MICROTAN 65

VIDEO 80/82-R

USER MANUAL

Contents

Introduction

The Video 80/82 module was an "intelligent" video terminal created by the Tangerine Users Group (TUG) and then adopted by Tangerine. Unlike other Microtan video cards it is equipped with its own CPU and memory leaving the system memory free for user applications. Communication with the card is via 2 IO locations, one to write data and one to read status.

Video 80/82-R design goals

In line with other Microtan-R projects the goal is to recreate the original board as faithfully as possible, incorporating fixes to known issues and addressing component obsolescence issues where practical. The board layout has been kept very similar to the original with most of the components numbered the same.

Changes from the original

- The original board used 6116 SRAM chips in a 150ns speed rating. These devices are hard to source, particularly in the fastest speed rating, and the board requires 9 of them. It is possible to run the board at double speed and this is outside the timing specification of even the 150ns parts. 80/82-R uses 6264 devices to replace the 6116. The 6264 was launched at about the time the original board was available, is still in production today and is readily available in speed ratings of 150ns and faster.
- The original design used a 6502A CPU which might need to be upgraded to a 6502B when running with the double speed modification (see below). An extra pullup resistor and a jumper have been fitted and a few other minor changes made so that 80/82-R should work with a 6502A/B or the 65C02 CPU that is still in production.
- As the system RAM device has been changed to a 6264 an extra address line has been added to give the system access to 4K of RAM rather than the original 2K, although current firmware makes no use of this.
- The 2764 socket is pin compatible with any 28 pin JEDEC standard device such as 27128, 27256 or 27512 devices. An extra address line has been added so that if one of these larger devices is used the system will have access to 16K of EPROM. Note the firmware must always be programmed into the top of larger devices.
- A number of modifications and fixes were published for the original board in TANDOC 1 and a document called More Microtan Modifications (MMM) the following of these have been incorporated.
    - Missing Characters fix from MMM page 8
    - Allow use of M command from MMM page 9
    - Decode of the spare socket from MMM page 10

    - 2X clock option (3MHz CPU clock) from TANDOC 1 page 6 (Method 1) – selectable by links.
- The DIP switches have been replaced by pin headers and jumpers
- The original board output video on pin b20 but many users required it to be on b19, this is now selectable by jumpers.
- A pin header or Molex KK connector has been added on the front of the board which exposes the TTL level video output and other key timing signals for maximum flexibility for video mixing.
- Improvements to the memory timing to support a wide range of devices.
- With the timing improvements in place and the right mix of devices it was found possible to operate the board at quadruple speed (6MHz CPU clock) so a link was added to support this option.

Copyright and Warranty

Acknowledgements

As with all of my Microtan projects, this would not have been possible without the archive of material maintained by Alan at http://www.microtan.ukpc.net/. His support and guidance have been invaluable throughout this project.

Thanks are due to James Price for spotting many typos and OCR issues in this manual

Getting started

Construction requires assembly of the PCB, programming an EPROM with the VBUG (V6 PTL) software and modification of the host Microtan CPU board to display the video output. For most users replacing the TANBUG EPROM on the Microtan with one programmed with TUGBUG will be required. For use with BASIC modification to the BASIC EPROMS (for compatibility with TUGBUG) and the 80/82 toolkit EPROM are required. Images for these EPROMS are available from http://www.microtan.ukpc.net/

Bill of Materials

Ident	Description	Qty	Notes
J1	DIN 41614 male right angle connector type B 64 way 2 row a+b	1	HARTING 0902 164 6921 is an example
	14 pin 0.3” DIP Socket	9	
	16 pin 0.3” DIP Socket	9	
	20 pin 0.3” DIP Socket	1	
	24 pin 0.6” DIP Socket	3	
	28 pin 0.6” DIP Socket	4	
	40 pin 0.6” DIP Socket	1	
U1,U4	M8212	2	
U2,U15	74LS139	2	
U3	74LS138	1	
U5	74LS163	1	
U6,U12	74LS00	2	See note 2
U7	74LS123	1	
U8	74LS04	1	See note 2
U9	74LS74	1	
U10	6502A	1	6502B required for 3MHz WDC65C02 will work at any speed.
U11	74LS02	1	
U13,U14,U20,U21	74LS157	4	
U16	74LS10	1	See note 2
U17,U18	74LS393	2	
U19	74LS32	1	
U23	74LS165	1	
U25	74LS245	1	
U28	2764	1	See note 1
U33,34,35	6264	3	See note 2
U31 Control link	2x3 Pin Header	1	0.1” pitch
B3 B7 link	1x3 Pin Header	1	0.1” pitch
B6 link	1x2 Pin Header	1	0.1” pitch
6/3/1.5 Mhz link	2x3 Pin Header	1	0.1” pitch
Reset link	1x3 Pin Header	1	0.1” pitch
Video Output	1x3 Pin Header	1	0.1” pitch
A1-A8 links	2x8 Pin Header	1	0.1” pitch
6502A link	1x2 Pin Header	1	0.1” pitch
TTL Video connector	1x6 right angle header	1	0.1” pitch – alternate molex KK
	0.1” Jumper	9	
C1-C28	0.1mfd ceramic capacitor	28	5mm or 5.08mm pitch
C30	47uf Tantalum capacitor	1	5mm or 5.08mm pitch
D1	1N4148	1	
R1,R2,R3,R4,R5,R7,R8	1K resistor	7	R8 only required with optional rest switch
SW-RESET1	C&K PTS645V vertical tactile switch	1	Optional local reset switch

Note 1: The EPROM U28 can be a 2764, 27128, 27256, or 27512. If 128, 256 or 512 parts are used the program must be in the upper 8K of the EPROM. For 1.5MHz operation a 250ns speed or faster is required. For 3MHz operation 27Cxx parts with a speed of 150ns or faster are required. 6MHz operation requires a speed of 50ns or better. EEPROM devices such as the Winbond 27C512 or OTP devices such as the Microchip AT27C512 also work and are available in access times as low as 45ns. A 45ns Winbond 27C512 was found to work at 6MHz.

Note 2: The 6264 is available in 0.6” and 0.3” ‘skinny DIP’ packages. The 0.6” device is required. For 1.5MHz operation 150ns or faster is required. For 3MHz operation 120ns devices are required. Hyundai -10 parts were tested and found to work at 1.5MHz and 3MHz. For 6MHz operation 60ns or better devices are required. If an Alliance 55ns device (still in production) is used for U34 then U6, U8 and U16 should be replaced by 74F series devices to avoid a timing conflict. These devices have been tested to work at 6MHz. 62256 devices follow the same JEDEC standard. A 62256 device can be used for the system memory U34. To use a 62256 for the video memory fit the device in U35 and leave U33 empty.

Component sourcing

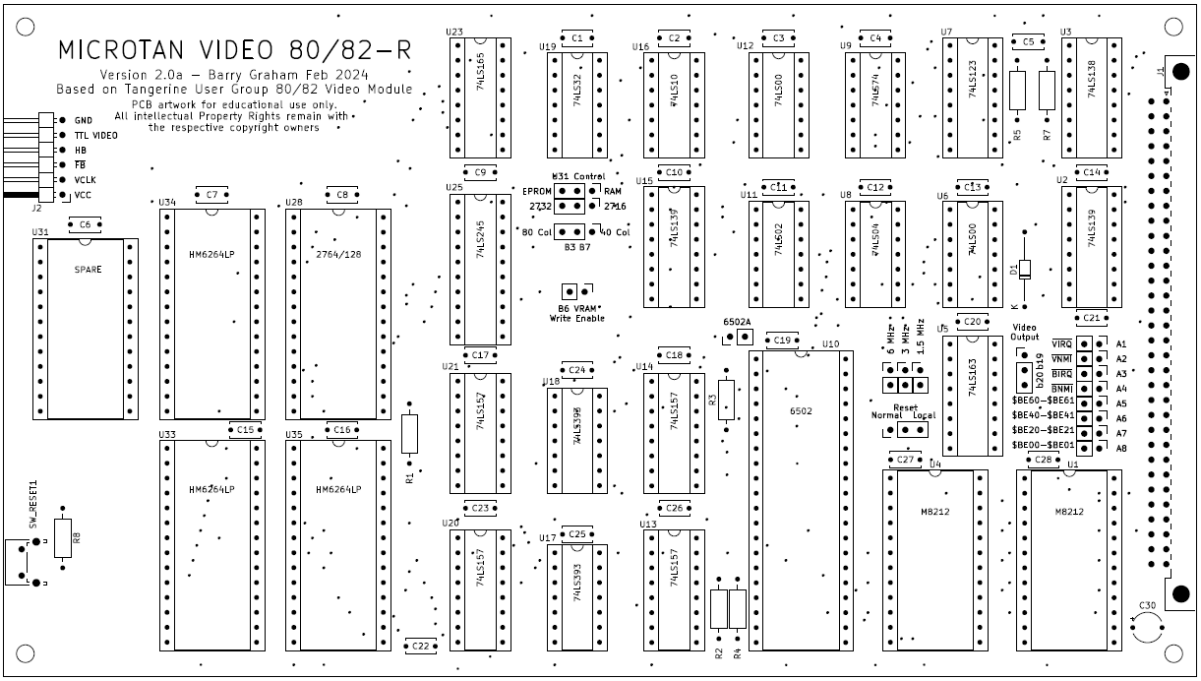
As of January 2024 all parts other than the M8212 and 6502A are still manufactured and readily available from suppliers such as Mouser and Farnell in the UK. The M8212 was originally an Intel part, other manufacturers made them and the in particular the Mitsubishi M5L8212P is available from several specialist suppliers selling on ebay. Both Intel and Mitsubishi devices have been tested and found to work.

The 6502A or 6502B are available used from specialists. The WDC65C02 is still manufactured available from Mouser and works.

6464 devices from Hitachi, Hyundai and others are available from specialist suppliers, Alliance Memory still manufacture a 55ns part which works but if used in U34 requires U6, 8 and 16 to be replaced by 74F series devices.

Memory device speed designations follow a ‘-‘ after the part number. Check the datasheet from the manufacturer as the designation is not always obvious. For example for the 6116 the Hitachti HM6116-3 part is rated at 150ns. LP and ALP versions of the SRAM chips can be found and should all work.

Construction



As with any PTH PCB it is best to assemble the board in sequence of increasing component height, so fit the sockets first, then the diode, resistors, capacitors, the DIN connector and finally the pin headers for the jumpers. Care should be taken with all components to ensure they lie flat to the board but this is especially critical with the DIN connector. If the mounting screws are used on the connector then they must be tightened before it is soldered.

Pin headers can be cut from longer strips. If single strips are being used to make up the larger blocks then fit a jumper to hold them upright when soldering.

Jumper settings

A1 to A4 - These control the routing of interrupts from the 8212s to the Microtan CPU (A3 and A4) and the Video 80/82 CPU (A1 and A2) current firmware requires only A1 to be fitted.

A5 to A8 – These control the IO address in the Microtan memory map that the 8212s appear at. Current TUGBUG firmware requires A8 to be fitted.

Video Output – This controls which TANBUS connection the video output is routed to. The original card was wired through switch B8 to b20. Select b20 or b19 to suit the installation.

6/3/1.5 MHz – One jumper should be fitted vertically in the right hand 1.5MHz position for normal operation. Placing a jumper in the centre 3MHz position implements the clock speed doubling described in TANDOC 1 page 6. Ensure that the CPU and Memory device speed ratings are suitable for the option chosen, see notes to the Bill of Materials for details.

Reset – With the jumper fitted in the normal position the board is reset by the TANBUS reset line. In the 'local' position the TANBUS line is disconnected, and a reset switch must be provided to reset the board.

U31 Control – These links must be set to suit any device installed in the "spare" socket U31, the use of which was described in MMM page 10.

B3/B7 – This controls if the board runs at 1.5MHz creating an 80 column display or 0.75MHz creating an 40 column display. Vbug V6 (PTL) only supports 80 column mode.

B6 – This link is normally fitted. Removing prevents the CPU from writing to the video ram.

6502A – This link should be fitted when a 6502A or B is used. It MUST NOT be fitted when using a WDC65C02.

Video connection

Video 80/80-R does not generate synchronisation signals and so it's output must be mixed with Microtan output before it can be displayed on a monitor. This is achieved by routing the output of the board over the backplane, there are several options depending on the other boards in the system, some common configurations are given below, others are possible.

The idea in every case is to perform a "diode or" of the Video 80/82-R signal, the Microtan video signal and synchronisation pulses.

Motherboards

Depending on the motherboard in use it may be necessary to add a connection between b20 on the expansion slots and a20 on the Microtan slot, or b19 on the expansion slots and a19 on the Microtan slot. The choice of b20/a20 or b19/a19 depends on the other boards in the system.

On Backplane-R this means fitting a jumper to link pins 3 and 5 on the video matrix for b20/a20 and pins 2 and 4 for b19/a19.

Operation without HRG / HRG-R

Microtan-R

The simplest approach is to set the Video 80/82-R to output on b20 and then fit link JP5 on the Microtan-R to route the video signal from TANBUS a20 to the Microtan-R video mixer. The mixed output from both boards will be available at the Microtan video output.

Microtan MT10016 Iss1 or MT001

Set the Video 80/82 to output on b19. Connect a wire from pin a19 on the Microtan TANBUS connector to the cathode of D3. The mixed output will be available across the 75R resistor.

Operation with HRG or HRG-R

This is complicated as there are now 3 video signals. On HRG-R ensure that JP8 is NOT fitted in either position so that the card is not outputting it's video onto the backplane.

Microtan-R

The HRG / HRG-R manuals instruct the user to fit a wire from pin 8 of U24 to 19a on the TANBUS connector. Move this wire from pin 8 (which is connected to the anode D2) to the cathode of D2. It will still provide the functionality required for HRG / HRG-R. Set the VIDEO 80/82-R to output on b19. The mixed output of Video 80/82-R and the Microtan-R will be available at the Microtan-R video output and the mixed output of all three boards will be available at the output of HRG/HRG-R

Microtan MT10016 Iss1 or MT001

The HRG / HRG-R manuals instruct the user to fit a wire from pin 8 of IC C3 to 19a on the TANBUS connector. Move this wire from pin 8 (which is connected to the anode D3) to the cathode of D3. It will still provide the functionality required for HRG / HRG-R. Set the VIDEO 80/82-R to output on b19. The mixed output of Video 80/82-R and the Microtan will be available at the Microtan across the 75R resistor and the mixed output of all three boards will be available at the output of HRG/HRG-R

TTL Video output

The 6 way right angle connector brings the direct TTL video output, the TANBUS blanking signals, clock and power to the edge of the board. This output can be used to drive the digital input of RGBtoHDMI project https://github.com/hoglet67/RGBtoHDMI/wiki and power the converter by using a cable as follows:

80/80-R TTL video connector pin	RGBtoHDMI 12 way IDC pin
1 (VCC)	12 (+5v)
6 (GND)	3 (GND)
4 (HB)	8 (HSYNC)
3 (FB)	10 (VSYNC)
5 (Video)	9 (GREEN3)

Setting the RGBtoHDMI clock to 12MHz, sync input to +H+V, input to 3 bit mode and then using the 'create new profile function' should correctly detect the signal, or a profile can be downloaded from microtan@thegrahamfamily.me.uk. The connections above create a green display. Although the 80/82-R does not create sync pulses the converter is flexible enough to be able to use the leading edge of the blanking signals for sync.

Firmware implications

At the time of writing all firmware referenced in this section is available from www.microtan.ukpc.net

TUGBUG

TUGBUG V1.1 replaces TANBUG on the Microtan 65 CPU card and is specifically designed to support the original 80/82 card. By default all characters are output to both screens. <ctrl>^ toggles output to 80/82 and <ctrl>S toggles output to the original Microtan screen.

BASIC

The original BASIC EPROMS are not compatible with TUGBUG. A modification to the BASIC EPROMs was originally published in TANDOC 1. This modification is better explained in the firmware section of www.microtan.ukpc.net (mod 1) and solves this problem.

TUG produced a BASIC Toolkit for the 80/82 board as an EPROM fitted in TANEX socket E2 (E800-EFFF). This toolkit works with 80/82-R but note it is NOT compatible with TANDOS if the firmware has been modified to work with the HRG board. It is compatible with the original TANDOS firmware. Note that the V6 toolkit is compatible with VBUG V6. The manual for the V6 toolkit is effectively an addendum to the V1 tookit manual and both should be read.

The toolkit requires either TUGBUG or alternatively TANBUG modified as described in TUGNews 29 on page 10.

TANDOS

Although DBASIC is a file stored on disk part of the code is resident in the TANDOS EPROM.

Because both DBASIC and the 80/82 toolkit extend the commands of BASIC they need to be compatible. 80/82 toolkit V6 is compatible with unmodified DBASIC file and unmodified TANDOS EPROM. There is a version of the TANDOS EPROM modified to work with the HRG toolkit and this is not compatible with 80/82 toolkit.

VBUG V6(PTL) Operating System

[The following chapter is reproduced word for word from the original manual]

The Video 80/82 Operating System is supplied in an 8K 2764 250ns Eprom. It provides for an 'Intelligent' exchange of information and commands between itself and the host computer over a wide range of facilities for both Text and Graphic handling routines and is supported with its own standard 96 Ascii character set, an additional three sets may be added at a later date.

At the heart of the operating system is the Status/Command Register and a 256 character Silo <First In - First Out> Buffer. Together these provide for a rapid exchange of data or commands between the Video module and the host computer.

VBUG uses a 2K working random access memory area for Zero page and Stack operations. This is provided separately from the video ram, thereby allowing the Vbug operating system to remain totally independent from its host computer system. Using this method of operation, the host computer is able to send a series of commands and/or data to the Video module thus allowing the Video module to process those instructions whilst the host computer carries out other tasks.

The development of VBUG over the past couple of years, coupled with practical experience has resulted in VBUG V6(PTL), offering a wide range of facilities which enable complex Graphic and Text display to be built limited only by the skill and imagination of the user.

By using an earlier version of VBUG it is possible to run the Video module in a 40 character mode (covered later in this manual. But normally it is accepted that the Video module will be fully expanded thus operating in 8O character mode. Therefore throughout this manual it will be assumed that the mode of operation will be 8O character 512 x 256 format. The examples given are legal for both types of operation subject to parameter changes.

The SILO

VBUG uses a 256 character SILO or (First In - First Out) Buffer. This facility allows the host computer to very quickly send a sequence of commands or data directly to the Video Module for processing, thereby al1 owing it to continue with other tasks whilst awaiting for the Video Module to complete those instructions. The 'Ready Bit' in the Status Register is 'SET' almost immediately after the character has been received, if there is still room in the Silo.

The rules governing the use of the Silo are that, when using the Graphics or ESC command modes, the sequence of commands plus parameters must remain unbroken and completed before the Silo returns to a "Standby' condition. Aborting the transfer of data to the Video module after it has received the graphics command will result in VBUG remaining in the 'Wait' state for the rest of those

graphics parameters to be passed over, which it expected in conjunction with the command instruction. It can be seen therefore that by aborting the program during this sequence of events, will leave the module waiting for data it is not likely to receive. A typical example of this in operation would be under Basic Language control where a "Break" command is issued during graphic plotting etc. The Silo may be 'Cleared Down' to a "Standby" condition by sending it a sequence of four single byte instructions such as 'Clear Screen'. This will replace the lost parameters caused by the 'Abort' command.

During Error handling when an "Error" condition occurs, - VBUG will automatically clear down the Silo, as it is likely that commands and data synchronisation will be lost at this point, thereby leaving an unknown sequence of commands and/or data which it is unable to differentiate between

Likewise, if a Test Point' command is issued and the result is returned to the Status Register, the result could easily be overwritten if an illegal command followed the original command, thereby losing the "Test Point' result. Due to this, a 'Silo Empty' Bit is provided in the Status Register. When a 'Test Point' command is issued it is recommended that no further commands be sent until the 'Silo Empty' bit is set and the result in DO is valid.

It should be noted that when using a Silo of this capacity, that VBUG operations will continue until the Silo is empty even though the host computer has ceased operations.

VBUG Status and Command register

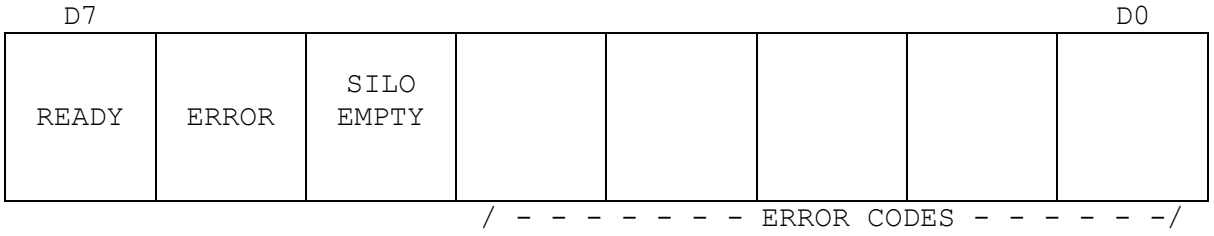
$BE00 – Status Register – Read

$BE01 – Command Register – Write

The host computer is able to write commands or data to the I/O address $BE01 only after ensuring that the Video module processor is free to accept that data. The operation is carried out by monitoring the Status Register until D7 is set. Thereafter the commands or data may be written to the Command Register, and there by way of the Silo, for processing. NOTE:- Commands or Data written to the Command Register before D7 is set will cause that information to be lost.

$BE00 – Status Register

Read Only



ERROR CONDITION

If an Error condition occurs either from previous programming or a processor busy condition, D6 in the Status Register will be set at the same time as D7. If this occurs,there will be an error code in the Low Order bits of the Status Register. The 6502 'Bit' instruction provides a convenient way of testing both the Ready and Error flags in the Status Register.

Example:-

```
WAIT: BIT SBEOO ;Test Status
      BPL WAIT ;Loop if not Ready
      BVS ERROR ;Branch if Error
      STA $BE01 ;Else write Data
```

COMMAND STRUCTURE

The Command Structure falls into three groups, these are as follows:-

1. SINGLE BYTE COMMANDS

2. GRAPHIC

3. ESCAPE SEQUENCES

1. SINGLE BVTE COMMANDS

The bulk of these commands will come in the form of the Ascii character set within the Hex codes $20 - $7F. A command byte within this range on the display will cause that Ascii character to be echoed at the current text cursor position.

HEX	DECIMAL	COMMAND	ACTION
01	1	CNTRL A	Home Cursor
02	2	B	Insert Character on current line
04	4	D	?????????? [show test card]
06	6	F	Clears Graphic Screen
08	8	H	Cursor Left (Back Space)
09	9	I	Cursor Right
0A	10	J	Cursor Down (Line Feed)
0C	12	L	Clear Screen, Cursor Home
0D	13	M	Carriage Return, Auto Line Feed
0E	14	N	Reverse Video On
0F	15	O	Reverse Video Off
1A	26	Z	Cursor Up
7F	127		Delete Character Before Cursor
A0	160		Super Script
A1	161		Sub Script
B0	176		Delete to end of Line
B1	177		Delete to end of Screen
C0	192		Push Parameters on Stack
C1	193		Pop, pull Parameters from Stack
C2	194		Set Text Cursor to that of Graphics Cursor
C4	196		Write Data direct to VDU Memory, Screen and Character Set Ram
20-7E	32-196		Ascii Character Set, Print Character Move Cursor Right One Position
80-98	128-152		User Defined Characters

Note that Reverse Video on/off command remains active until changed.

2. GRAPHIC

Graphic commands fall into two groups, linear and angular.

The linear group require 3 bytes of data following the command, while angular commands require 6 bytes of data.

When using the linear draw commands the actual execution depends upon a mode flag which is set via an ESC sequence.

On power up the draw flag is set for absolute mode. In this mode the actual co-ordinates for drawing are used.

When the flag is set to the relative mode, via an ESC sequence then all drawing takes place to the current graphics pointer and in the 2's complement form.

EXAMPLE:-

If you wish to draw with an X co-ordinate of -1 and Y co-ordinate of -5 then the following data should be dispatched immediately after the draw command

```
FF    LOW BYTE X
FF    HIGH BYTE X
FB    LOW BYTE Y
```

LINEAR COMMANDS

HEX	DECIMAL	ACTION
1C	28	Set Pixel and X,Y On
1D	29	Clear Pixel at X,Y (Off)
18	24	Test Point at X,Y
15	21	Invert Point at X,Y
1E	30	Draw Line from Current Position to X,Y
19	25	Undraw Line from Current Position to X,Y
17	23	Invert Line from Current Position to X,Y
1F	31	Move Co-ordinate Pointer to X,Y

The four parameters must be given in the following order, which Horizontal 'X' axis - Vertical 'Y' axis

1. Low Byte X Co-ordinate within range (0 - FF)
2. Hi. Byte X Co-ordinate within range (o or 1)
3. Low Byte Y Co-ordinate within range (0 - FF)
4. Hi. Byte Y Co-ordinate within range (O Always 0)

ERROR CODE 2 = X Out of Range
ERROR CODE 3 = Y Out of Range

When line drawing, an invisible graphics 'PEN' is used to the co-ordinates, these co-ordinates are set to ZERO on a Power On, thereafter they will have the value of the most recent X and Y co-ordinates

EXAMPLE:-

$1E,$80,$1,$80,$0 will cause a line to be drawn from the current 'PEN' position to location x = 384 /10 and y = 128 /10 which will make this the new 'PEN' position.

ANGULAR COMMANDS

HEX	DECIMAL	ACTION
A2	162	Draw ARC
A3	163	Undraw ARC
A5	165	Origin of ARC

The above commands have the following structure:-

COMMAND STAL STAH EAL EAH R AR

STAL	=	Start angle low range	0 - 255	0 - FF
STAH	=	Start angle high range	0 - 1	0 - 1
EAL	=	End angle low range	0 - 255	0 - FF
EAH	=	End angle high range	0 - 1	0 - 1
R	=	Radius range	0 - 255	0 - FF
AR	=	Aspect ratio range	0 - 255	0 - FF

If the start or end angle exceeds 360 degrees the arc will not be drawn and the ERROR CODE 6 will be returned.

Note: Error codes are only returned whilst in parallel mode.

Radius is obvious, but some arcs may generate Out of Range X or Y co-ordinates at certain points on the arc. If this happens, the co-ordinates will be truncated to the appropriate screen edge.

The Aspect Ratio varies the width/height ratio of the arc enabling ellipses to be drawn. A value of 12 will cause the true co-ordinates to be plotted. A value of 16 will give the appearance of a true circle on the screen.

Larger values widen the arc, whereas lower values heighten it. A value of O will cause a straight line to be drawn.

3. ESCAPE SEQUENCES

Escape sequences are used to set up system parameters that seldom require changing, i.e. blinking cursor or character size.

All escape sequences are ASCII coded and must be dispatched without intervening spaces.

ESC[PN b	PN = 0 PN = 1 PN = 2	Non Blinking Cursor Blinking Cursor Cursor Off
ESC[PN u	PN = 0 PN = 1	Underlining Off Underlining On
ESC[PN w	PN = 1-5	Character Size
ESC[PN d	PN= 128-152	User Defined Characters
ESC[PN l	PN=0 PN=1	No Auto Line Feed Auto LF after CR
ESC[PN i		Set Circle Increment to PN
ESC[PN1;PN2 c	PN1 = 1-24 PN2 = 1-80	Move Cursor to Row PN1 - Column PN2
ESC[PN;PN r	PN = 1-24	Scrolling Window from Row PN to Row PN inclusive
ESC[PN;PN h	PN2 = 1-80	Set Window Width
ESC[PN a	PN = 0 PN = 1	Draw Absolute Draw Relative to current X,Y
ESC[PN s	PN = 1-4	Select Character set
ESC q		Bi-directional printing
ESC x		Page/Scroll Mode (Toggle)

Summary

b = Blinking
u = Underlining
w = Character Width
d = Define Character
1 = Line Feed Control
c = Cursor Positioning
r = Scrolling Window
h = Window Width
a = Draw Control
s = Character Set Selection
i = Circle Increment
q = Bi-directional Printing
x = Page/Scroll Toggle

THE SCROLLING WINDOW

VBUG allows a section of the screen display to be set aside for a Scrolling Window. In perspective, the entire screen is in fact the true window at all times unless defined otherwise by this facility under ESC command.

For text purposes, the window will be set to the area from between row "TOP" to row "BOTTOM", thereafter, all commands such as Clear Screen and Cursor movement will take place inside this window area only. Data which has been set up outside this area will remain unaffected.

The width of the scrolling window can also be set via ESC command, the rules apply as above.

```
ESC[ 0;24 r      :Set window to whole screen lines O - 24
ESC[ 0;10 r      :Set from line O - line 10
ESC[ 10;24 r     :Set from line 10 - line 24

ESC[ 0;B0 h      :Set window width to full screen, column O - 80
ESC[ 10;70 h     :Set window between columns 10 - 70
```

When a window has been defined, all the text will scroll off the screen at the pseudo top line. On system Reset or power on, the window will be set to full size.

If the window size is altered from normal screen size, the new co-ordinates will remain until the operator changes them, either by issuing new co-ordinates or by issuing a System Reset. Graphics remain unaffected and legal throughout the entire screen display area irrespective of the window size setting.

ON LINE

When power is applied to the host computer and/or a Reset issued to the system, the Video modules processor will be initialised along with the VBUG monitor. This will result in a Clear Screen, displayed at top left, the message VBUG V6 (PTL), followed by the word Monitor and Flashing Cursor. Should the screen remain filled with random high resolution graphic bit patterns, then issue system reset, this should not normally be necessary. Once the Video Module messages have appeared at the top left of the screen, then the Video module has initialised and is On-Line awaiting further instructions from the host computer. The Video module is now subject to its Operating System Rules which govern the communications link between itself and the host computer.

The communication link is via the two User selected I/O locations i.e. $BE00 - Status Register and $BE01 – Command Register.

An On-Line test may be provided by sending a single byte instruction to the Command Register. A convenient single byte instruction is $0C - Clear Screen.

MBE01,0C, (CR) :Clear Screen instruction to Command Register.

Further testing may now be carried out with this simple keyboard interface program.

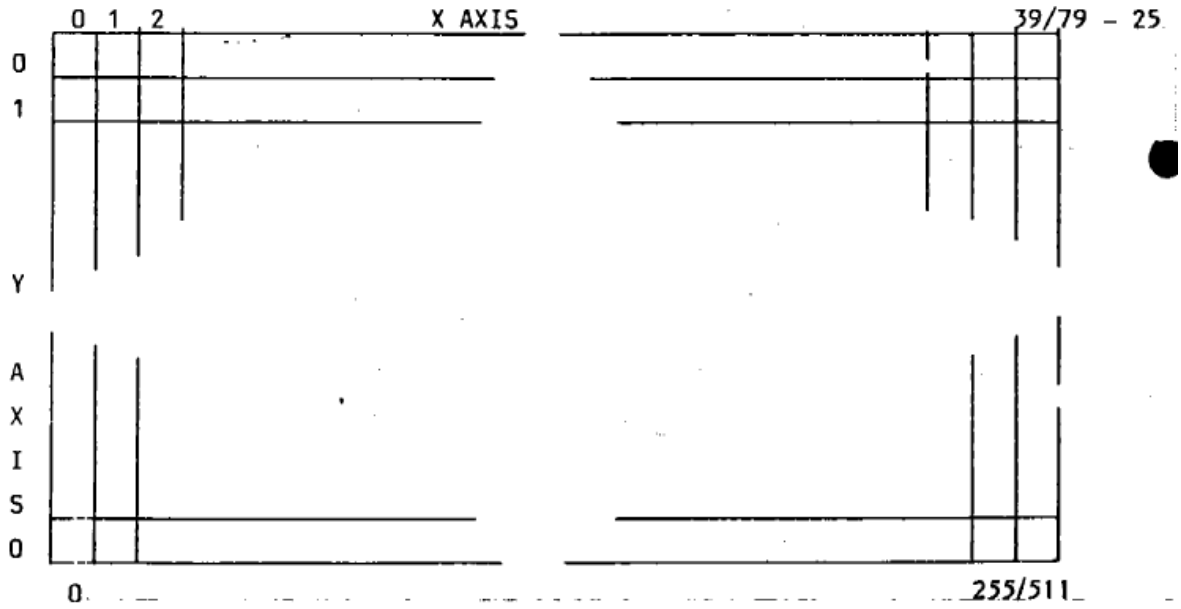
```
START:     SEI
LOOP:      LDA $BFF3 ;Read Keyboard
           BPL START+1 ;Do it
           AND #$7F ;Mask top bits
           STA SBFFO ;Clear KB flag
           STA $BE01 ;Vbug command register
           JMP START+1 ;Do it again
```

All data now typed in from the keyboard will be directed to the Vbug Command Register for processing. _CTRL commands may also be sent, likewise the ESC sequence commands. Remember care should be taken as this routine does not read the Vbug Status Register.

- HAVE FUN -

THE VIDEO DISPLAY

There are two displays to be considered with the Video module, one for Text and the other for Graphics. Taking the Text display first, it must be noted that the co-ordinates of the rows and columns that are referred to in this manual are in accordance with the accepted standard for text handling. Graphic display co-ordinates are slightly different being that the 0,0 X,Y co-ordinates commence from the bottom left hand corner of the screen. Any programming by the user should be guided by these standards to maintain compatibility with commercial software/firmware packages.



TEXT MODE 40/80 Character Mode

The co-ordinates are given as Line PN by Column PN.

If Line PN = O and Column PN = 0 then those co-ordinates would place the text cursor on the top line in the leftmost column.

If Line PN = 0 and Column PN = 2, then this would place the text cursor on the top line in the third column.

GRAPHICS MODE

In the graphics mode the bottom left hand corner co-ordinates commence from the bottom of the screen. The X and Y axis remain legal in both cases.

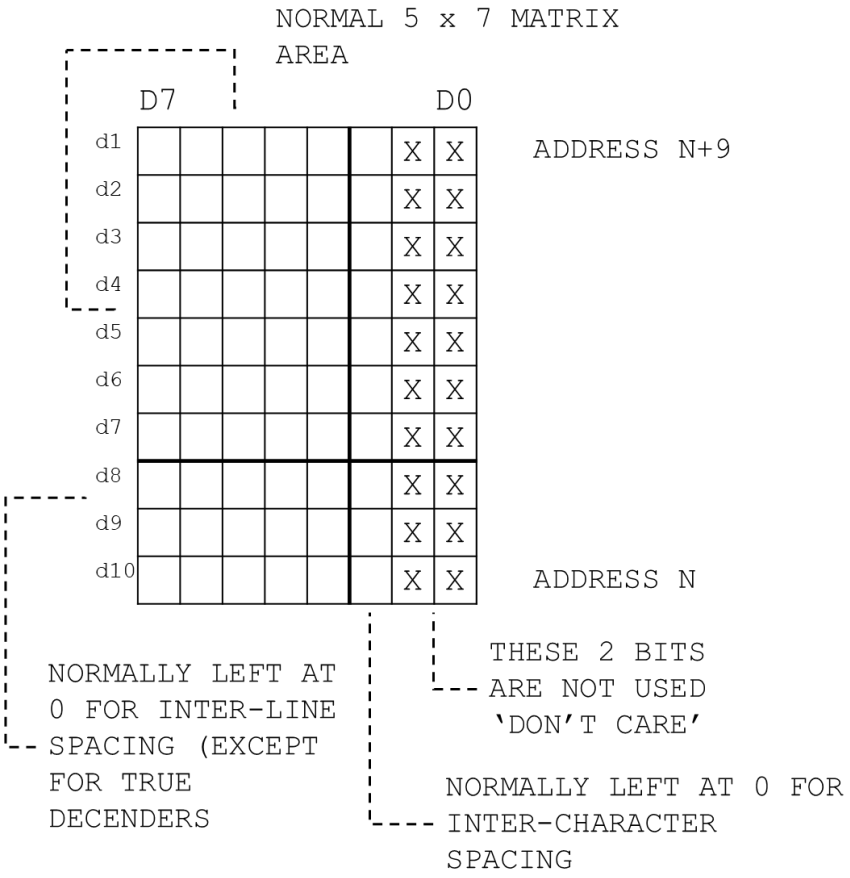
If X axis PN = 0 and Y axis PN = 0, then these co-ordinates would place the Graphics pen in the bottom left hand corner of the screen.

If X PN = 511 and Y PN = 255, then these co-ordinates would place the Graphics pen in the top right hand corner of the screen.

THE ASCII CHARACTER SET

The full standard 96 Ascii character set has been provided in firmware format within the VBUG Operating System Monitor, also there is a user option for the operator to create his/her own character set. The Ascii characters supplied are formatted on a 5 x 7 matrix pixels and include the required Inter-Line and Inter-Character spacing. The character matrix is shown below.

The character set table is at Rom address $FOOO - SFC2F and commences with the definition of the Ascii space code $20.



[This is the end of the section reproduced verbatim from the original manual]

Appendix A – Design Notes

Memory Changes

The original video memory used 8 x 6116 2K x 8 SRAMs and a 74LS138 3 line to 8 decoder to decode address bit 11 to 13 to select the appropriate device.

For 80/82-R these have been replaced with 2 X 6264 8K x 8 SRAMs. The 74LS138 has been removed and address bits 11 and 12 are routed to the address lines of the RAM devices, bit 13 is routed to the active high chip select CS2 of U35 and the active low chip select of U33 to select the devices. For full JEDEC compatibility pin 1 which is nc on the 2764 but A14 on a 6256 32K x 8 SRAM is connected to VCC. Pin 26, which is CS2 on the 6264 is A13 on the 62256 and so a single 62256 in U35 will appear as a 16K device and replace the 6264s in U35 and U33

The original system RAM was provided by a single 6116. A11 was not decoded and so the device would appear twice in the memory map.

For 80/82-R the 6116 has been replaced by a 6264 8K x 8 SRAM with the A11 line connected and A12 connected to ground to give 4K of RAM. Pin 26 (A13 / CS2) and pin 1 (A14 / nc) are connected to VCC so a 62256 can be used in the socket.

The original EPROM was a 2764 8K x 8. A13 was not decoded and so the device would appear twice in the memory map.

For 80/82-R A13 has been routed to the EPROM to allow a 27128 to provide 16K of EPROM. For Full JEDEC compatibility pin 1 (VPP / A14) and pin 27 (PGM / A15) have been connected to VCC so that a 27256 or 27512 can be used.

Timing changes

The address decoding is performed by IC15 a 74LS139. The chip enable outputs of this device are gated by the enable input which is connected to Ø1 so they are asserted when Ø1 is low. Because the address lines are shared with the video circuitry, they only carry the address from the CPU when Ø2 is high. On a 6502 Ø1 and Ø2 are deliberately timed by the CPU not to overlap with at least 5ns gap. This means that in the original design the chip enable outputs to the memory devices from IC15 are enabled before the address lines are stable and inevitably have glitches causing false enables in the first few nanoseconds of the period Ø2 is high.

The original board uses an additional clock input to IC16 at double the CPU clock to gate the R/W signal such that it only valid for the second half of the period Ø2 is high. This was presumably added to ensure that the glitches created clocking IC15 from Ø1 do not create false writes to the RAM.

By only setting R/W low for the second half of the time Ø2 is high, the 8212s and memory devices perform a read operation in the first half of Ø2 followed by a write, but the CPU is also driving the bus during the read so bus contention can occur. On the prototype this contention was observed and lead to unpredictable behaviour.

For 80/82-R the enable input from U15 is driven from Ø2 through an inverter. Using Ø2 delayed one gate delay by the inverter ensures that the address line input to U15 will be stable before the enable is asserted and prevents any glitches being present on the memory device enable lines.

With the address decode glitches removed there is no longer a need to delay R/W and so the additional clock input has been deleted and the 'read then write' contention is removed.

Note: The original board parts list stated a requirement for 150nS RAM devices even though the cycle would have been about 300ns, presumably this was because the write cycle had been halved. In fact the write pulse width requirement stated on the datasheet is only 90ns for the 150ns version of the device, so this was very conservative. This possibly explains why some 150ns devices worked at double clock speed.

WDC65C02 Changes

Pin 1 of an original 6502 is identified as Vss (GND)on the WDC65C02 this pin is an output VPB and so a link has been added to allow this to be disconnected.

Pin 36 is not connected on the original 6502 but is BE (Bus Enable on the WDC65C02 and so has been connected to VCC

Pin 38 SO (Set Overflow – active low) is on both the 6502 and WDC65C02. This pin is not mentioned on the circuit diagram of the original board but has now been connected to VCC.

On a 6502A the timing reference is based on Ø2 output from the CPU. Address and Data out from the CPU will be held stable for a few nanoseconds after the falling edge of Ø2 and as R/W is gated by this signal address and data will be stable until the end of the write cycle.

With the WDC65C02 the timing reference changes to the Ø0 input to the CPU. Address and Data out are stable only until the falling edge of this signal which occurs before the falling edge of the Ø2 output. On the prototype when using 55ns memory chips this caused occasional errors at the end of the write cycle.

The R/W logic is now gated by U16B such that the write cycle requires both Ø0 input and Ø2 output to be high which satisfies the timing requirements for any version of the 6502.

6MHz Operation

The prototype functions well at 3MHz in line with the modifications published in TANDOC2. With 55ns RAM and 45ns EEPROM devices available, and current generation 65C02 devices capable of operation at 14MHz it is interesting to consider if the board could be run at 6MHz.

All memory access must occur whilst Ø2 is high, which is half the clock cycle. For compatibility with 65C02 memory access must be completed before the falling edge of Ø0 which occurs 8 – 10 ns before the falling edge of Ø2. This results in a memory decode and access time widow of slightly less than half the cycle time.

Measurements taken from the prototype give the memory decode times with 74LS logic for the System RAM as 32ns, EPROM 37ns and the 8212 I/O devices 48ns. The following table shows the device access times at various clock speeds (the window minus the decode time) [Note these times are approximate, the logic probes used apply an additional 8pF of load which increases switching time]

Timing with LS series address decode (ns)	1.5MHz	3MHz	6MHz
Ø2 AND Ø0 window	325	158	76
System RAM Access time	293	126	45
EPROM Access time	288	121	40
8212 Read (U4 MD Pin)	277	110	28

The Mitsubishi M5L8212P device datasheet states 45ns from MD to data out. The Intel datasheet does not specify this time but other timings are similar.

Most of the decode delay is contributed by U15 the 74LS139 however replacing this with a F series part to reduce this time allows switching glitches from the 74LS157 devices (U13,14,20,21) through to the chip selects even if these devices are changed to F devices.

The 8212 decode time is longer than the memory chips because of the three gates U6, U16 and U8. Replacing these with F series devices reduces the decode time considerably and in this configuration the prototype operated at 6MHz. The timings at 6MHz are almost certainly just outside the specification for several of the devices and so reliable operation may not be possible with every combination of devices.

Although there is a 6MHz clock available on pin 14 of U5 this does not have a 50% duty cycle due to the imprecise nature of the clock doubler. The 6MHz Microtan clock from TANBUS is therefore used for the 6MHz jumper.

Other changes

The original circuit diagram has an error in that it shows the enable pin of the four 74LS 157 devices IC13, IC14, IC20 and IC21 connected to +5v. This enable input is active low and so clearly should be connected to GND.