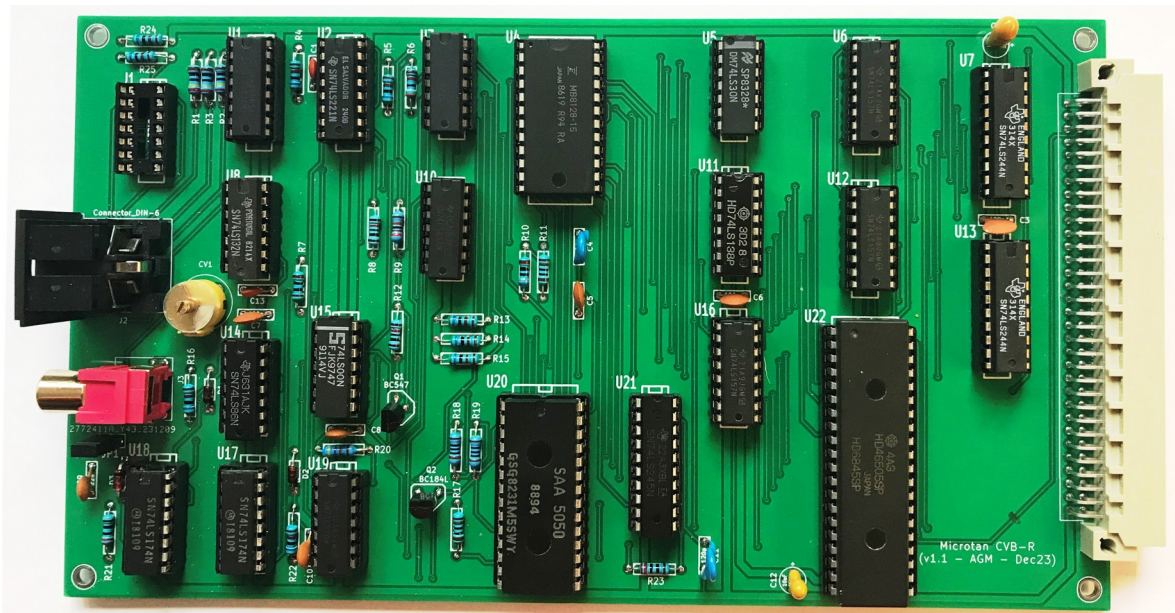


Microtan-R

CVB-R

Manual



Contents

Table of Contents

Contents.....	2
Specification.....	3
Requirements.....	3
Construction.....	4
Modification.....	4
Assembly.....	4
Layout.....	5
Bill of Materials.....	6
Installation.....	7
CVB Screen / Memory Organistion.....	8
Screen Layout.....	8
Microtan Screen Use.....	8
Configuration.....	9
CRT Controller Registers R10/R11.....	9
CRT Controller Registers R14/R15.....	9
Usage.....	10
Control codes.....	10
Character Generation.....	11
Graphics.....	11
BASIC usage.....	11
On screen editing in BASIC.....	12
Cassette Tape Routines.....	12
XBUG Translator and Disassembler.....	12
Appendix A.....	13
Output connections.....	13
J1 – 14 pin DIL socket.....	13
J2 – 6 pin DIN Socket.....	13
J3 – RCA Phono Socket.....	13
SCART Plug.....	13
Appendix B.....	14
SAA5050/55 Character/Control Codes.....	14

Specification

- Stable flicker-free 64 Teletext alpha-numeric or graphic character by 25 row display
- Video output is non-interlaced, 312 lines per 20ms frame.
- Monochrome composite and colour video
- Flashing, Inverse video, 6 colours + black & white characters
- Programmable cursor format & position
- Outputs for monochrome composite video and SCART/RGB colour monitors.
700mV positive output and 300mV negative (or positive) sync with 75 ohms load.
- Software switch to output to Microtan screen
- Compatible with TANDOS
- Compatible with Mouse Packet Colour Video Board Toolkit
- On-board memory mapped screen (\$A000-\$A63F)
- CRT controller registers (\$A7FE-\$7FF)

Note. In order to free up space on the TANBUG v3B EPROM to include specific CVB routines, it has been necessary to remove the Hexpad routines. It is not therefore possible to use the CVB with the Hexpad.

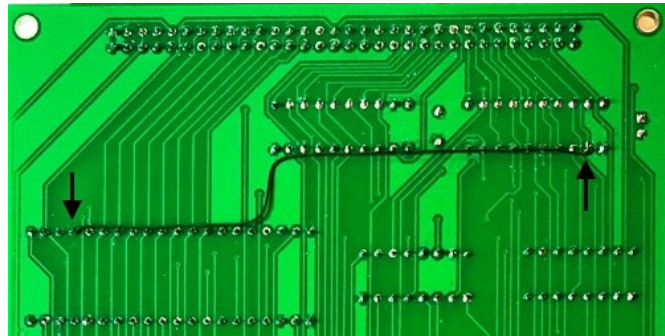
Requirements

- MT65 CPU with TANBUG V3B EPROM
- Spare Microtan system expansion slot
- SCART/RGB monitor or converter (eg Linkfor SCART-HDMI converter with HDMI monitor)
- Modified BASIC D3 EPROM – if BASIC programming is required

Construction

Modification

Before assembly of the Version 1.1 PCB, a modification to the PCB upper surface is necessary. The modification breaks the connection between U22 (6845) pin 24 and TANBUS O2 (pin a3). After assembly, a new connection is made between U22 (6845) pin 24 to U7 (74LS244) pin 2 on the lower surface of the PCB.

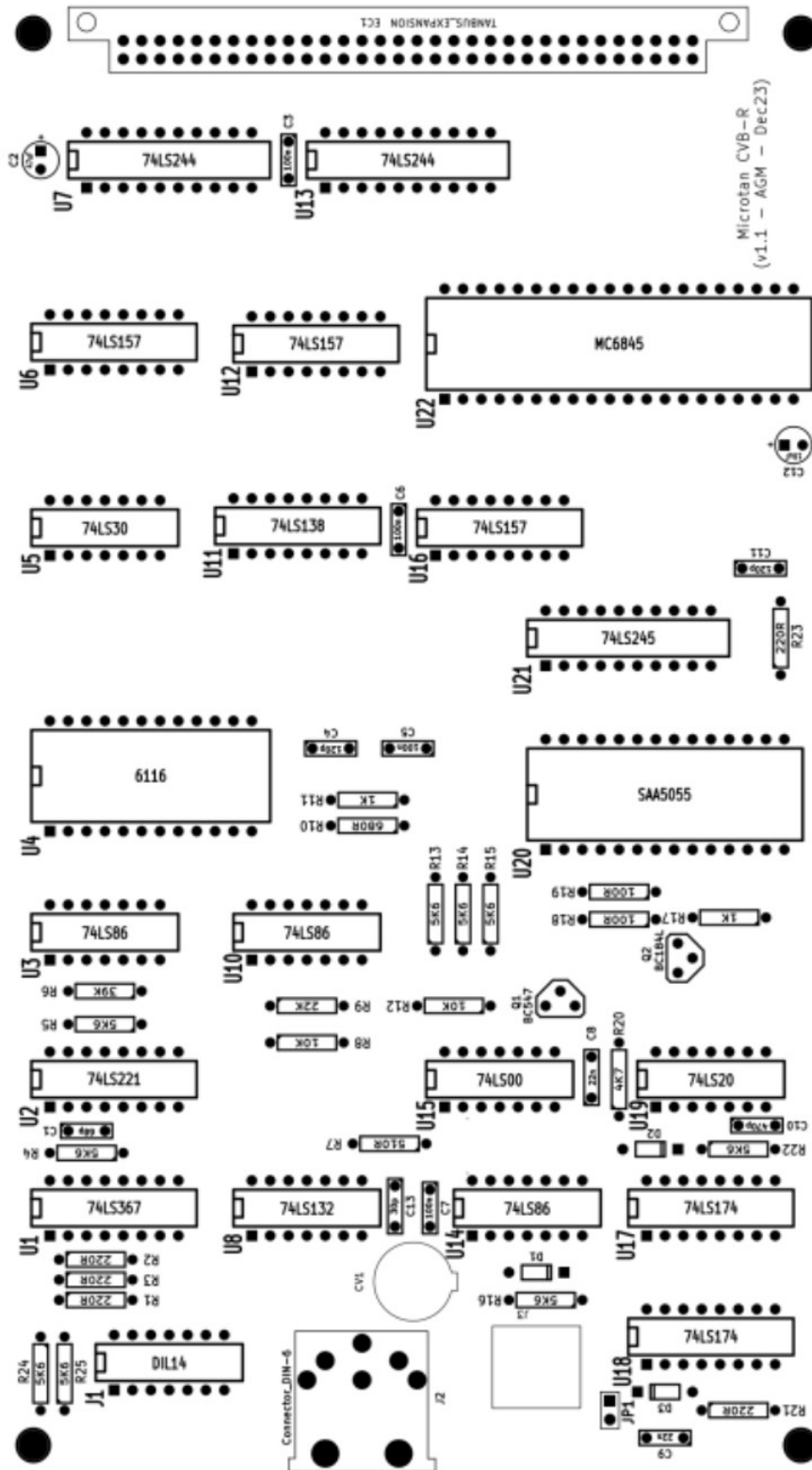


Assembly

Use the PCB Layout and the Components List below to identify the location of the components on the PCB. Ensure the transistors, electrolytic capacitors are orientated correctly. The longer leads on the electrolytic capacitors are the positive leads. Square pads on the PCB denote the location of the components' Pin 1 or positive lead as appropriate.

After all components and sockets have been soldered, check all IC pins are fully inserted into their sockets and check the underside for bridged tracks and leads/pins not soldered.

Layout



Bill of Materials

Designator	Qty	Designation	Package
C1	1	68p	C2
C2	1	47uF	C1V5
C3,C5,C6,C7	4	100n	C2
C4,C11	2	120p	C2
C9,C8	2	22n	C2
C10	1	470p	C2
C12	1	10uF	C1V5
C13	1	30p	C2
CV1	1	5-65pF	BFC2808
D1,D2	2	1N4148	DO-35
D3	1	5v6 Zener	DO-35
EC1	1	TANBUS_2x32 (a+b)	DIN41612 R/A Plug
J1	1	DIL14	DIP-14__300
J2	1	DIN-6	240_DIN_6p
J3	1	RCA PCB socket	PSG01546
JP1	1	JUMPER	PIN_ARRAY_2X1
Q1	1	BC547	TO92_vert
Q2	1	BC184L	TO92_vert
R1,R2,R3,R21,R23	5	220R	R4
R4,R5,R13,R14,R15,R16,R22,R24,R25	9	5K6	R4
R6	1	39K	R4
R7	1	510R	R4
R8,R12	2	10K	R4
R9	1	22K	R4
R10	1	680R	R4
R11,R17	2	1K	R4
R18,R19	2	100R	R4
R20	1	4K7	R4
U1	1	74LS367	DIP-16__300
U2	1	74LS221	DIP-16__300
U3,U10,U14	3	74LS86	DIP-14__300
U4	1	6116	DIP-24__600
U5	1	74LS30	DIP-14__300
U6,U12,U16	3	74LS157	DIP-16__300
U7,U13	2	74LS244	DIP-20__300
U8	1	74LS132	DIP-14__300
U11	1	74LS138	DIP-16__300
U15	1	74LS00	DIP-14__300
U17,U18	2	74LS174	DIP-16__300
U19	1	74LS20	DIP-14__300
U20	1	SAA5055	DIP-28__600
U21	1	74LS245	DIP-20__300
U22	1	MC6845	DIP-40__600

Installation

Replace the monitor EPROM on the MT65 CPU board with TANBUG v3B.

If using an original TANEX and BASIC is required, replace the BASIC D3 EPROM with the CVB modified D3 EPROM.

Plug the CVB-R into a spare expansion slot in the Microtan system.

Connect CVB J1 or J2 to either:

- RGB monitor,
- SCART TV, or
- SCART to HDMI converter + HDMI monitor

Output connections are detailed in Appendix A.

Note if a positive-going CSYNC signal is required to display a stable picture, remove the jumper from JP1.

Switch on power to the Microtan system. The text "TANBUG V3B" should be displayed on the bottom row of the CVB screen. Fine adjustment of CV1 may be necessary to eliminate any flickering of the individual characters.

CVB Screen / Memory Organisation

The CVB display occupies memory space from \$A000 to \$A63F (Hex), 40960 to 42659 (Decimal). The CRT controller (U22) occupies the top 16 locations ie \$A7F0 to A7FF. It only need 2 locations, but these are repeated as they are only partially decoded.

Screen Layout

64 characters

```
A000 A001 A002 .....A03F
A040 A041.....A07F
A080 .....A0BF
-
-
-    25 rows
-
-
A5C0 .....A5FF
A600 .....A63F
```

Microtan Screen Use

If the user has a TV or Monitor connected to the video output of the MT65 CPU board, TANBUG v3B can also be used to drive this. It will be noticed that when using the CVB screen the bottom row of the Microtan screen echoes the output characters. However, typing '<Control> X' will transfer all output to the Microtan screen which will behave exactly as it did before. To return output to the CVB screen, type '<Control> X' again. The cursor is suppressed on the unused screen.

A change of screen can also be made from within a programme by storing a number other than zero in location \$A642 Hex, 42562 Decimal followed by a dummy print statement, eg

Assembly

```
LDA #$1
STA $A642
JSR $FE75
```

BASIC

```
100 POKE42562,1:PRINT
```


Configuration

The 6845 CRT Controller contains a number of registers which may be written to in order to modify the operating conditions. Most of the registers are concerned with the CVB screen format and system timing and should not be modified. A few however may be re-configured by the user.

Register	R10	Cursor Start
	R11	Cursor End
	R14	Cursor High
	R15	Cursor Low

The method of addressing these registers is first to write to the register number to location \$A7FE (decimal 43006), then read or write data from/to location \$A7FF (decimal 43007). This must be done from within a program, not by a TANBUG 'M' command or direct POKE in BASIC from the keyboard.

The CRT Controller uses its own memory address map from \$000 to \$7FF. These correspond to the CVB screen map \$A000 to \$A7FF.

CRT Controller Registers R10/R11

These define the height, position and blink rate of the cursor. The cursor occupies one character width and can consist of any block of scan lines within the 10 line character height. The scan lines are numbered from 0 at the top to 9 at the bottom. R10 contains the line number at the top of the cursor and R11 contains the line number at the bottom.

In addition, Bits 5 and 6 of R10 control the blink rate and display status of the cursor as follows:

Bit 6	Bit 5	Cursor Display
0	0	Non-blinking
0	1	Not displayed
1	0	Fast blink
1	1	Slow blink

On RESET, R10 contains \$69 and R11 contains \$09. This gives a slow blinking cursor occupying only the bottom scan line.

If R10 is modified by the user, the new value will remain set until the user changes it again or a system RESET occurs. However, it is regularly updated by TANBUG v3B which reads location \$A641 and stores the value in R10. Thus the user may modify R10 by writing a new value to location \$A641 (Decimal 42561). A system RESET will restore this to its original value.

CRT Controller Registers R14/R15

These registers contain the high and low byte respectively of the cursor address and are regularly updated by TANBUG v3B. If the user wishes to position the cursor at the top left hand corner of the CVB screen, writing \$00 to both R14 and R15 will do this. Note this will have no effect on the

position of characters output to the screen, which are positioned according to parameters set by TANBUG. Also, the next time a character is output, the cursor will return to its normal position on the screen. If it is desired to output characters at a different position, this may be achieved by setting the screen pointers (\$0A and \$0B in the Zero page, low byte in \$0A and the high byte in \$0B). The current position of the cursor can be read from R14 and R15.

Usage

Each location on the 64 x 25 screen can be a character or a control code. A full list of these codes are provided in Appendix A.

The attributes of the characters are defined by control codes. These control codes are displayed as spaces and control all the characters succeeding it on that row or until another control character is encountered before the end of the row is reached.

The default attributes assumed at the beginning of every row are as follows:

Alphanumeric white, Black background, steady, contiguous graphics, release graphics.

Control codes

Hex	Dec	Control Function
01	1	Alphanumeric RED
02	2	Alphanumeric GREEN
03	3	Alphanumeric YELLOW
04	4	Alphanumeric BLUE
05	5	Alphanumeric MAGENTA
06	6	Alphanumeric CYAN
07	7	Alphanumeric WHITE
08	8	FLASH
09	9	STEADY
11	17	Graphics RED
12	18	Graphics GREEN
13	19	Graphics YELLOW
14	20	Graphics BLUE
15	21	Graphics MAGENTA
16	22	Graphics CYAN
17	23	Graphics WHITE
18	24	CONCEAL DISPLAY
19	25	CONTIGUOUS Graphics
1A	26	SEPARATED Graphics
1C	28	BLACK background
1D	29	NEW background
1E	30	HOLD Graphics
1F	31	RELEASE Graphics

Character Generation

The CVB uses a Teletext Character Generator ROM (SAA5050/55) to provide the characters in any of 6 colours or white. The background can be any of the 6 colours or black or white. In addition, characters can be made to flash or appear in inverse video (ie complimentary colours).-

An 8 bit word defines the character, the lower 7 defining the character or control code with the highest order bit defining normal (0) or inverse video (1).

Graphics

Graphics characters comprise 6 pixels (2 wide, 3 high). The bit allocation is as follows:

Bit Number		BitValue (Decimal)	
0	1	1	2
2	3	4	8
4	6	16	64

Note Bit 5 is always 1 for graphics characters. If Bit 5 is set to 0 for a character in a row succeeding a graphics control code, the capital alphanumeric character will be displayed instead. This allows mixing of graphics characters with upper-case letters without additional codes. To illuminate a group of pixels, add their bit values together, add 32 to the total and poke this value to the relevant screen location.

Graphics characters are either Contiguous or Separated. Contiguous graphics occupy the full character cell space. For Separated graphics, a blank border appears around each pixel in the cell. Graphics characters are normally contiguous unless the control character ‘Separated Graphics’ precedes the graphics characters on a row.

Control characters are displayed as spaces. However if the control code ‘Hold Graphics’ is included earlier in the row, the character that immediately precedes the control character is repeated to fill this control code ‘space’. The ‘Release Graphics’ control code character countermands this.

BASIC usage

When in BASIC, use is made of the CHR\$() function to implant these control characters within a program eg

```
100 PRINT CHR$(1)CHR$(29)CHR$(3)"YELLOW ON A RED BACKGROUND"
```

which will display exactly as it states. Note however the red background continues to the end of the row. The code below demonstrates the use of the flashing and steady codes.

```
200 PRINT CHR$(8)"FLASHING"CHR$(9)"STEADY"
```

Control characters are displayed as spaces and it is not possible therefore to change colour in the middle of a word without leaving a space. It is however acceptable to substitute control characters for spaces as in the following example:

```
300 PRINT CHR$(2)"GREEN"CHR(3)"YELLOW"
```

would display the words GREEN and YELLOW in the appropriate colour with a space between them.

On screen editing in BASIC

The replacement BASIC extension EPROM contains re-written editing routines to allow editing on either the Microtan or CVB screen. Exactly the same protocols have been retained and all the user has to do is follow normal procedures whichever screen is in use. Editing actually takes place on the Microtan screen but the revised routines routinely copy the old screen onto the middle of the CVB screen. Note that BASIC editing corrupts the memory areas of both screens so any data or code stored in these areas should be saved before editing.

Cassette Tape Routines

When undertaking cassette tape dump and fetch routines, both XBUG and BASIC look only on the bottom row of the Microtan Screen for user responses. When using the CVB screen, there are no such restrictions, user responses can be anywhere on the screen.

XBUG Translator and Disassembler

The Translator and Disassembler work in the normal way in the CVB screen except that the ASCII equivalents of the code are not displayed. The flashing cursor is not suppressed and appears under the exclamation mark prompt.

Appendix A

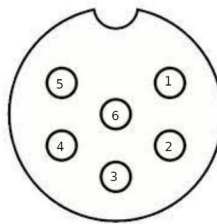
Output connections

J1 – 14 pin DIL socket

pin:	1	Composite SYNC
	2	6 MHz clock
	3	HB (Horizontal Blanking)
	7	GROUND
	8	BLUE Video
	9	GREEN Video
	10	RED Video
	14	2.5v

J2 – 6 pin DIN Socket

Pin:	1	2.5v
	2	Composite SYNC
	3	BLUE video
	4	GREEN video
	5	RED video
	6	GROUND



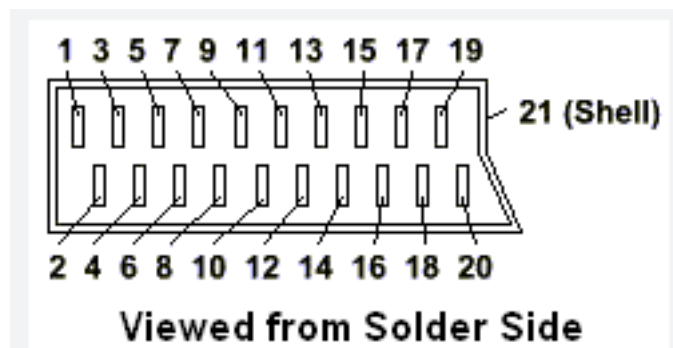
J3 – RCA Phono Socket

Pin:	Inner	Monochrome composite video & sync
	Outer	Ground

Power up the system and press RESET. The message TANBUG V3B will appear at the bottom of the CVB screen followed by the Microtan prompt and a flashing underline cursor. If a monitor is plugged into the MT65 CPU board, the message TANBUG V3B will be momentarily displayed on the Microtan screen and immediately replaced with the Microtan prompt.

SCART Plug

Pin:	7	BLUE
	9	GROUND
	11	GREEN
	15	RED
	16	2.5v
	20	CSYNC
	9-16	3K3 Resistor



Appendix B

SAA5050/55 Character/Control Codes

Bits					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1				
b7	b6	b5	Col	Row	0	1	2	2a	3	3a	4	5	6	6a	7	7a
0	0	0	0	0	<u>NUL</u> *	<u>DLE</u> *			0		@	P	-		p	
0	0	0	1	1	Alpha ⁿ Red	Graphics Red	!		1		A	Q	a		q	
0	0	1	0	2	Alpha ⁿ Green	Graphics Green	"		2		B	R	b		r	
0	0	1	1	3	Alpha ⁿ Yellow	Graphics Yellow	£		3		C	S	c		s	
0	1	0	0	4	Alpha ⁿ Blue	Graphics Blue	\$		4		D	T	d		t	
0	1	0	1	5	Alpha ⁿ Magenta	Graphics Magenta	%		5		E	U	e		u	
0	1	1	0	6	Alpha ⁿ Cyan	Graphics Cyan	&		6		F	V	f		v	
0	1	1	1	7	Alpha ⁿ White	Graphics White	'		7		G	W	g		w	
1	0	0	0	8	Flash	Conceal Display	(8		H	X	h		x	
1	0	0	1	9	** Steady	** Contiguous Graphics)		9		I	Y	i		y	
1	0	1	0	10	** End Box	** Separated Graphics	*		:		J	Z	j		z	
1	0	1	1	11	Start Box	** ESC	+		;		K	←	k		¼	
1	1	0	0	12	** Normal Height	** Black Background	,		<		L	½	l			
1	1	0	1	13	Double Height	New Background	-		=		M	→	m		¾	
1	1	1	0	14	** <u>SO</u>	Hold Graphics	.		>		N	↑	n		÷	
1	1	1	1	15	** <u>S1</u>	** Release Graphics	/		?		O	#	o		□	