAMPLE

```
  10 REM CU-GRAPH DEMO
  20 MODE3
  30 COLOUR132    :REM Text background blue
  40 CLS
  50 VDU19,&46,0 :REM Largish letters
  60 VDU31,27,12 :REM Move cursor
  70 COLOUR3      :REM Text foreground yellow
  80 PRINT"HELLO!"
  90 REPEAT
 100 VDU26,7-1    :REM Turn-off red
 110 IF INKEY 50 :REM Half second delay
 120 VDU26,7      :REM Turn on RGB
 130 IF INKEY 50
 140 UNTILFALSE
```

VDU29,X;Y;

Define graphics origin

VDU29 has been added to emulate the BBC Micro. This command now moves the graphics origin to the position X,Y.

(b) VDU23,0 Cursor ON/OFF

The commands to turn the cursor on and off are now the same as those on the BBC Micro:

```
VDU23,1,0;0;0;0; turns the cursor off
VDU23,1,1;0;0;0; turns the cursor on
```

In general, VDU23 commands which are hardware dependent, e.g. those that access the 6845 CRTC, will not be recognised by CU-GRAPH.

VDU25,K,X;Y; (PLOT K,X,Y)

Plotting function

(a) The PLOT commands are now compatible with the BBC micro:

PLOT K,X,Y where K is defined as follows:

| binary | decimal | definition |
|---|---|---|
| | | relative co-ordinates |
| 0 0 0 0 0 0 0 0 | 0 | Move |
| 0 1 | 1 | Plot foreground graphics colour |
| 1 0 | 2 | Plot inverted foreground graphics colour |
| 1 1 | 3 | Plot background graphics colour |
| 0 0 0 0 0 1 0 0 | +4 | absolute co-ordinates (as above) |
| 0 0 0 0 1 0 0 0 | +8 | replot last dot of line (removes gap between connecting lines when in 'invert pixel' mode |
| 0 0 0 1 0 0 0 0 | +16 | dotted line |
| 0 0 1 0 0 0 0 0 | +32 | dashed line |
| 0 1 0 0 0 0 0 0 | +64 | single dot only |
| 1 0 0 0 0 0 0 0 | +128 | not used |

NOTES

1. The BBC Micro's PLOT 85 (plot and fill a triangle) is not supported by CU-GRAPH.

2. The MOVE command now writes the co-ordinates to the graphics chip (useful when used in conjunction with a pixel read).

CU-GRAPH ISSUE 2

EFFECT OF THE VDU COMMANDS

In Issue 2 of the CU-GRAPH peripheral driver, most of the VDU commands have been brought into line with those of the BBC Micro.

The VDU commands are ASCII characters between 0 and 31 which are passed to the output channel. In the table below, 'Variable' refers to the character(s) passed to the output channel immediately following the VDU command.

| VDU | Variable | Comments | BASIC Word |
|---|---|---|---|
| 0 | | Ignores | |
| 1 | ,P | Send P to printer only | |
| 2 | | Turn printer ON | |
| 3 | | Turn printer OFF | |
| 4 | | Write text at text cursor | |
| 5 | | Write text at graphics cursor | |
| 6 | | Turn video ON | |
| 7 | | Make short bleep | |
| 8 | | Backspace | |
| 9 | | Forward space | |
| 10 | | Linefeed | |
| 11 | | Move up one line | |
| 12 | | Clear screen | CLS |
| 13 | | Move cursor to line start | PRINT' |
| 14 | | Page Mode ON | |
| 15 | | Scroll Mode ON (Page Mode OFF) | |
| 16 | | Clear to graphics background colour | CLG |
| 17 | ,C | Define text colour | COLOUR C |
| 18 | ,B,C | Define graphics colour | GCOL B,C |
| 19 | ,M,S | Define text size and style | |
| 20 | ,M,S | Define graphic text size and style | |
| 21 | | Turn video OFF | |
| 22 | ,L | Select screen mode | MODE L |
| 23 | ,C,Dt,--------,Db | Programmable character | |
| 24 | | Ignored | |
| 25 | ,P,Xl,Xm,Yl,Ym | Plotting function | PLOT P,X,Y |
| 26 | ,G | Control RGB output gate | |
| 27 | | Ignored | |
| 28 | | Ignored | |
| 29 | ,X,Y | Define graphics origin X,Y | |
| 30 | | Move cursor to top left (home) | |
| 31 | ,X,Y | Tab X,Y | TAB X,Y |
| 32-127 | | Normal ASCII codes | |
| 129-223 | | Gap block | |
| 224-255 | | User programmable | |

VDU12 (CLS)

This command clears the screen and homes the cursor (top left of the screen). It does not restore the cursor if it is has been switched off with VDU23,1,0;0;0;0; (see VDU23).

VDU14

Page Mode ON

This makes the computer wait at the bottom of the page and is used mainly when listing long programs to allow the page to be read. It can also be used for systems with minimum RAM. CU-GRAPH uses a RAM scroll buffer which is copied on to the screen during scrolling; VDU14 disables this scroll buffer. When the computer is waiting at the bottom of the page, the user must press the space bar in order to continue. Page mode is automatically selected if only one 8K RAM is fitted.

In Issue 2, the video RAM buffer is now completely disabled when CU-GRAPH is in Page mode (in Issue 1 it continued to be written to while CU-GRAPH was in Page mode).

VDU15

Scroll Mode ON

If this is used without scroll RAM, unpredictable characters will be copied on to the screen. If more than 8K of RAM is fitted, the CU-GRAPH peripheral driver will claim 5K of scroll RAM on initialisation. Otherwise page mode will automatically be selected.

VDU16 (CLG)

This clears the screen and moves the graphics cursor to the bottom left of the screen. The text cursor is not changed and is still displayed after a CLG command. It may only be removed by the VDU23,1,0;0;0;0; command (see VDU23).

The position of the graphics cursor is affected by graphics commands such as MOVE and PLOT, etc. Graphics text will be printed at the graphics cursor, following the commands VDU20 (change text size and style) or VDU5 (write text to graphics cursor). Graphics text does not have a cell background, i.e. the characters are displayed in the current graphics foreground on the current graphics background.

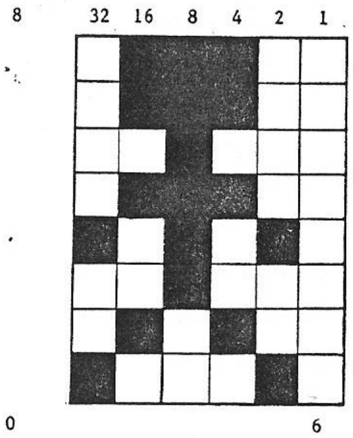
VDU23

(a) VDU23,C,Dt,--------,Db

User-programmable characters

VDU23,224-255 is reserved for 32 programmable characters. If you want to use this facility, first draw your character on an 8 x 6 grid (see below). As you can see in the example below, the six horizontal boxes are each assigned a number. Starting with the top line, add up the numbers of all the shaded boxes in a given line, and incorporate the sum of each line in the VDU23 command. The character shown below would thus be programmed in the following way:

```
VDU23,224,28,28,8,28,42,8,20,34
```



16 + 8 + 4 = 28

16 + 8 + 4 = 28

= 8

16 + 8 + 4 = 28

32 + 8 + 2 = 42

= 8

16 + 4 = 20

32 + 2 = 34

Try the following example program:

```
 10 REM exercise
 20 *FX3
 30 VDU23,224,28,28,8,28,42,8,20,34
 40 VDU23,225,28,28,42,28,8,42,20,0
 50
 60 GCOL0,7
 70 GCOL0,128+16+7
 80
 90 VDU20,&08,0
100 CLG
110
120 FORC=0TO20
130 MOVE 500,300
140 VDU224+(C MOD 2)
150 IF INKEY50
160 NEXT
170
180 VDU4
190 *FX3,7
```

You should see the example character 'jumping up and down'.

VDU22,L (MODE L)

Select screen mode

The CU-GRAPH display has been programmed to emulate the BBC modes. However, with CU-GRAPH all colours are available to all modes. A mode change provides a new row/column specification and character size, clears the screen and homes the cursor.

The MODE statement has the following effect:

1. Clear screen
2. Home text cursor (top left of screen)
3. Home graphics cursor (bottom left of screen)
4. Clear background text and graphics to black
5. Set foreground text and graphics to white
6. Switch on the cursor even if it has previously been switched off
7. Revert to default character sizes for the chosen mode
8. Graphics character size defaults to the MODE 0 text character size.

MODE does not affect the position of the scroll buffer or the character buffer.

| Mode | columns x rows |
|---|---|
| 0 | 80 x 32 |
| 1 | 40 x 32 |
| 2 | 20 x 32 |
| 3 | 80 x 25 |
| 4 | 20 x 16 |
| 5 | 10 x 16 |
| 6 | 40 x 25 |
| 7 | 40 x 25 |

NOTE

VDU22,L is equivalent to Mode L, but does not change the position of HIMEM. This is particularly useful in the case of EPROMed BASIC programs where both LOMEM and HIMEM have been assigned to another area of RAM. If Mode L were used, it would create the error 'Bad Mode' because of the position of HIMEM. VDU22,L overcomes this problem because it ignores HIMEM.

The use of both graphics and text cursors is demonstrated in the following example:

```
 10 GCOL 0,1          :REM RED graphics foreground
 20 GCOL 0,128+4      :REM BLUE graphics background
 30 COLOUR 3          :REM YELLOW text foreground
 40 COLOUR 128+5      :REM MAGENTA text background
 50 CLS
 60 PRINT"CU-GRAPH"
 70 MOVE 500,500
 80 VDU5:PRINT "GRAPHICS TEXT"   :REM print at graphics cursor
 90 VDU4:PRINT "TEXT"  :REM print at text cursor
100 CLG
110 VDU23,1,0;0;0;0;0  :REM remove text cursor
```

NOTE

The method of generating a text character in the MOS software is as follows:

1. clear cell to black,
2. write background colour to cell (if black do nothing),
3. create the absolute colour required by inverting the character on to the cell. (This is possible because the background colour is already known.)

VDU17,C (COLOUR C)

The text colour commands (COLOUR C) have been changed to emulate more closely those found on the BBC Micro.

Foreground text (COLOUR0-COLOUR7) can now be printed in DIRECT COLOUR. For example, COLOUR3 would produce yellow text regardless of the colour of the existing pixels on the screen. Background text (COLOUR128-COLOUR135) is also implemented in DIRECT colour, i.e. a cell of 6 x 8 pixels is written in the current text background before the character is applied.

EXAMPLE: Print blue characters on a cyan background

```
10 COLOUR 128+6      :REM cyan background
20 COLOUR 4          :REM blue characters
30 PRINT "CU-GRAPH"
```

VDU18,B,C (GCOL B,C)

GCOL and VDU commands are now compatible with those on the BBC Micro.

B can take values in the range 0 to 4:

0 Write in absolute colour

1 OR the specified colour with the colour already there

2 AND the specified colour with the colour already there

3 Exclusive-OR the specified colour with the colour already there

4 INVERT the colour already there

C can take on any value in the range from 0 to 7 for foreground colours, and in the range from 128 to 135 for background colours.

NOTES:

1. GCOL0,C (write in absolute colour) is implemented with a double write in the CU-GRAPH software. The result of this is that PLOTs following a GCOL0,C command take approximately twice as long as those using the other GCOL commands. With practice, the user will find that any CU-GRAPH display can be constructed using the other GCOL commands for maximum speed.

2. The effect of GCOL4,C on the BBC Micro depends on the number of colours available in the current mode. On CU-GRAPH there are eight colours independent of the mode.

For both COLOUR and GCOL statements the colours are defined as follows:

| | C2 | C1 | C0 | |
|---|---|---|---|---|
| | B | G | R | |
| 0 | 0 | 0 | 0 | black |
| 1 | 0 | 0 | 1 | red |
| 2 | 0 | 1 | 0 | green |
| 3 | 0 | 1 | 1 | yellow |
| 4 | 1 | 0 | 0 | blue |
| 5 | 1 | 0 | 1 | magenta |
| 6 | 1 | 1 | 0 | cyan |
| 7 | 1 | 1 | 1 | white |

where C0 - C2 are the least significant bits of the parameter C.

For background colours add 128 to the values given.

The default value is 7 for foreground text and 128 for background text.

EXAMPLE:

```
10 GCOL 0,2       :REM green foreground
20 GCOL 0,128+4   :REM blue background
30 CLG
40 MOVE 500,500
50 PRINT "Control Universal"
```

NOTE: Logical colour definitions and windows are not supported.

VDU19 and VDU20

Syntax: VDU19,<size>,<style>

VDU20,<size>,<style>

These VDU codes have been redefined and are completely different from the BBC VDU codes. VDU19 is used to select text size and style, while VDU20 is used to select graphic text size and style. Both codes also select the current cursor ON (cf. VDU4 and VDU5 which both select cursors but do not define size and style). When writing text at the graphics cursor, the cursor move codes are ignored.

Size byte

| | |
|---|---|
| MS nibble | 1 to F, 0 multiplies the X axis from 1 to 16 times |
| LS nibble | as for MS, but multiplies the Y axis from 1 to 16 times |

Style

| | |
|---|---|
| 0 | normal character |
| 1 | tilted |
| 2 | vertical (use only with graphics cursor) |
| 3 | tilted and vertical |
| 4-255 | no change |

For example:

1.
```
VDU19,&42,0
PRINT "H"
```

will print the letter H. Compared to the smallest available characters (MODE 0) this letter is four times wider in the horizontal direction and twice taller in the vertical direction.

2.
```
VDU19,&11,1
PRINT "HELLO"
```

will print a small sized sloped version of the word HELLO.

3.
```
MOVE 200,200
VDU20,&21,2
PRINT "Magenta"
```

will print the word "Magenta" vertically.