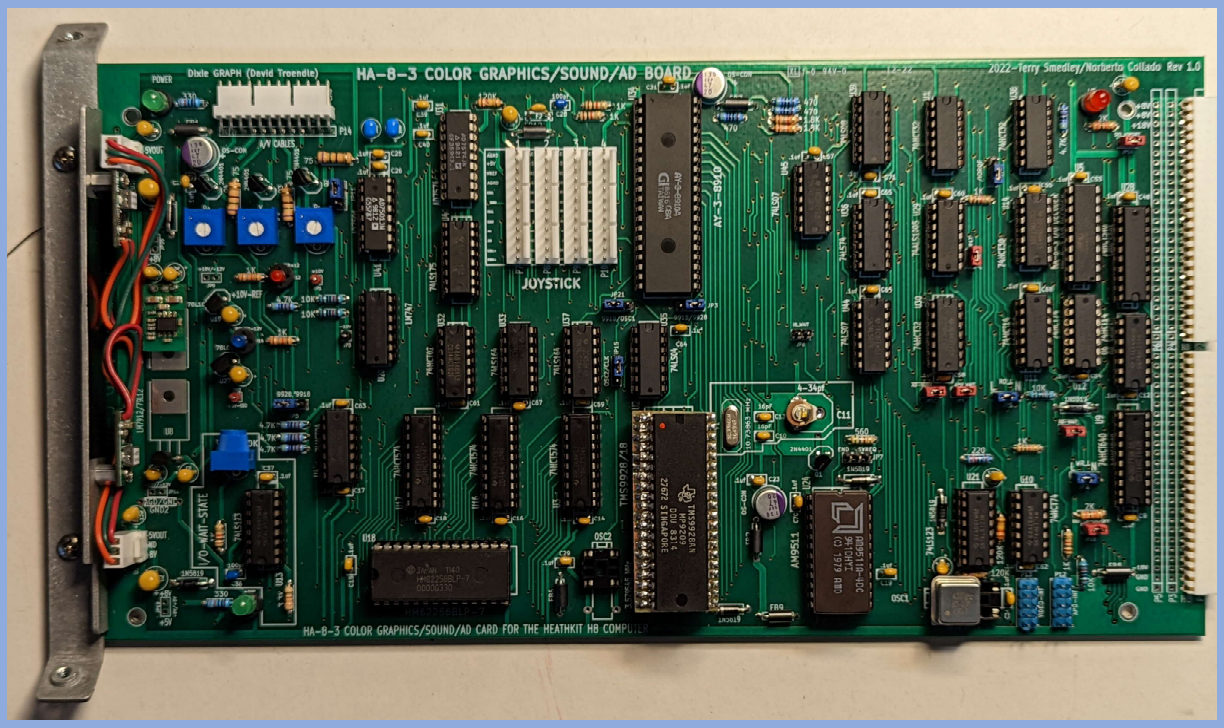
HA-8-3-1 Color Graphics Card for the Heathkit H8 Computer



HA-8-3 Color Graphics Card
Construction, Testing, and Setup Guide

Table of Contents

Rev 1.0
4/25/2022

HA-8-3-1 Color Graphics Board for the Heathkit H8 Computer

Compatible with and based on the original HA-8-3 design by New Orleans General Data Systems

Video Display Processor

TI TMS9928, or optionally TMS9918
YPbPr component video (TMS9928) or composite video (TMS9918)
16K static VRAM
256x192 video resolution

Programmable Sound Generator

AY-3-8910 PSG
3 noise channels + 3 tone channels
2 x 8-bit digital input/output ports
Stereo output: Ch0 - A + 50%C, Ch1 - B + 50% C

Analog Inputs

8-bit ADC (AD7574)
8 channel multiplexer (AD7501)
10v reference voltage

Arithmetic Processing Unit

AMD AM9511 or Intel C8231A
2, 3 and 4MHz APU supported

Joystick/Controller Inputs

Four 12-pin headers, each with
GND, +5v, and +10v Vref
2 digital inputs (0-5v)
2 digital outputs (0-5v direct, 1 TTL load only)
2 buffered digital outputs (74LS07, 40ma open collector, 30v maximum)
2 analog inputs (0-10v)

General I/O Wait State Generator

Adjustable ~700ns wait state on IORQ
Most legacy H8 boards will run at 16MHz with I/O wait states

Soft Reset Capability

Write to port 066Q forces reset of TMS99x8 without a hardware system reset

HA-8-3 Board Resources

Memory

The HA-8-3 does not use any part of the H8 memory space

I/O Ports

All I/O ports are assigned using GAL U5

066Q	Soft Reset (any write to this port resets the VDP)
270-271Q	Video Display Processor (VDP)
272-273Q	Programmable Sound Generator (PSG)

Default Assignments, Jumper JP14 Removed

364-365Q	Analog to Digital Converter (ADC)
366-367Q	Arithmetic Processing Unit (APU)

Alternate Assignments, Jumper JP14 Installed

274-275Q	Analog to Digital Converter (ADC)
276-277Q	Arithmetic Processing Unit (APU)

Interrupts

By default, interrupts are not used on the HA-8-3
The VDP and APU are capable of generating interrupts
Jumper selected for INT 3,4,5 or 6 using P12 and P13

	Value	Qty	Description	Specifications	Part Number	Stock No.
Capacitors	0.1	36	0.1uf ceramic	0.1"		
	1	2	1uf tantalum	0.1" 25v	TAP105K025SCS	Mouser 581-TAP105K025SCS
	2.2	7	2.2uf tantalum	0.1" 35v	TAP225K035SCS	Mouser 581-TAP225K035SCS
	10	3	10uf tantalum	0.1" 25v	TAP106K025SCS	Mouser 581-TAP106K025SCS
	33	2	33uf tantalum	0.1" 25v	TAP336K025SCS	Mouser 581-TAP336K025SCS
	47	3	47uf OS-CON	SEP series 20v	20SEP47M	Digikey P16310-ND
	10 cer	2	10uf ceramic	0.1" 25v	RCER71E106K3A2H03B	Digikey 490-12869-ND
	100pf	2	100pf ceramic	0.1" C0G	SR151A101JAATR1	Mouser 581-SR151A101JAATR1
	16pf	2	16pf ceramic	0.1" C0G	C315C150J1G5TA (15pf OK)	Digikey 399-9717-ND
	4-34pf	1	4-34pf trimmer		*Do not install*	Jameco 32839
Resistors	220	1	220 1/4w 5%			
	330	2	330 1/4w 5%			
	470	3	470 1/4w 5%			
	560	1	560 1/4w 5%			
	1.8K	2	1.8K 1/4w 5%			
	10K	4	10K 1/4w 5%			
	120K	3	120K 1/4w 5%			
	1K	8	1K 1/4w 5%			
	2K	2	2K 1/4w 5%			
	3.3K	1	3.3K 1/4w 5%			
	4.7K	5	4.7K 1/4w 5%			
	75 (1/2w)	3	75 1/2w 5%	1/2 watt		
Integrated Circuits	74AHCT123	1	DIP-14	do not substitute		
	74ALS1005	1	DIP-14	do not substitute		
	74HCT02	1	DIP-14			
	74HCT04	1	DIP-14			
	74HCT14	2	DIP-14			
	74HCT30	1	DIP-14			
	74HCT32	3	DIP-14			
	74HCT540	1	DIP-20			
	74HCT574	3	DIP-20			
	74HCT640	1	DIP-20			
	74HCT74A	2	DIP-14			
	74LS04	1	DIP-14			
	74LS07	2	DIP-14			
	74LS08	1	DIP-14			
	74LS123	1	DIP-16	do not substitute		
	74LS164	2	DIP-14			
	74LS175	1	DIP-16			
	74LS74A	1	DIP-14			
	78L05	1				
	78L10	1			UA78L10ACLP	
	79L12	1		Can sub 7912 TO220		
	AD7501	1	DIP-16	JN,KN are OK		
	AD7574	1	DIP-18	JN,KN are OK		
	AM9511	1	DIP-24	Optional		
	AY-3-8910	1	DIP-40			
	GAL	1	DIP-20	F16V8BQL-15PU	F16V8B-15PU, GAL16V8D-15LP	
	HM62256BLP-7	1	DIP-28	Can sub AS6C62256-55PCN		Digikey 1450-1033-ND
	LM747	1	DIP-14	Can sub UA747		
	LM78012	1		Can sub switching		
	LM7805	2		Can sub switching		
	TMS9928		DIP-40			ArcadeShop.com TMS9928

	Value	Qty	Description	Specifications	Part Number	Stock No.
IC Sockets	DIP-14	16	IC socket			
	DIP-16	4	IC socket			
	DIP-18	1	IC socket			
	DIP-20	8	IC socket			
	DIP-24	1	IC socket			
	DIP-28	1	IC socket			
	DIP-40	3	IC socket			
Semiconductors	2N4401	4				
	LED	5	Color and size optional, adjust Rdrop if necessary (see "LEDs" tab)			
	1N5819	6				
	1K Trimpot	3	top adjust	Bourns	3386F-1-102LF	Mouser 652-3386F-1-102LF
	10K Trimpot	1	side adjust	Bourns	3386H-1-103LF	Mouser 652-3386H-1-103LF
	Ferrite Bead	7		Beads-on-Leads	2743005111	Mouser 623-2743005111
	1x12 header	4	Molex KK	Polarized w/ramp	22-23-2121	
	1x12 header	1	Molex KK rt ang	Polarized w/ramp	22-05-3121	
	2x5 header	2				
	25-pin bus conn	2	Molex	MX 22-16-2251	Samtec BCS-125-L-S-HE	Digikey SAM1009-25-ND
	Crystal	1	10.73863MHz			Mouser 449-LFXTAL029962REEL
	Oscillator (OSC1)	1	2/3/4MHz	See "Oscillators"	ECS-2100A-040	Digikey X202-ND
	Oscillator (OSC2)	1	3.579545MHz	See "Oscillators"	ECS-2100A-035	Digikey XC235-ND
	PTC fuse	1	200ma	Littelfuse	60R020XU	Mouser 576-60R020XU
	Heat sink	1	for DIP-24		Optional, with AM9511 only	Digikey AE10816-ND
	Adhesive	1	(see instructions)		Optional, with AM9511 only	Digikey 1944-1369-ND
	Jumper block	22				
	1x2 header	11	(see instructions)			
	1x3 header	9	(see instructions)			

HA-8-3 Jumper Settings

			LEFT/BOTTOM/ON	RIGHT/TOP/OFF	Default	
1	1X3	RD_L	Latched	Normal	LEFT	
2	1X3	WR_L	Latched	Normal	LEFT	
3	1X3	9918/9928 Pin 38	Pin 38=VDPCLK (9918)	Pin 38=Pr (9928)	RIGHT	LEFT for 9918, RIGHT for 9928
4	1X3	/IORQF	Z80 v4 IORQ	"Fake" IORQ=IORD+IOWR	BOTTOM	BOTTOM for Z80v4, TOP for others
5	1X3	9928/9918 Pin 35	Pin 35=Pb (9928)	Pin 35=GND (9918)	LEFT	LEFT for 9928; RIGHT if no ext video in for 9918; not installed if using 9918 external video
6	1X2	/APU_WAIT	APU Waits On	No APU Waits	ON	
7	1X3	END/SVREQ	APU Int on END	APU Int on SVREQ	1-2	See P12 for APU interrupt jumper
8	1X2	/IO_WAIT	IO Waits On	No I/O Waits	OFF	If using I/O Wait States, adjust RV3 for ~700ns pulse width at JP8
9	1X2	+18v/+12v	Bypass board +12v	Use on-board regulator	OFF	CAUTION! Do not install unless the H8 bus has been modified for PC-style power supply
10	1X2	-18v/-12v	Bypass board -12v	Use on-board regulator	OFF	CAUTION! Do not install unless the H8 bus has been modified for PC-style power supply
11	1X3	MUX Enable	AD7503 - Enable=GRD	AD7501 - Enable=+5v	TOP	Set according to MUX IC being used
12	1X2	/AD_WAIT	ADC Waits On	No ADC Waits	ON	
13	1X2	/AD_BUSY	ADC Busy Waits On	No ADC Busy Waits	ON	
14	1X2	SEL_IOPORT	Use Original HA-8-3 Ports	Use triple-disk ports	OFF	OFF: uses triple disk compatible ports, ON: uses original HA-8-3 addresses
15	1X3	OSC2/Clk	Use OSC2 for PSG clock	Use JP21 for PSG clock	(*)	See PSG clock selection matrix
16						
17	1X2	/PSGWAIT	PSG Waits On	No PSG Waits	ON	
18						
19	1X2	+8v/+5v	Bypass Primary +5v	Use on-board regulator	OFF	CAUTION! Do not install unless the H8 bus has been modified for PC-style power supply
20	1X2	+8v/+5v	Bypass Secondary +5v	Use on-board regulator	OFF	CAUTION! Do not install unless the H8 bus has been modified for PC-style power supply
21	1X3	OSC1/9918	Use OSC1 (APU) for PSG clock	Use 9918 for PSG clock	(*)	See PSG clock selection matrix
22						
23	1X2	BUFFER-ON/OFF	Enable bus buffers	Buffers OFF for DEBUG	ON	Off only for debug testing

Default Configuration:

TMS9928 VDP
No AM9511 APU installed
AD7501 MUX IC
Z80 v4 CPU board
No general I/O wait states, or wait states produced by other board

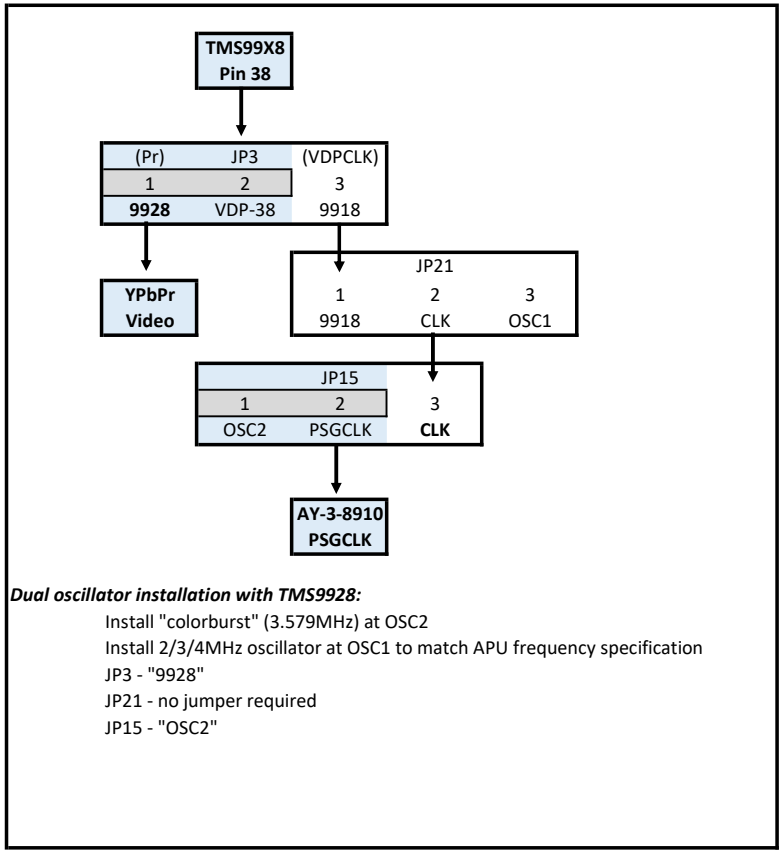
HA-8-3 CLOCK OSCILLATOR INSTALLATION

On the HA-8-3, can oscillators are used to clock the AY-3-8910 sound chip and the AM9511 APU
The original HA-8-3 board used a TMS9918 video chip, which outputs a 3.57954MHz "colorburst" frequency that is used to clock the AY-3-8910 sound chip
The TMS9928 YPbPr upgrade chip does not output the colorburst clock so an alternative clock for the AY-3-8910 must be used
The original HA-8-3 board used the 2MHz bus clock for the AM9511 APU
4MHz AM9511 chips are readily available, and require a clock source other than the bus clock for full performance

OSC1 Clock for the AM9511 APU, alternate clock for the AY-3-8910
OSC2 Primary clock for the AY-3-8910

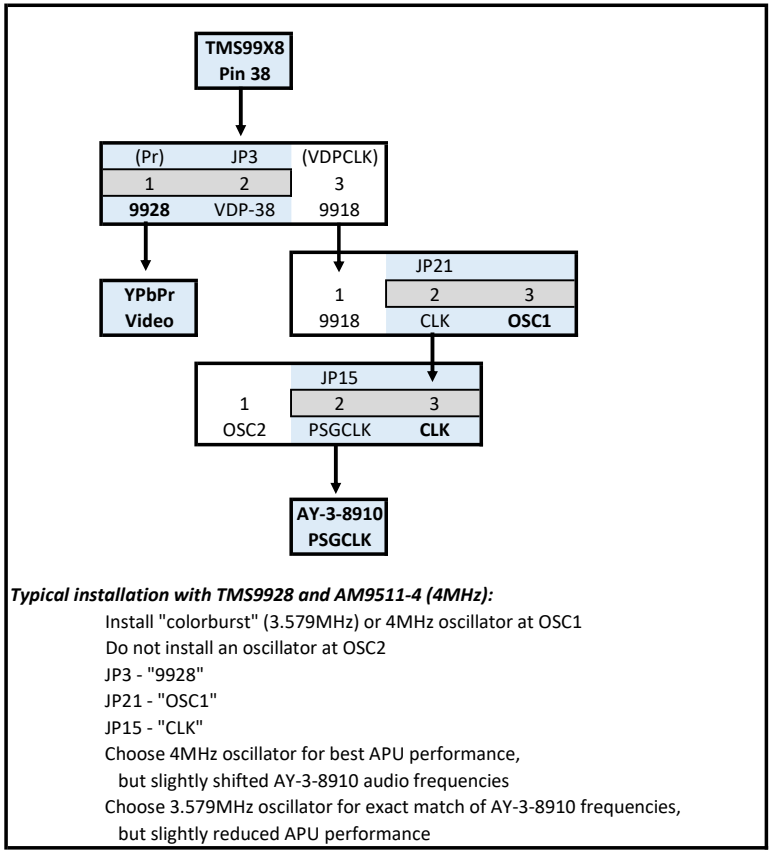
Jumpers JP3, JP15, and JP21 allow flexible selection of the oscillator(s) to be used on the HA-8-3 board

JP3 Selects Pin 38 of the VDP as either a clock signal (TMS9918) or a Pr signal (TMS9928)
JP21 Selects the CLK source - either OSC1, or Pin 38 of the VDP (TMS9918)
JP15 Selects the clock source for the PSG - either OSC2, or the CLK signal selected at JP21



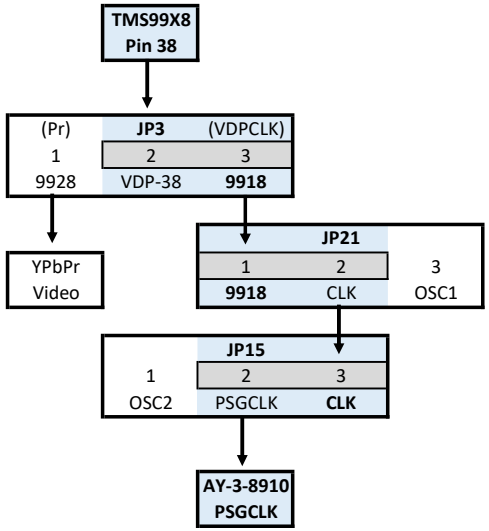
Dual oscillator installation with TMS9928:

Install "colorburst" (3.579MHz) at OSC2
Install 2/3/4MHz oscillator at OSC1 to match APU frequency specification
JP3 - "9928"
JP21 - no jumper required
JP15 - "OSC2"



Typical installation with TMS9928 and AM9511-4 (4MHz):

Install "colorburst" (3.579MHz) or 4MHz oscillator at OSC1
Do not install an oscillator at OSC2
JP3 - "9928"
JP21 - "OSC1"
JP15 - "CLK"
Choose 4MHz oscillator for best APU performance,
but slightly shifted AY-3-8910 audio frequencies
Choose 3.579MHz oscillator for exact match of AY-3-8910 frequencies,
but slightly reduced APU performance



Typical installation with TMS9918:

Do not install oscillator at OSC2

Install 2/3/4MHz oscillator at OSC1 to match APU frequency specification

JP3 - "9918"

JP21 - "9918"

JP15 - "CLK"

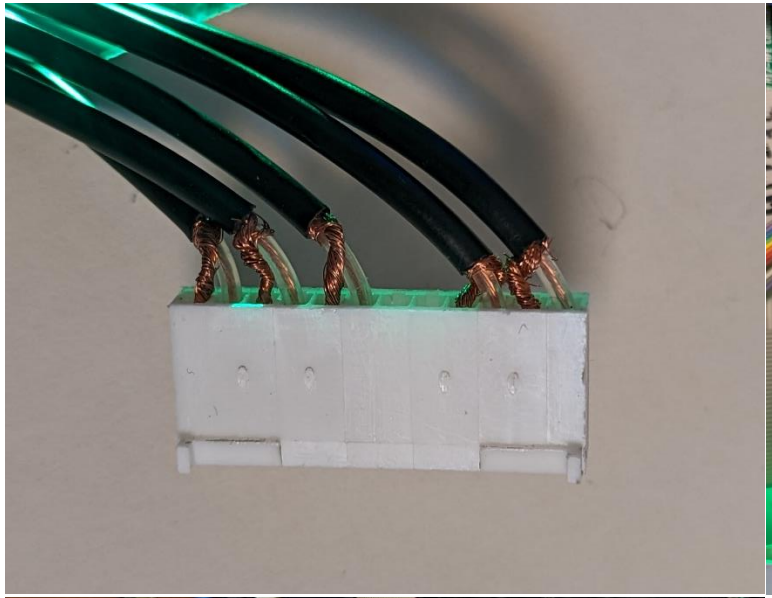
HA-8-3 Connector Pinouts

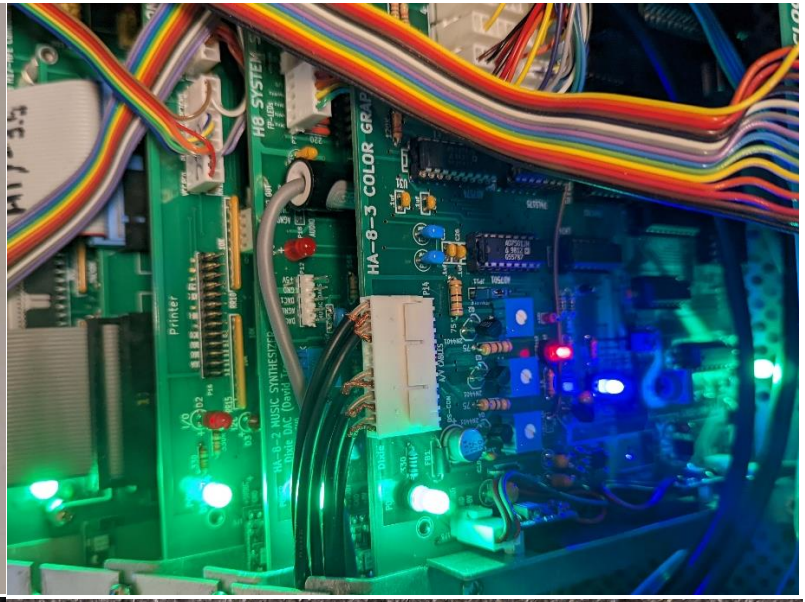
A/V Connector (P14)		
1	GND	Logic ground (shield)
2	Pb-BLUE	Pb component signal
3	GND	Logic ground (shield)
4	Y-GREEN	Y component signal (TMS9928) or composite video (TMS9918)
5	GND	Logic ground (shield)
6	Pr-RED	Pr component signal
7	GND	Logic ground (shield)
8	EXT-TV	External composite video input (TMS9918 only)
9	AGND	Analog ground (shield)
10	CH1	Audio CH1 (AY-3-8910 B_OUT + 50% C_OUT)
11	AGND	Analog ground (shield)
12	CH0	Audio CH0 (AY-3-8910 A_OUT + 50% C_OUT)
		Use shielded cable for video connections Recommend using shielded cable for audio connections

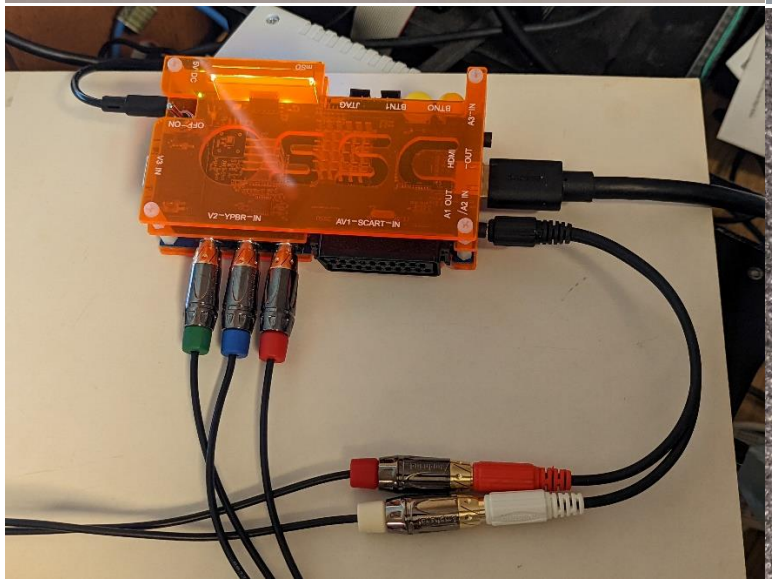
Joystick Connectors (P7-P10)			
Pin	ID	Description	Power Restrictions
1	AGND	Analog Ground	
2	+5V	+5v from +5VA supply (U23)	200ma PTC fuse F2
3	VREF	Analog Reference Voltage	+10v @ 100ma (limited by capacity of 78L10 U19)
4	AGND	Analog Ground	
5	IOB1/3/5/7	AY-3-8910 IOB1/3/5/7	1.6ma (one TTL load)
6	IOB0/2/4/6	AY-3-8910 IOB0/2/4/6	1.6ma (one TTL load)
7	IOA1/3/5/7	AY-3-8910 IOA1/3/5/7	1.6ma (one TTL load)
8	IOA0/2/4/6	AY-3-8910 IOA0/2/4/6	1.6ma (one TTL load)
9	Ai1/3/5/7	Analog Input 1/3/5/7	0-VREF (10v)
10	Ai0/2/4/6	Analog input 0/2/4/6	0-VREF (10v)
11	IOB1/3/5/7+	Buffered AY-3-8910 IOB1/3/5/7	40ma open collector (74LS07 U44, U45)
12	IOB0/2/4/6+	Buffered AY-3-8910 IOB0/2/4/6	40ma open collector (74LS07 U44, U45)
		The usual AY-3-8910 DIO configuration will have Port "B" as OUTPUT, Port "A" as INPUT AY-3-8910 DIO inputs have internal pull-up resistors to +5v AY-3-8910 IOBn is connected through 74LS07 to IOBn+ The '+' pins can be used to drive most external loads without any additional circuitry	

Audio/Video Cable Preparation for HA-8-3 Color Graphics Card

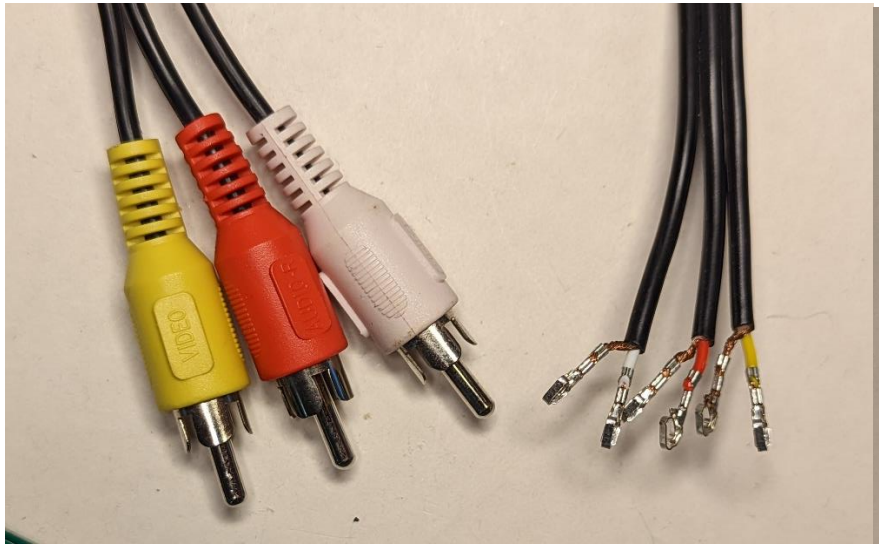
Terry Smedley
March 30, 2022

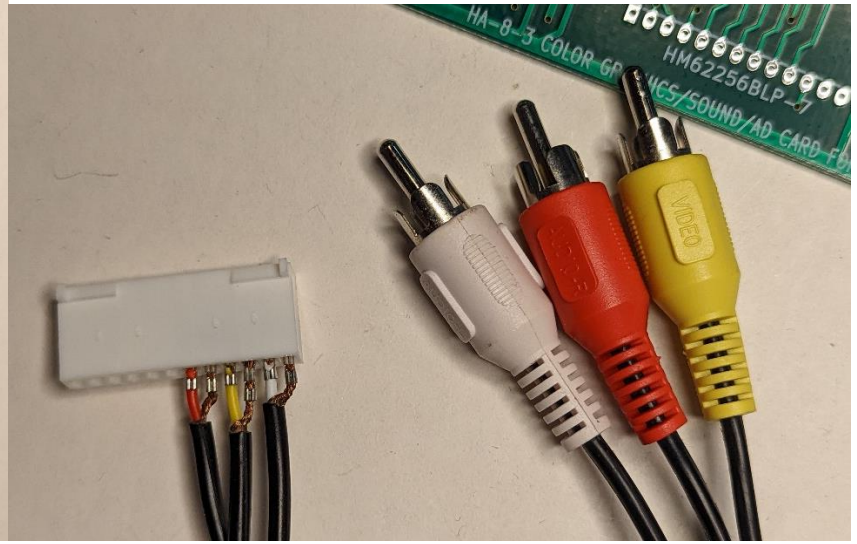






This shows custom made discrete cables for connection between the HA-8-3 and the OSSC. RG174 coax cable was used. This is 50 ohm cable so it is a technical mismatch with the 75 ohm load at the HA-8-3. I judged that for the very few feet from the H8 to the OSSC this would not be a significant quality issue. RG174 is thin and very flexible. One end was stripped and crimped with Molex KK terminals for insertion into the 12-pin polarized shell. At the other end, Amphenol ACPL-Cxx connectors were soldered on. In the H8, the cables were routed underneath the outboard mounting bracket and out an empty connector space on the rear panel. The OSSC uses a 1/8” stereo mini-jack for audio, so an RCA to miniplug adapter was used. A total of five cables is needed – Y, Pb, Pr, L audio, R audio.





An alternative (and less expensive) method is to modify molded cable sets. I used a ten-foot A/V cable and cut it in the middle to make two five-foot, three conductor cables. The KK terminals are crimped as before. I used A/V cable (composite yellow, white/red audio) because those cables are usually cheaper, smaller diameter, and easier to find than a component cable set with red, green, and blue connectors. Some of those component cable sets (often called “RGB” cables) have very thick conductors that would be awkward to manage inside the H8. To use this method, you have to be willing to accept that the video cables will likely not have the “right” color coding.

OSSC

Open Source Scan Converter

Converts and scales TMS9928 YPbPr 256x192 video output to crystal-clear HDMI output for display on a modern monitor, TV, or projector.

https://videogameperfection.com/support-files/ossc-quickstart.pdf

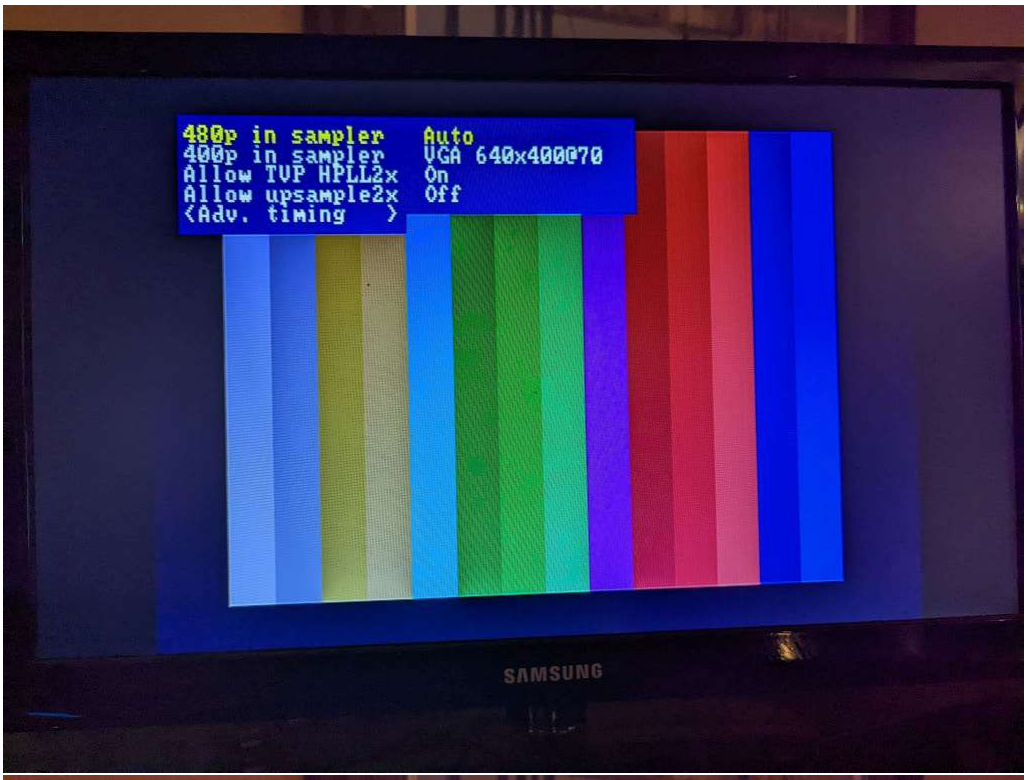
May be purchased on Amazon.com:

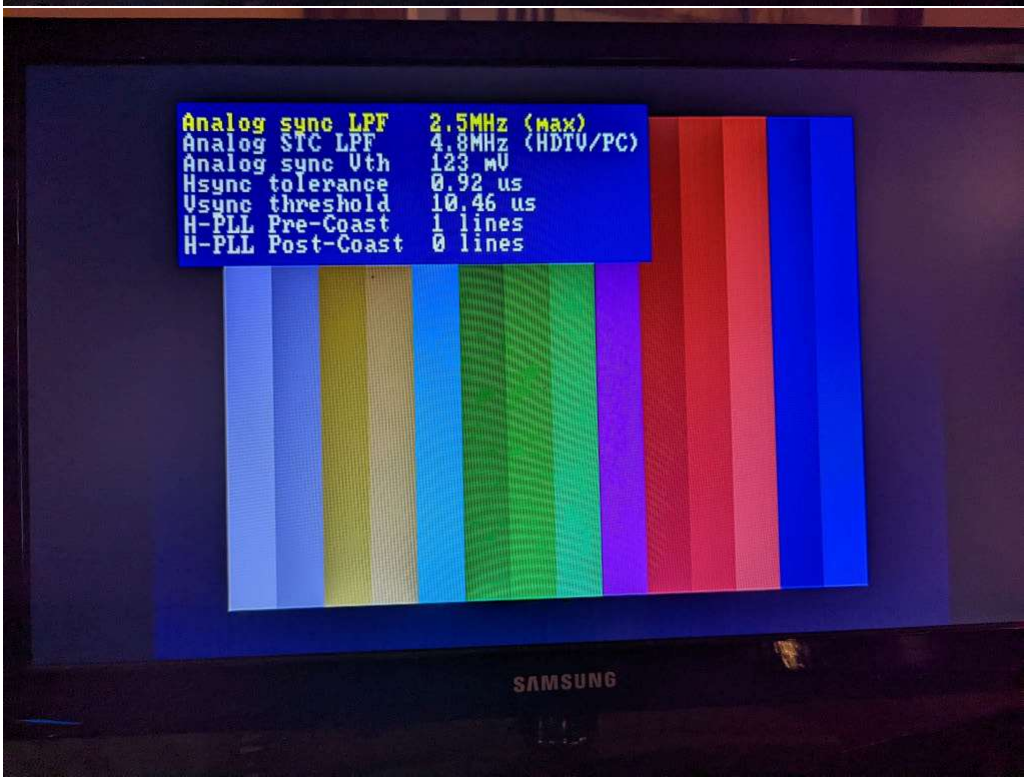
https://www.amazon.com/dp/B083FG6PXH

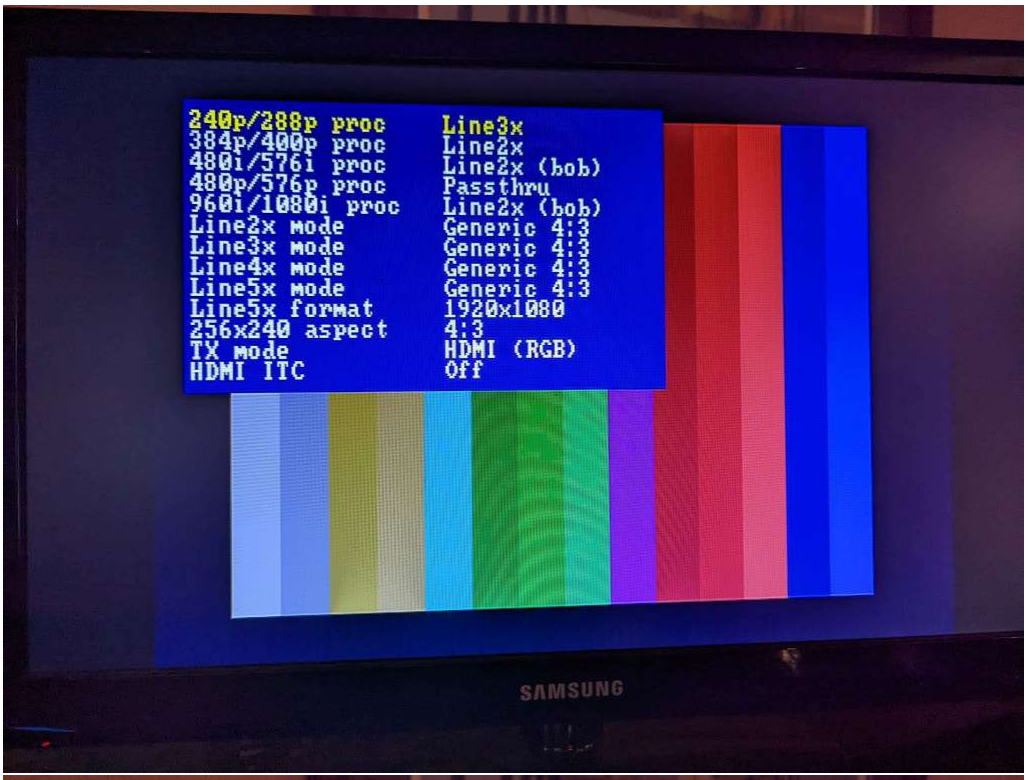
Sample configuration pages follow….

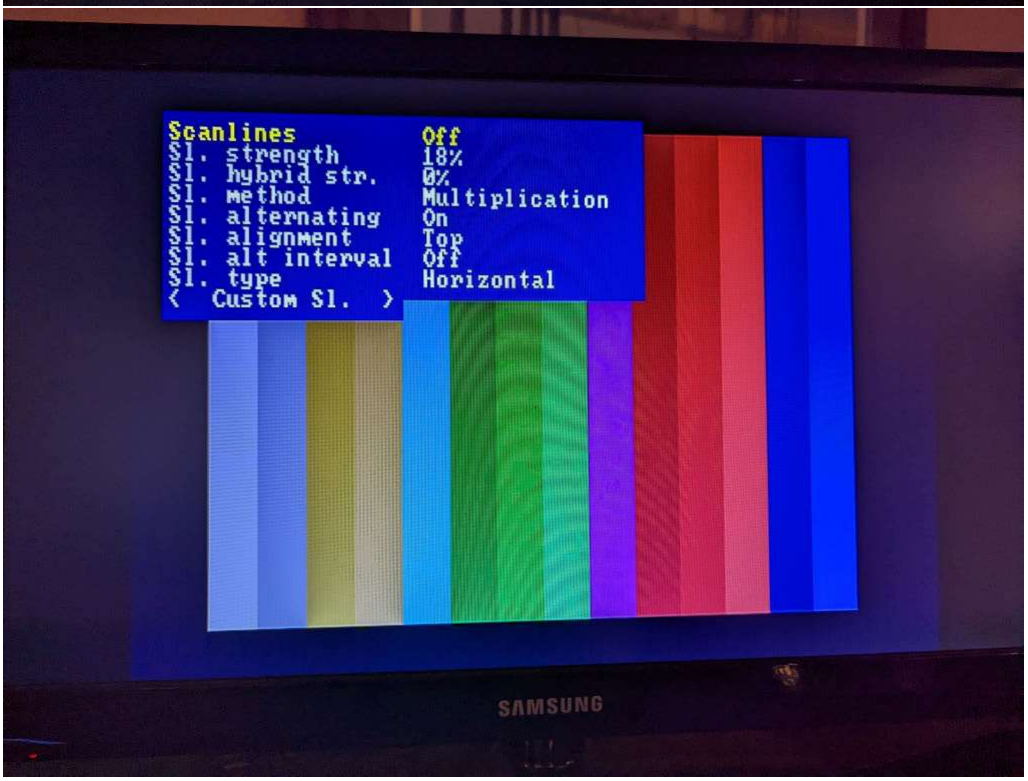

Video in proc >
Sampling opt. >
Sync opt. >
Output opt. >
Scanline opt. >
Post-proc. >
Compatibility >
Audio options >
Settings opt >
SAMSUNG

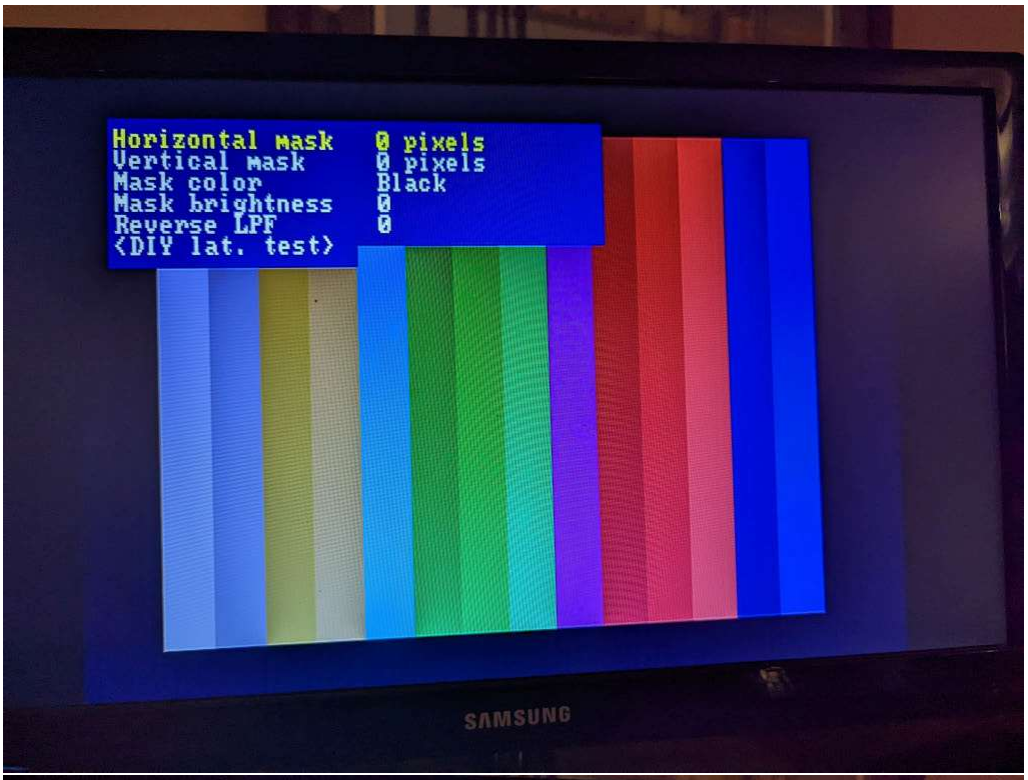

Video LPF Auto
YPbPr in ColSpa Auto
R/Pr offset 128
G/Y offset 128
B/Pb offset 128
R/Pr gain 26
G/Y gain 26
B/Pb gain 26
Pre-ADC Gain 5
Clamp/ALC offset 20
ALC V filter 512 lines
ALC H filter 16 pixels
SAMSUNG


480p in sampler Auto
400p in sampler VGA 640x400@70
Allow TVP HPLL2x On
Allow upsample2x Off
<Adv. timing >
SAMSUNG

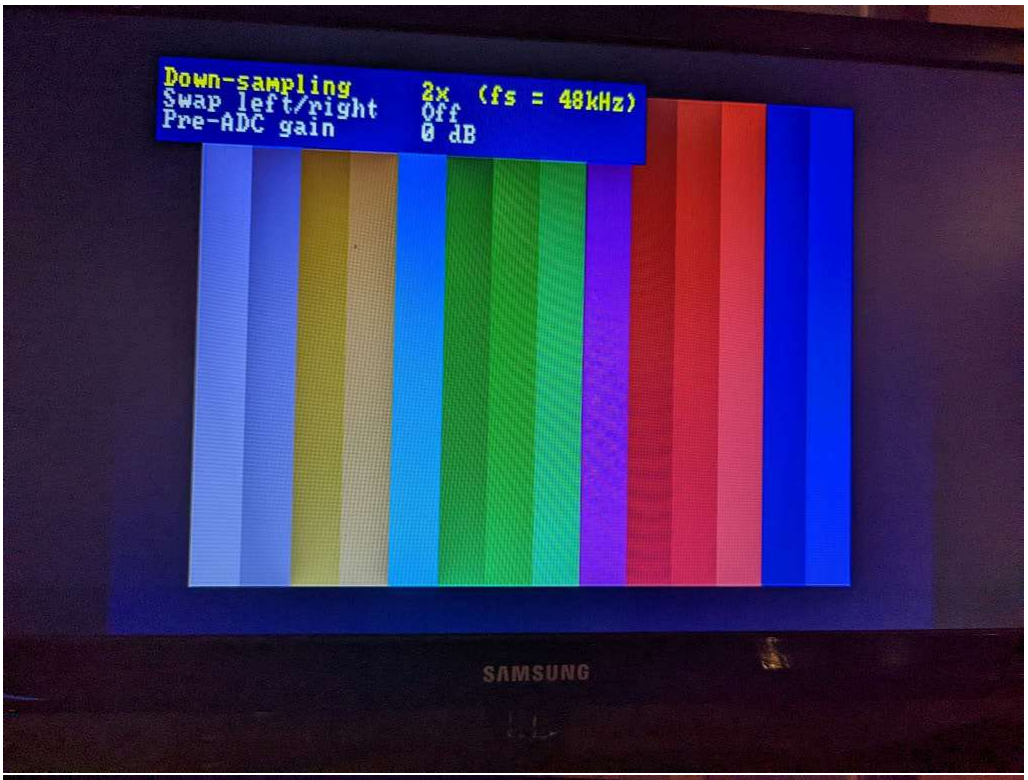

Analog sync LPF 2.5MHz (max)
Analog STC LPF 4.8MHz (HDTV/PC)
Analog sync Vth 123 mV
Hsync tolerance 0.92 us
Vsync threshold 10.46 us
H-PLL Pre-Coast 1 lines
H-PLL Post-Coast 0 lines
SAMSUNG


240p/288p proc Line3x
384p/400p proc Line2x
480i/576i proc Line2x (bob)
480p/576p proc Passthru
960i/1080i proc Line2x (bob)
Line2x mode Generic 4:3
Line3x mode Generic 4:3
Line4x mode Generic 4:3
Line5x mode Generic 4:3
Line5x format 1920x1080
256x240 aspect 4:3
TX mode HDMI (RGB)
HDMI ITC Off
SAMSUNG


Scanlines Off
Sl. strength 18%
Sl. hybrid str. 0%
Sl. method Multiplication
Sl. alternating On
Sl. alignment Top
Sl. alt interval Off
Sl. type Horizontal
< Custom Sl. >
SAMSUNG


Horizontal mask 0 pixels
Vertical mask 0 pixels
Mask color Black
Mask brightness 0
Reverse LPF 0
<DIY lat. test>
SAMSUNG


Full TX setup Off
AV3 interlacefix Off
AV3 use alt. RGB Off
Default HDMI VIC 0
Panasonic hack Off
SAMSUNG

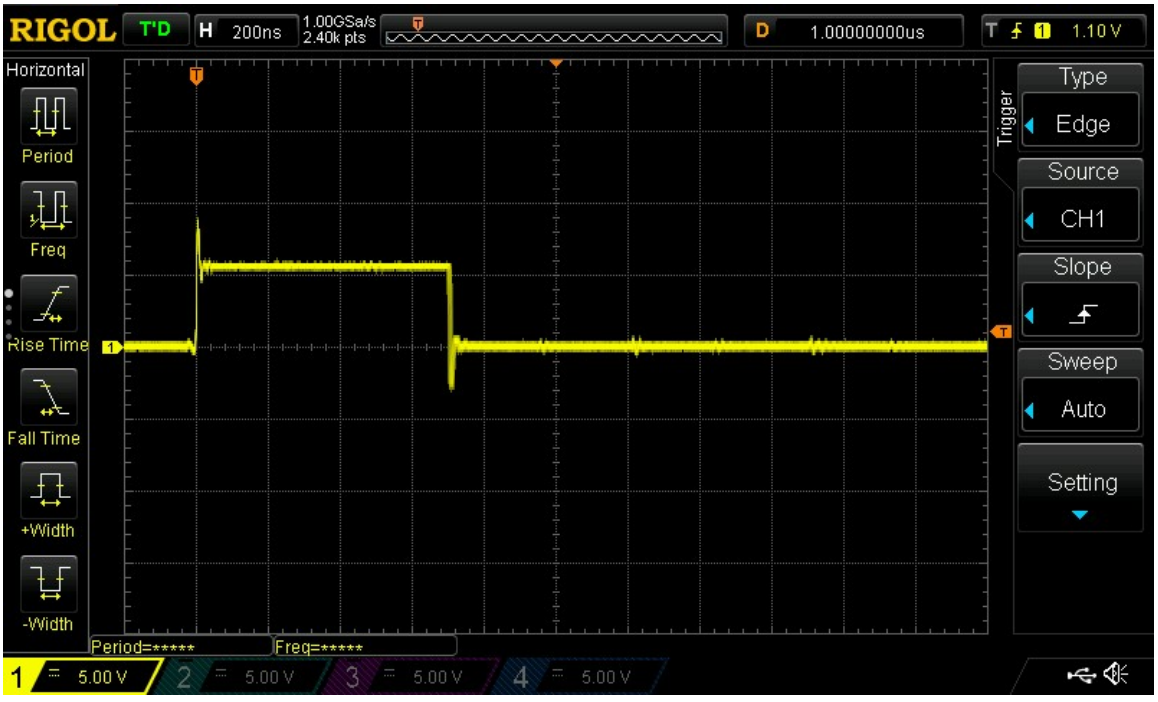

Down-sampling 2x (fs = 48kHz)
Swap left/right Off
Pre-ADC gain 0 dB
SAMSUNG


<Load profile > 6: <used>
<Save profile >
<Reset settings>
Link prof->input No link
Link input->prof Off
Initial input AV2_YPbPr
Autodetect input Off
Auto AV1 Y/Gs RGsB
Auto AV2 Y/Gs YPbPr
Auto AV3 Y/Gs RGsB
LCD BL timeout Off
OSD Full
OSD status disp. 5s
<Import sett. >
<Fw. update >
SAMSUNG

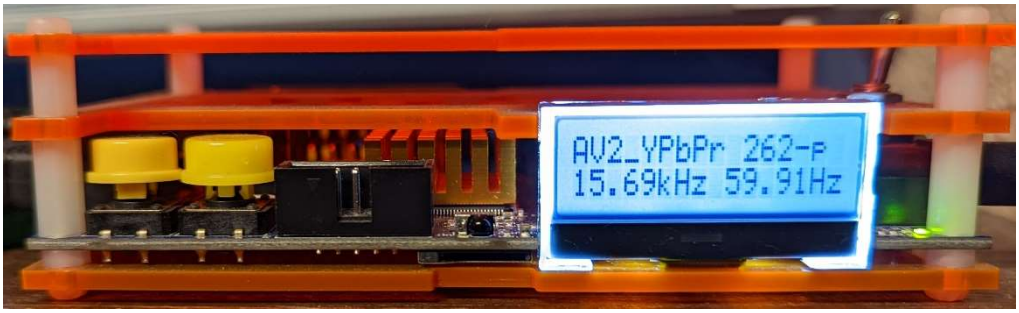
Initial Testing and Final Assembly of HA-8-3 Board

Solder all components to the board, but do not install any socketed ICs or oscillators until the following power supply checks are made. Install the board into any open H8 bus slot.

1) Power up the H8 and verify that all power LEDs are lit: +5va (D1), +5vp (D13), +12v (D12), and -12v (D14)
2) Using a voltmeter, test for proper regulated voltages:
   JP19 Pin 1 +5v
   JP20 Pin 1 +5v
   JP9 Pin 2 +12v
   JP10 Pin 2 -12v
   TP1 +10v (ADC reference voltage)
   TP2 +5v (ADC isolated +5v supply)
3) If voltage checks are OK, power off the H8, remove the board and populate all socketed components
4) Set all jumpers according to the Jumper Settings tab in the HA-8-3 Excel worksheet
5) Reinstall the board, power on the H8 and verify another ADC reference voltage:
   TP3 -10v
6) If you are enabling I/O wait states on the HA-8-3 card, be sure you have followed the instructions for modifying the v3 or v4 CPU or H17/37/67 controller card: http://koyado.com/heathkit/New-H8-Website/download/reworks_for_h17_h67_h37_z80-v3_z80_v4_boards.pdf
7) If you are enabling I/O wait states, connect an oscilloscope to pin 1 of U29A (74ALS1005). Using 10K trimmer pot RV3, adjust the length of the I/O wait pulse to 700ns. This wait length has been found to support 16MHz operation with most cards.



8) Set the three video level trimmers (RV1, RV2, RV4) to their middle position.
9) If you are using a component monitor (without OSSC scan converter), connect the YPbPr Video and Ch0/Ch1 Audio outputs to your equipment using shielded cables at P14.
10) If you are using the OSSC, attach an HDMI display to the OSSC and the three video cables to the AV2 input. Configure the OSSC to use the AV2_YPbPr input using either the OSSC buttons or the OSSC remote control. If you wish to use the OSSC to send the HA-8-3 audio to your HDMI display device, use a 1/8" miniplug to RCA adapter to connect the two audio outputs from the HA-8-3 to the OSSC. With the H8 powered on, video cables connected, and the AV2 input selected, the front panel of the OSSC should show a display similar to this:



Software Verification of the HA-8-3 Board

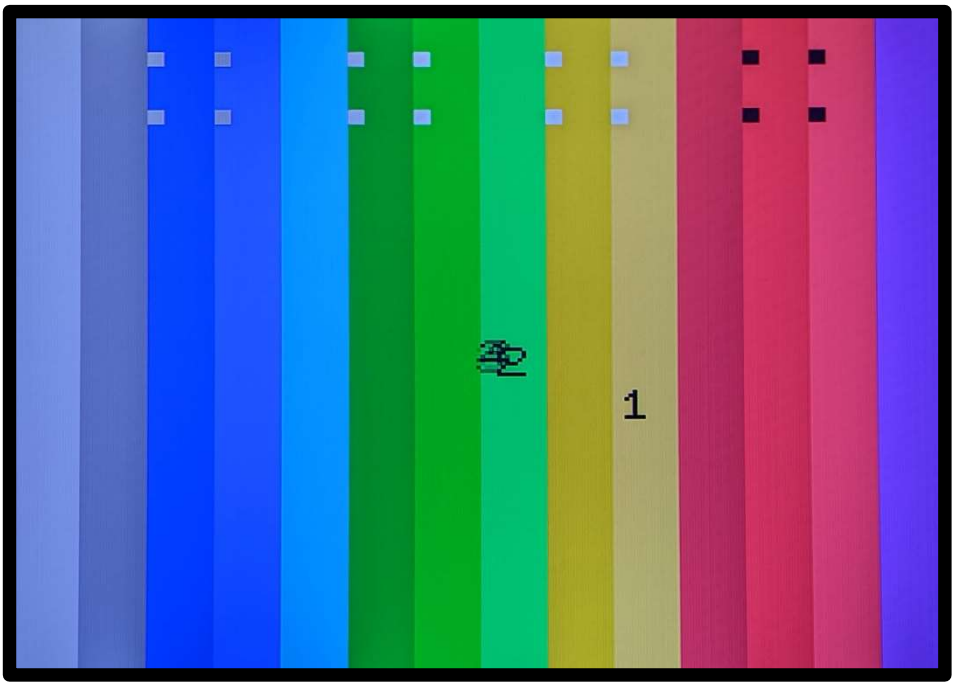
The HA-8-3-1 board by default uses different port addresses for the ADC and APU sections of the card than the original board. This was done to allow use of the board in a "three drive" (H17, H37 plus H67) system. The table below shows the port assignments for both the original and new cards:

Device	**Original**	**New Board**
VDP	270-271Q	270-271Q
PSG	272-273Q	272-273Q
APU	274-275Q	364-365Q
ADC	276-277Q	366-367Q
S/W Reset	N/A	066Q

Any software originally written for the HA-8-3 card that uses only the VDP and PSG sections will work exactly as with the original board. Software using the APU or ADC (for example, joysticks) must be modified to run with the new board in a three-drive environment by changing the port assignments as shown above.

The fastest way to initially test the new board is to run the HA83DIAG program from the original HA-8-3 distribution diskette (an H8D disk image is available from the SEBHC). With no changes to this program, you can immediately verify the operation of the VDP and PSG sections of the card. HA83DIAG is fully documented in the original NOGDS manual for the board. Start the tests with the CPU at 2MHz.

1) Run the HA83DIAG program from HDOS2.0 or later
2) The program silently tests the VRAM which takes a while at 2MHz. It may appear to be hung but check the I/O activity LED on the HA-8-3 board. It should be lit constantly as the board is accessed to do the VRAM tests. It takes about 60 seconds to complete the VRAM test at 2MHz, during which time you will see nothing happening on either the H19 or the color display.
3) At the completion of the VRAM tests, a message should appear on the H19 console:
   "No VRAM failures detected"
4) A color bar pattern should appear on the color display, and in succession you should hear four sound effects:
   Laser
   Whistling Bomb
   Wolf Whistle
   Race Car
5) The color monitor should now show two rows of eight white squares at the top of the screen. The squares indicate the status of the sixteen digital I/O pins of the AY-3-8910. An unconnected pin will show as a white square. If the pin is grounded, the corresponding square will be black. You can connect jumpers or switches to the joystick headers to test the I/O behavior.
6) That completes the testing that can be done without modifying the ADC port assignment in the HA83DIAG program. Use ^Z^Z to exit the program.
7) For further video exploration, you can run the "KALEIDO" program on the NOGDS diskette. It will produce a constantly varying kaleidoscope-style display on the color monitor.
8) To modify the HA83DIAG program for the new port assignments, use an editor to change this line in the ADDEF.ACM file on the NOGDS diskette from:
   ```
   ADADR       EQU   276Q        ;A/D I/O port
   ```
   to:
   ```
   ADADR       EQU   366Q        ;A/D I/O port
   ```
9) No other changes are necessary. Reassemble the program using the HDOS assembler:
   ```
   >ASM SYx:HA83DIAG,=SYx:HA83DIAG
   ```
   (specify disk device and number as needed.)
10) Rerun the HA83DIAG program. When the test program reaches the final stage, with the two rows of eight squares at the top, you should see the numbers '1', '2', '3', '4' somewhere near the middle of the screen. These numbers correspond to the X-Y position determined by measuring the voltage at each pair of analog inputs at each joystick connector. If you vary the voltage on these pins between 0 and +10v (VRef), you should see the corresponding number move on the screen.
11) You can fabricate an ADC/joystick test using a 5K or 10K potentiometer. One end goes to AGND (pins 1 or 4 of the joystick connector), the other end goes to Vref (pin 3 of the connector), and the wiper goes to one of the analog inputs (pin 9 or 10).



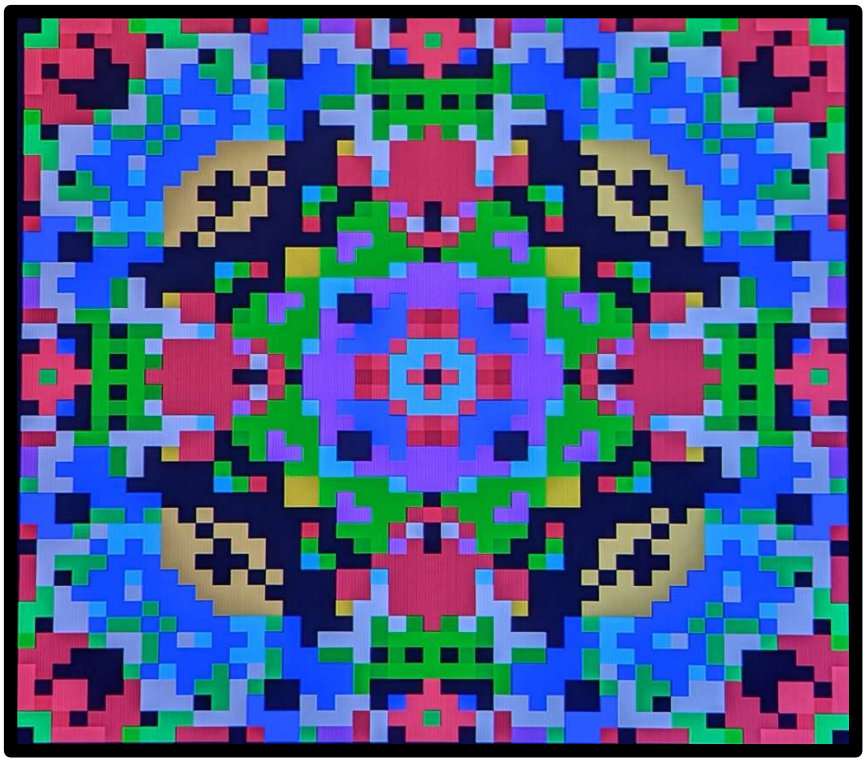
This is a photograph of the image from the HA83DIAG program displayed on an Epson projector. The actual projected image is about 100"diagonal. Even at that magnification, the edges are crisp. The image was upscaled from the HA-8-3 using an OSSC. You can see four of the AY-3-8910 DIO bits are grounded ('0', displayed as black), while the others are open ('1', displayed as white). Joystick 1 is being held slightly to the right of center. The analog inputs for joysticks 2-4 are connected to +5v for this test, so they appear directly in the center of the screen (because 5v is exactly half the 10v reference voltage).

Other Software for the HA-8-3

NOGDS Distribution Diskette

KALEIDO	continuous kaleidoscope pattern
BARS	the same color bar display shown in HA83diag

There are also individual programs to display a full screen of each color (LBLUE, DRED, GRAY, etc). You can use these programs to help adjust the relative amounts of Green, Blue, and Red in the video signal using the three trimmer pots on the HA-8-3-1 board. You can experiment with the full range of settings to see how the image and color are affected. You will probably find that you get the best color balance when all three trimmers are in about the same position, close to their midpoint. There are also color adjustments (dozens!!) possible within the OSSC menus.



Photograph of the projected image from the KALEIDO program. Transitions between colored squares are crisp. Using the OSSC scan converter and an Epson projector.

HUG COLOR GRAPHIC SOFTWARE 885-1098

MUSICK plays HA-8-2 .PLA music files through the HA-8-3 PSG with kaleidoscope video

HUG COLOR RAIDERS AND GOOP 885-1114

The two games on this diskette do not use the ADC (analog inputs), so they will run just fine on the new board with no modification.

Programs Written by SEBHC Members

WARLORDS by Les Bird

WARLORDS is a fast-paced game that runs under CP/M 2.2 It takes full advantage of the capabilities of the HA-8-3, with great graphics, sound effects, and real analog joystick control.

Because this program uses the ADC to read the joystick position, it needs a very small change for the new port addresses. In WARLORDS.ASM, locate these EQUates and change from:

```
JOYDAT      EQU   0BCH
JOYCTL      EQU   0BCH
```

to:

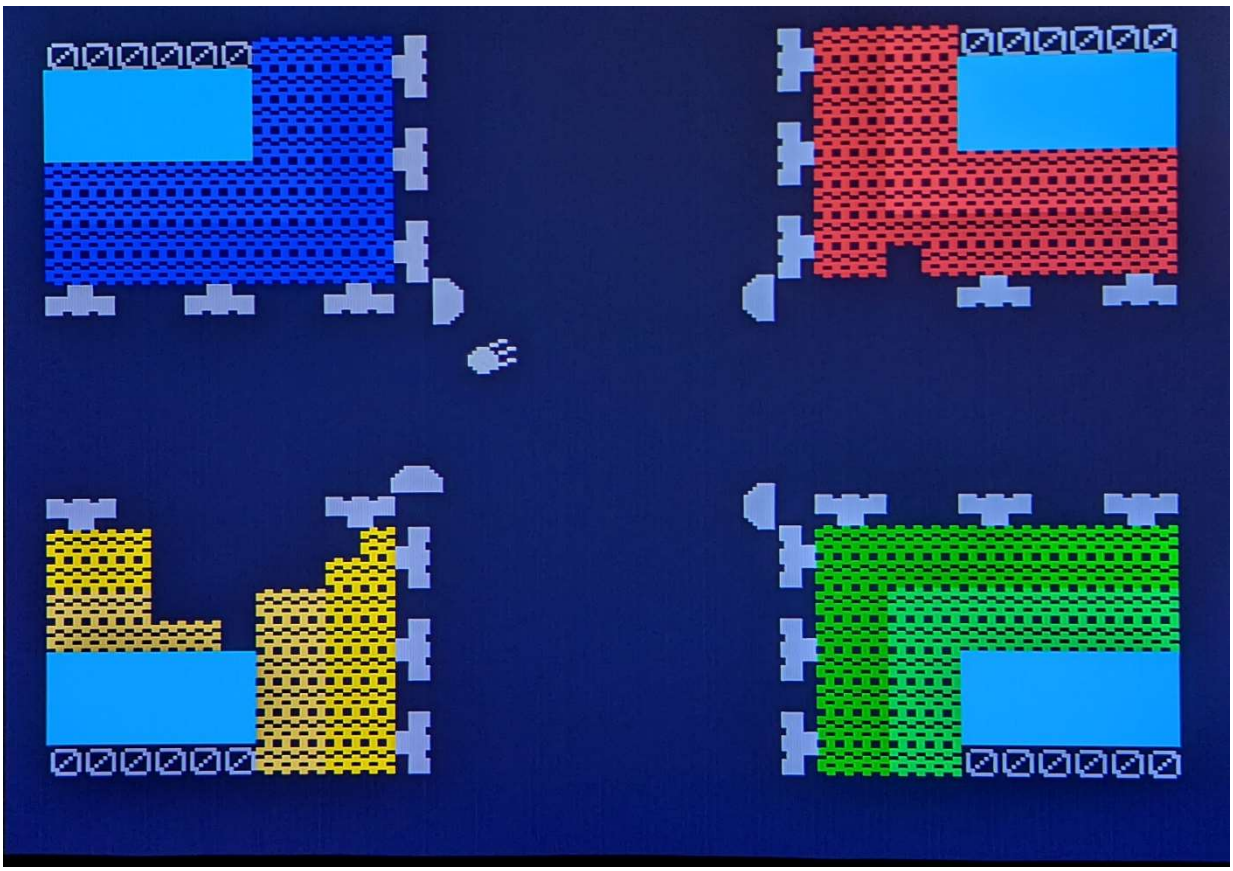
```
JOYDAT      EQU   0F6H
JOYCTL      EQU   0F6H
```

Reassemble with the CP/M assembler:

```
ASM WARLORDS.AAZ
LOAD WARLORDS
```

(you may need to change .AAZ to the drive letter you are using for the WARLORDS source files)

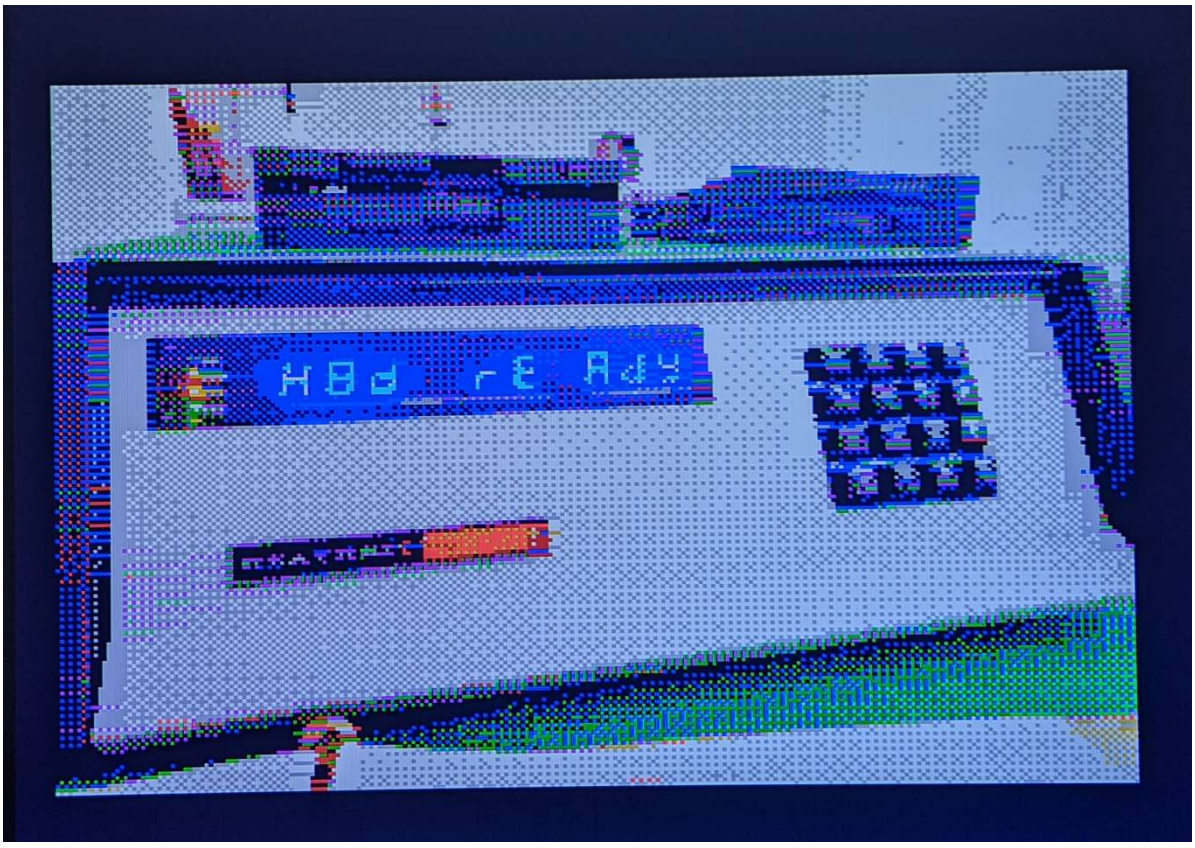
Les Bird has written other games for the HA-8-3 as well.



SLIDESHO

This program was written at the suggestion of Norberto Collado to display a pre-processed photographic image through the HA-8-3. A Windows PC program (CONVERT9918.EXE, from Harmless Lion) processes a .JPG file into a pattern and color file in the TMS99x8 format. The processed files can be quickly copied to the H8 by USB transfer using the V-Utilities by Glenn Roberts. SLIDESHO was a quick proof-of-concept program, so it has no bells and whistles. It continuously displays three image files (named TM99181.TIC/TIP, TM99182.TIC/TIP, and TM99183.TIC/TIP). The images are remarkably good for a VDP with just 256x192 16-color capability.

Get the CONVERT9918.EXE file here: http://harmlesslion.com/cgi-bin/onesoft.cgi?2



CLS – Clear Screen

Sometimes simple things help a lot. The TMS99x8 VDP on the original HA-8-3 can only be reset by a Master System Reset – in other words, a complete system restart. Especially during software development, that's a pain. A "soft reset" has been implemented on the HA-8-3. Any WRITE to port 066Q will reset the VDP, PSG, and APU without doing a full system reset. This tiny program does just that – writes an arbitrary value to port 066Q. This very simple program also clears the color display completely, a handy feature because many programs end and leave the color display forever showing whatever was last written to it.

HA-8-3 Troubleshooting using the H8 Front Panel

The H8 Front Panel can be used to verify and troubleshoot basic board operation
These instructions assume you are already familiar with how to use the
front panel to OUTput values to a port, and INput values from a port

ADC and 8-channel MUX

ADC port is 366Q by default (JP14 off), 276Q in NOGDS mode (JP14 on)

Connect 5v from pin 2 of any joystick header to an analog input on pins 9 or 10
OUTput an ADC channel number (0-7) to the ADC port

000 366	OUT	selects analog input 0
000 366	IN	reads the value from the ADC
xxx 366		xxx should be approximately half-scale (between 170Q-210Q)

You can also connect GND from pin 1 or Vref from pin 3 of any joystick header
GND should result in an Input value close to 000Q, Vref should be close to 377Q

APU Basic Function Test

APU port is 364Q by default (JP14 off), 274Q in NOGDS mode (JP14 on)

To quickly test basic APU function, you can push several values onto
the APU stack and verify they read back correctly. For example:

000 364	OUT	
001 364	OUT	
123 364	OUT	
377 364	OUT	
000 364	IN	
377 364	IN	377 is TOS (top of stack)
123 364	IN	readback in reverse order
001 364	IN	
000 364		first value written, last value read

VDP VRAM Function Test

VDP port is 270Q

There are easier ways to test VRAM, but it can be done from the FP!
To test VRAM address 000.000:

000 271	OUT	low order address byte
100 271	OUT	high order address byte with high order bit set
123 270	OUT	value to store at address 000.000
000 271	OUT	low order address byte
000 271	OUT	high order address byte with high order bit cleared
000 270	IN	
123 270		should read back the value you wrote

PSG Function Test

PSG port is 272Q

Connect amplified speakers to the CH0 and CH1 audio outputs

010 273	OUT	select "C" channel volume register
017 272	OUT	highest volume
005 273	OUT	select "C" channel frequency register
004 272	OUT	coarse frequency (period) value
007 273	OUT	select ENABLE register
373 272	OUT	enable the "C" sound generator

You should hear sound from both the CH0 and CH1 outputs

(On this board, the "C" channel is mixed to both CH0 and CH1 outputs)

Turn it off by a FP reset, or

007 273	OUT	ENABLE register
377 272	OUT	disable everything

HA-8-3 Game Console/Joystick Ideas

Thumb sticks from Adafruit or Sparkfun
https://www.adafruit.com/product/512

Connections to HA-8-3 Joystick Headers P7-P10:

Vcc	Pin 3, Vref, +10v
XOut	Pin 10, Ai0/2/4/6
YOut	Pin 9, Ai1/3/5/7
Sel	Pin 8, IOA0/2/4/6
GND	Pin 1, GND

This is for the typical HA-8-3 configuration, where Port A is input and Port B is output. The configuration/direction of DIO pins on the AY-3-8910 is software configurable.

https://www.adafruit.com/product/3489

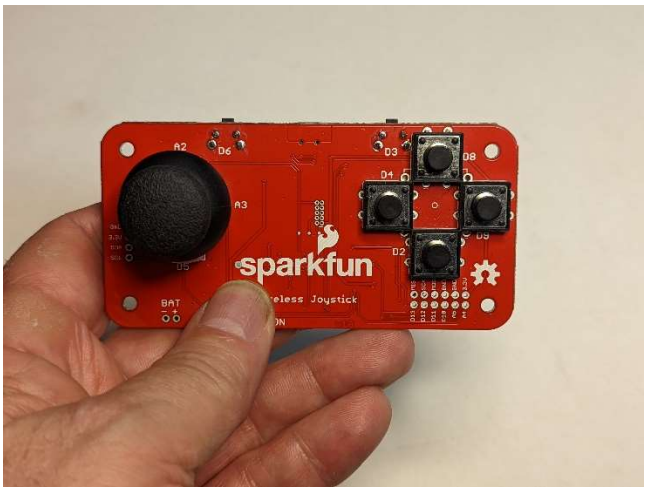
Available in five colors. Very rugged pushbutton – designed for gameplay. Integral LED with built-in dropping resistor for easy hookup to 5v circuit.

https://www.servocity.com/2-function-joystick-ball-stick/

10K potentiometers – much easier to control than the thumbstick.

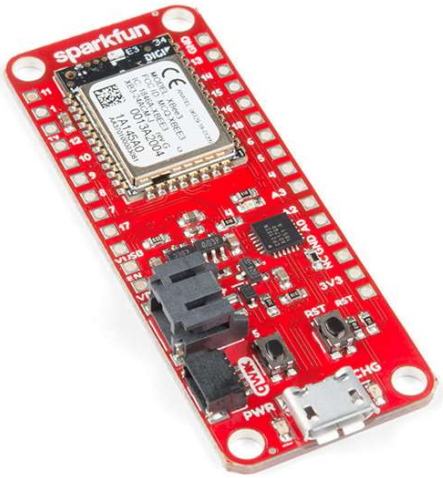
https://www.servocity.com/4-function-joystick/

THREE 10K potentiometers. The pushbutton at the top of the stick is especially convenient for gameplay.



https://www.sparkfun.com/products/14051

WIRELESS remote control! A terrific “starter kit” to cut the cord. Requires a though-hole XBEE 2 or 3 RF module for the transmitter, and an XBEE 2 or 3 + breakout board for the receiver. The receiver must be scratch-built (easy, just a few components). This Sparkfun kit can be assembled in many different ways – two thumbsticks, or one thumbstick plus four pushbuttons. Includes two front “fire” buttons in all configurations. The XBees allow direct connection to the HA-8-3 because they can translate the analog voltage input at the remote end to a PWM varying-voltage output at the receiving end. As far as the HA-8-3 and software is concerned, it’s looking at a directly attached potentiometer – no software changes are required, and the XBee remote can be plugged in directly to the joystick connector.



https://www.sparkfun.com/products/15454

Sparkfun XBee 3 “Thing” contains just about everything needed to build a custom wireless transmitter OR receiver for a wireless joystick. Has on-board LiPO battery charging circuitry.

Here’s a receiver built from the Sparkfun XBee “THING”, put in a small plastic enclosure:

This same wireless receiver works with either the Sparkfun wireless remote, or the custom built wireless remote shown here:

The “fire” button located at the top of the joystick is very convenient to use with gameplay. You can play with one hand, and don’t have to remove your hand from the joystick control to send button commands.

Here’s a wired controller, with five potentiometers and three pushbuttons:

Tensility ultra-flexible UL2464 28awg cable was used to connect to the HA-8-3-1.

HA-8-3 Software Transmittal (HDOS 2.0)

"CLEAR SCREEN" utility for HA-8-3
CLS.ASM
CLS.ABS

Many programs written for the HA-8-3 don't clear the screen when the program terminates. This program does nothing more than write an arbitrary byte to Port 066Q, which triggers a soft reset of the APU, PSG, and VDP on the HA-8-3 card, as well as the APU on the System Support I card.

Assembly/Compile Dependencies:

HOSDEF.ACM	HDOS 2.0 distribution
HOSEQU.ACM	HDOS 2.0 distribution

Quick VRAM Test for TMS9918/TMS9928
NCVRAM.ASM
NCVRAM.ABS
HEXOUT.ACM

Does a simple test for HA-8-3 VRAM write/read. For any byte in error, shows the memory location, expected byte, and received byte. This is not an exhaustive test - a single byte is written and verified at each memory location. Tests 16K of VRAM (0000H – 3FFFH).

Assembly/Compile Dependencies:

HDOS.ACM	HDOS 2.0 distribution
ASCII.ACM	HDOS 2.0 distribution
HEXOUT.ACM	attached
VDPDEF.ACM	HA-8-3 NOGDS distribution
VDPIO.ACM	HA-8-3 NOGDS distribution

TMS99x8 “Slide Show” using pre-processed graphics images

SLIDESHO.ASM
SLIDESHO.ABS
TM99181.TIC
TM99182.TIP
TM99182.TIC
TM99182.TIP
TM99183.TIC
TM99183.TIP

Continuously loops through three images pre-processed by the HarmlessLion Convert9918 program on a Windows PC: http://harmlesslion.com/cgi-bin/onesoft.cgi?2. The image files (two per image, a Color and a Pattern file for each) must reside on the same disk drive as the program. ^C will terminate the program.

Assembly/Compile dependencies:

HOSDEF.ACM	HDOS 2.0 distribution
HOSEQU.ACM	HDOS 2.0 distribution
ASCII.ACM	HDOS 2.0 distribution
H17ROM.ACM	HDOS 2.0 distribution
MTR.ACM	HDOS 2.0 distribution
HEXOUT.ACM	attached
VDPDEF.ACM	HA-8-3 NOGDS distribution
VDPMEM.ACM	HA-8-3 NOGDS distribution
VDPMOVES.ACM	HA-8-3 NOGDS distribution
VDPIO.ACM	HA-8-3 NOGDS distribution

Wait State Processing on the Heathkit HA-8-3 Color Graphics Card

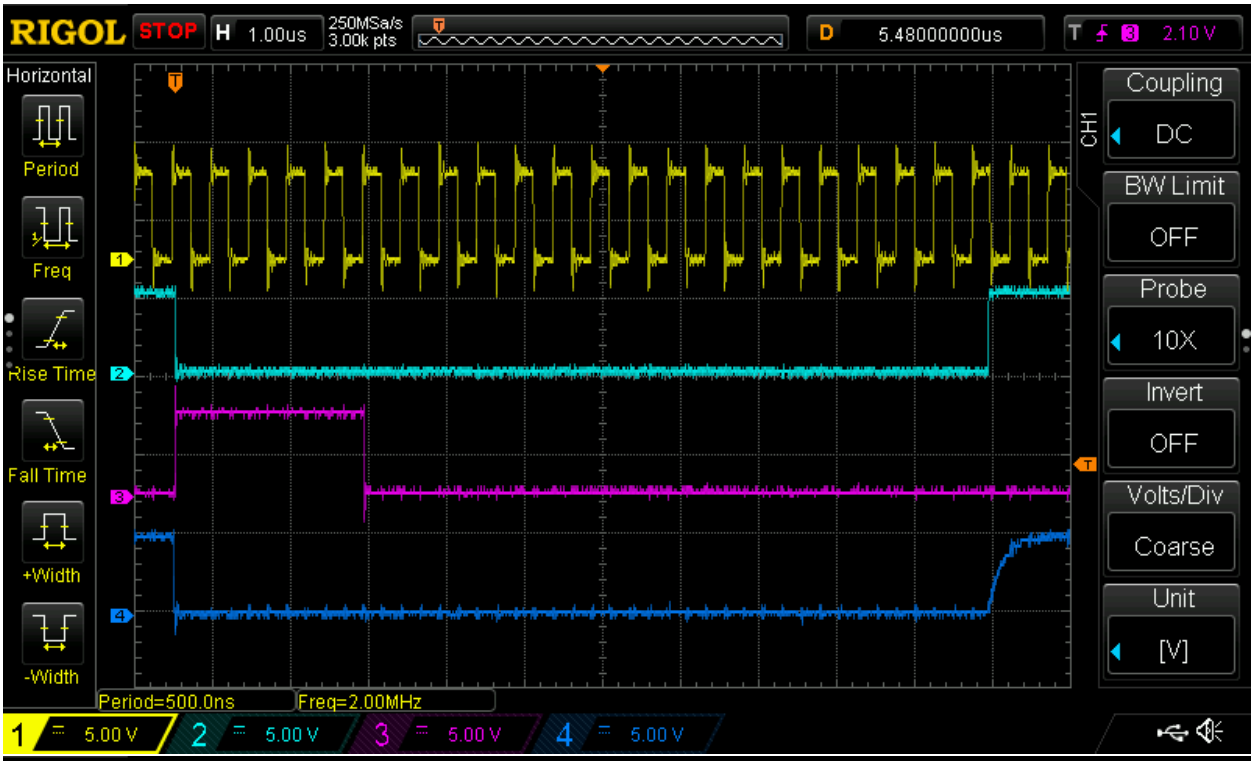
Terry Smedley
March 29, 2022

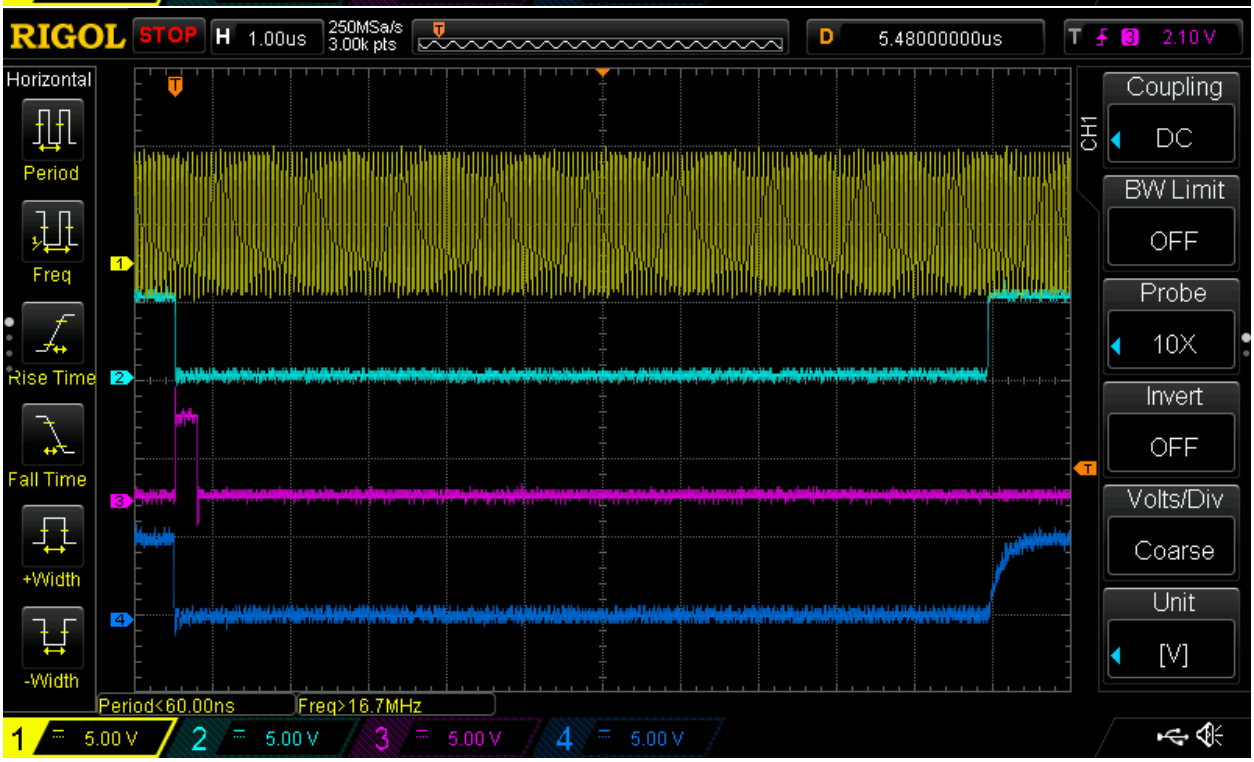
ADC Wait States

Yellow: Bus Clock, Cyan: AD_BUSY, Magenta: AD_WAIT, Blue: RDYIN*

TOP: 2MHz CPU speed, BOTTOM: 16MHz CPU speed

NOTE: The AD7574 BUSY signal causes the CPU to enter a wait state until the conversion is complete – this period is not affected by the CPU speed. The purpose of the AD_WAIT signal was to guarantee the CPU entered the WAIT state while waiting for the AD7574 to assert its BUSY signal. The length of AD_WAIT is CPU speed dependent.



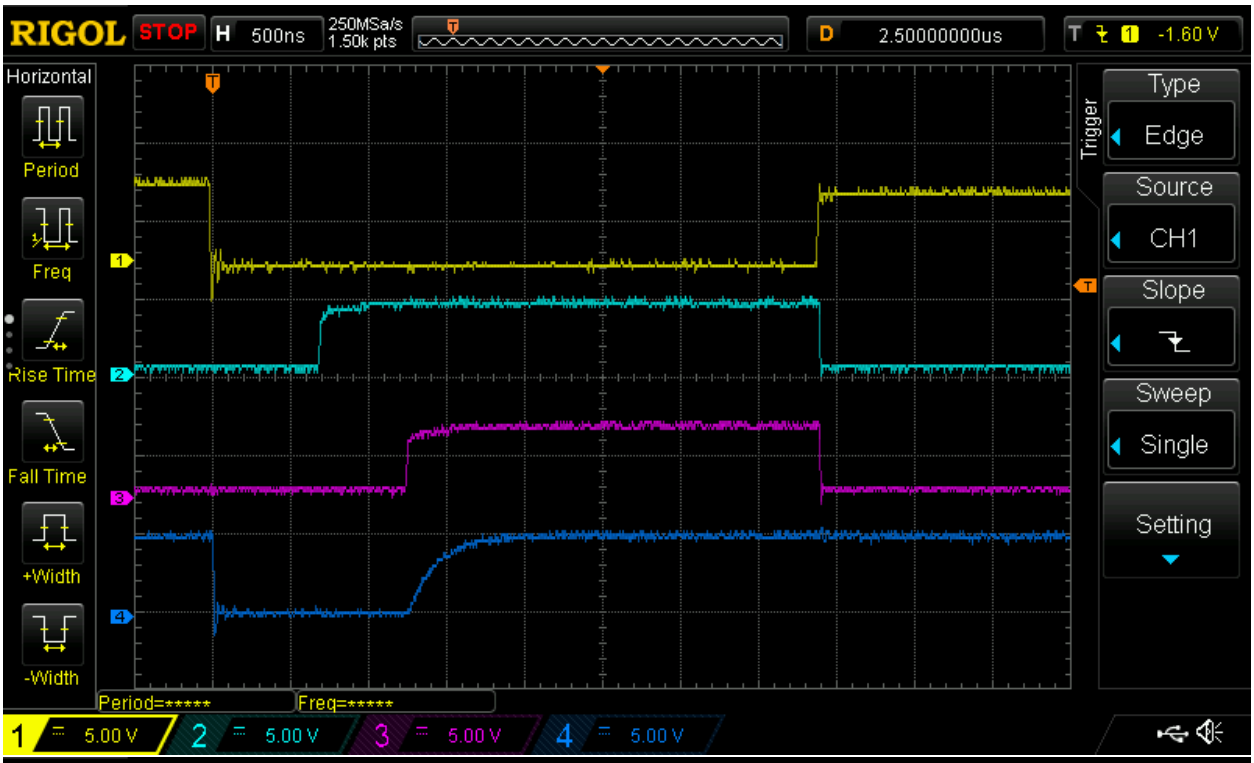


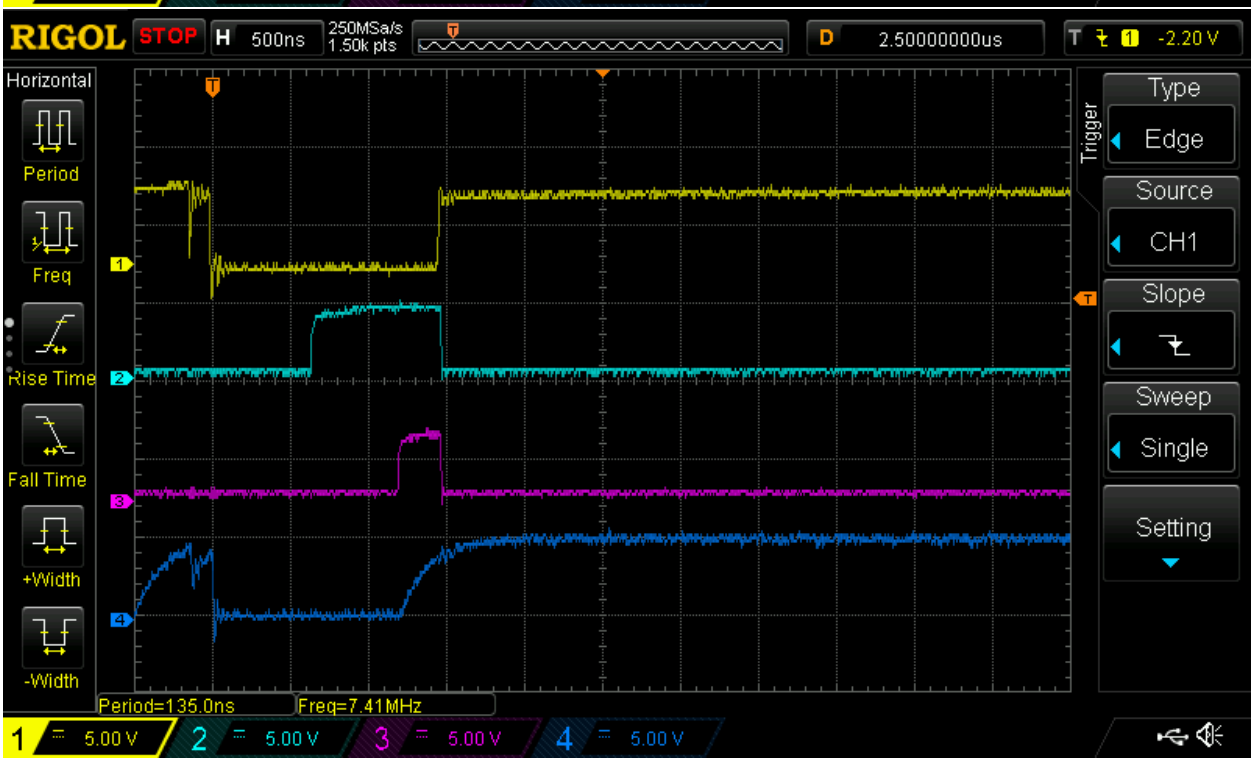
PSG Wait States

Yellow: PSGSEL, Cyan: PSG ENABLE, Magenta: PSG DONE, Blue: RDYIN*

TOP: 2MHz CPU speed, BOTTOM: 16MHz CPU speed

NOTE: Shift register U37 counts “Colorburst Clock” (JP15) cycles starting with the PSGSEL pulse. Three clock cycles after PSGSEL, PSG ENABLE goes active, two cycles after that PSG DONE goes active. The CPU is held in wait state until PSG DONE. The trailing edge of PSGSEL resets the ENABLE and DONE signals. The length of the wait state is independent of the CPU clock (compare the BLUE trace between the two images). The PSG has two clock cycles between ENABLE and DONE to access the bus.



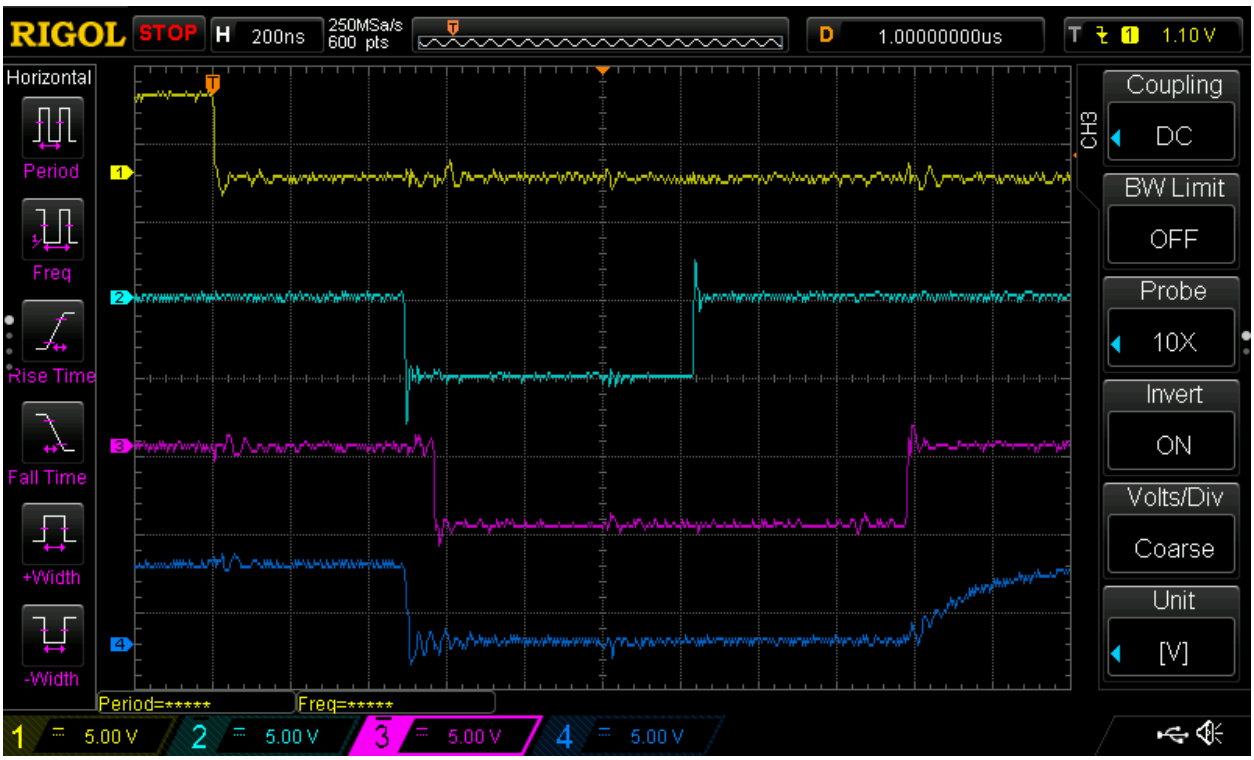


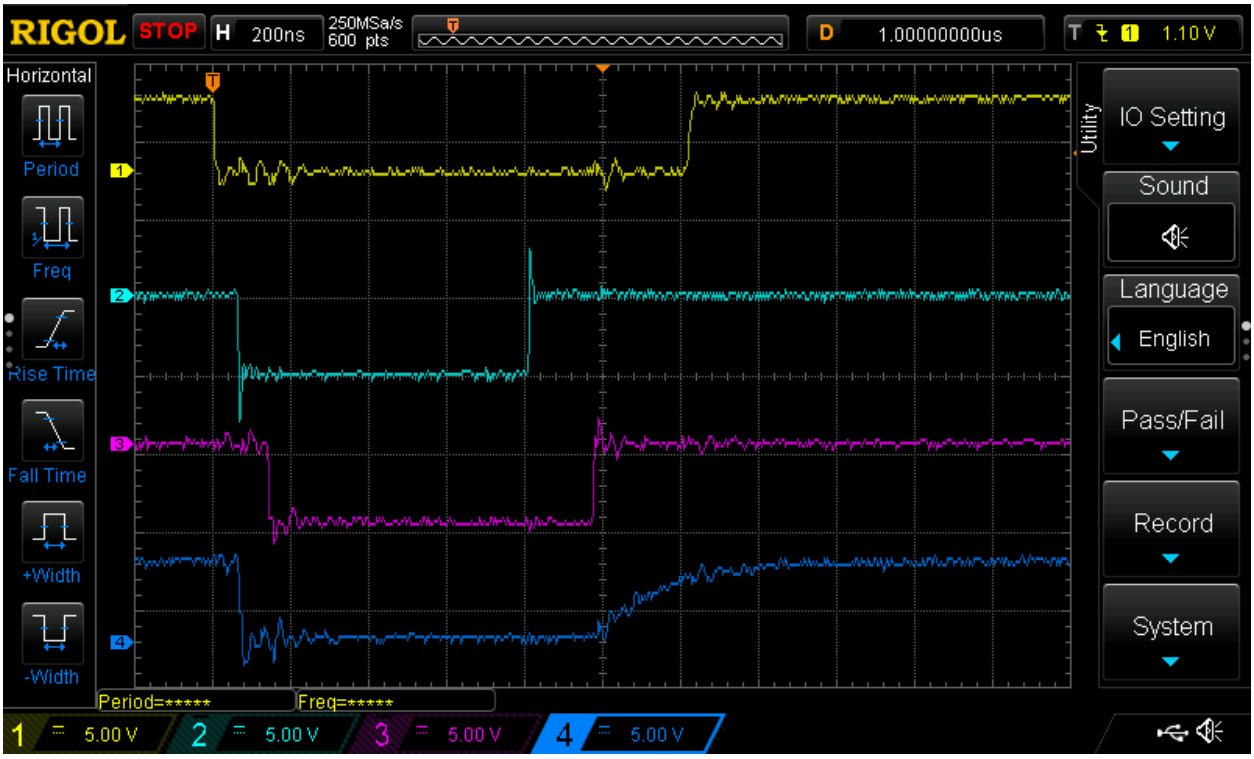
APU Wait States

Yellow: APUSEL, Cyan: I/O WAIT, Magenta: APU_WAIT, Blue: RDYIN*

TOP: 2MHz CPU speed, BOTTOM: 16MHz CPU speed

NOTE: The general I/O WAIT (from the onboard wait state generator, or the comparable System Support I generator) holds the CPU until the APU can assert its PAUSE (APU_WAIT) signal. The length of the PAUSE depends upon the APU instruction being executed. None of this depends upon the CPU clock speed.





These APU traces also show the difference between generating Chip Select based only on address lines (APUSEL, yellow line) compared to ANDing the address selection with IORQ (the leading edge of the general I/O Wait pulse, Cyan line, aligns with the leading edge of the IORQ pulse). The chip select can be generated one full clock cycle ahead of the IORQ pulse. This is required to meet the AM9511 timing specification, and it was also found to be necessary for some older, slower 8255s to work on the PPIO card.

Terry Smedley
Humptulips, WA
March 29, 2022

```
Date            08/07/2021              ;
Revision        01                              ;
Designer        T.SMEDLEY               ;
Company         KOYADO.COM              ;
Assembly        None                    ;
Location        for Koyado clone        ;
Device          g16v8a                  ;
/* This GAL is for the HA-8-3 clone                                    */
/* ******************** INPUT PINS *************************/
PIN  1 = !IORQ          ;       /*                                     */
PIN  2 = A1                     ;       /*                             */
PIN  3 = A2                     ;       /*                             */
PIN  4 = A3                     ;       /*                             */
PIN  5 = A4                     ;       /*                             */
PIN  6 = A5                     ;       /*                             */
PIN  7 = A6                     ;       /*                             */
PIN  8 = A7                     ;       /*                             */
PIN  9 = !IO_SEL                ;       /* SELECT SIGNAL FOR PORT RANGE */
PIN 11 = M1                     ;
/* ******************* OUTPUT PINS *************************/
PIN 15 = VDC                    ;       /* CHIP SELECT FOR TMS9918      */
PIN 14 = PSG                    ;       /* CHIP SELECT FOR AY-3-8910    */
PIN 13 = ADC                    ;       /* CHIP SELECT FOR ADC          */
PIN 12 = APU                    ;       /* CHIP SELECT FOR AM9511       */
PIN 16 = SPARE1                 ;
PIN 17 = SPARE2                 ;
PIN 18 = SPARE3                 ;
PIN 19 = RST                    ;       /* SOFT RESET FOR AM9511        */
/* Intermediate equation                                               */
IoEnable = !M1                                                          ;
A_066Q = !A7 & !A6 & A5 & A4 & !A3 & A2 & A1                            ;
A_270Q = A7 & !A6 & A5 & A4 & A3 & !A2 & !A1                            ;
A_272Q = A7 & !A6 & A5 & A4 & A3 & !A2 & A1                             ;
A_274Q = A7 & !A6 & A5 & A4 & A3 & A2 & !A1                             ;
A_276Q = A7 & !A6 & A5 & A4 & A3 & A2 & A1                              ;
A_364Q = A7 & A6 & A5 & A4 & !A3 & A2 & !A1                             ;
A_366Q = A7 & A6 & A5 & A4 & !A3 & A2 & A1                              ;
/*************************************************************/
VDC = !(IoEnable & A_270Q)                                              ;
PSG = !(IoEnable & A_272Q & IORQ)                                       ;
RST = !(IoEnable & A_066Q)                                              ;
ADC = !(IoEnable & ((A_276Q & IO_SEL) # (A_366Q & !IO_SEL)))            ;
APU = !(IoEnable & ((A_274Q & IO_SEL) # (A_364Q & !IO_SEL)))            ;
SPARE1 = 'b'1                                                           ;
SPARE2 = 'b'1                                                           ;
SPARE3 = 'b'1                                                           ;
/*************************************************************/
```