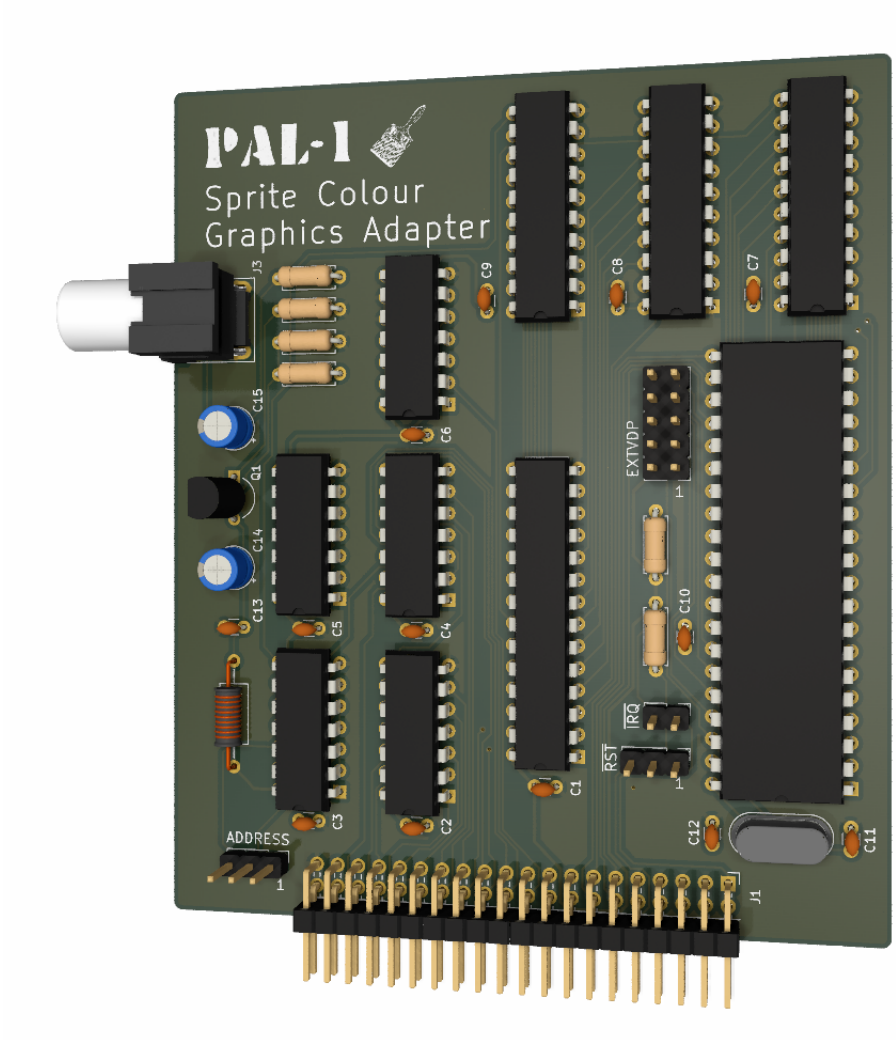


PAL-1 Sprite Colour Graphics Adapter

Plastic Objects Limited

Published by **Plastic Objects Limited**
Beccles, UK

September 2024

Disclaimer

Contents

List of Figures

Introduction

The PAL-1 Sprite Colour Graphics Adapter (SCGA) is an expansion module for the PAL-1 system[1] based on the Texas Instruments TMS9918A Video Display Processor (VDP)[2]. The TMS9918A VDP was used in video systems to provide data display on raster scanned colour television sets or monitors. The IC generates all necessary video, control, and synchronisation signals and also controls the storage and retrieval of display data in its own dedicated screen memory.

The TMS9918A VDP generates a 525-line NTSC encoded colour video signal. While the TMS9928A VDP, which is intended for use with PAL video systems, is functionally compatible with the TMS9918A, it is not pin-for-pin compatible. Thus it is not possible to use the TMS9928A VDP in the PAL-1 Sprite Colour Graphics Adapter. Instead, users who do not have access to an NTSC TV or monitor are recommended to use AV to HDMI or AV to VGA converters as appropriate. Such converters are widely available from the usual e-commerce and auction web sites.

This expansion module was inspired by a Ciarcia's Circuit Cellar project for a 'High-Resolution Sprite-Oriented Color Graphics' expansion card for the Apple II computer, published in BYTE magazine.[3] The design for the PAL-1 SCGA also incorporates Static RAM (SRAM) circuitry based on work by Tom LeMense and Dan Werner.[4,5]

---

[1] Liu Ganning, 'PAL-1 Microcomputer User Manual' (November 2020), ⟨`http://pal.aibs.ws/assets/PAL_en.pdf`⟩ accessed 2024-08-18.

[2] Texas Instruments, 'TMS9918A/TMS9928A/TMS9929A Video Display Processors' (1982).

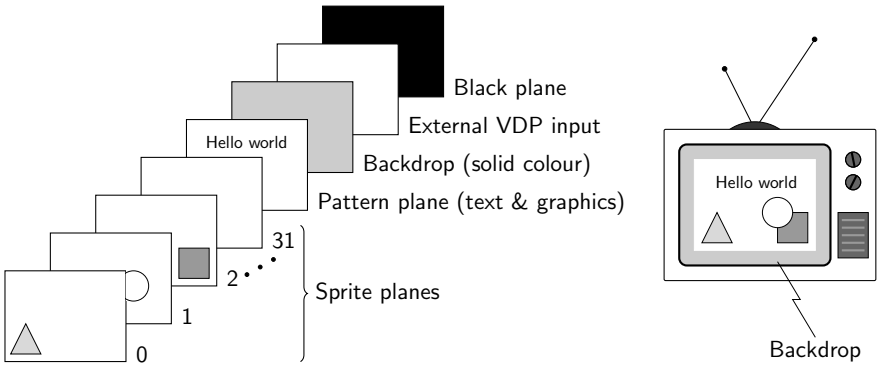
[3] Steve Ciarcia, 'High-Resolution Sprite-Oriented Color Graphics' (August 1982), pp. 57–80.

[4] Tom LeMense, 'SRAM Replacement for TMS99x8 VRAM', Technical report (January 2010), ⟨`https://retrobrewcomputers.org/n8vem-pbwiki-archive/0/35845334/48860720/33053543/SRAM%20Replacement%20for%20TMS99x8%20VRAM.pdf`⟩ accessed 2024-08-18.

[5] Dan Werner, 'Video Display Interface Board', ⟨`https://www.retrobrewcomputers.org/doku.php?id=boards:ecb:scg:start`⟩ accessed 2024-08-18.

Overview of the TMS9918A VDP

The TMS9918A VDP contains all necessary circuitry to generate a video display with up to 16 colours and a maximum resolution of 256 x 192 pixels. Additionally, the VDP supports the display of up to 32 sprites. The image displayed on the screen can best be envisioned as a number of display planes sandwiched together, where each plane has a different priority. For an entity on a specific plane to show through, all planes with higher priorities must be transparent at that point (see Figure 1).



(a) Individual display planes.

(b) Image shown on screen.

Figure 1: TMS9918A VDP overlapping display planes.

In descending priority order, the display planes are made up of 32 *Sprite Planes*, each of which may contain a single sprite. Below these sprite planes is the *Pattern Plane*, which is used for the display of textual and graphics images. Below this is the *Backdrop Plane*, which provides a single colour border around all other planes and sets the default colour for the active display area. Finally, with the lowest priority is the *External Video Plane*, which displays a video image from a compatible external source (see also appendix A).

The VDP supports four video colour display modes that determine the appearance of the Pattern Plane. In both graphics modes, *Graphics 1* and *Graphics 2*, the pattern plane is split into groups of 8x8 pixels, referred to as pattern positions. Graphics and images using multiple colours may be created by defining unique patterns for each pattern position. In *Text* mode, the Pattern Plane is split into groups of 6x8 pixels, resulting in 40x24 text positions on the screen. Sprites are not displayed in *Text* mode and only two colours are allowed. Finally, in *Multicolor* mode, the screen is split into a grid of 64x48 positions, each of which is 4x4 pixels. In this mode, each position may have a unique colour.

The *Sprite Planes* are used for the display of up to 32 sprites in the *Multicolor* and *Graphics* modes. Sprites are automatically transparent in

Text mode and thus cannot be used in this mode. Each sprite covers a 8x8, 16x16 or 32x32 pixel area and may be positioned anywhere on its plane. Any part of the plane not covered by the sprite is transparent, as may be all or part of a sprite.

The VDP registers define the base addresses for several sub-blocks within Video RAM (VRAM). The sub-blocks form *tables* which are used to define sprites and produce the desired text or images on the screen. Before use, the contents of these tables must be initialised by applications running on the PAL-1. Displaying text, drawing images and moving sprites across the screen can be achieved by choosing the relevant register parameters and changing the pointers.

1. *Getting Started*

Board assembly

Before assembling your PAL-1 Sprite Colour Graphics Adapter board check the package contents against the Bill of Materials on page 15, and contact your distributor as soon as possible if any items are missing.

No specialist tools are required for assembling the board, though care must be taken when handling ESD sensitive components, especially the ICs. Before inserting the ICs into their sockets, check the board for dry joints and solder bridges. Also be sure to pay special attention to the orientation of the ICs, ensuring pin 1 of each IC is correctly aligned. Refer to the board layout on page 13 for the orientation of ICs, diodes and electrolytic capacitors.

Installation and configuration

Because of its width, the PAL-1 Sprite Colour Graphics Adapter cannot be installed in all slots of the PAL-1 Motherboard.[1] Rather, it must be installed in any of slots 4, 5, or 6. Alternatively, it may be directly connected to the PAL-1 using a 40-pin IDC cable.

To prevent damage to your system, *always* power off your PAL-1 before installing or removing the PAL-1 Sprite Colour Graphics Adapter board. Ensure that the pins are correctly aligned when inserting the board into the PAL-1 motherboard or when directly connecting it to the PAL-1 expansion port using an IDC cable.

Data and register addresses

The TMS9918A VDP has an 8-bit data bus, which is memory mapped on the PAL-1 system at addresses determined by the position of the ADDRESS jumper, as shown in Table 1.1. Up to two PAL-1 Sprite Colour Graphics

[1] Liu Ganning, 'PAL-1 MOTHERBOARD' (2021), ⟨`http://pal.aibs.ws/assets/Motherboard_manual.pdf`⟩ accessed 2024-08-18.

Pins	VDP data	VDP register
1 – 2	$16EC	$16ED
2 – 3	$16EE	$16EF

Table 1.1: ADDRESS jumper settings.

Adapters can be installed in one system by assigning each adapter unique data and register addresses.

Option jumpers

The PAL-1 Sprite Colour Graphics Adapter has two option jumpers to configure its behaviour:

$\overline{\text{RST}}$ **1 – 2** Disables TMS9918A $\overline{\text{RESET}}$/SYNC using pull-up resistor $R1$.

2 – 3 Connects the TMS9918A $\overline{\text{RESET}}$/SYNC line with the $\overline{\text{RESET}}$ line of the PAL-1 MOS6502 CPU, ensuring it is used as a reset signal on the VDP.

No jumper This jumper *must* be removed when using the EXTVDP external video pin header.

$\overline{\text{INT}}$ Enable CPU interrupts. If enabled, interrupts are raised each time a frame is completed, while the beam is tracing up to the upper left corner of the screen. This occurs every $1/60^{\text{th}}$ of a second (approximately).

2. *Sample Code*

Programming the PAL-1 Sprite Colour Graphics Adapter simply requires the ability to write to and read from the memory mapped data and register addresses. Many programming languages provide statements to achieve this, including the `POKE` and `PEEK` statements in BASIC, and `C!` and `C@` in Forth.

Note that the examples included below use the default SCGA data and register addresses (\$16EC, 5868_{10} and \$16ED, 5869_{10}). Further examples may be found on Github at `https://github.com/dimitrit/pal1vdp`.

BASIC

TODO!

6502 assembly language

The assembly language example is taken from the '*High-Resolution Sprite-Oriented Color Graphics*' article in the August 1982 issue of BYTE magazine.[1] The code has been modified where required to allow it to run on the PAL-1.

The program starts by initialising the eight VDP registers and clearing the VRAM. In this example the 9918A is set to the following operating specifications: Graphics II mode, external video input disabled, and 16 x 16 pixel sprites, with selectable magnification to twice their normal size (32 x 32 pixels) under keyboard control. When the program starts, four different sprites are displayed, as shown in Figure 2.1. The display can be changed as follows. When you press the GO key, the sprites' position coordinates are incremented and the sprites move. Pressing the 0 key and then a hexadecimal digit 1 through F will set one of the fifteen background colours or transparency. Pressing the AD or DA keys will vary the size of the sprites between 16 x 16 and 32 x 32 pixels.

```
;
;*******************************************************
;
```

[1] Ciarcia, pp. 57–80.

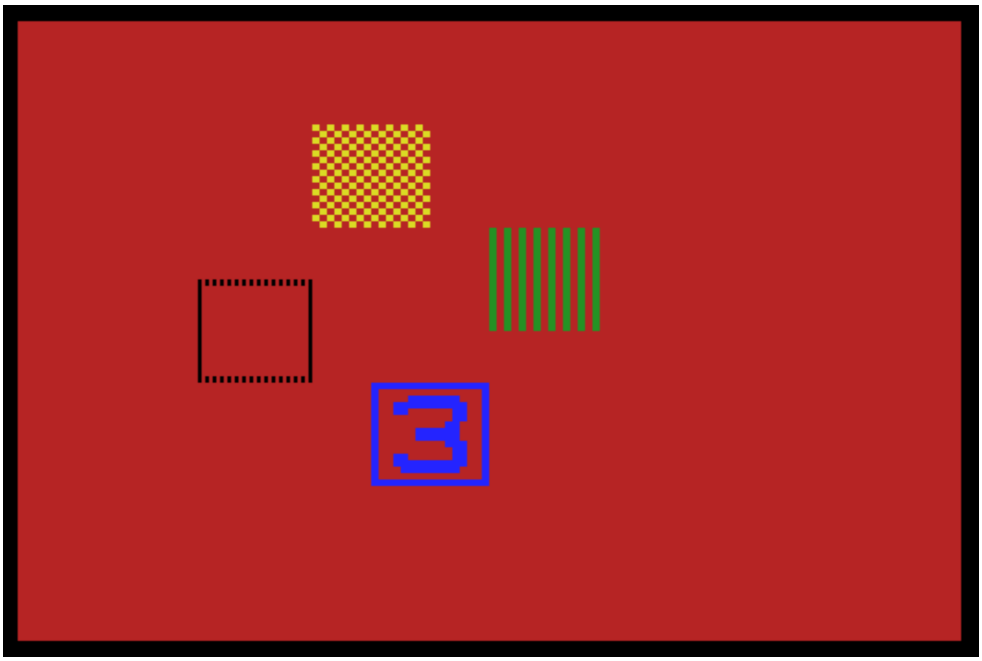


Figure 2.1: 6502 Assembly language *Video Demo* showing four sprites.

```
;                  *** VIDEO DEMO ***
;
;

KBD      = $C000                  ;APPLE KEYBOARD DATA
KSTRB    = $C010                  ;KEYBOARD DATA CLEAR
VREG     = $C081                  ;VDP REGISTER
VDATA    = $C080                  ;VDP RAM
;
         *= $200                  ;PROGRAM STARTING ADDRESS
;
;**************** INITIALIZE VDG *************************
;
         LDY #$87                 ;REGISTER SELECT
         LDX #$07                 ;INITIALIZE COUNTER
INIT1    LDA ITAB,X               ;LOAD INIT TABLE
         JSR SREG                 ;WRITE TO VDP
         DEY                      ;DECREMENT REGISTER
         DEX                      ;DECREMENT COUNTER
         BNE INIT1                ;DONE?
;
;************** CLEAR ALL MEMORY *************************
;
         LDY #$40                 ;BYTE2 ADDRESS SET UP
         LDA #$00                 ;BYTE1 ADDRESS SET UP
         JSR SREG                 ;WRITE TO VDP
         LDX #$C0                 ;COUNTER HIGH BYTE
```

```
NEXF    LDY #$00                ;COUNTER LOW BYTE
FILL    STA VDATA               ;WRITE TO VDP RAM
        INY                     ;INCREMENT LOW COUNTER
        BNE FILL                ;LOW COUNTER FULL?
        INX                     ;INCREMENT HIGH COUNTER
        BNE NEXF                ;HIGH COUNTER FULL?
;
;******** LOAD SPRITE ATTRIBUTES ************************
;
LOOP    LDY #$47                ;BYTE2 AT 0700 HEX
        LDA #$00                ;BYTE1 ADDRESS SET UP
        JSR SREG                ;WRITE TO VDP
        LDX #$00                ;INITIALIZE COUNTER
NEXA    LDA ATAB,X              ;LOAD ATTRIBUTE
        STA VDATA               ;STORE TO VDP RAM
        INX                     ;INCREMENT COUNTER
        TXA
        CMP #$10                ;TEST COUNTER
        BNE NEXA                ;DONE?
;
;********** LOAD SPRITE PATTERNS ************************
;
        LDY #$40                ;BYTE2 AT 0000 HEX
        LDA #$00                ;BYTE1 ADDRESS SET UP
        JSR SREG                ;WRITE TO VDP
        LDX #$00                ;INITIALIZE COUNTER
NEXS    LDA PTAB,X              ;LOAD PATTERN BYTE
        STA VDATA               ;STORE TO VDP RAM
        INX                     ;INCREMENT COUNTER
        TXA
        CMP #$80                ;TEST COUNTER
        BNE NEXS                ;DONE?
;
;************* CHANGE BACKGROUND ************************
;
CBACK   LDA KBD                 ;TEST FOR
        CMP #$CF                ;"O" KEY INPUT
        BNE CSIZE               ;TO SET BACKGROUND COLOR
        JSR LOADN               ;READ KEYBOARD
        LDY #$87                ;BYTE1 REGISTER 7
        JSR SREG                ;STORE TO VDP
;
;****************** CHANGE SIZE ************************
;
CSIZE   LDA KBD                 ;TEST FOR LEFT ARROW
        CMP #$88                ;MAGNIFICATION X 1
        BNE ONE
        LDA ITAB+1              ;LOAD REGISTER 1
        AND #$FE                ;MASK 0 ON LSB
        LDY #$81                ;BYTE1 REGISTER 1
        JSR SREG                ;STORE TO VDP
ONE     CMP #$95                ;TEST FOR RIGHT ARROW
        BNE MOVE                ;MAGNIFICATION X 2
        LDA ITAB+1              ;LOAD REGISTER 1
```

```
        ORA #$01                ;MASK 1 ON LSB
        LDY #$81                ;BYTE1 REGISTER 1
        JSR SREG                ;STORE TO VDP
;
;****************** MOVE SPRITES *************************
;
MOVE    LDA KBD                 ;MOVE?
        CMP #$CD                ;TEST FOR "M" KEY
        BNE JUMP
        INC ATAB                ;SPRITE0 UP
        DEC ATAB+1              ;SPRITE0 LEFT
        INC ATAB+4              ;SPRITE1 UP
        INC ATAB+5              ;SPRITE1 RIGHT
        DEC ATAB+8              ;SPRITE2 DOWN
        DEC ATAB+9              ;SPRITE2 LEFT
        DEC ATAB+$C             ;SPRITE3 DOWN
        INC ATAB+$D             ;SPRITE3 RIGHT
JUMP    BIT KSTRB               ;CLEAR KEYBOARD
        JMP LOOP                ;JUMP TO START
;
;
;********* STORE VIDEO REGISTERS ************************
;
SREG    STA VREG                ;STORE BYTE1
        STY VREG                ;STORE BYTE2
        RTS                     ;RETURN
;
;************* LOAD KEYBOARD INPUT ***********************
;
LOADN   BIT KSTRB               ;CLEAR KEYBOARD
WAIT    BIT KBD                 ;TEST KEYBOARD
        BPL WAIT                ;IS KEY PRESSED ?
        LDA KBD
        AND #$F0                ;TEST IF NUMERICAL INPUT
        CMP #$C0
        BEQ LETER
        LDA KBD
        AND #$0F                ;MASK OFF HIGH NIBBLE
        RTS                     ;RETURN
LETER   LDA KBD
        CLC
        ADC #$09                ;CONVERT INPUT TO HEX VALUE
        AND #$0F                ;MASK OFF HIGH NIBBLE
        RTS                     ;RETURN
;
;********************** TABLES ***
;
ITAB    .BYT $02,$C2,$01,$80    ;INITIALIZE TABLE
        .BYT $01,$0E,$00,$01
;
ATAB    .BYT $40,$60,$00,$03    ;SPRITE 0 ATTRIBUTE
        .BYT $60,$60,$04,$07    ;SPRITE 1 ATTRIBUTE
        .BYT $40,$80,$08,$0B    ;SPRITE 2 ATTRIBUTE
        .BYT $60,$80,$0C,$0F    ;SPRITE 3 ATTRIBUTE
```

```
;
PTAB    .WORD $FF80,$8080,$8080,$8080 ;SPRITE 0 PATTERN
        .WORD $8080,$8080,$8080,$80FF ;16 X 16 PIXELS
        .WORD $FF01,$0101,$0101,$0101 ;32 BYTES / SPRITE
        .WORD $0101,$0101,$0101,$01FF
;
        .WORD $FF80,$879F,$9880,$8083 ;SPRITE 1 PATTERN
        .WORD $8380,$8098,$9F8F,$80FF
        .WORD $FF01,$F1F9,$1919,$31F1
        .WORD $F139,$1919,$F9F1,$01FF
;
        .WORD $AA55,$AA55,$AA55,$AA55 ;SPRITE 2 PATTERN
        .WORD $AA55,$AA55,$AA55,$AA55
        .WORD $AA55,$AA55,$AA55,$AA55
        .WORD $AA55,$AA55,$AA55,$AA55
;
        .WORD $AAAA,$AAAA,$AAAA,$AAAA ;SPRITE 3 PATTERN
        .WORD $AAAA,$AAAA,$AAAA,$AAAA
        .WORD $AAAA,$AAAA,$AAAA,$AAAA
        .WORD $AAAA,$AAAA,$AAAA,$AAAA
;
        .END
```

Listing 2.1: Ciarcia's Circuit Cellar Video Demo

A. *Connector Pin Outs*

External Video Pin Header (EXTVDP)

The PAL-1 Sprite Colour Graphics Adapter (SCGA) external video pin header (EXTVDP) can be used to connect a compatible external video source. Alternatively, it allows the SCGA to be used as the source for other systems.[1]

Note that the pins of this header are numbered 'odd-even' with pin 1 clearly marked on the PCB, as shown in figures A.1 and C.1. **Be sure to check the TTL voltage levels and signals before connecting any external systems as incorrect voltages and/or incompatible signals may result in permanent damage to your PAL-1 Sprite Colour Graphics Adapter.**

[1] Texas Instruments, pp. 3.7–3.8.

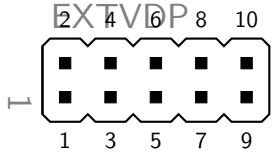


Figure A.1: EXTVDP external video pin header.

Pin	Signal	Description
1	GND	Ground
2	VCC	+5V Supply Voltage
3	GND	Ground
4	COMVID	Composite Video Output
5	GROMCLK	VDP Output Clock = $XTAL/24$
6	COMVID	Composite Video Output
7	EXTVDP	External Video (VDP) Input
8	CPUCLK	VDP Colour Burst Frequency Clock
9	$\overline{\text{RESET}}$/SYNC	This is a tri-level input pin. When it is below 0.8 volts, $\overline{\text{RESET}}$/SYNC initialises the VDP. When it is above 9 volts $\overline{\text{RESET}}$/SYNC is the synchronising input for external video.
10	COMVID_r	Unconditioned Composite Video Output of the TMS9918A VDP.

Table A.1: EXTVDP external video pin header.

B. *Schematic*

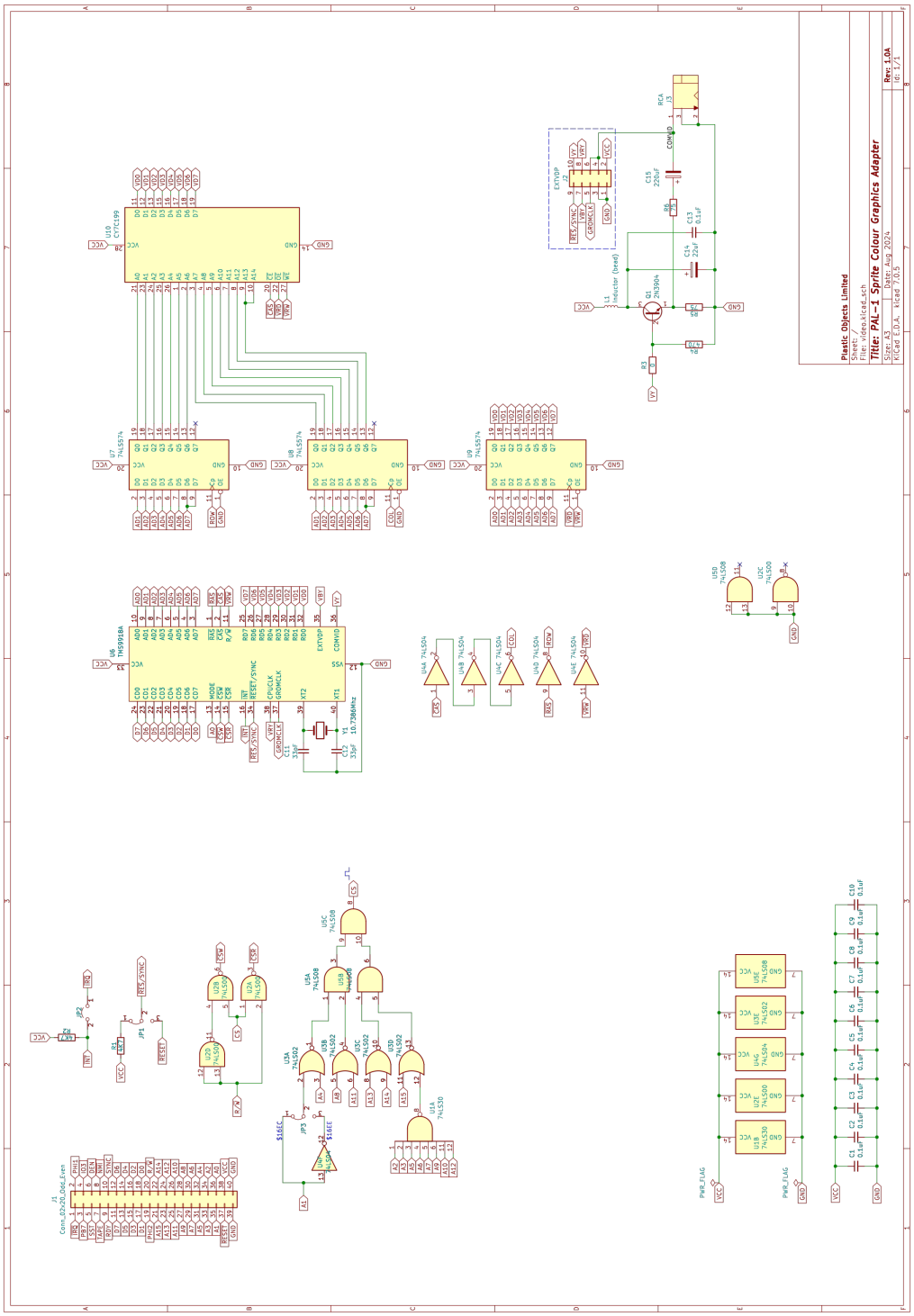


Figure B.1: PAL-1 Sprite Colour Graphics Adapter schematic.

C. *Board layout*

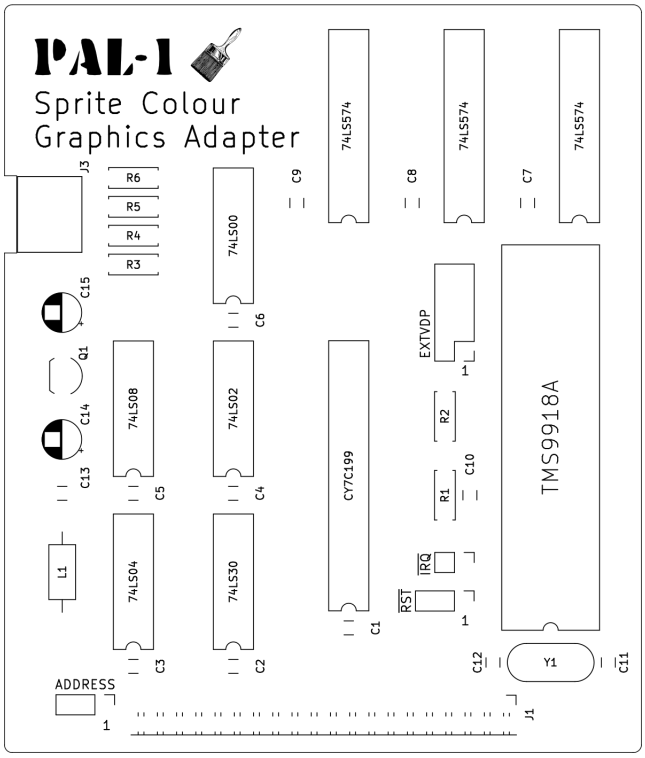


Figure C.1: PAL-1 Sprite Colour Graphics Adapter component locations.

D. *Bill of Materials*

Part	Qty	Value	Description
C1 – C10, C13	11	0.1uF	Unpolarized capacitor
C11, C12	2	33pF	Unpolarized capacitor
C14	1	22uF	Polarized capacitor
C15	1	220uF	Polarized capacitor
J1	1	Connector	Right angle connector, double row, 2x20
J2	1	EXTVDP	Connetor, double row, 2x5
J3	1	RCA	RCA jack
JP1	1	/RST	Pin header, 1x3
JP2	1	/IRQ	Pin header, 1x2
JP3	1	ADDRESS	Pin header, 1x3
L1	1	Inductor	Inductor, small bead
Q1	1	2N3904	Small Signal NPN Transistor, TO-92
R1, R2	2	4K7	Resistor, 250mA
R3	1	0	Resistor, 250mA
R4	1	470	Resistor, 250mA
R5, R6	2	75	Resistor, 250mA
U1	1	74LS30	8-input NAND
U2	1	74LS00	Quad 2-input NAND
U3	1	74LS02	Quad 2-input NOR
U4	1	74LS04	Hex Inverter
U5	1	74LS08	Quad 2-input AND
U6	1	TMS9918A	Video Display Processor
U7, U8, U9	3	74LS574	8-bit Register
U10	1	CY7C199	Static RAM, 32K x 8
Y1	1	10.7386Mhz	Two pin crystal

Table D.1: PAL-1 Sprite Colour Graphics Adapter v1.0A bill of materials.

References

Ciarcia, Steve, 'High-Resolution Sprite-Oriented Color Graphics' (August 1982).

Ganning, Liu, 'PAL-1 Microcomputer User Manual' (November 2020), ⟨http://pal.aibs.ws/assets/PAL_en.pdf⟩ accessed 2024-08-18.

—— 'PAL-1 MOTHERBOARD' (2021), ⟨http://pal.aibs.ws/assets/Motherboard_manual.pdf⟩ accessed 2024-08-18.

LeMense, Tom, 'SRAM Replacement for TMS99x8 VRAM', Technical report (January 2010), ⟨https://retrobrewcomputers.org/n8vem-pbwiki-archive/0/35845334/48860720/33053543/SRAM%20Replacement%20for%20TMS99x8%20VRAM.pdf⟩ accessed 2024-08-18.

Texas Instruments, 'TMS9918A/TMS9928A/TMS9929A Video Display Processors' (1982).

Werner, Dan, 'Video Display Interface Board', ⟨https://www.retrobrewcomputers.org/doku.php?id=boards:ecb:scg:start⟩ accessed 2024-08-18.