* Display, GUI, and Multimedia options with the P2

(It’s way better now with P2!)

Wednesday, 24 June 2020

Raymond Allen, PhD EE (AKA Rayman in the forum)
Embedded Systems Architect
Rayslogic.com, LLC

*Outline

- New features with P2
- Native Digital Video output & HDMI. It’s new with P2!
- Analog VGA output & LUT
- Digital video output using TFP410 & ADV7513
- External VGA → HDMI adapters
- HyperRam and HyperFlash as image buffers
- LCD display module interfaces (the regular kind)
- HDMI, LVDS and EVE displays
- EVE2, EVE3, EVE4 GPUs
- Native GUI example
- Audio and mass storage for Multi-media use
- Movie player in 480p using eMMC

*P2 hardware features greatly expands options, compared to P1

- P2 hardware is quantum leap, not incremental update from P1
- New Propeller Hardware Features of interest for Display, GUI, and Multimedia, here are some:
  - Digital Video Output (AKA HDMI, DisplayPort, etc.)
  - DAC pin output modes that can output high quality VGA at 1080p
  - Lots of more I/O pins!   (64 vs. 32)
    - Can add HyperRam and eMMC each with 8-bit bus to add fast memory
    - Can use 4, 8, 16 or even 24 bit interface to LCD modules
  - Higher raw clock speed (300 MHz vs. 80..100 MHz)
  - Even more MIPS as most instructions only take 2 clocks (vs. 4 clocks for P1)
  - FIFO buffer to HUB RAM allows fast 2 clock writes or reads
  - Streamer allows simple output data stream with low overhead
  - Interrupts allow a single cog to juggle multiple tasks (important as still 8 cogs)
  - LUT RAM in COGs allow easy indexed color modes (512 longs can also run code!)
  - New instructions for Pixel Operations and CORDIC math operations
  - USB input enabled by Smart Pins (wireless combo can add KB + Mouse with 1 cog)

New Digital Video output mode allows easy use of modern video connections

- P2 can directly output digital video using a group of 8 I/O pins
  - Uses TDMS (Transition-minimized differential signaling) protocol just like HDMI, DVI, DisplayPort, PanelLink, PanelBus, etc.
  - Uses four pairs of differential signal wires & lots of digital magic

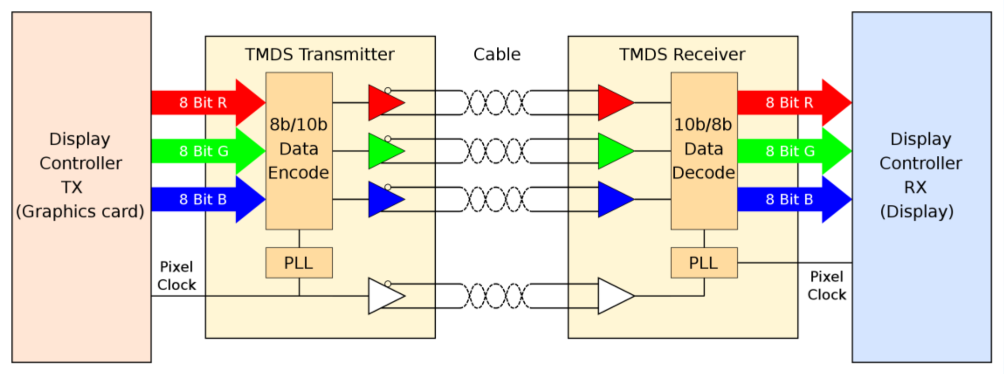


Image from Wikipedia

- Note that pixel clock output is limited to 1/10 of P2 clock, meaning that only low resolution or low refresh rate output is possible. Output as 10-bit codes.
  - Mainly for VGA resolution (25 MHz pixel clock), requires ~250 MHz P2 system clock
  - The output is always full 24-bit color, but can use the P2 LUT for indexed modes
  - 720p works for televisions that accept 24 Hz refresh rate (movie speed).
  - 1080p works at 11 Hz refresh rate, accepted by some DLP projectors

*Example output photos

- With HyperRam or HyperFlash memory (described later) can show high quality images over HDMI
- 16bpp can be upscaled by TVs and monitors

Images from Wikipedia

1080p @ 11Hz on DLP projector

800x480 @ 56Hz on monitor

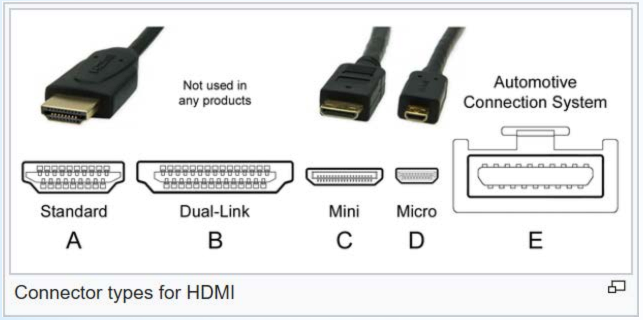
720p @ 24Hz on OLED TV

There are several connector options

(You can use an HDMI connector, just don’t label it as “HDMI”)

- **You can use one of many different kind of connectors interchangeably and use passive adapters to switch between them**

HDMI:



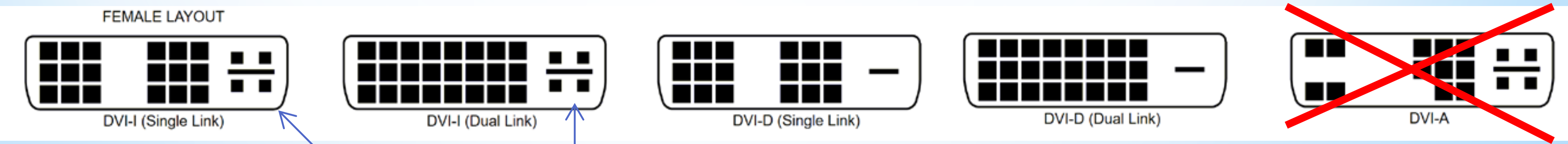
Connector types for HDMI

DisplayPort:

USB-C:

Images from Wikipedia

DVI:



Images from Wikipedia

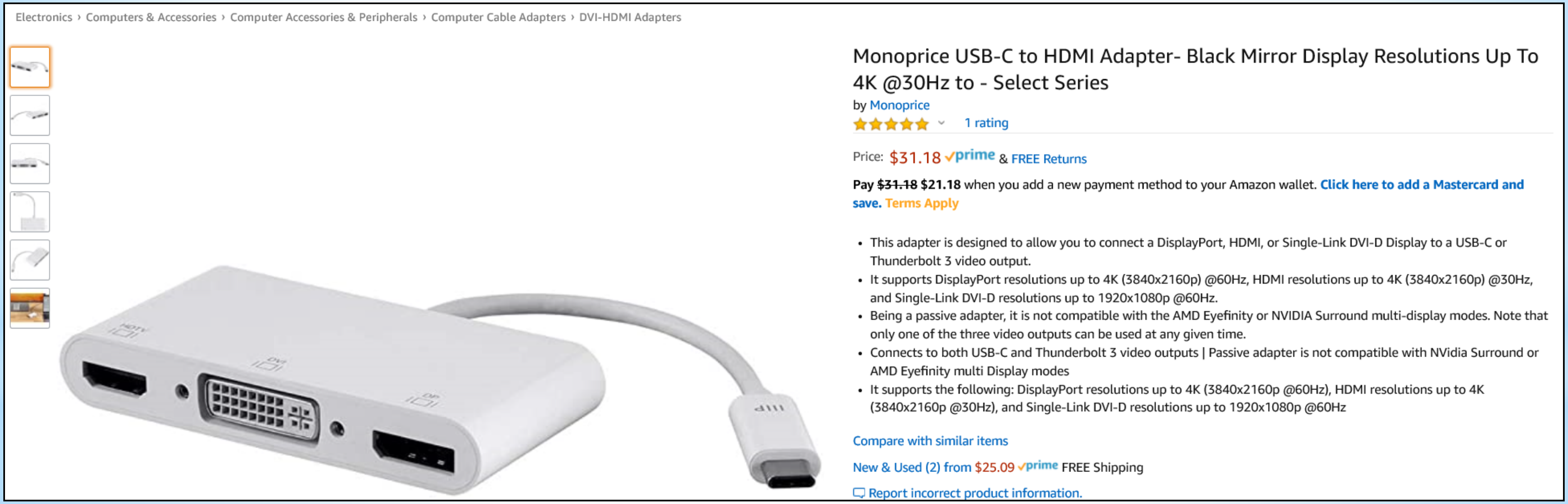
Analog VGA

Dual Link for SVGA?

- **Regular digital video only uses a subset of the 10-bit codes for video pixel data and sync signals, free to use!**
- **HDMI (High-Definition Multimedia Interface) uses the extra codes to add other stuff, like audio, to the output stream, need to pay $$$ to use!**
  - **Maybe we can test this out?**

*You can find “passive” adapters to switch between connectors

- But, there are some caveats…
- May not support dual-link
- May not support audio
- Sometimes 1-way adapters



Electronics › Computers & Accessories › Computer Accessories & Peripherals › Computer Cable Adapters › DVI-HDMI Adapters

Monoprice USB-C to HDMI Adapter- Black Mirror Display Resolutions Up To 4K @30Hz to - Select Series

by Monoprice

★★★★★ 1 rating

Price: $31.18 ✓prime & FREE Returns

Pay ~~$31.18~~ **$21.18** when you add a new payment method to your Amazon wallet. **Click here to add a Mastercard and save.** **Terms Apply**

- This adapter is designed to allow you to connect a DisplayPort, HDMI, or Single-Link DVI-D Display to a USB-C or Thunderbolt 3 video output.
- It supports DisplayPort resolutions up to 4K (3840x2160p) @60Hz, HDMI resolutions up to 4K (3840x2160p) @30Hz, and Single-Link DVI-D resolutions up to 1920x1080p @60Hz.
- Being a passive adapter, it is not compatible with the AMD Eyefinity or NVIDIA Surround multi-display modes. Note that only one of the three video outputs can be used at any given time.
- Connects to both USB-C and Thunderbolt 3 video outputs | Passive adapter is not compatible with NVidia Surround or AMD Eyefinity multi Display modes
- It supports the following: DisplayPort resolutions up to 4K (3840x2160p @60Hz), HDMI resolutions up to 4K (3840x2160p @30Hz), and Single-Link DVI-D resolutions up to 1920x1080p @60Hz

Compare with similar items

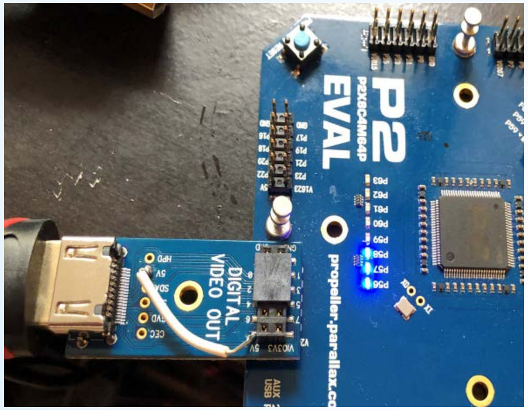
New & Used (2) from $25.09 ✓prime FREE Shipping

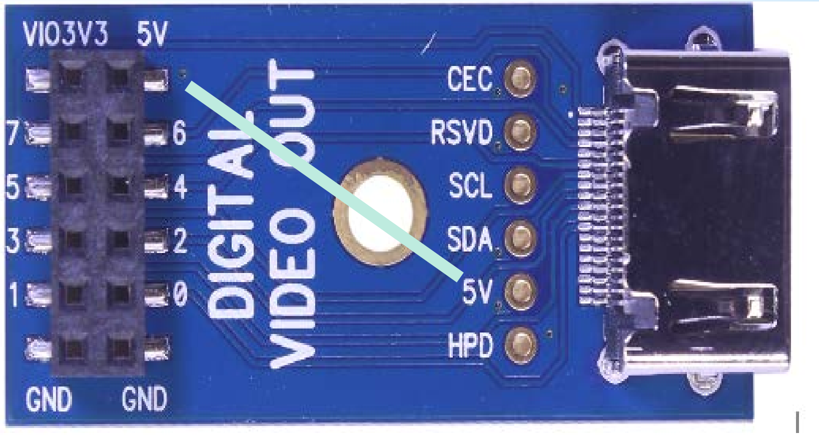
Report incorrect product information.

Image from Amazon

Parallax offers a Digital Video Output accessory for P2 Eval board

- The Parallax accessory makes it easy. Also provides access to other pins.
  - Some TVs seem to require the 5V. I added this jumper:

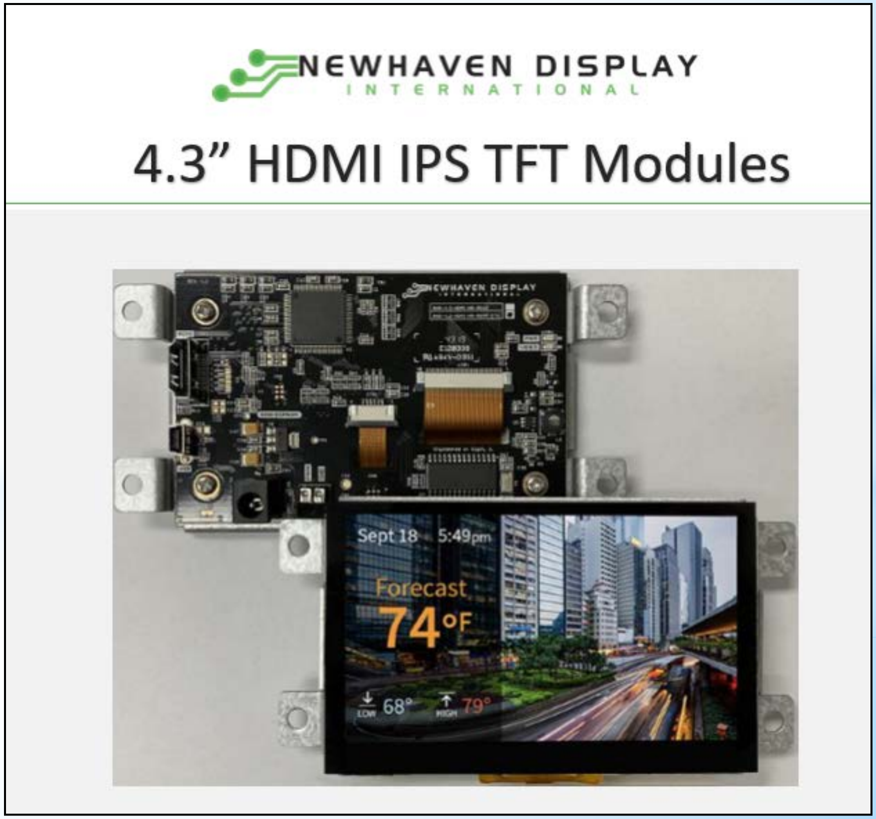




- We could use the DDC (Display Data Channel) pins (I2C with SCL & SDA) to communicate with monitor, but don’t need to.
  - Also used for High-bandwidth Digital Content Protection (HDCP) encryption.
  - Probably need pull-up on Hot Plug Detect (HPD) if using this.
  - Possible to get audio from SmartTVs using this pin (in some cases with RSVD pin).
- May be able to use CEC (Consumer Electronics Control) pin as a way to interact with a TVs remote control. It’s a 1-wire type bus.

*There are also LCD panels that accept HDMI input (Works? It should…)

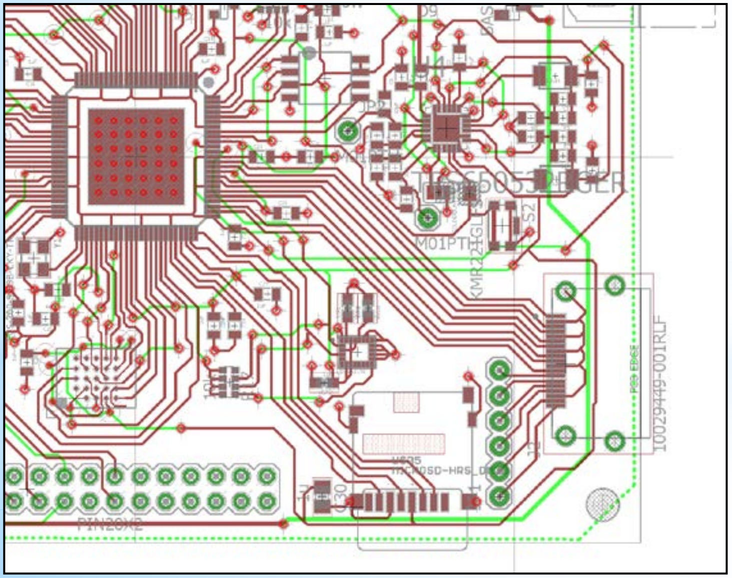
- Many of them have 800x480 resolution, which should be ideal.
- Some say LVDS/MIPI which may or may not work.
- Most appear to use Texas Instruments TFP401 chip to decode the digital video into 24-bit color pins (I’ll mention the companion encoder, TFP410, later…). No sound.
- Some have USB touch panels
- Available from several suppliers
- Several sizes from 4.3” to 7”

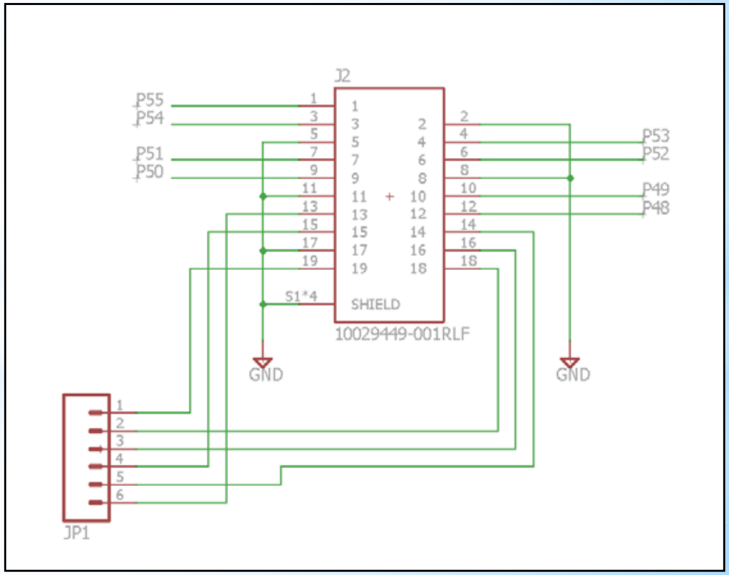


4.3” HDMI IPS TFT Modules

*It’s really easy to add HDMI connector to your own PCB design

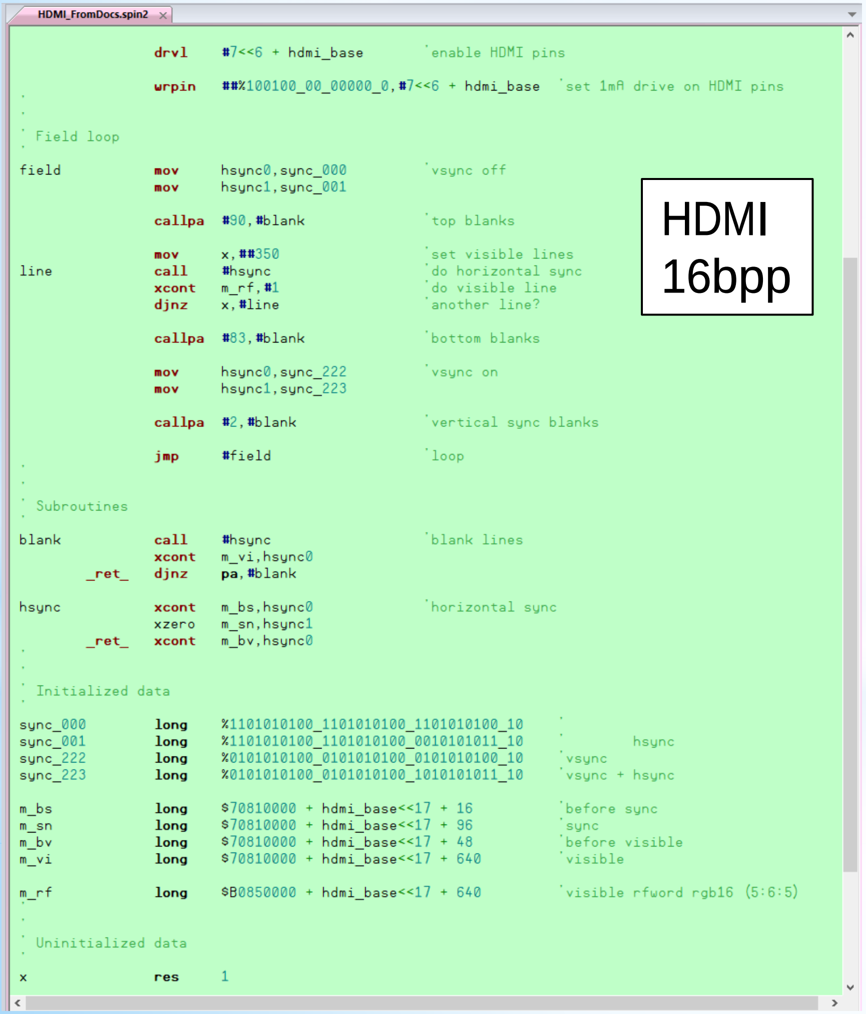
- This Eagle design has been proven. Design files posted in the P2 forum.
- Don’t see to need to worry too much about matching signal path length
- ***But, VGA only needs 5 pins and can go to higher resolution with P2 and with similar quality***





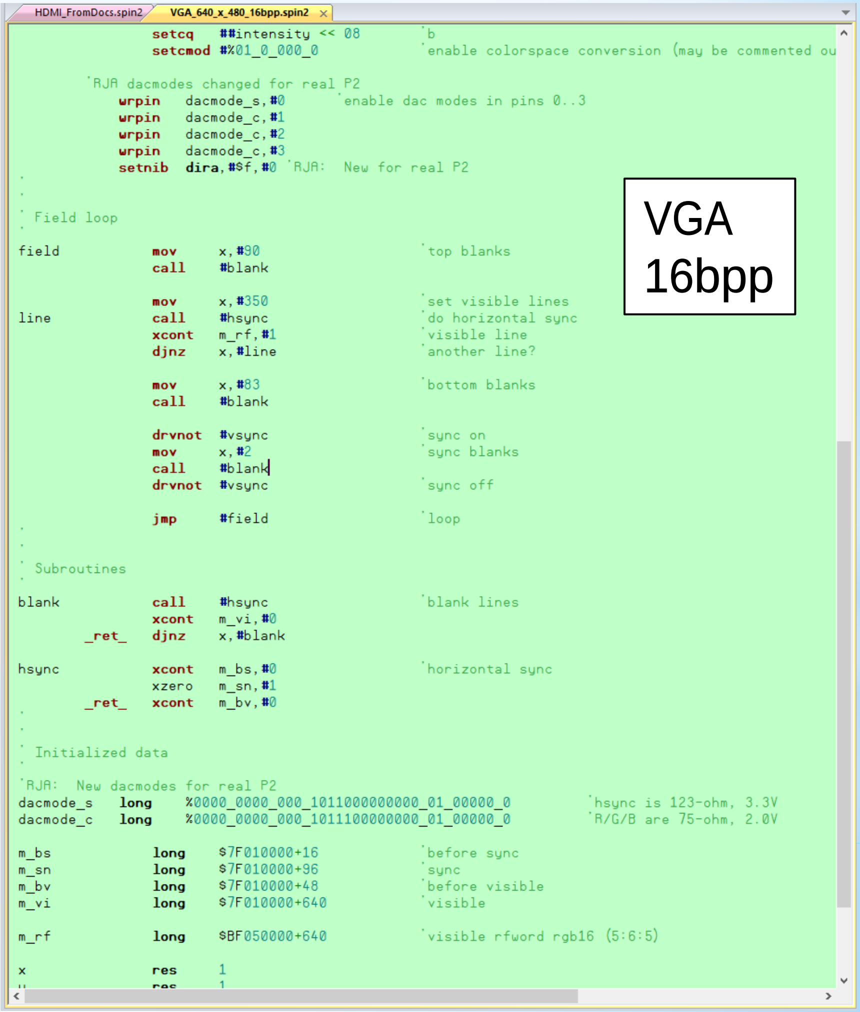
* Assembly code for HDMI is very similar to VGA

- **Main difference is the setup and the sync code…**



HDMI 16bpp

```
                drvl    #7<<6 + hdmi_base        'enable HDMI pins

                wrpin   ##%100100_00_00000_0,#7<<6 + hdmi_base  'set 1mA drive on HDMI pins
'
'
' Field loop
'
field           mov     hsync0,sync_000          'vsync off
                mov     hsync1,sync_001

                callpa  #90,#blank               'top blanks

                mov     x,##350                  'set visible lines
line            call    #hsync                   'do horizontal sync
                xcont   m_rf,#1                  'do visible line
                djnz    x,#line                  'another line?

                callpa  #83,#blank               'bottom blanks

                mov     hsync0,sync_222          'vsync on
                mov     hsync1,sync_223

                callpa  #2,#blank                'vertical sync blanks

                jmp     #field                   'loop
'
'
' Subroutines
'
blank           call    #hsync                   'blank lines
                xcont   m_vi,hsync0
        _ret_   djnz    pa,#blank

hsync           xcont   m_bs,hsync0              'horizontal sync
                xzero   m_sn,hsync1
        _ret_   xcont   m_bv,hsync0
'
'
' Initialized data
'
sync_000        long    %1101010100_1101010100_1101010100_10    '
sync_001        long    %1101010100_1101010100_0010101011_10    '        hsync
sync_222        long    %0101010100_0101010100_0101010100_10    'vsync
sync_223        long    %0101010100_0101010100_1010101011_10    'vsync + hsync

m_bs            long    $70810000 + hdmi_base<<17 + 16          'before sync
m_sn            long    $70810000 + hdmi_base<<17 + 96          'sync
m_bv            long    $70810000 + hdmi_base<<17 + 48          'before visible
m_vi            long    $70810000 + hdmi_base<<17 + 640         'visible

m_rf            long    $B0850000 + hdmi_base<<17 + 640         'visible rfword rgb16 (5:6:5)
'
'
' Uninitialized data
'
x               res     1
```

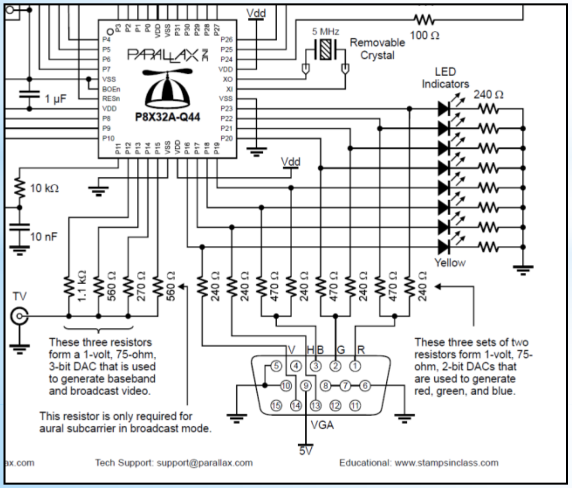


VGA 16bpp

```
                setcq   ##intensity << 08        'b
                setcmod #%01_0_000_0             'enable colorspace conversion (may be commented ou

        'RJA dacmodes changed for real P2
            wrpin   dacmode_s,#0        'enable dac modes in pins 0..3
            wrpin   dacmode_c,#1
            wrpin   dacmode_c,#2
            wrpin   dacmode_c,#3
            setnib  dira,#$f,#0 'RJA:  New for real P2
'
'
' Field loop
'
field           mov     x,#90                    'top blanks
                call    #blank

                mov     x,#350                   'set visible lines
line            call    #hsync                   'do horizontal sync
                xcont   m_rf,#1                  'visible line
                djnz    x,#line                  'another line?

                mov     x,#83                    'bottom blanks
                call    #blank

                drvnot  #vsync                   'sync on
                mov     x,#2                     'sync blanks
                call    #blank
                drvnot  #vsync                   'sync off

                jmp     #field                   'loop
'
'
' Subroutines
'
blank           call    #hsync                   'blank lines
                xcont   m_vi,#0
        _ret_   djnz    x,#blank

hsync           xcont   m_bs,#0                  'horizontal sync
                xzero   m_sn,#1
        _ret_   xcont   m_bv,#0
'
'
' Initialized data
'
'RJA:  New dacmodes for real P2
dacmode_s   long    %0000_0000_000_1011000000000_01_00000_0       'hsync is 123-ohm, 3.3V
dacmode_c   long    %0000_0000_000_1011100000000_01_00000_0       'R/G/B are 75-ohm, 2.0V

m_bs            long    $7F010000+16             'before sync
m_sn            long    $7F010000+96             'sync
m_bv            long    $7F010000+48             'before visible
m_vi            long    $7F010000+640            'visible

m_rf            long    $BF050000+640            'visible rfword rgb16 (5:6:5)

x               res     1
y               res     1
```

*VGA output is a whole lot better than it is with P1!

- Easy to do 1080p with a single cog
- Only needs 5 pins, thanks to DACs (Digital to Analog Converter) in every pin
- Simple to do 24-bit color and can use 512 longs of LUT (Look Up Table) built into every cog to convert 8-bit indexed color palette to 24-bit color
- There's enough HUB RAM to store a full frame of VGA at 8bpp
  - ~301 kB .bmp file size with 512 kB HUB RAM available
  - Much better than 32 kB of P1!

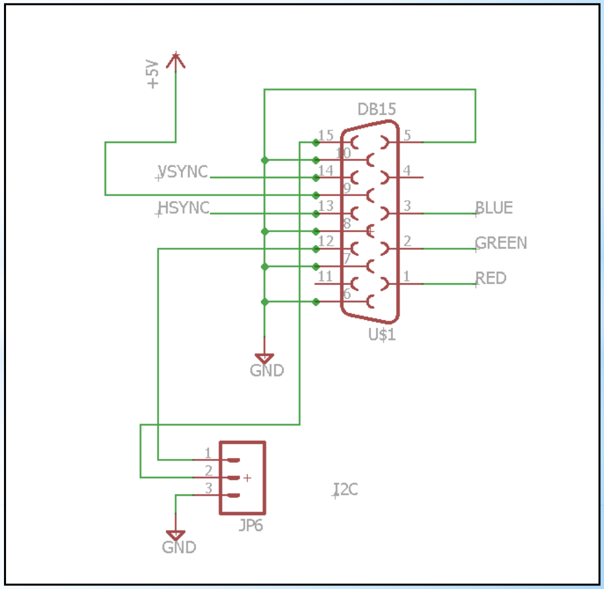


These three resistors form a 1-volt, 75-ohm, 3-bit DAC that is used to generate baseband and broadcast video.

This resistor is only required for aural subcarrier in broadcast mode.

These three sets of two resistors form 1-volt, 75-ohm, 2-bit DACs that are used to generate red, green, and blue.

x.com Tech Support: support@parallax.com Educational: www.stampsinclass.com

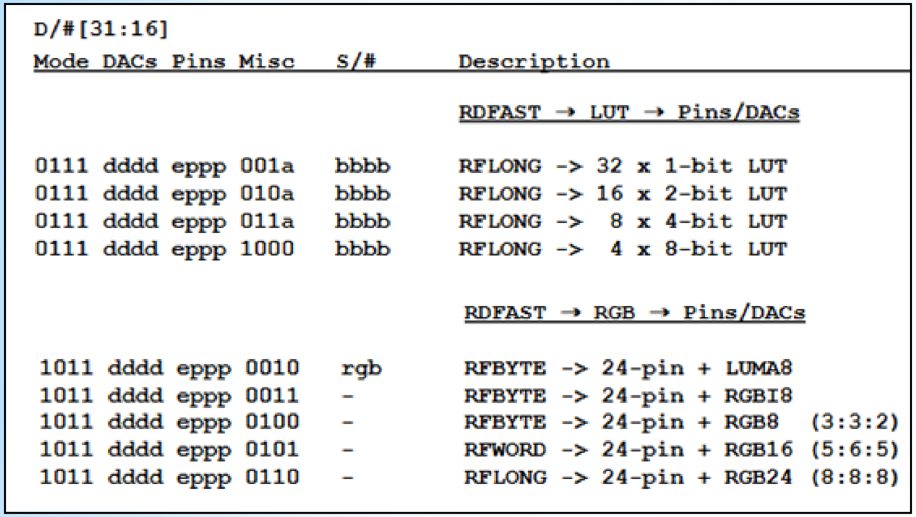


The usual 6-bit color with P1

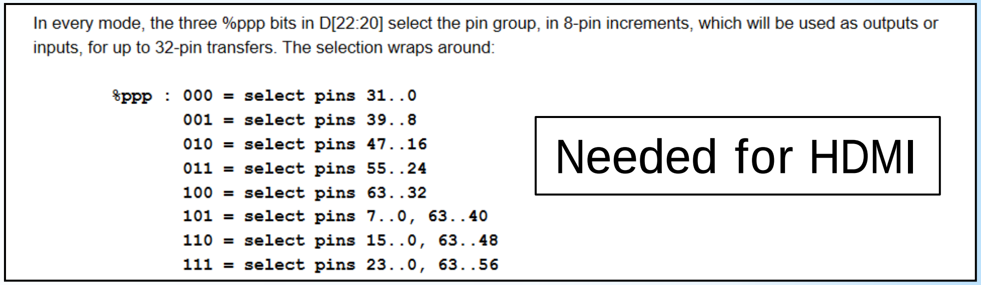
RGB signals connected directly to P2 pins

Video streamer modes and Pin Groups for VGA & HDMI

- **Here are streamer modes used for VGA and HDMI**



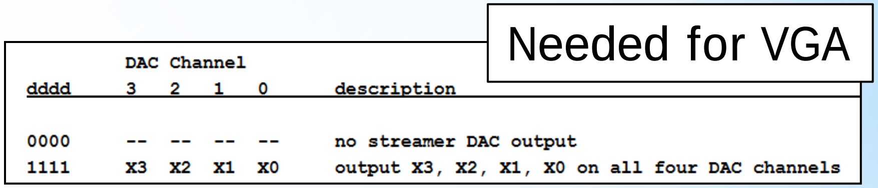
```
D/#[31:16]
Mode DACs Pins Misc   S/#       Description

                                RDFAST → LUT → Pins/DACs

0111 dddd eppp 001a   bbbb      RFLONG -> 32 x 1-bit LUT
0111 dddd eppp 010a   bbbb      RFLONG -> 16 x 2-bit LUT
0111 dddd eppp 011a   bbbb      RFLONG ->  8 x 4-bit LUT
0111 dddd eppp 1000   bbbb      RFLONG ->  4 x 8-bit LUT

                                RDFAST → RGB → Pins/DACs

1011 dddd eppp 0010   rgb       RFBYTE -> 24-pin + LUMA8
1011 dddd eppp 0011   -         RFBYTE -> 24-pin + RGBI8
1011 dddd eppp 0100   -         RFBYTE -> 24-pin + RGB8  (3:3:2)
1011 dddd eppp 0101   -         RFWORD -> 24-pin + RGB16 (5:6:5)
1011 dddd eppp 0110   -         RFLONG -> 24-pin + RGB24 (8:8:8)
```



In every mode, the three %ppp bits in D[22:20] select the pin group, in 8-pin increments, which will be used as outputs or inputs, for up to 32-pin transfers. The selection wraps around:

```
%ppp : 000 = select pins 31..0
       001 = select pins 39..8
       010 = select pins 47..16
       011 = select pins 55..24
       100 = select pins 63..32
       101 = select pins 7..0, 63..40
       110 = select pins 15..0, 63..48
       111 = select pins 23..0, 63..56
```

Needed for HDMI

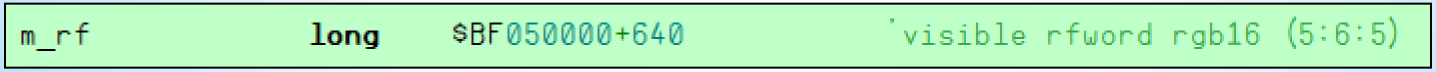


Needed for VGA

```
        DAC Channel
dddd    3   2   1   0      description

0000    --  --  --  --     no streamer DAC output
1111    X3  X2  X1  X0     output X3, X2, X1, X0 on all four DAC channels
```

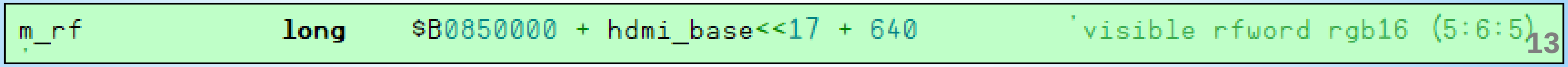
- **The %ppp bits define a group of 8 pins to be used**
  - **So, there are 8 options for where to put VGA/HDMI bus (convenient or Eval Board VGA & HDMI accessories with 8-pin interface).**

VGA, 16bpp, basepin=0, 8, 16, 24, etc.:



```
m_rf                long     $BF050000+640                'visible rfword rgb16 (5:6:5)
```

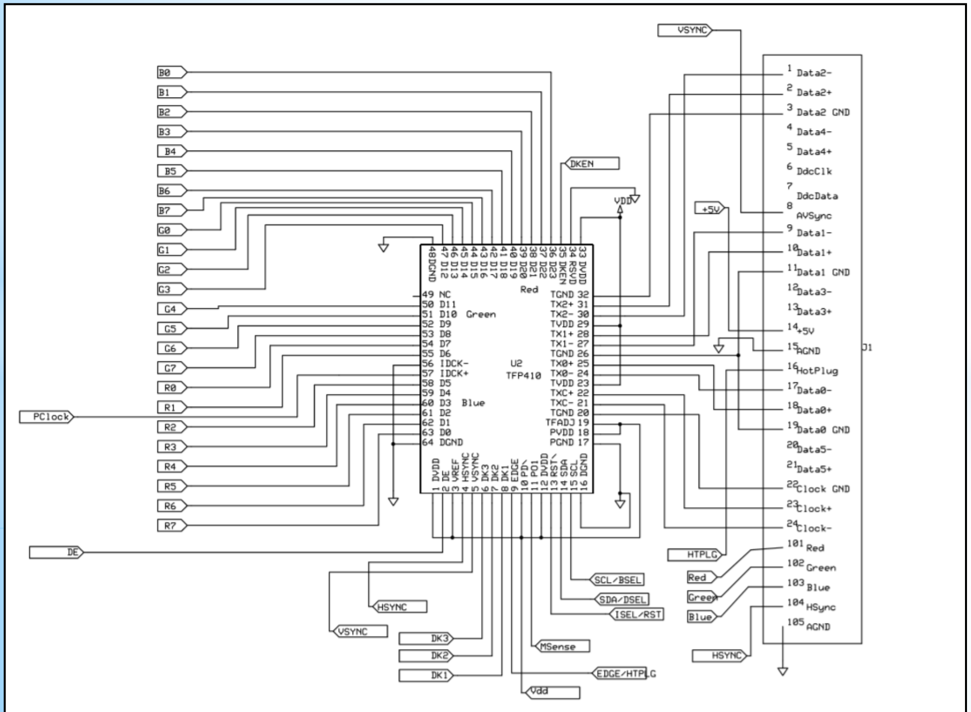
HDMI, 16bpp, basepin=0, 8, 16, 24, etc. (need to specify pins, not using DACS):

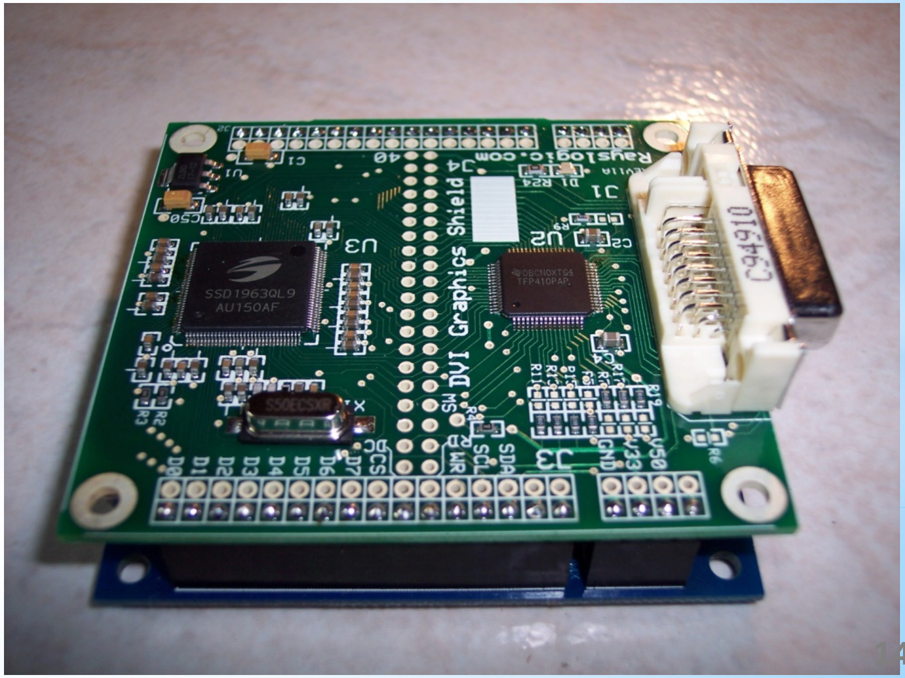


```
m_rf                long     $B0850000 + hdmi_base<<17 + 640        'visible rfword rgb16 (5:6:5)
```

*VGA → DVI using TFP410

- The Texas Instruments TFP410 (encoder) is widely used and has been around for a while.
  - No license or NDA (Non-Discloser Agreement) needed to use.
  - Often paired with TFP401 (decoder)
  - Video Only, no audio
- Successfully used along with SSD1963 (meant for LCD modules) with P1 for 24-bit, up to 800x480 resolution, output
  - Configured by either I2C or pull up/down resistors





*Would like to use ADV7513 to add audio

- The ADV7513 takes video input much like the TFP410, but adds in audio input for true HDMI output
    - SPDIF or I2S audio
    - Also ADV7511W for automobile

FUNCTIONAL BLOCK DIAGRAM

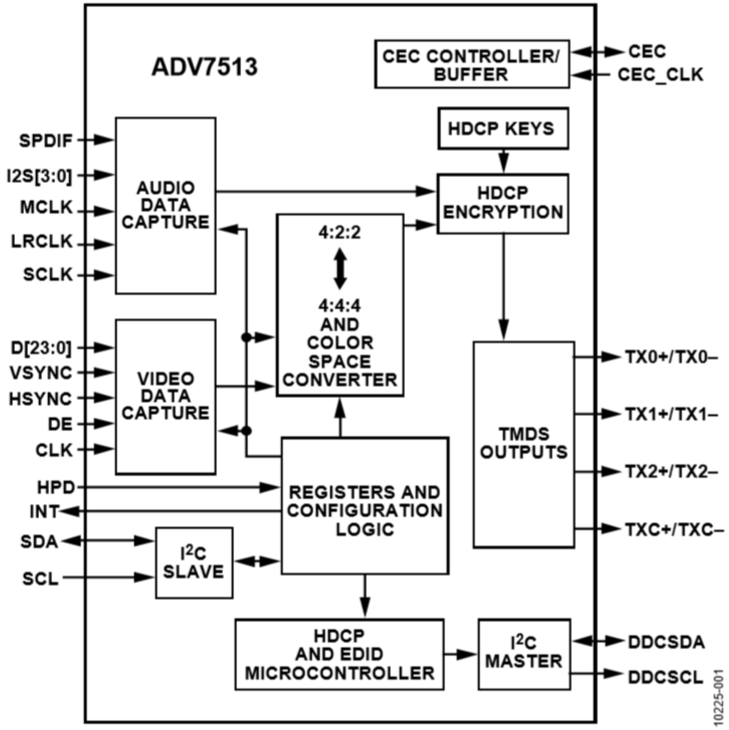
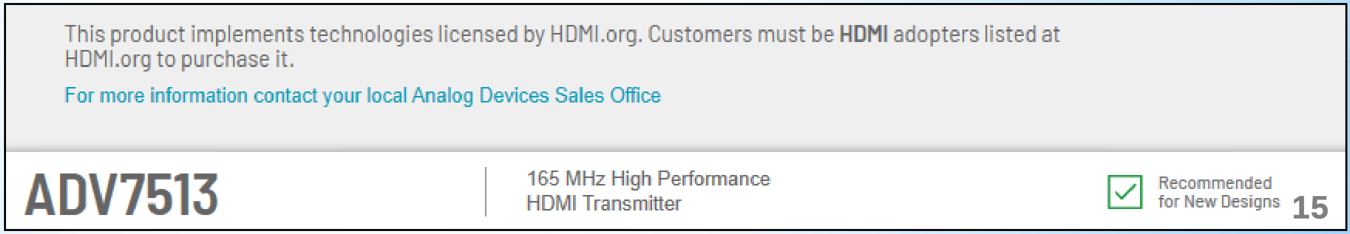


Figure 1.

- Here’s the problem:
    - Adopter=$$$$
    - Buy from eBay?



This product implements technologies licensed by HDMI.org. Customers must be **HDMI** adopters listed at HDMI.org to purchase it.

For more information contact your local Analog Devices Sales Office

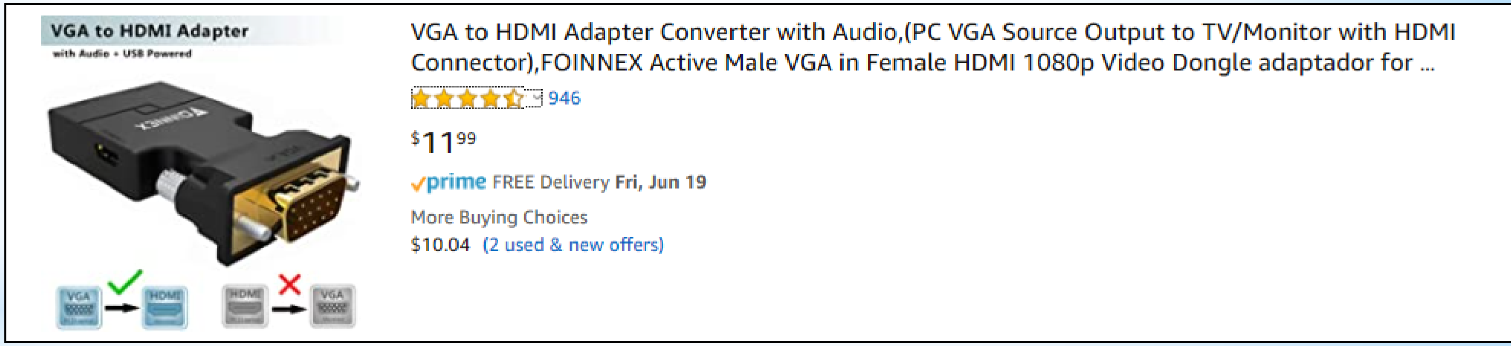
ADV7513

165 MHz High Performance
HDMI Transmitter

☑ Recommended for New Designs

*Easiest solution is just to use VGA → HDMI Adapter

- Low cost and readily available
- Stereo Audio input with regular 3.5mm jack
- USB powered
- Works at 1080p, 60 Hz
- A few different styles:



VGA to HDMI Adapter Converter with Audio,(PC VGA Source Output to TV/Monitor with HDMI Connector),FOINNEX Active Male VGA in Female HDMI 1080p Video Dongle adaptador for ...

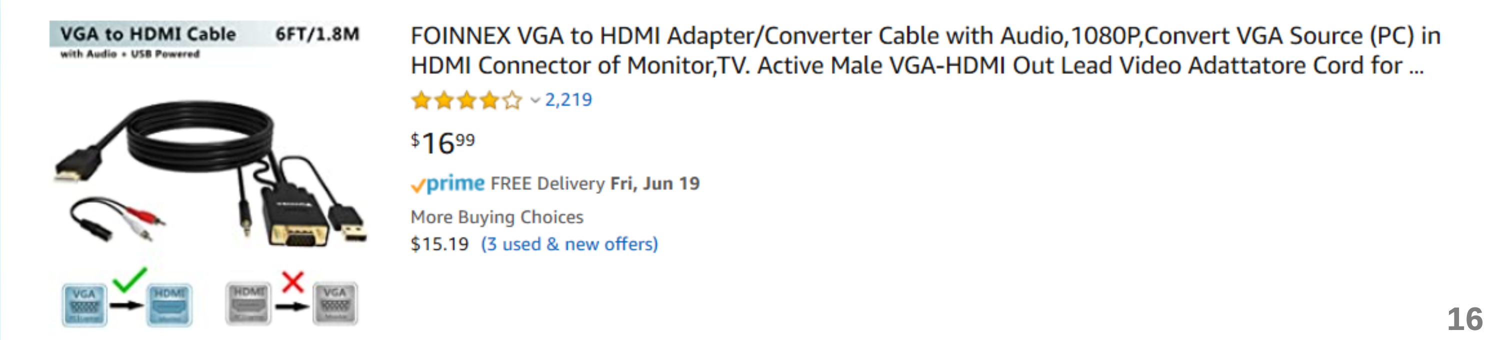
★★★★☆ 946

$11.99

✓prime FREE Delivery **Fri, Jun 19**

More Buying Choices

$10.04 (2 used & new offers)



FOINNEX VGA to HDMI Adapter/Converter Cable with Audio,1080P,Convert VGA Source (PC) in HDMI Connector of Monitor,TV. Active Male VGA-HDMI Out Lead Video Adattatore Cord for ...

★★★★☆ 2,219

$16.99

✓prime FREE Delivery **Fri, Jun 19**

More Buying Choices

$15.19 (3 used & new offers)

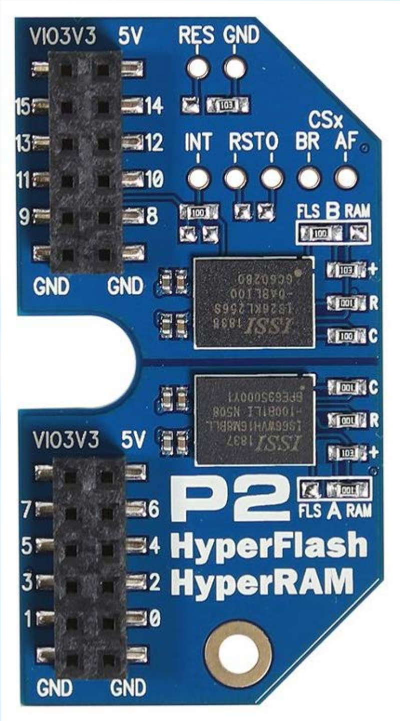
Images from Amazon

* HyperRam and HyperFlash can be used as image buffers, enabling high quality image display

- Both allow low latency reading of 8-bit data streams at clock/2 speed (150 MBPS reads with 300 MHz P2 clock).
- HyperFlash is harder to write to, but easier to read from…
  - Also, HyperFlash is static (keeps data when power off)
- Parallax sells a module with HyperRam and HyperFlash onboard.
- Code posted to forum that switches between two images, up to 1080p at 8bpp, loaded from uSD card.
  - Takes about a second to load with uSD (faster with eMMC!)
- Can store 7 HD images in HyperRam, 14 in HyperFlash
  - If I did the math right…

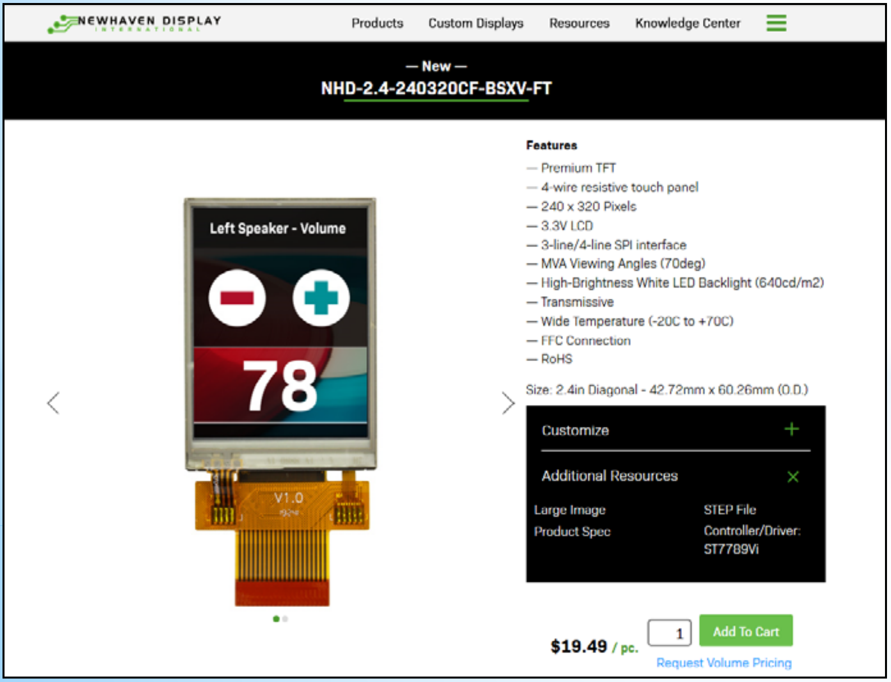
1080p VGA at 8bpp

@rogloh working on driver



* Working with regular display modules nicer due to more pins and faster speed

- Generally, there are two main types of regular display modules:
  - Small ones, 2.4” and below with integrated RAM
    - Often with 8-bit, 16-bit, or SPI interface. Often with resistive touch screen.
  - Large ones, 3.5” and above with no RAM
    - Usually 24-bit interface. Resistive or Capacitive touch screen.
    - Works in a similar way as VGA monitor except with 24-bit digital color pins
- Writing GUI code for these should be easy (and fun!)



— New —
NHD-2.4-240320CF-BSXV-FT

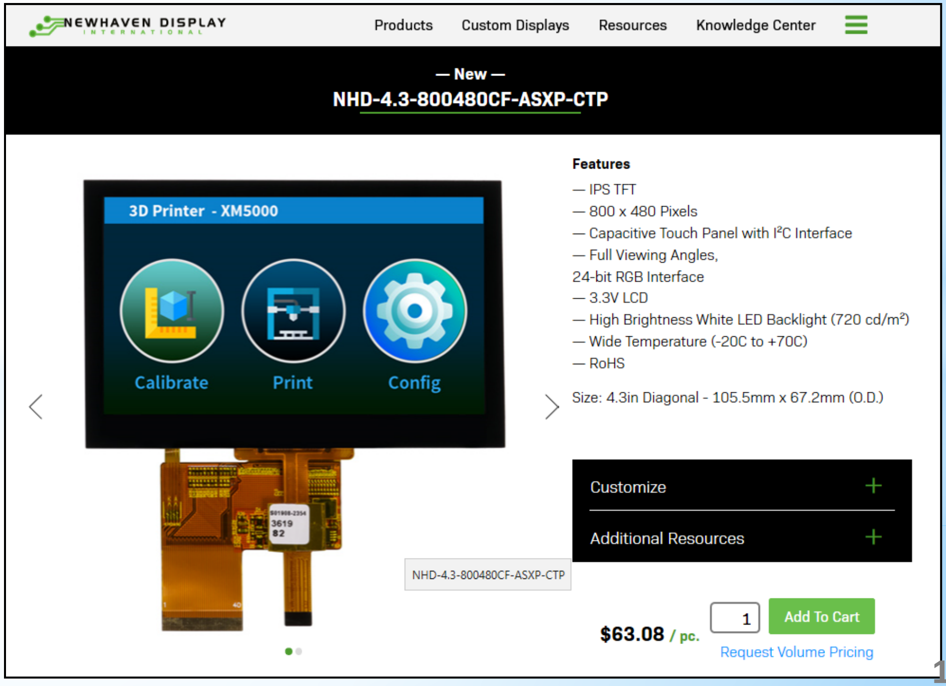
Features
- — Premium TFT
- — 4-wire resistive touch panel
- — 240 x 320 Pixels
- — 3.3V LCD
- — 3-line/4-line SPI interface
- — MVA Viewing Angles (70deg)
- — High-Brightness White LED Backlight (640cd/m2)
- — Transmissive
- — Wide Temperature (-20C to +70C)
- — FFC Connection
- — RoHS

Size: 2.4in Diagonal - 42.72mm x 60.26mm (O.D.)

Large Image
Product Spec
STEP File
Controller/Driver: ST7789Vi

$19.49 / pc. 1

Request Volume Pricing



— New —
NHD-4.3-800480CF-ASXP-CTP

Features
- — IPS TFT
- — 800 x 480 Pixels
- — Capacitive Touch Panel with I²C Interface
- — Full Viewing Angles,
24-bit RGB Interface
- — 3.3V LCD
- — High Brightness White LED Backlight (720 cd/m²)
- — Wide Temperature (-20C to +70C)
- — RoHS

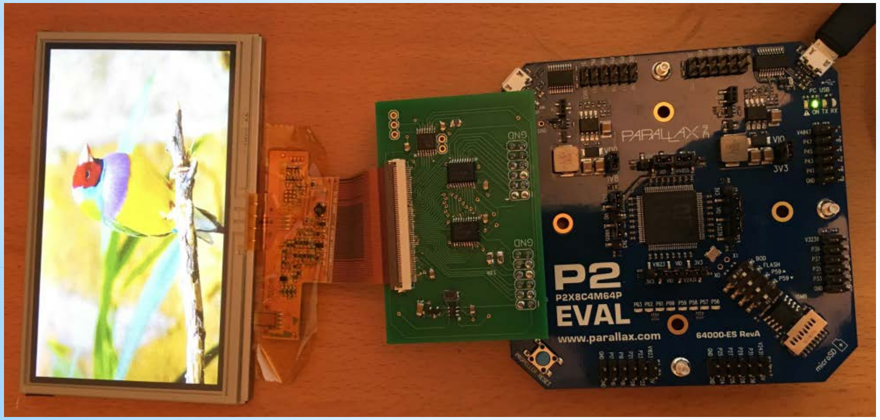
Size: 4.3in Diagonal - 105.5mm x 67.2mm (O.D.)

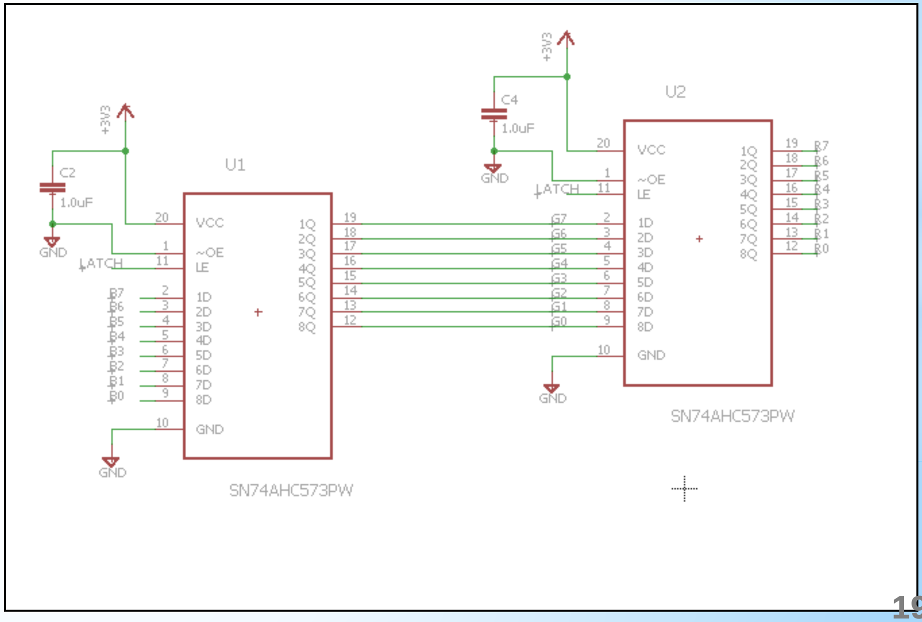
$63.08 / pc. 1

Request Volume Pricing

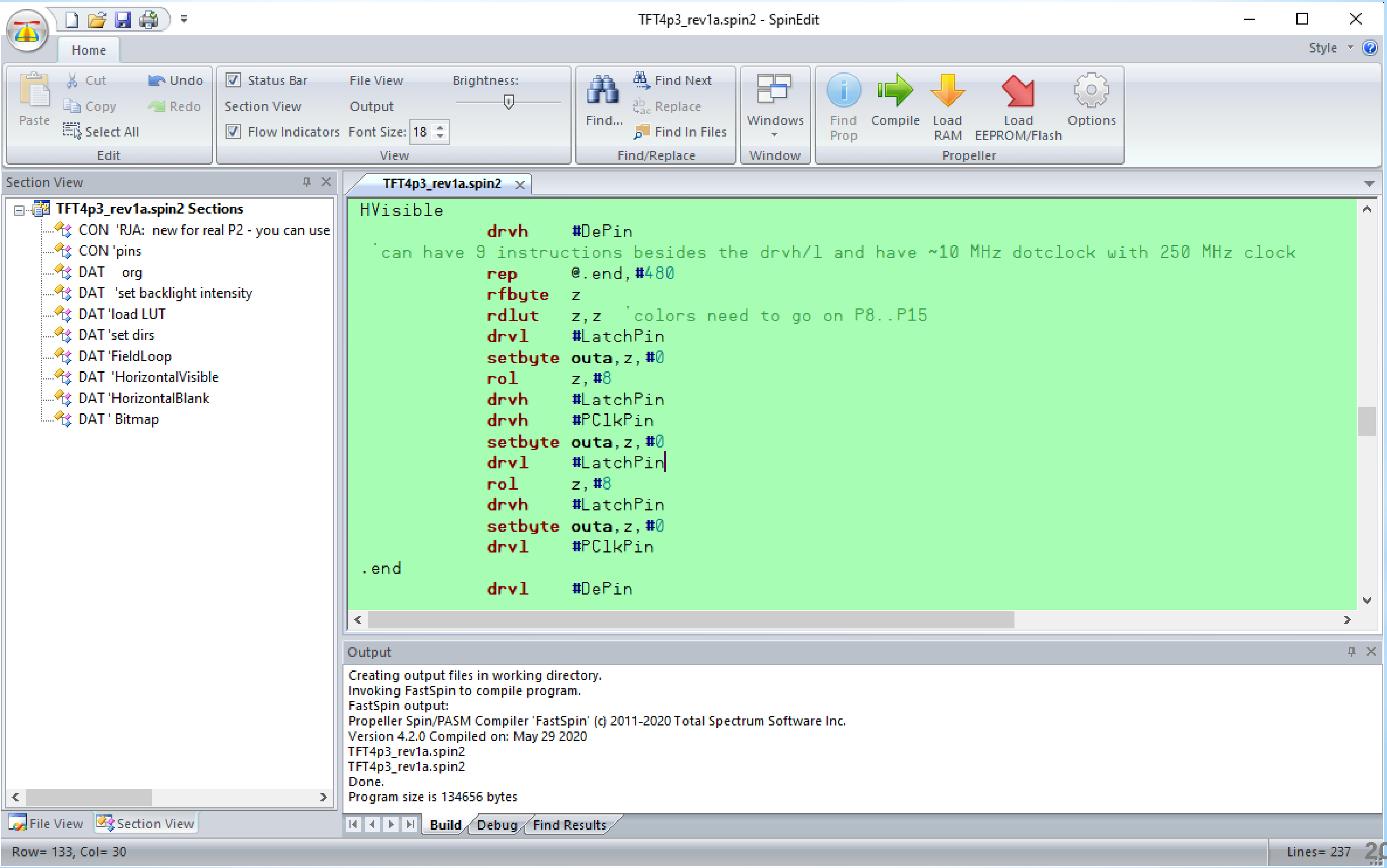
* P2 is fast enough to do 24-bit color with 8 pins and two extra chips

- 4.3” TFT LCD module with 24-bit interface can be driven by 8-pins for color using two octal latches, SN74LVC574A, in series
  - I was going to use 3 latches until @jmg pointed out could just use two!
  - Would have tried P2 ADCs for touchscreen if had more pins (I2C chip uses less pins)
- The 480x272 pixel screen only needs of pixel clock of 9 MHz, so have a lot of time with 300 MHz P2 to shift bytes into place
- More than enough room to double buffer 8bpp screen (~131 kB each)
- Could do 16bpp…





* Easy to get 10 Mhz dot clock for 4.3” TFT LCD with 8-pin to 24-pin circuit



TFT4p3_rev1a.spin2 Sections
- CON 'RJA: new for real P2 - you can use
- CON 'pins
- DAT org
- DAT 'set backlight intensity
- DAT 'load LUT
- DAT 'set dirs
- DAT 'FieldLoop
- DAT 'HorizontalVisible
- DAT 'HorizontalBlank
- DAT ' Bitmap

```
HVisible
            drvh    #DePin
 'can have 9 instructions besides the drvh/l and have ~10 MHz dotclock with 250 MHz clock
            rep     @.end,#480
            rfbyte  z
            rdlut   z,z   'colors need to go on P8..P15
            drvl    #LatchPin
            setbyte outa,z,#0
            rol     z,#8
            drvh    #LatchPin
            drvh    #PClkPin
            setbyte outa,z,#0
            drvl    #LatchPin
            rol     z,#8
            drvh    #LatchPin
            setbyte outa,z,#0
            drvl    #PClkPin
.end
            drvl    #DePin
```

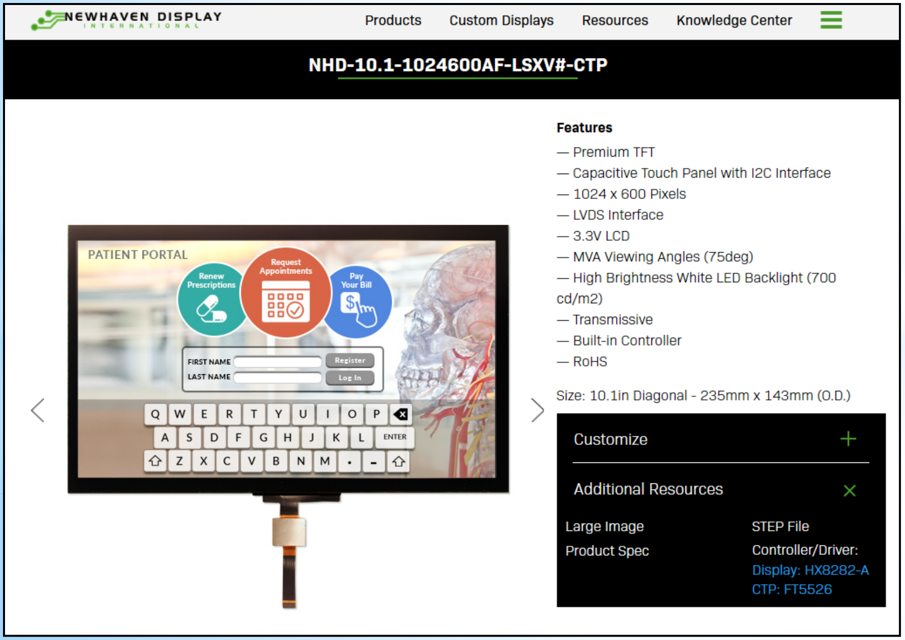
```
Creating output files in working directory.
Invoking FastSpin to compile program.
FastSpin output:
Propeller Spin/PASM Compiler 'FastSpin' (c) 2011-2020 Total Spectrum Software Inc.
Version 4.2.0 Compiled on: May 29 2020
TFT4p3_rev1a.spin2
TFT4p3_rev1a.spin2
Done.
Program size is 134656 bytes
```

Row= 133, Col= 30

Lines= 237

Newer types of interfaces for large display modules include HDMI, LVDS, and FTDI EVE

- LVDS (low-voltage differential signaling) seems to be mostly for the 7” and larger displays. Similar but not same as TDMS used by HDMI.
  - Not clear if can be used by P2 directly, but there are TTL to LVDS chips.
- HDMI is also mainly on the bigger screens. WVGA resolution is good fit for P2.
- EVE2 based displays are like a SPI/QPI based GPU with 1 MB onboard RAM
  - Some EVE3 displays now available, EVE4 displays coming…



NHD-10.1-1024600AF-LSXV#-CTP

Features

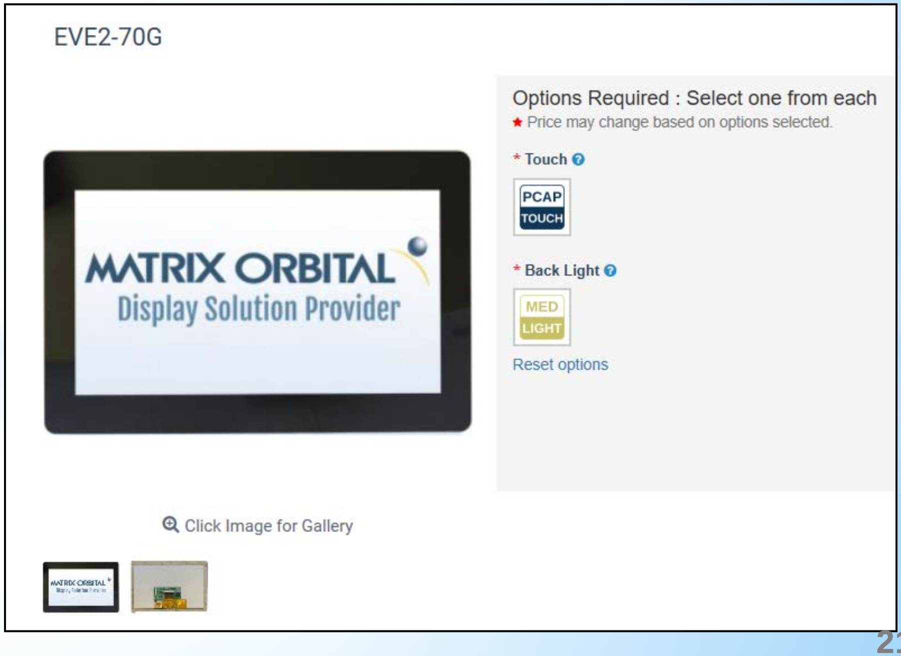
— Premium TFT
— Capacitive Touch Panel with I2C Interface
— 1024 x 600 Pixels
— LVDS Interface
— 3.3V LCD
— MVA Viewing Angles (75deg)
— High Brightness White LED Backlight (700 cd/m2)
— Transmissive
— Built-in Controller
— RoHS

Size: 10.1in Diagonal - 235mm x 143mm (O.D.)

Customize +

Additional Resources ×

Large Image	STEP File
Product Spec	Controller/Driver: Display: HX8282-A CTP: FT5526



EVE2-70G

Options Required : Select one from each

* Price may change based on options selected.

* Touch

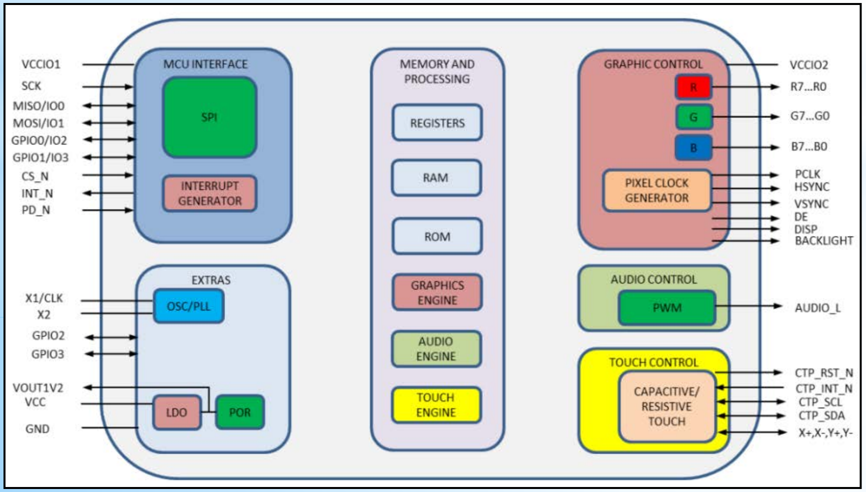
* Back Light

Reset options

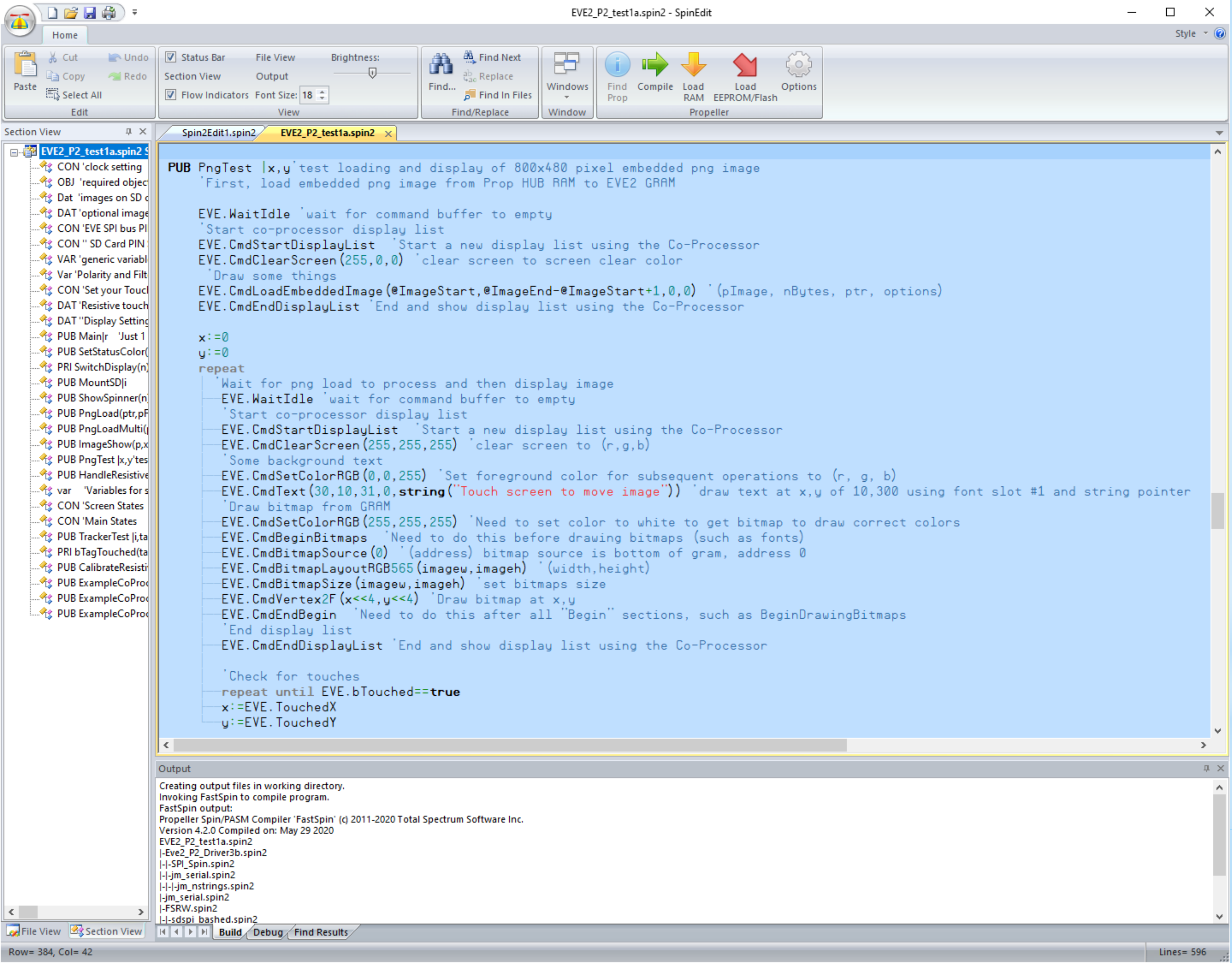
Click Image for Gallery

The FTDI EVE display modules offload display and GUI operations, works pretty good with P1 already

- Spin1 SPI code for EVE2 posted to forum some time ago
- With P2, now makes sense to switch to SQI mode (4-bit interface) for faster updating speed. Also, even SPI is much faster with P2.
    - Spin1 code already working with slight modifications on P2 thanks to FastSpin! (Thanks Eric S.!)
- EVE2 displays make GUI work easier
    - Easy to implement buttons, dials, rotary knobs, sliders, etc.
    - Interfaces with touchscreen to tell you what buttons were pushed for you, slider and rotary knob positions…
    - There is audio processor, but it’s mono and not as good as can be done with P2
    - Display list code is similar to OpenGL



* Currently adapting P1 driver code to P2 using FastSpin



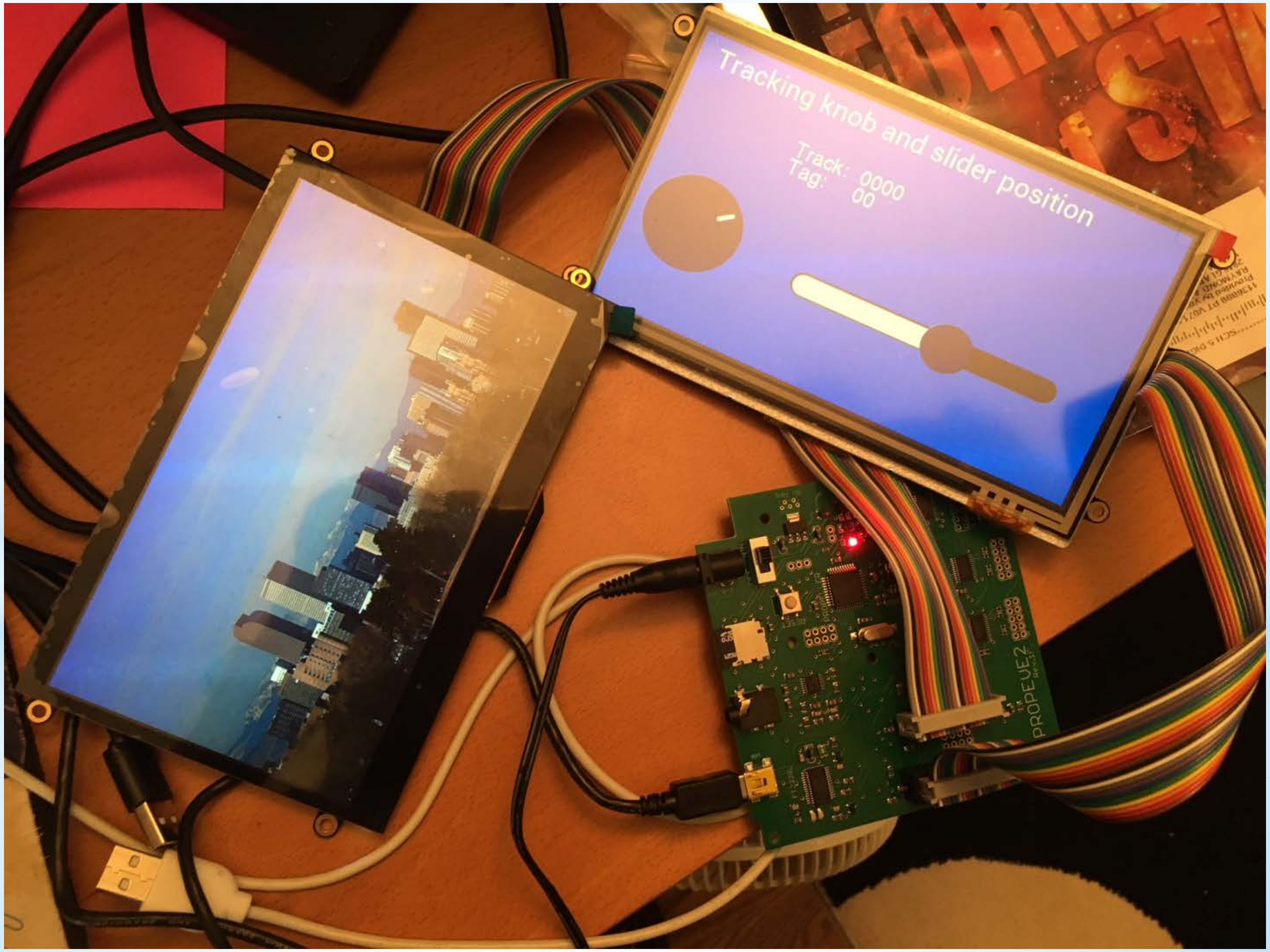
```
EVE2_P2_test1a.spin2 S
  CON 'clock setting
  OBJ  'required objec
  Dat  'images on SD c
  DAT 'optional image
  CON 'EVE SPI bus PI
  CON '' SD Card PIN :
  VAR 'generic variabl
  Var 'Polarity and Filt
  CON 'Set your Toucl
  DAT 'Resistive touch
  DAT ''Display Setting
  PUB Main|r    'Just 1
  PUB SetStatusColor(
  PRI SwitchDisplay(n)
  PUB MountSD|i
  PUB ShowSpinner(n)
  PUB PngLoad(ptr,pF
  PUB PngLoadMulti(
  PUB ImageShow(p,x
  PUB PngTest |x,y'tes
  PUB HandleResistive
  var     'Variables for s
  CON 'Screen States
  CON 'Main States
  PUB TrackerTest |i,ta
  PRI bTagTouched(ta
  PUB CalibrateResisti
  PUB ExampleCoProc
  PUB ExampleCoProc
  PUB ExampleCoProc
```

```
PUB PngTest |x,y'test loading and display of 800x480 pixel embedded png image
    'First, load embedded png image from Prop HUB RAM to EVE2 GRAM

    EVE.WaitIdle 'wait for command buffer to empty
    'Start co-processor display list
    EVE.CmdStartDisplayList  'Start a new display list using the Co-Processor
    EVE.CmdClearScreen(255,0,0) 'clear screen to screen clear color
     'Draw some things
    EVE.CmdLoadEmbeddedImage(@ImageStart,@ImageEnd-@ImageStart+1,0,0) '(pImage, nBytes, ptr, options)
    EVE.CmdEndDisplayList 'End and show display list using the Co-Processor

    x:=0
    y:=0
    repeat
      'Wait for png load to process and then display image
       EVE.WaitIdle 'wait for command buffer to empty
       'Start co-processor display list
       EVE.CmdStartDisplayList  'Start a new display list using the Co-Processor
       EVE.CmdClearScreen(255,255,255) 'clear screen to (r,g,b)
       'Some background text
       EVE.CmdSetColorRGB(0,0,255) 'Set foreground color for subsequent operations to (r, g, b)
       EVE.CmdText(30,10,31,0,string("Touch screen to move image")) 'draw text at x,y of 10,300 using font slot #1 and string pointer
       'Draw bitmap from GRAM
       EVE.CmdSetColorRGB(255,255,255) 'Need to set color to white to get bitmap to draw correct colors
       EVE.CmdBeginBitmaps  'Need to do this before drawing bitmaps (such as fonts)
       EVE.CmdBitmapSource(0) '(address) bitmap source is bottom of gram, address 0
       EVE.CmdBitmapLayoutRGB565(imagew,imageh) '(width,height)
       EVE.CmdBitmapSize(imagew,imageh) 'set bitmaps size
       EVE.CmdVertex2F(x<<4,y<<4) 'Draw bitmap at x,y
       EVE.CmdEndBegin  'Need to do this after all "Begin" sections, such as BeginDrawingBitmaps
       'End display list
       EVE.CmdEndDisplayList 'End and show display list using the Co-Processor

       'Check for touches
       repeat until EVE.bTouched==true
       x:=EVE.TouchedX
       y:=EVE.TouchedY
```

```
Creating output files in working directory.
Invoking FastSpin to compile program.
FastSpin output:
Propeller Spin/PASM Compiler 'FastSpin' (c) 2011-2020 Total Spectrum Software Inc.
Version 4.2.0 Compiled on: May 29 2020
EVE2_P2_test1a.spin2
|-Eve2_P2_Driver3b.spin2
|-|-SPI_Spin.spin2
|-|-jm_serial.spin2
|-|-|-jm_nstrings.spin2
|-jm_serial.spin2
|-FSRW.spin2
|-|-sdspi_bashed.spin2
```

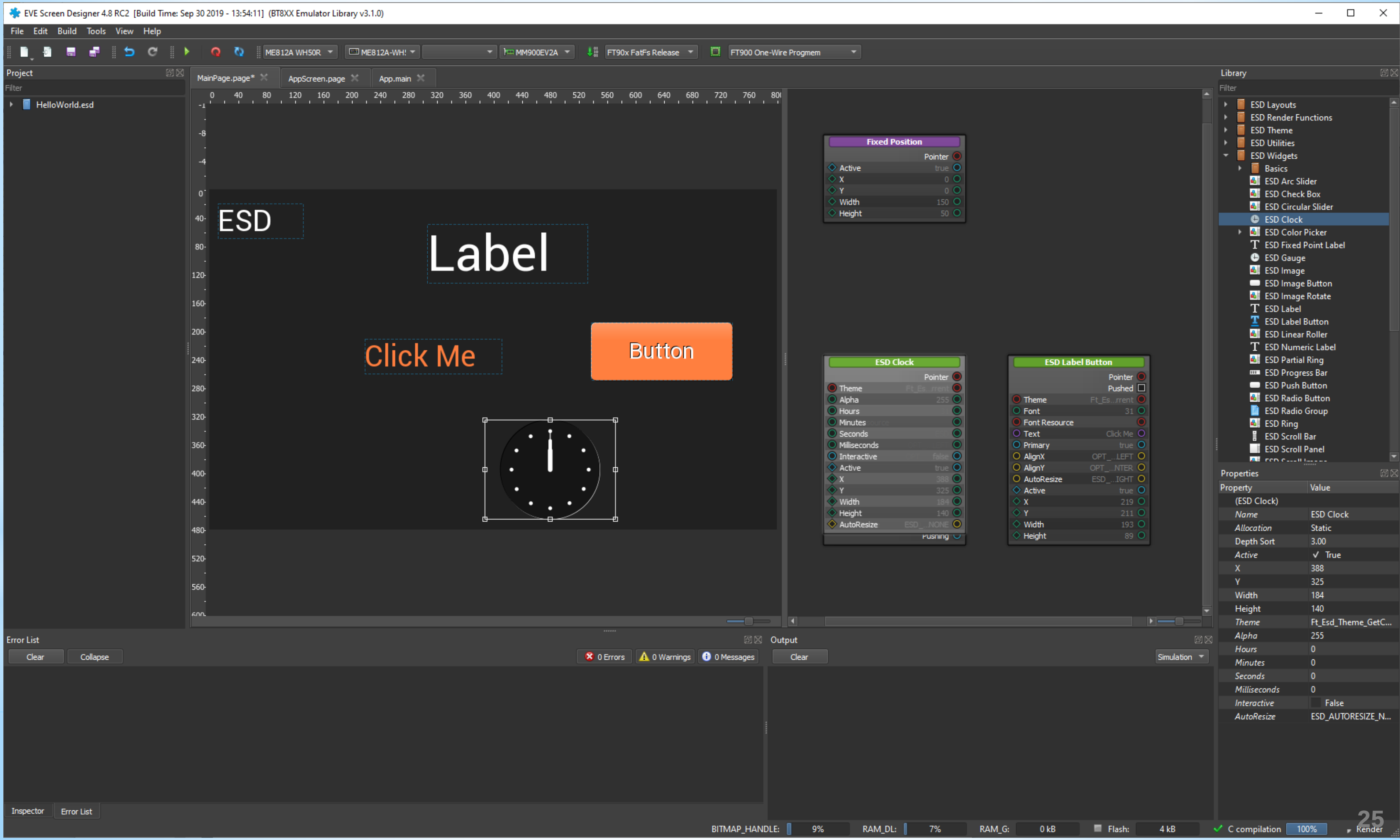
* EVE2 also includes a compressed image decoder

- Works with jpg and png file formats and some others too.
- Can play mpeg movies with audio
  - Not yet demonstrated with Propeller…



* There’s a “Screen Designer” program for EVE

- It’s strange and slow, but maybe useful for laying out your GUI



EVE Screen Designer 4.8 RC2 [Build Time: Sep 30 2019 - 13:54:11] (BT8XX Emulator Library v3.1.0)

Fixed Position: Active true, X 0, Y 0, Width 150, Height 50

ESD Clock: Alpha 255, Interactive false, Active true, X 388, Y 325, Width 184, Height 140, AutoResize ESD_...NONE

ESD Label Button: Theme Ft_Es...rrent, Font 31, Text Click Me, Primary true, AlignX OPT_...LEFT, AlignY OPT_...NTER, AutoResize ESD_...IGHT, Active true, X 219, Y 211, Width 193, Height 89

Properties

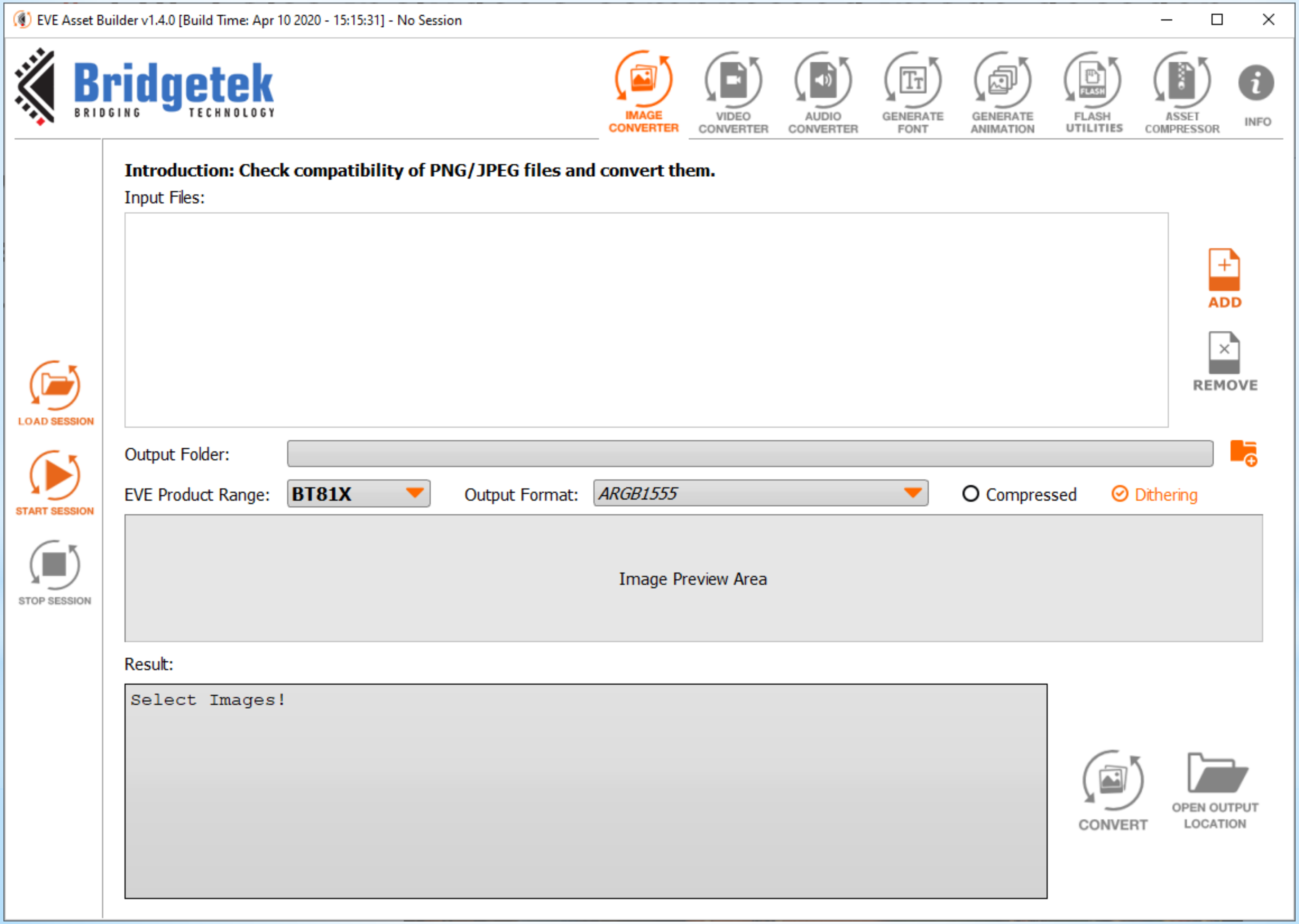
Property	Value
(ESD Clock)	
Name	ESD Clock
Allocation	Static
Depth Sort	3.00
Active	✓ True
X	388
Y	325
Width	184
Height	140
Theme	Ft_Esd_Theme_GetC...
Alpha	255
Hours	0
Minutes	0
Seconds	0
Milliseconds	0
Interactive	False
AutoResize	ESD_AUTORESIZE_N...

0 Errors 0 Warnings 0 Messages

BITMAP_HANDLE: 9% RAM_DL: 7% RAM_G: 0 kB Flash: 4 kB C compilation 100% Render

* There's a new "Asset Manager" program for EVE

- Maybe I'll finally be able to get video to work…



EVE Asset Builder v1.4.0 [Build Time: Apr 10 2020 - 15:15:31] - No Session

Introduction: Check compatibility of PNG/JPEG files and convert them.

Input Files:

Output Folder:

EVE Product Range: BT81X

Output Format: ARGB1555

○ Compressed

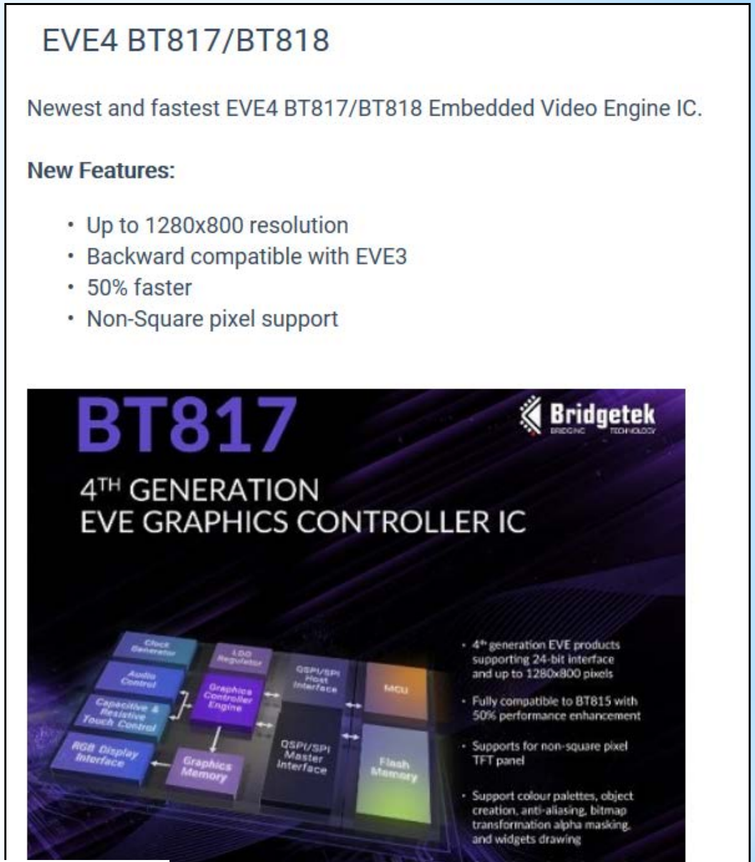
✓ Dithering

Result:

```
Select Images!
```

* New versions of EVE up and coming

- **EVE3 is out and adds SQI flash chip interface**
  - **Also boosts main SQI interface to 30 MHz**
- **EVE4 boosts speed to enable 1280x800 resolution, “coming soon”**
  - **Main SQI interface to 40 MHz**
  - **Think was going to be announced at Embedded World 2020 in Nuremberg, Germany**



EVE4 BT817/BT818

Newest and fastest EVE4 BT817/BT818 Embedded Video Engine IC.

New Features:

- Up to 1280x800 resolution
- Backward compatible with EVE3
- 50% faster
- Non-Square pixel support

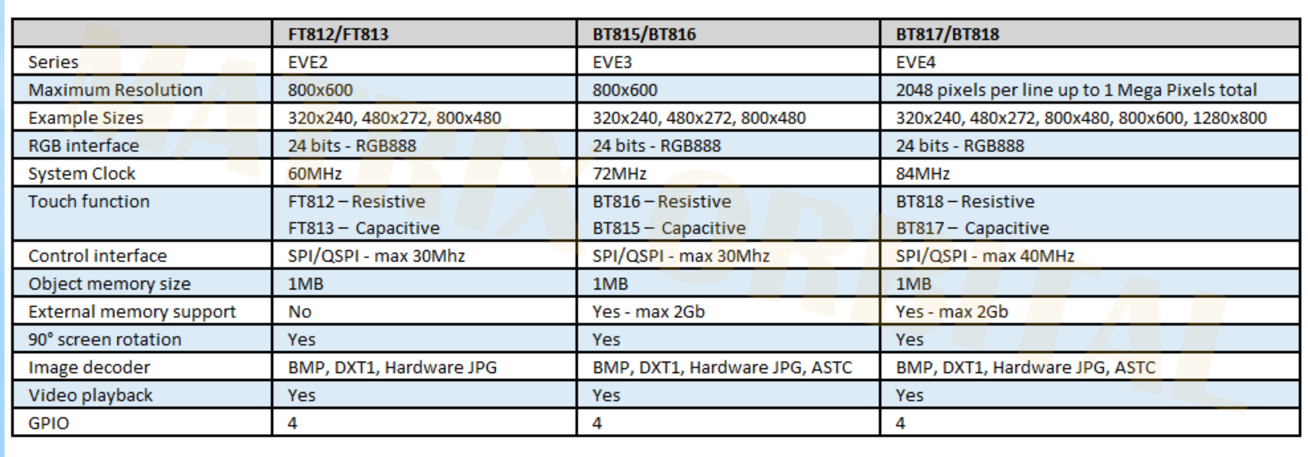
BT817

Bridgetek

4TH GENERATION
EVE GRAPHICS CONTROLLER IC

- 4th generation EVE products supporting 24-bit interface and up to 1280x800 pixels
- Fully compatible to BT815 with 50% performance enhancement
- Supports for non-square pixel TFT panel
- Support colour palettes, object creation, anti-aliasing, bitmap transformation alpha masking, and widgets drawing
- Capacitive and resistive touch options available

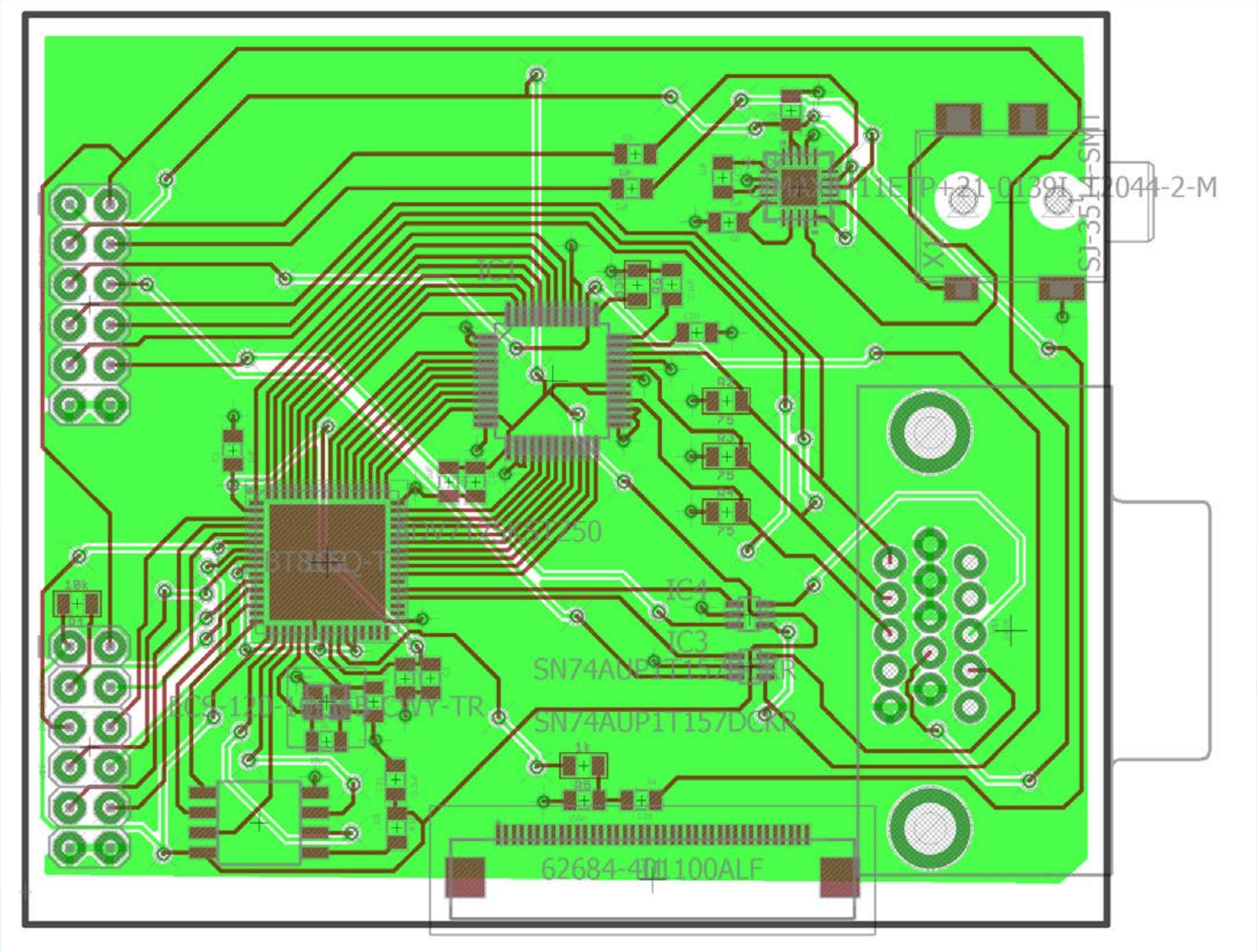
Clock Generator
Audio Control
Capacitive & Resistive Touch Control
RGB Display Interface
LDO Regulator
Graphics Controller Engine
Graphics Memory
QSPI/SPI Host Interface
QSPI/SPI Master Interface
MCU
Flash Memory



	FT812/FT813	BT815/BT816	BT817/BT818
Series	EVE2	EVE3	EVE4
Maximum Resolution	800x600	800x600	2048 pixels per line up to 1 Mega Pixels total
Example Sizes	320x240, 480x272, 800x480	320x240, 480x272, 800x480	320x240, 480x272, 800x480, 800x600, 1280x800
RGB interface	24 bits - RGB888	24 bits - RGB888	24 bits - RGB888
System Clock	60MHz	72MHz	84MHz
Touch function	FT812 – Resistive FT813 – Capacitive	BT816 – Resistive BT815 – Capacitive	BT818 – Resistive BT817 – Capacitive
Control interface	SPI/QSPI - max 30Mhz	SPI/QSPI - max 30Mhz	SPI/QSPI - max 40MHz
Object memory size	1MB	1MB	1MB
External memory support	No	Yes - max 2Gb	Yes - max 2Gb
90° screen rotation	Yes	Yes	Yes
Image decoder	BMP, DXT1, Hardware JPG	BMP, DXT1, Hardware JPG, ASTC	BMP, DXT1, Hardware JPG, ASTC
Video playback	Yes	Yes	Yes
GPIO	4	4	4

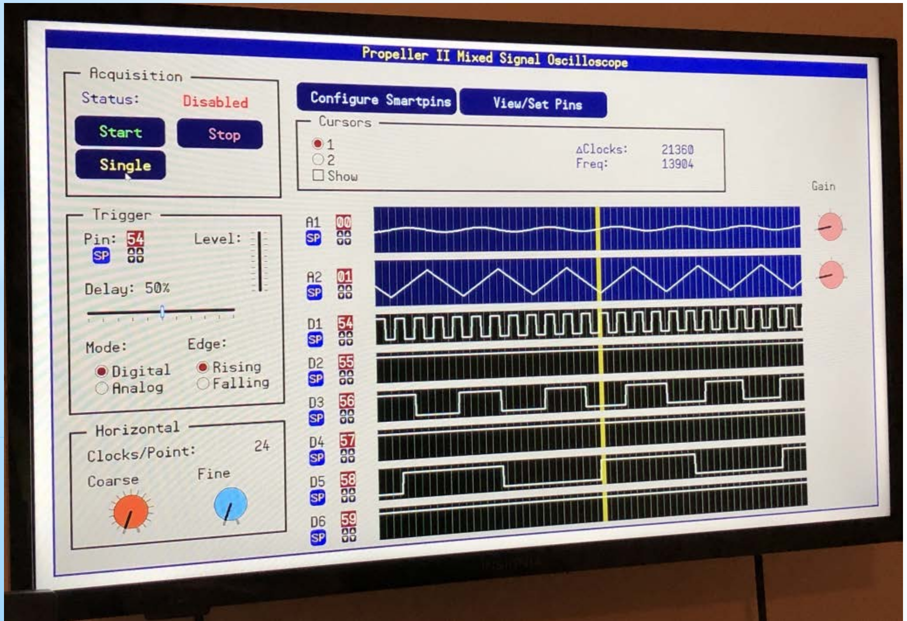
Use ADV7125 (triple DAC) to output EVE over VGA? Or, perhaps TFP410 to output over HDMI?

- Designed a board to try the ADV7125 option..



Native P2 GUI Example: Tiled 1080p @ 2bpp w/Graphics

- Standard resolutions for PC monitors and such are 480p, 720p and 1080p; but resolution is king, so targeted 1080p
- Not enough onboard memory for super nice graphics at 1080p, so using the P1 style of 16x16 pixel tiles for graphics and font
- Uses the P1 graphics.spin, adapted for P2, doesn't need a cog for simple graphics
- "Mixed Signal Oscilloscope" example demonstrates key features
- Uses USB mouse (Thanks @garryj!)
- Each tile can pick from 256, 24-bit colors for both foreground and background



Propeller II Mixed Signal Oscilloscope

Status: Disabled

ΔClocks: 21360
Freq: 13904

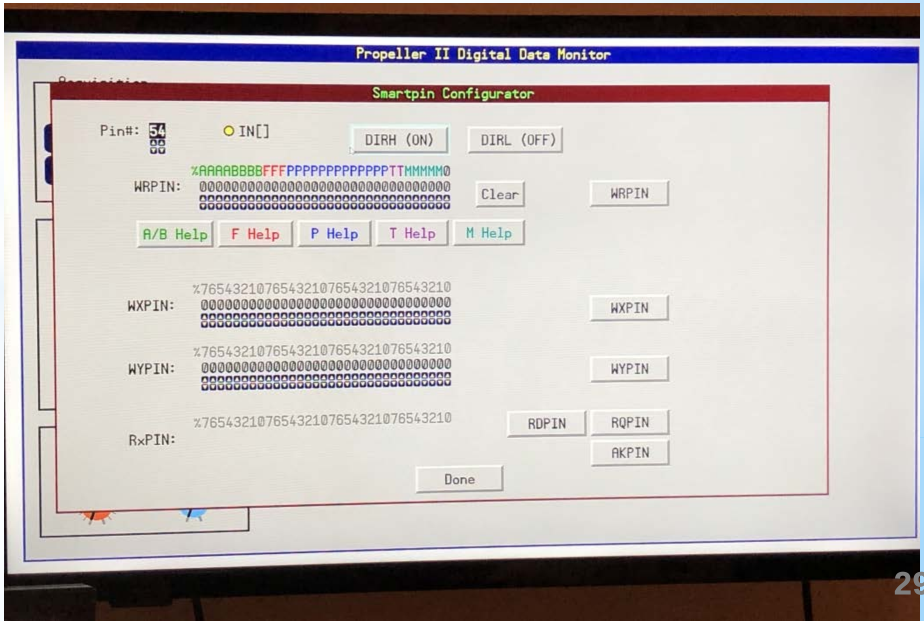
Pin: 54

Delay: 50%

Mode: ● Digital ○ Analog

Edge: ● Rising ○ Falling

Clocks/Point: 24



Propeller II Digital Data Monitor

Smartpin Configurator

Pin#: 54

WRPIN: %AAAABBBBFFFPPPPPPPPPPPPPTTMMMMM0
00000000000000000000000000000000

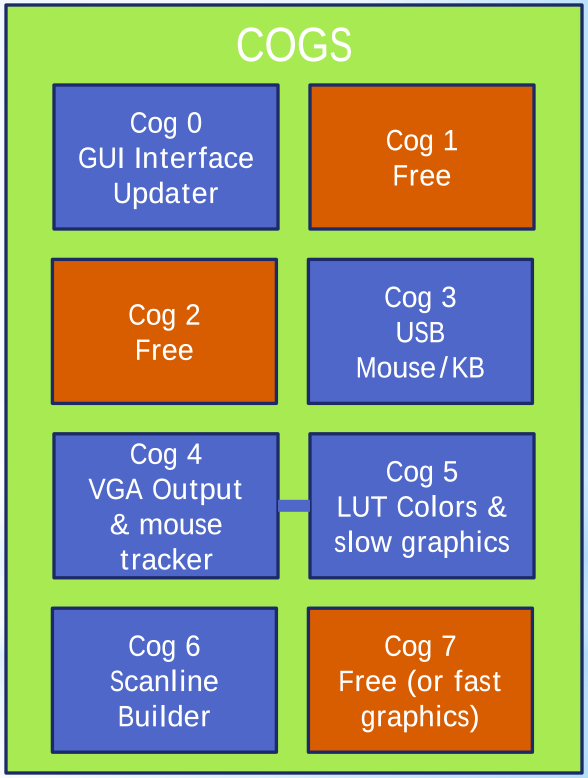
WXPIN: %76543210765432107654321076543210
00000000000000000000000000000000

WYPIN: %76543210765432107654321076543210
00000000000000000000000000000000

RxPIN: %76543210765432107654321076543210

Tiled 2-bit GUI uses about half of P2 resources, leaving the other half for the application

- Uses 5 of 8 cogs and ~200 of 512 kB HUB Ram
- Cog #0 is main Spin2 cog and responds to mouse clicks
- Cog #6 reads from tile array to build 16 rows of scanline for each row of tiles
- Cog #4 outputs the scanlines while mixing in the cursor pixels
  - Adding in the cursor was not easy!
- Cog #5 updates the shared LUT for tile colors and also does simple graphics
  - Good thing we have LUT sharing!
  - Cog #4 just cycles through 8 sets of 4 colors
  - Cog #5 updates these colors tile by tile in the row of tiles
- Cog #3 runs @garryj's USB mouse/kb code
- Cogs #1, 2, and 7 are free, but one is sometimes needed if form has complex graphics

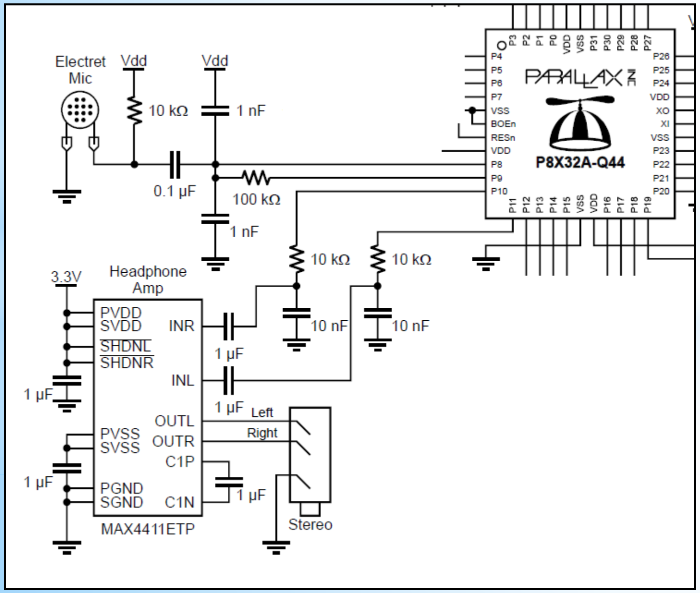


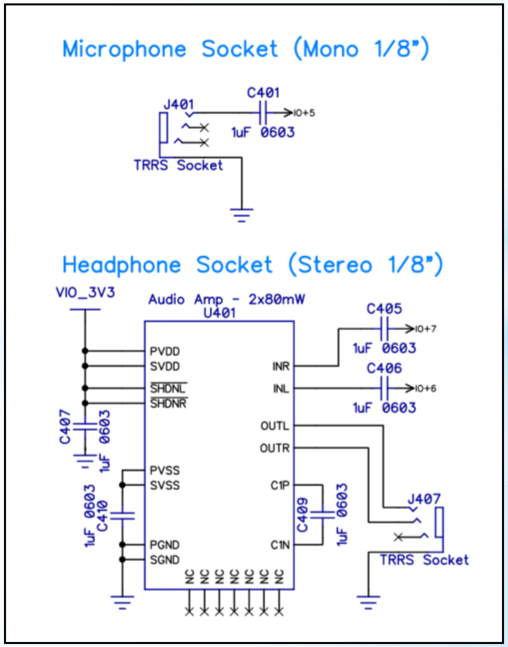
* Multimedia: Just add in audio and mass storage, and perhaps a camera and mic (Needs Work!)

- Audio is much better on P2 because of the DACs in every pin.
  - Seems that we’re just getting started with this…
- There are a few uSD driver options in the forum
  - FSRW (File System Read/Write) was made working by @cheesuz
  - Read speed improved to 2.4 MB/s
  - Still a lot more to do…
- An eMMC (embedded Multi-Media Controller) driver was posted to forum
  - Read speed at 28 MB/s opens up more options…
  - FSRW for eMMC (read only) code posted in forum
- Camera code for FPGA version of P2 posted some time ago
  - Needs work still to look for best options…
- Don’t think I’ve seen much microphone code yet. Except a demo by Chip.

* Audio is simpler and better

- P1 audio output was actually a digital signal that needed to be heavily filtered to pass just the audio frequencies you want
- For P2, just need a DC blocking capacitor. Smartpins do almost all the work for you.
- P1 microphone input needed two pins and external components to do sigma-delta ADC.
- For P2, you just need a DC blocking capacitor and can use ADC in every pin to

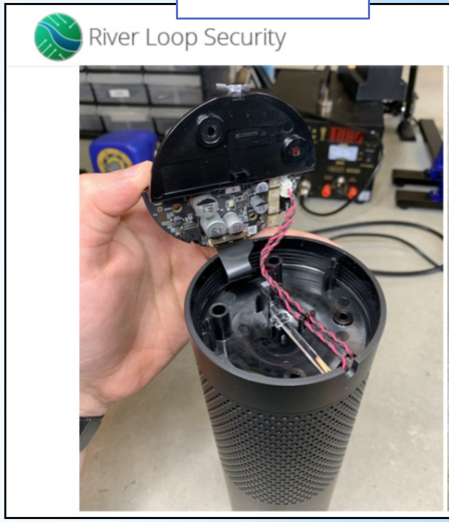


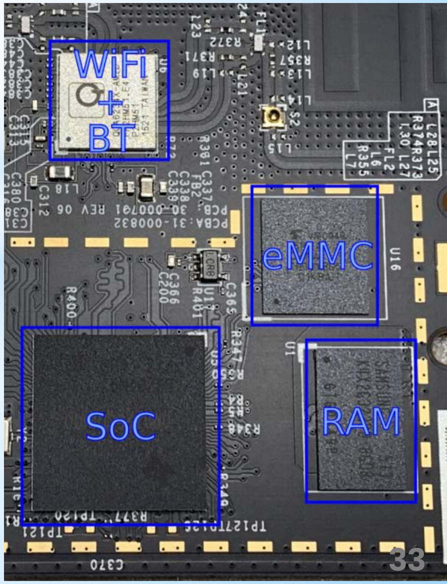


eMMC opens to door to better and faster multimedia

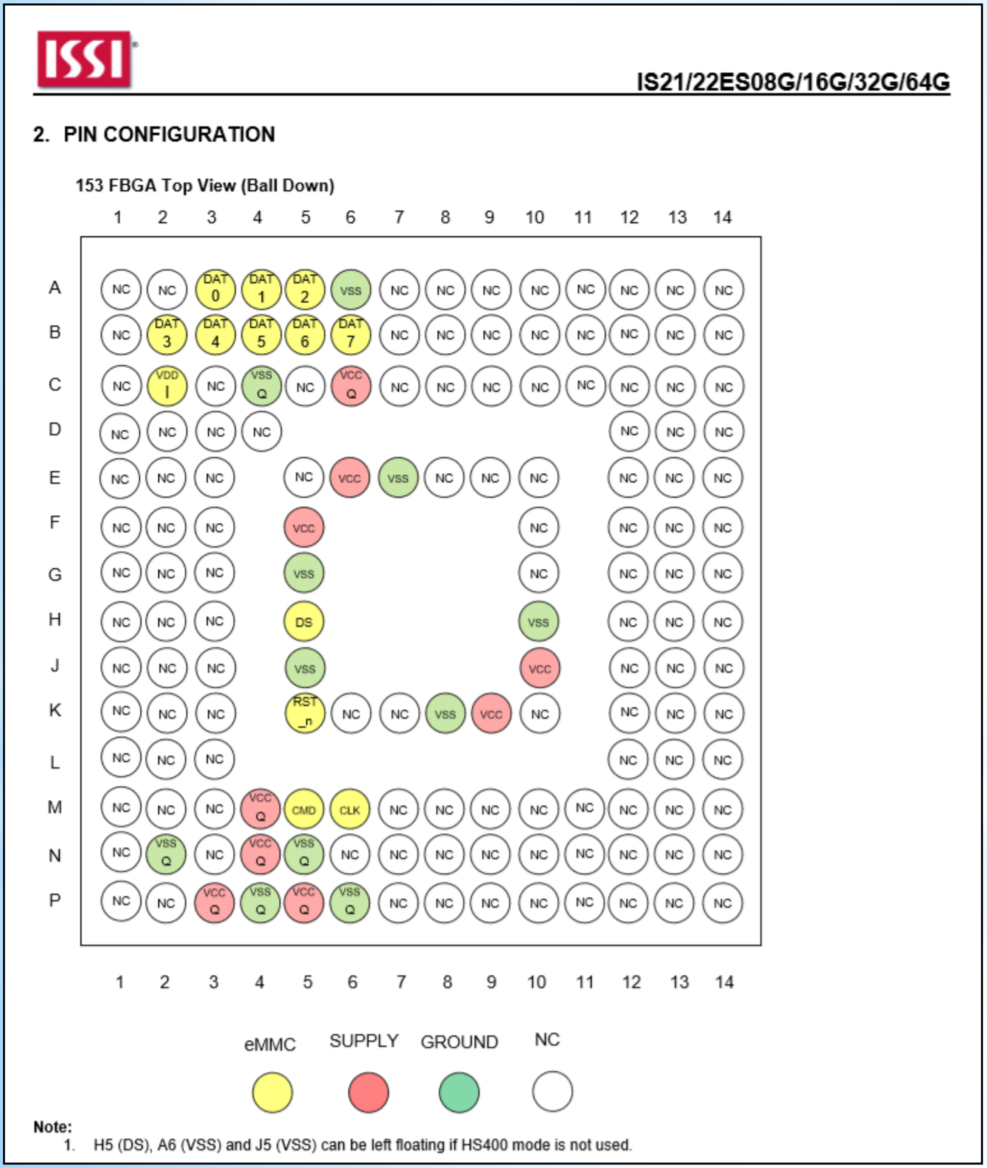
- eMMC is actually based on the same standard as the old MMC cards (before SD cards)
  - Very similar to uSD in SPI mode, but with 1, 4, or 8-bit I/O bus and bidirectional CMD line (Note that CRC is required and can’t be turned off)
  - The 8-bit bus option is what makes this so attractive
  - Open standard, no license required (uSD has a 4-bit bus option but legal usage is murky).
  - Just like with uSD, fastest speed achieved by reading sequential blocks
  - FSRW uSD code adapted for eMMC, interface is nearly the same.
  - The 28 MB/s read rate allows for 480p widescreen video at up to 60 Hz (but you’re run out of disk space and files are limited to 4 GB each)
  - Can load 1080p 8bpp image into HyperRam in ~ 100 ms, allowing for slideshows with good quality
  - There are also super high speed modes, HS200 and HS400, that would let this work at same speed as HyperFlash
    - Sadly, this only works when I/O powered at 1.8 V, might not work without a level shifter…

Alexa





* The giant BGA package looks intimidating, but only a few pins are actually used



ISSI

IS21/22ES08G/16G/32G/64G

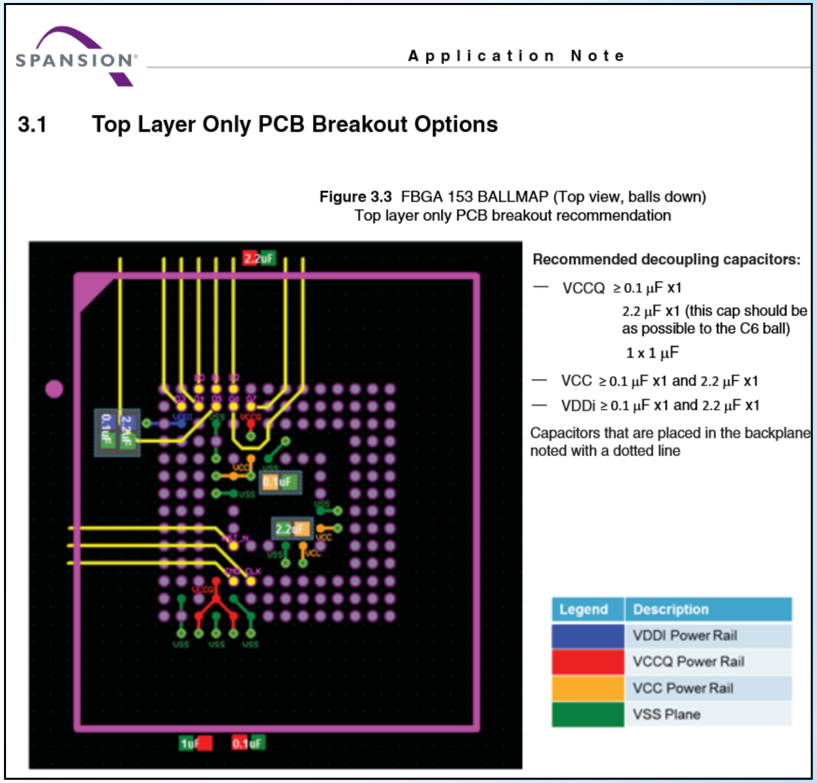
2. PIN CONFIGURATION

153 FBGA Top View (Ball Down)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	NC	NC	DAT0	DAT1	DAT2	VSS	NC	NC	NC	NC	NC	NC	NC	NC
B	NC	DAT3	DAT4	DAT5	DAT6	DAT7	NC	NC	NC	NC	NC	NC	NC	NC
C	NC	VDDI	NC	VSSQ	NC	VCCQ	NC	NC	NC	NC	NC	NC	NC	NC
D	NC	NC	NC	NC								NC	NC	NC
E	NC	NC	NC		NC	VCC	VSS	NC	NC	NC		NC	NC	NC
F	NC	NC	NC		VCC					NC		NC	NC	NC
G	NC	NC	NC		VSS					NC		NC	NC	NC
H	NC	NC	NC		DS					VSS		NC	NC	NC
J	NC	NC	NC		VSS					VCC		NC	NC	NC
K	NC	NC	NC		RST_n	NC	NC	VSS	VCC	NC		NC	NC	NC
L	NC	NC	NC									NC	NC	NC
M	NC	NC	NC	VCCQ	CMD	CLK	NC	NC	NC	NC	NC	NC	NC	NC
N	NC	VSSQ	NC	VCCQ	VSSQ	NC	NC	NC	NC	NC	NC	NC	NC	NC
P	NC	NC	VCCQ	VSSQ	VCCQ	VSSQ	NC	NC	NC	NC	NC	NC	NC	NC
	1	2	3	4	5	6	7	8	9	10	11	12	13	14

eMMC	SUPPLY	GROUND	NC
🟡	🔴	🟢	○

Note:
1. H5 (DS), A6 (VSS) and J5 (VSS) can be left floating if HS400 mode is not used.



SPANSION

Application Note

3.1 Top Layer Only PCB Breakout Options

Figure 3.3 FBGA 153 BALLMAP (Top view, balls down)
Top layer only PCB breakout recommendation

Recommended decoupling capacitors:

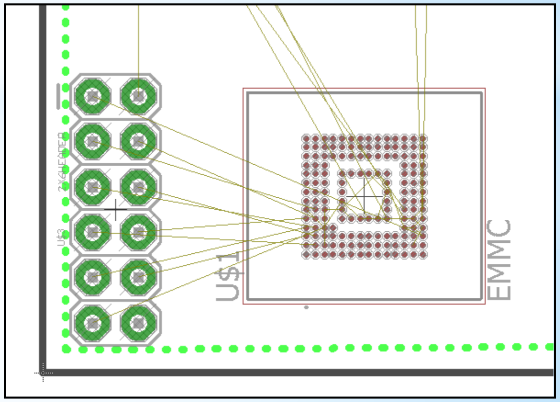
— VCCQ ≥ 0.1 μF x1
2.2 μF x1 (this cap should be as possible to the C6 ball)
1 x 1 μF

— VCC ≥ 0.1 μF x1 and 2.2 μF x1

— VDDi ≥ 0.1 μF x1 and 2.2 μF x1

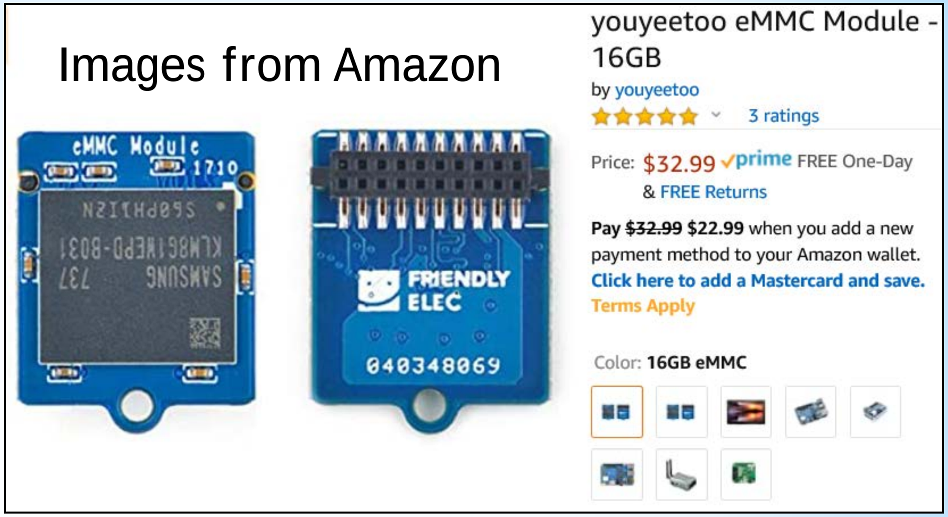
Capacitors that are placed in the backplane noted with a dotted line

Legend	Description
🟦	VDDI Power Rail
🟥	VCCQ Power Rail
🟧	VCC Power Rail
🟩	VSS Plane



* Might be easier to just use a NanoPi eMMC Module…

- Was going to make an adapter, but found one on Amazon
  - Thought was .1” space header, but actually .05”…
  - Comes with uSD adapter!



Images from Amazon

youyeetoo eMMC Module - 16GB

by youyeetoo

★★★★★ 3 ratings

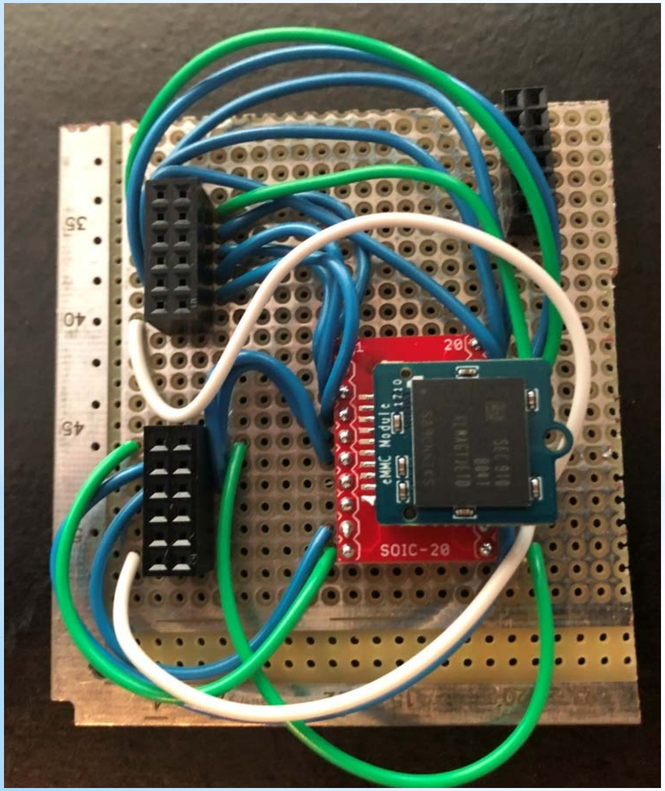
Price: $32.99 ✓prime FREE One-Day & FREE Returns

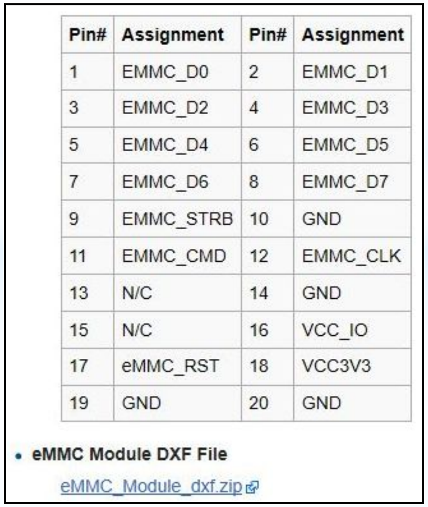
Pay ~~$32.99~~ $22.99 when you add a new payment method to your Amazon wallet.

Click here to add a Mastercard and save.

Terms Apply

Color: 16GB eMMC

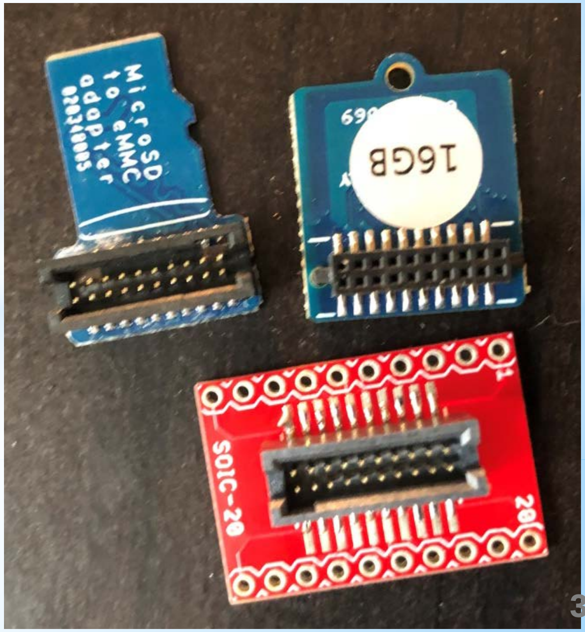




Pin#	Assignment	Pin#	Assignment
1	EMMC_D0	2	EMMC_D1
3	EMMC_D2	4	EMMC_D3
5	EMMC_D4	6	EMMC_D5
7	EMMC_D6	8	EMMC_D7
9	EMMC_STRB	10	GND
11	EMMC_CMD	12	EMMC_CLK
13	N/C	14	GND
15	N/C	16	VCC_IO
17	eMMC_RST	18	VCC3V3
19	GND	20	GND

- eMMC Module DXF File
  eMMC_Module_dxf.zip

Table from FriendlyArm



* Can now play 480p movie at 16 bpp!

- This is decent quality, maybe similar to DVD…
- Uses FSRW adapted for eMMC (Note: 4 GB file size limit with FAT32)
- A widescreen frame fits in HUB ram (640x273x16bpp ~ 350 kB)
- Windows video tool groups uncompressed video and audio into interleaved integer sector sized packets for each frame

Example frame from Blender’s open source movie, “Sintel”

*The End

- Thanks for watching!
- Special thanks to Ken, Chip and Parallax!
- Thank you to all the forum members who supported
- Get started at https://propeller.parallax.com/
- Questions?