www. microtan. UKPC.net

VIDEO 80/82 MODULE

Incorporating

VBUG V6 (PTL)

No part of this software package may be copied or reproduced in any way without the prior written permission of the author.
(C) MICROTANIC COMPUTER SYSTEMS LTD. 1984

********** IMPORTANT **********

If you have purchased the Video Module for use in the 40 column Mode, you will have been supplied with the

VBUG 1.2 OPERATING SYSTEM

Please note as VBUG 1.2 is less comprehensive than than VBUG V6 (PTL), then the following commands listed in your manual are INVALID:-

SINGLE BYTE COMMMANDS

HEX	DECIMAL	COMMAND	ACTION
02	2	CNTRL B	Insert character on current line
04	4	CNTRL D	????????????
B0	176		Delete to end of line
B1	177		Delete to end of screen
C0	192		Push parameters on stack
C1	193		Pull parameters from stack
C2	194		Set text cursor to that of graphics cursor
80-98	128-152		User defined characters

GRAPHIC

ALL ANGULAR COMMANDS

ESCAPE SEQUENCES

ESC[2 b	Cursor off
ESC[PN l	Line feed control
ESC[PN i	Set circle increment
ESC[PN;PN h	Set window width
ESC[PN a	Draw
ESC[PN s	Select character set
ESC q	Bi-directional printing
ESC x	Page/scroll mode

ALSO NOTE THAT THE CORRESPONDING COMMANDS IN THE VIDEO TOOLKIT ARE ALSO INVALID AND SCREEN EDITING MUST BE DONE ON THE MICROTAN SCREEN.

VIDEO 80/82

PREFACE

The Video 80/82 Module is an 'Intelligent' video terminal capable of complex activities and functions limited only by the imagination and programming skills of the individual. This manual contains the fundamental rules of operational use of the Video module and its VBUG V6 (PTL) Operating System Monitor.

Alternative Operating Systems may be used and can completely change the operational characteristics of this module to a degree outside the scope of this manual.

Primarily, this manual is concerned with the operation and use of the VBUG V6 (PTL) Operating System. This complex and powerful monitor is at the heart of this innovative hardware configuration. Coupled together they provide an advanced terminal dedicated towards graphics or video displays.

As this module requires the host computer to have a software driver routine it is recommended that TUGBUG is used as the Microtan Operating System, as it automatically communicates with this Video module.

Due to the high resolution output of this module it is recommended that the video output be coupled to a Video Monitor rather than a RF Device, thus ensuring optimum performance.

Constructional advice has been kept to a minimum as it is assumed that the purchaser of such a complex module has the required experience to complete assembly without the need for repetition of the basic rules governing construction.

Brian Gibbs
27.09.84

CONTENTS

VBUG V6 (PTL) OPERATING SYSTEM

The Video 80/82 Operating System is supplied in a 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 addition 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 displays to be built, limited only by the skill and imagination of the user.

By using an eariler version of VBUG it is possible to run the Video module in an 40 character mode (covered later in this manual).But normally it is accepted that the Video module will be fully expanded thus operating in 80 character mode. Therefore througout this manual it will be assumed that the mode of operation will be 80 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 allowing 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 theGraphics 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 remaing in the 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.

The Silo is automatically cleared on a Reset.

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 eturned to the Status Register, the result could easily be overwritten if an illegal command followed the orginal 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 usig a Silo of this capacity, that VBUG operations will continue until the Silo is empty even though the host computer has ceasd operations.

VBUG STATUS & 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 proceesing. 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

D7							D0
READY	ERROR	SILO EMPTY					

/ - - - - - - - - ERROR CODES - - - - - /

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 $BE00 ;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 BYTE 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 will cause that Ascii character to be echoed on the display 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	??????????
06	6	F	Clears Graphics 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	32-126		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 at 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 " " " "
17	23	Invert Line from " " " "
1F	31	Move Co-ordinate Pointer to X,Y

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

1. Low Byte X Co-ordinate within range		(0 - FF)
2. Hi. Byte X	" " "	(0 or 1)
3. Low Byte Y	" " "	(0 - FF)
4. Hi. Byte Y	" " "	(OAlways 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 position the co-ordinates, these co-ordinates are set to ZERO on a Reset or 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 elipses 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 0 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	Non Blinking Cursor
	PN = 1	Blinking Cursor
	PN = 2	Cursor Off
ESC[PN u	PN = 0	Underlining Off
	PN = 1	Underlining On
ESC[PN w	PN = 1-5	Character Size
ESC[PN d	PN = 128-152	User Defined Characters
ESC[PN l	PN = 0	No Auto Line Feed
	PN = 1	Auto LF after CR
ESC[PN i		Set Circle Increment to PN
ESC[PN1;PN2 c	PN1 = 1-24	Move Cursor to Row PN1 -
	PN2 = 1-80	Column PN2
ESC[PN;PN r	PN = 1-24	Scrolling Window from Row PN to Row PN inclusive
ESC[PN;PN h	PN = 0-80	Set Window Width
ESC[PN a	PN = 0	Draw Absolute
	PN = 1	Draw Relative to current X,Y co-ordinates
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
l = 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 0 - 24
ESC[ 0;10 r :Set from line 0 - line 10
ESC[ 10;24 r :Set from line 10 - line 24

ESC[ 0;80 h :Set window width to full screen, column 0 - 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 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 $BFF0 ;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 -

I/O ADDRESS SELECTION

Four I/O selections of the host computer systems memory map are provided on the Video module. A summary of the address locations of the respective DIP switch setting are listed below.

Both the Status and Command registers are selected by the appropriate DIP switch.

I/O Address Selection - DIP Switch 'A'

SW pin No.	Function	ON	OFF
5	I/O Address $BE60 - $BE61		OFF
6	I/O Address $BE40 - $BE41		OFF
7	I/O Address $BE20 - $BE21		OFF
8	I/O Address $BE00 - $BE01	ON	

It is suggested that the module be selected for addresses $BE00 and $BE01 DIP switch No. 8 to maintain compatability with commercially available host computer software i.e. TUGBUG.

Interrupt Pin Mnemonics - DIP Switch 'A'

SW pin No.	Function	ON	OFF
1	VDU Interrupt Request	ON	
2	VDU Non Maskable Interrupt		OFF
3	Not used		OFF
4	Not used		OFF

For normal use it is recommended that the standard Video modules IRQ (Interrupt Request) line be used, unless specialist applications require the Non-maskable interrupt facilities.

OPERATIONAL CONFIGURATIONS

The Video module permits the choice of two modes of operation, although now with the introduction of the latest VBUG V6 (PTL) it is restricted to one mode.(80 COLUMN MODE)

40 Column Mode

This can only be used with earlier version Vbugs (NOT PTL).

40 characters per line by 25 text lines.
256 x 256 Bit mapped graphics display.

The minimum requirements for this mode are as follows:-

8K of Video Ram + 2K Operating System Ram.

80 Column Mode (Standard)

80 characters per line by 25 text lines.
512 x 256 Bit mapped graphics display.

Fully expanded Video Module required.

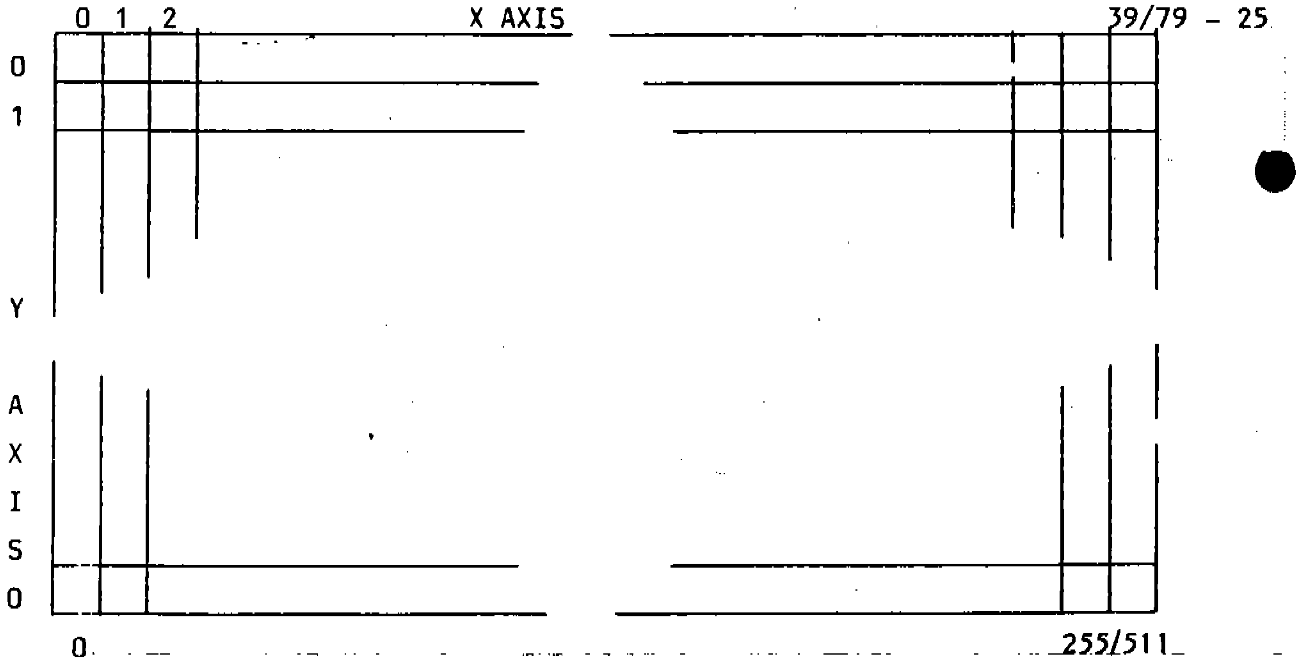
Operational Configurations - DIP Switch 'B'

SW pin No.	Function	ON	OFF
1	Not used		OFF
2	Not used		OFF
3	80 Column Mode 16K Ram	Optional	
4	Not used		OFF
5	Not used		OFF
6	Write Protect	ON	
7	40 Column Mode 8K Ram	Optional	
8	Video Select Disable	ON	

Included is the choice of Video 'ON/OFF' selection and a 'WRITE PROTECT' facility, both of which are assumed to be in the 'ON' position. CAUTION DIP Switch pin 3 and 7 must not be 'ON' at the same time.

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 refered 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 compatability with commercial software/firmware packages.



VIDEO SCREEN DISPLAY FORMAT

TEXT MODE 40/80 Character Mode

The co-ordinates are given as Line PN by Column PN.
If Line PN = 0 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.

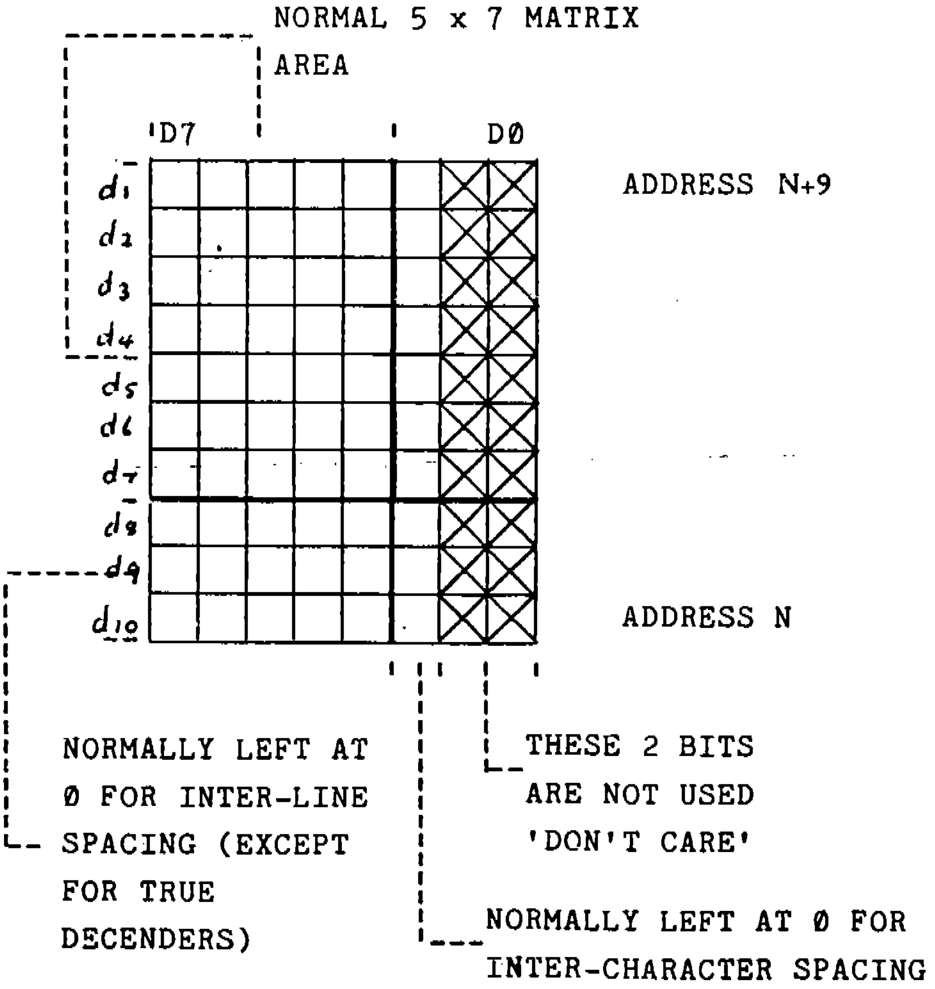
GRAPHICS MODE

In the graphics mode the co-ordinates commence from the bottom left hand corner 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 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 $F000 - $FC2F and commences with the definition of the Ascii space code $20.


NORMAL 5 x 7 MATRIX
AREA
D7
DØ
d1
d2
d3
d4
d5
d6
d7
d8
d9
d10
ADDRESS N+9
ADDRESS N
NORMALLY LEFT AT Ø FOR INTER-LINE SPACING (EXCEPT FOR TRUE DECENDERS)
THESE 2 BITS ARE NOT USED 'DON'T CARE'
NORMALLY LEFT AT Ø FOR INTER-CHARACTER SPACING

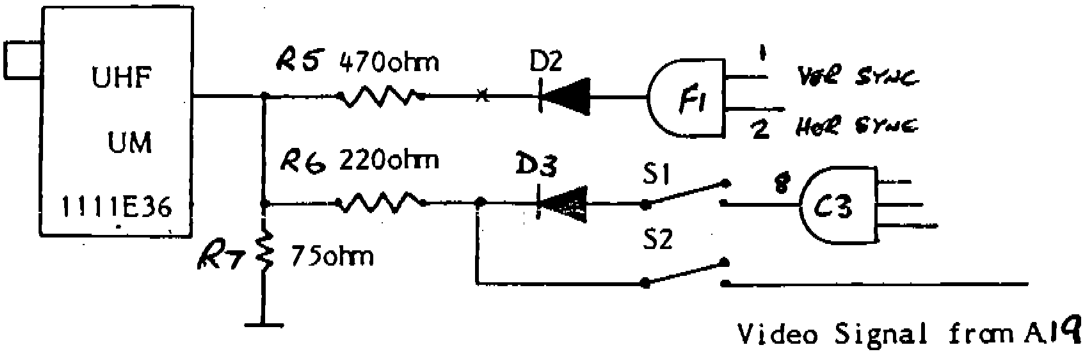
INSTALLATION & CONSTRUCTION

If you have purchased the Video module fully assembled it will have been preset to meet immediate service. All that is required by the user is the installation within the Microtan 65 system of the Video signal line. As Pin B19 on the additional slots of the system motherboard goes direct to Pin A19 on the Microtan slot, therefore it shall be deemed that this is the Video Signal line. It will be necessary to link B19 and B20 on the rear of the edge connector on the Video module.

Video Module:-
Video Signal Out - Pin B19.

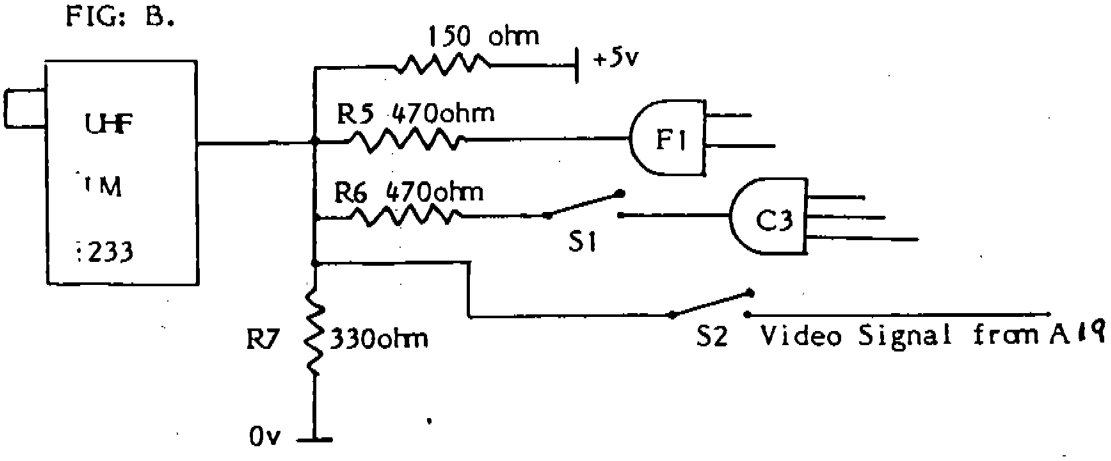
Microtan 65.
Video Signal In - Pin A19.

Video Sync picked up from Microtan 65 as shown below.



ISSUE 1. Low Bandwidth Modulator - Video Signal taken from acros 75 ohm resistor.
* If Monitor incorporates internal 75 ohm load - R7 should be removed.

FIG: B.



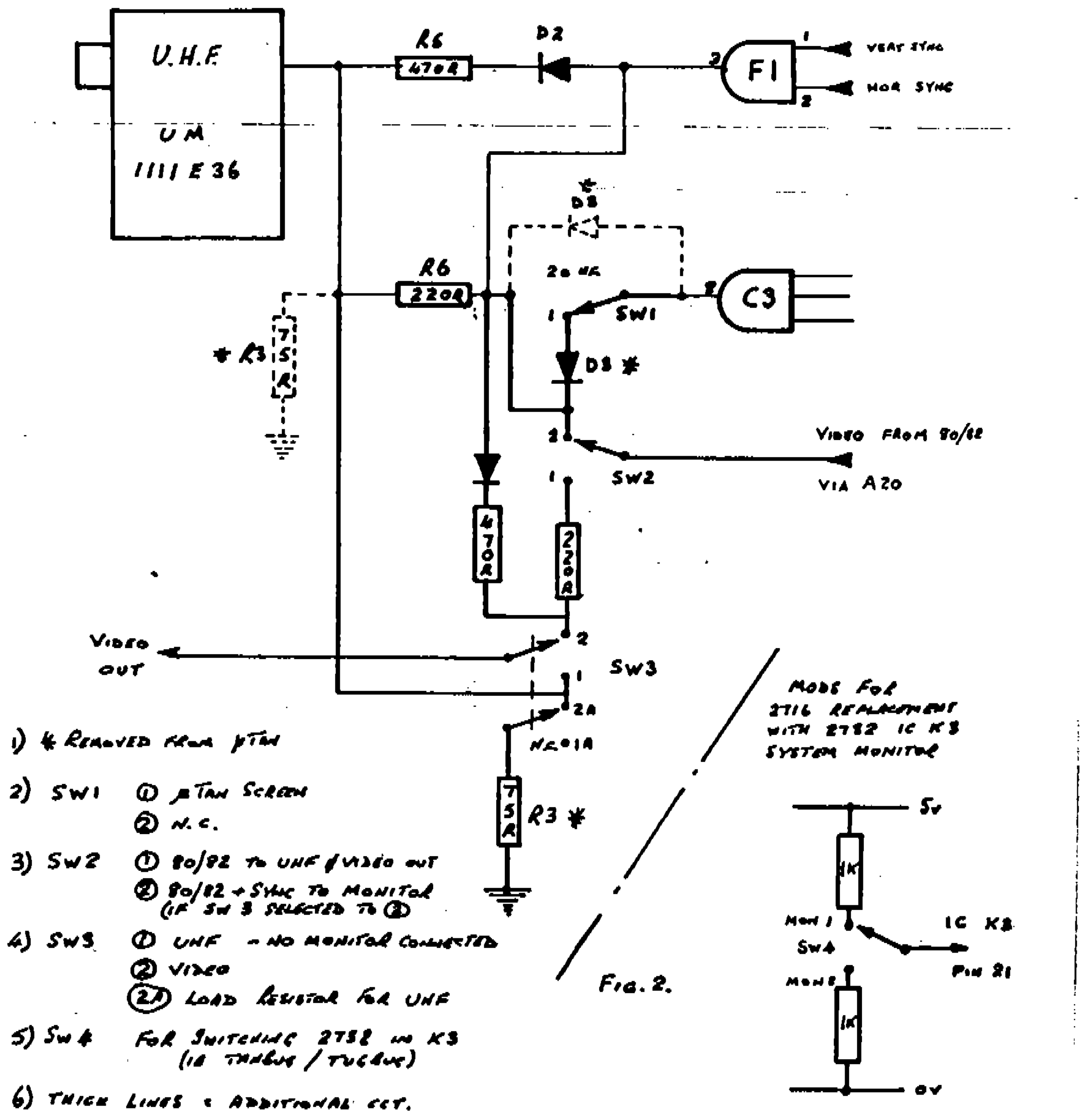
ISSUE 2. High Bandwidth Modulator - Video Signal taken from across 330 ohm resistor.
* D1 on Video module should be replaced with a 470 ohm resistor.

Switches S1 and S2 have been incorporated in the circuits to allow access to either video display individually or mixed.

Although a video signal is available through the UHF modulator, the quality of that display will be subject to varying degrees of performance and in some cases deteriorate to a point where the display is unreadable. Much will depend on the quality of the equipment used. We recommend therefore that Video Monitors be used for this purpose.

The above diagrams are of cicuits found to be in circulation. We cannot guarantee that your equipment will be the same as that shown above. Care should therefore be taken before connecting theVideo Module to your equipment. If in doubt refer to your systems manual or wiring diagrams.

OPERATIONAL μ65 CONFIGURATION



1) * REMOVED FROM μTAN

2) SW1 ① μTAN SCREEN
② N.C.

3) SW2 ① 80/82 TO UHF / VIDEO OUT
② 80/82 + SYNC TO MONITOR (IF SW 3 SELECTED TO ②)

4) SW3 ① UHF - NO MONITOR CONNECTED
② VIDEO
②A LOAD RESISTOR FOR UHF

5) SW4 FOR SWITCHING 2732 IN K3 (IE TANBUG / TUGBUG)

6) THICK LINES = ADDITIONAL CCT.

FIG. 2.

VIDEO 80/82 BUS CONNECTIONS

		HOST SYSTEM		
	b		a	
HOST SYSTEM	+5v	1	+5v	HOST SYSTEM SUPPLY
6MHz CLOCK	CLK	2		
		3		
SYSTEM RESET	RST	4	I/O	
ADDRESS BUS	A1	5	A0	ADDRESS BUS
" "	A3	6	A2	" "
" "	A5	7	A4	" "
" "	A7	8	A6	" "
" "	A9	9	A8	" "
		10		
		11		
		12		
		13	IRQ	INTERRUPT REQUEST
		14	NMI	NON-MASKABLE INTERRUPT
		15		
		16		
FIELD BLANKING	FB	17	R/W	READ NOT WRITE
		18	HB	HORIZONTAL BLANKING
VIDEO SMB	VIDEO	19	DB0	BUFFERED DATA BUS
VIDEO OUT TO HOST	VIDEO	20	DB1	" " "
		21	DB2	" " "
		22	DB3	" " "
		23	DB4	" " "
		24	DB5	" " "
		25	DB6	" " "
		26	DB7	" " "
		27		
		28		
		29		
		30		
		31		
EARTH RETURN	0v	32	0v	EARTH RETURN

NOTE: Originally b20. on the System Motherboard was used for Video signals, this has now been changed to b19, although the Video out from the Video module is still on b20. Therefore it is necessary to link b19 to b20 on the Video module should it not be already done.

£85

COMPONENT LISTING

I.C.No.		DEVICE		No.Pins	REMARKS
01	495	8212		24	Support Device
02		74LS 139		16	TTL
03		74LS 138		16	TTL
04		8212		24	Support Device
05	70	74LS 163	60	16	TTL
06	100	74LS 00		14	TTL
07		74LS 123	70	16	TTL
08		74LS 04		14	TTL
09		74LS 74		14	TTL
10	845	6502A	2Ma	40	CPU 2MHz
11		74LS 02		14	TTL
12		74LS 00		14	TTL
13		74LS 157		16	TTL
14		74LS 157		16	TTL
15		74LS 139		16	TTL
16		74LS 10		14	TTL
17	100	74LS 393	92	14	TTL
18		74LS 393		14	TTL
19		74LS 32		14	TTL
20		74LS 157		16	TTL
21		74LS 157		16	TTL
22		74LS 138		16	TTL
23	110	74LS 165	95	16	TTL
24		6116		24	Video Ram 150 nsec
25	160	74LS 245	88	20	TTL
26		6116		24	Video Ram 150 nsec
27		6116		24	Video Ram 150 nsec
28		2764 MONITOR		28	EPROM 250 nsec
29		6116		24	Video Ram 150 nsec
30		6116		24	Video Ram 150 nsec
31		6116		24	Video Ram 150 nsec
32		6116		24	Video Ram 150 nsec
33		6116		24	Video Ram 150 nsec
34		6116		24	2K Operating System Ram
35		6116		24	Video Ram 150 nsec

Watford

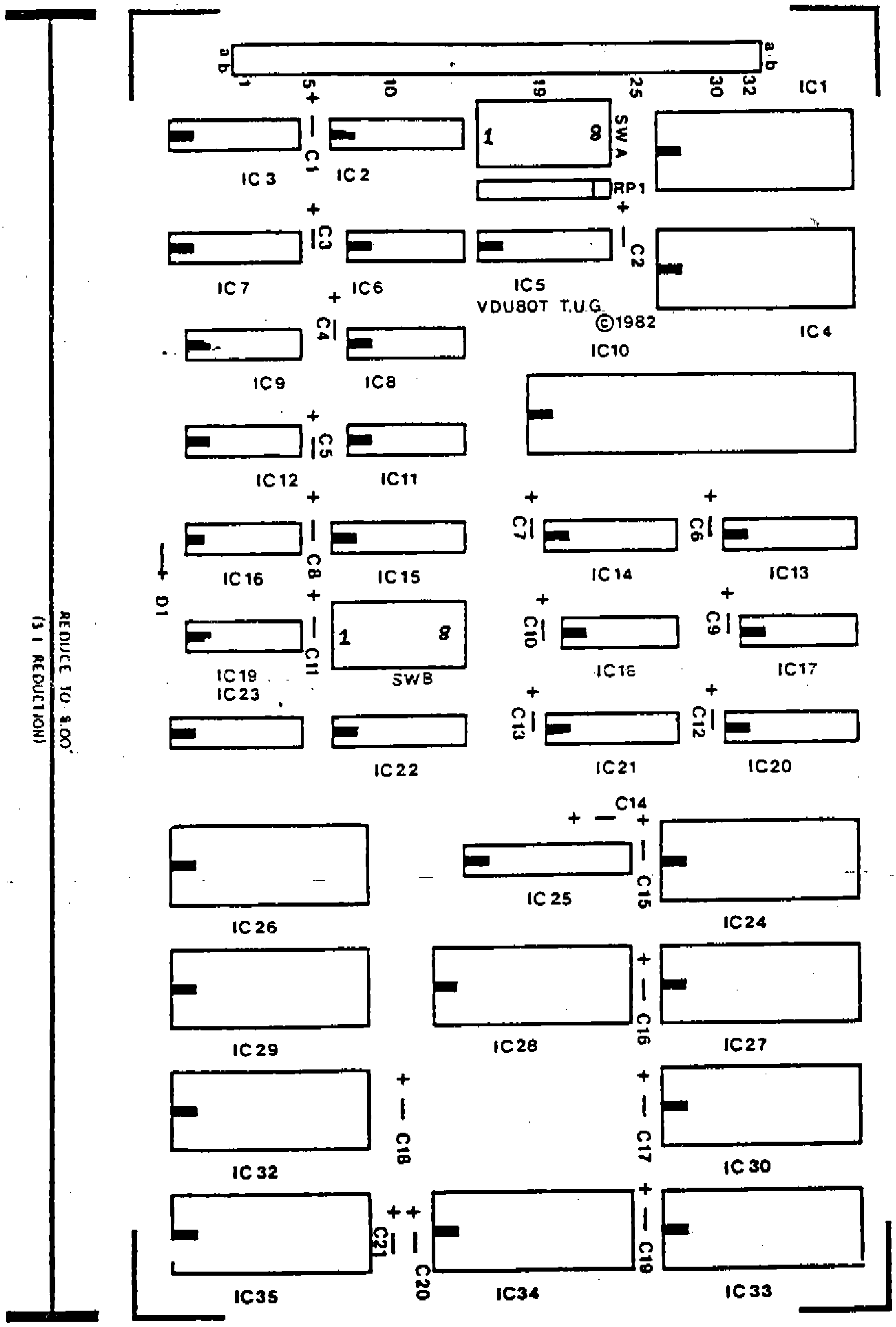
NOTE: 2016's may used instead of 6116's

ADDITIONAL

64 Way AB Edge Connector
1K ohm 0.25 watt 8 way Resistor Network (Including Common Pin)
Diode 1N4148
Sockets as indicated
2 x DIP Switch 8 way: Optional, hard wire links may be used
Capacitors - Disc Ceramic / Tantalum Bead C1 - C21 0.047uF 12 Volts

8K 40 Column mode 256 x256 requires I.C.s 30,32,33,35, installed.
16K 80 Column mode 512 x 256 requires all Ram installed.

1x 8212
1x 6502A
10 x 6116


a b
1
5+
10
19
25
30
32
a b
IC1
SW A
1
8
IC 3
C1
IC 2
RP1
C3
C2
IC 7
IC 6
IC5
VDU80T T.U.G.
©1982
IC4
C4
IC 9
IC 8
IC10
C5
IC 12
IC11
C7
C6
IC 16
C8
IC 15
IC 14
IC13
D1
C10
C9
IC 19
IC 23
C11
SWB
IC18
IC17
C13
C12
IC 22
IC 21
IC 20
C14
C15
IC 25
IC 26
IC 24
C16
IC 29
IC 28
IC 27
C18
C17
IC 32
IC 30
C21
C20
C19
IC35
IC34
IC 33
REDUCE TO 4.00"
(3:1 REDUCTION)

EXAMPLES

1) Print H_2O

Data sent to module:-

HEX	DECIMAL	REMARKS
48	72	Print H
A1	161	Command SUBSCRIPT
32	50	Print 2
~~50~~ 4F	80	Print O

2) Draw an arc from 25 to 60 degrees with a radius of 50 and aspect ratio of 16

A2	162	Command
19	25	STAL
00	0	STAH
3C	60	EAL
00	0	EAH
32	50	R
10	16	AR

3) Draw a circle with centre at 128,128 radius 40 and aspect ratio 16

1F	311	Command MOVE
80	128	LBX
00	0	HBX
80	128	LBY
A2	162	Command DRAW ARC
00	0	STAL
00	0	STAH
68	104	EAL
01	1	EAH
28	40	R
10	16	AR

4) Draw an elipse with a tilt of 23 degrees from the vertical

A5	165	Command ORIGIN
17	23	RAL
00	0	RAH
A2	162	Command DRAW ARC
00	0	STAL
00	0	STAH
68	104	EAL
01	1	EAH
50	80	R
04	4	AR

5) Set cursor to non blinking

1B	27	ESC
5B	91	[left hand square bracket
30	48	ASCII zero
62	98	b

6) Set character size to x 2

1B	27	ESC
5B	91	[
32	50	ASCII 2
77	119	w

7) Move cursor to row 10 column 18

1B	27	ESC
5B	91	[
31	49	1 row 10
30	48	0
3B	59	;
31	49	1 column 18
38	56	8
63	99	c

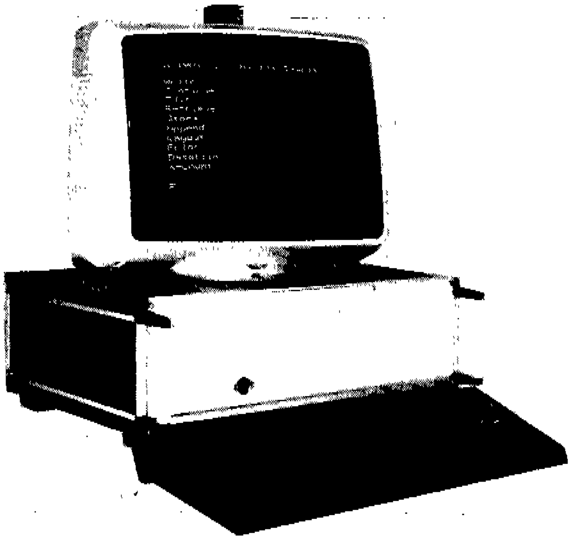
8) Set scrolling window from row 8 to 22

1B	27	ESC
5B	91	[
38	56	8 row 8
3B	59	;
32	50	2 row 22
32	50	2
72	114	r

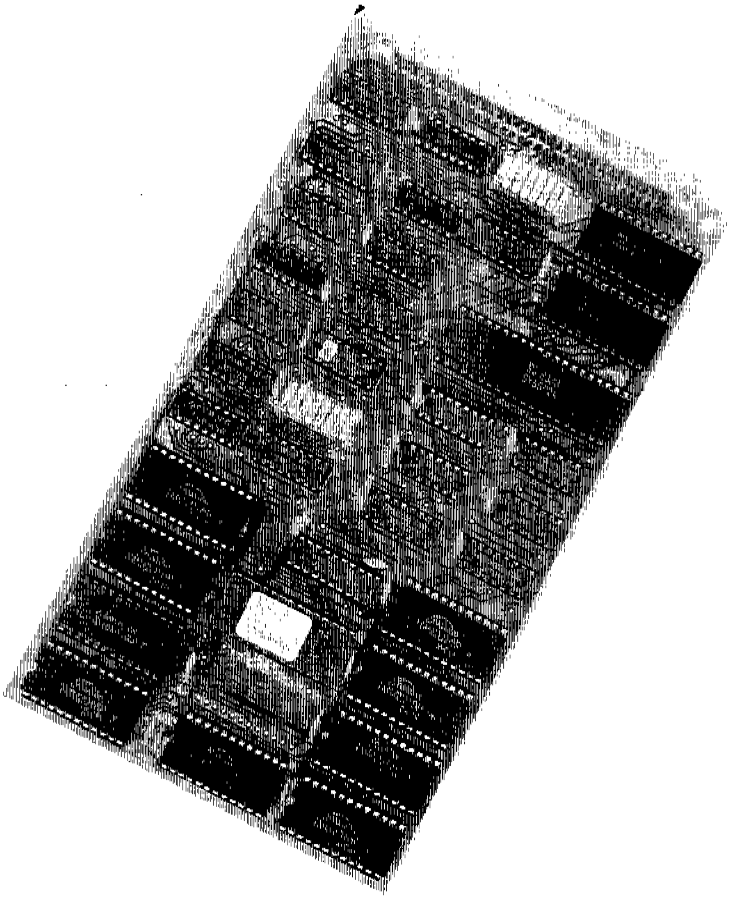
9) Set up the first user character, 128

1B	27	ESC
5B	91	[
31	49	ASCII 1
32	50	ASCII 2
38	56	ASCII 8
64	100	ASCII d

+ 10 bytes of data



80/82 VIDEO CARD



80/82 Video card

An 'intelligent' video terminal (EX–TUG) for use on the Microtan giving :–

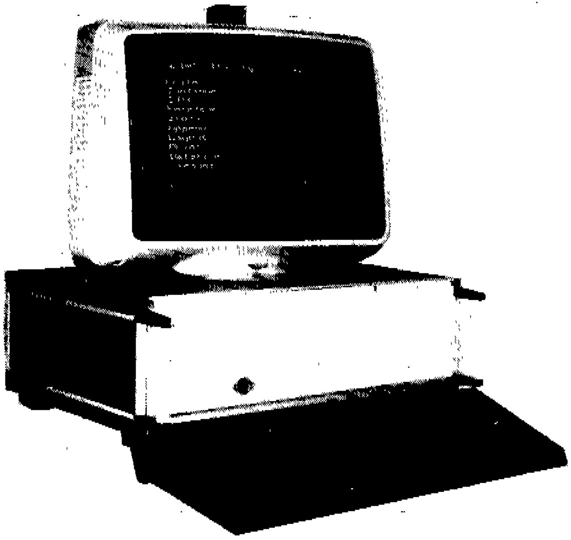
40 or 80 columns text
256 x 256 or 512 x 256 bit mapped graphics display
8K on board video RAM (4x6116)
for 40 column or 256 x 256 graphics
16K on board video RAM (8 x 6116)
for 80 column or 512 x 256 graphics
+2K operating system.

Includes a scrolling window and a 2K RAM for zero page in stack operations. The operating system is totally independant of the host computer.

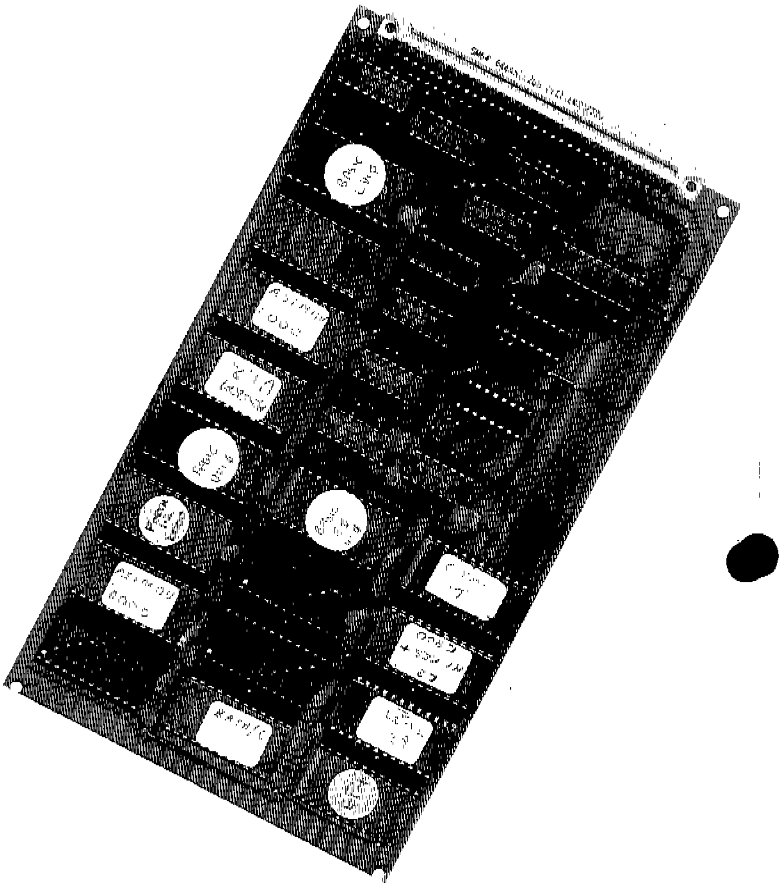
The full standard 96 ASCII Character set is produced on a 5 x 7 matrix pixels and include interline and inter character spacing.

Ideal for word processing ,printer and editing operation, graphics and general use. Excellent as a terminal for both the CP/M card and the 6809 single board controller.

The monitor allows a section of the screen to be set aside as a scrolling window. The size and position of this window can be defined and set. Once set 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 will remain unaffected. Text will scroll off the screen at the psuedo top line. On a system reset or on power up the window will be set to its normal size i.e. the full screen size. VBUG uses a 256 character S1LO 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 allowing it to continue other tasks whilst awaiting for the module to complete those instructions.



EPROM STORAGE CARD



EPROM STORAGE CARD

This very useful addition to the microtan 65 system is supplied as a high quality double sided plated thru' the hole, solder resisted screen printed circuit board. It can be purchased either ready built or as a bare board or kit.

With this board fitted to the system it is possible to select up to four Eproms out of a population of 16, from the keyboard (and also with care, from a program).

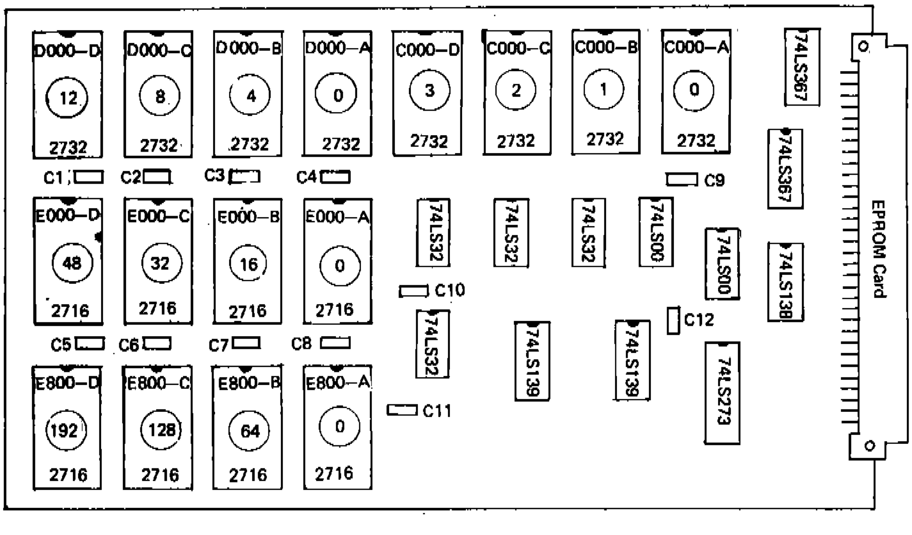
No modification to any other board is necessary other than removing the relevant Eproms from Tanex and transfer them to the new board. Part of the memory map located on Tanex is as follows:—

C000—CFFF	2732	Eprom	4K	Basic int
D000—DFFF	2732	Eprom	4K	Basic Interpreter
E000—E7FF	2716	Eprom	2K	Basic Int
E800—EFFF	2716	Eprom		User choice

The ESC also covers this part of the memory map but allows the user to select a particular combination of Eproms, one of four for each bank. Each Eprom is given a pixel number and the sum of the numbers of those chosen is written to a write only latch located at C000 (HEX) on reset there is a default position, bank A and this usually accommodates Basic.

So if you cannot afford to purchase discs this is a very versatile way of storing programs.

No Eproms supplied with this board


D000—D
12
2732
D000—C
8
2732
D000—B
4
2732
D000—A
0
2732
C000—D
3
2732
C000—C
2
2732
C000—B
1
2732
C000—A
0
2732
74LS367
C1
C2
C3
C4
74LS367
C9
E000—D
48
2716
E000—C
32
2716
E000—B
16
2716
E000—A
0
2716
74LS32
74LS32
74LS32
74LS00
74LS00
74LS138
EPROM Card
C10
C12
C5
C6
C7
C8
74LS32
74LS139
74LS139
74LS273
E800—D
192
2716
E800—C
128
2716
E800—B
64
2716
E800—A
0
2716
C11