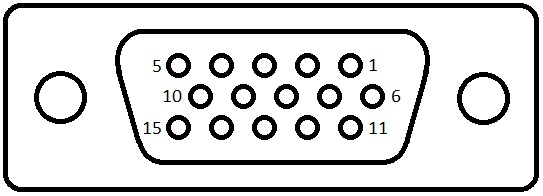
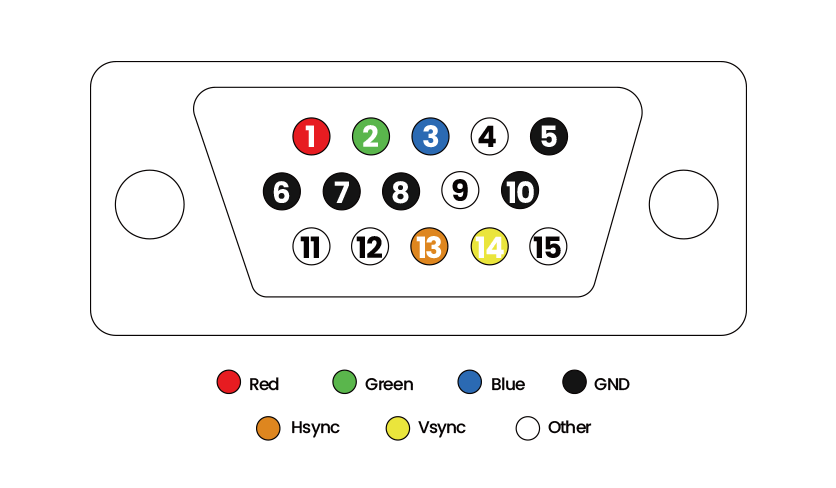
# VGA

The connector is like this:



And the pins are:

|  |  |  |
| --- | --- | --- |
| **Pin No.** | **Pin Name** | **Description** |
| 1 | RED | Red video (75 ohm, 0.7V peak-to-peak) |
| 2 | GREEN | Green video (75 ohm, 0.7V peak-to-peak) |
| 3 | BLUE | Blue video (75 ohm, 0.7V peak-to-peak) |
| 4 | ID2 / RES | Monitor ID Bit 2 / Reserved |
| 5 | GND | Ground |
| 6 | RGND | Red Ground |
| 7 | GGND | Green Ground |
| 8 | BGND | Blue Ground |
| 9 | KEY | +5V DC output from graphic card |
| 10 | SGND | Sync Ground |
| 11 | ID0 / RES | Monitor ID Bit 0 / Reserved |
| 12 | ID1 / SDA | Monitor ID Bit 1 / I2C bi-directional data line |
| 13 | HSYNC | Horizontal Sync |
| 14 | VSYNC | Vertical Sync |
| 15 | ID3 / SCL | Monitor ID Bit 3 / I2C data clock |



Lets start with 640\*480 @60Hz

Signals are:

Red, Green, Blue, HSync, VSync.

A screen begins a new line when it receives a **horizontal sync** and a new frame on a **vertical sync**.



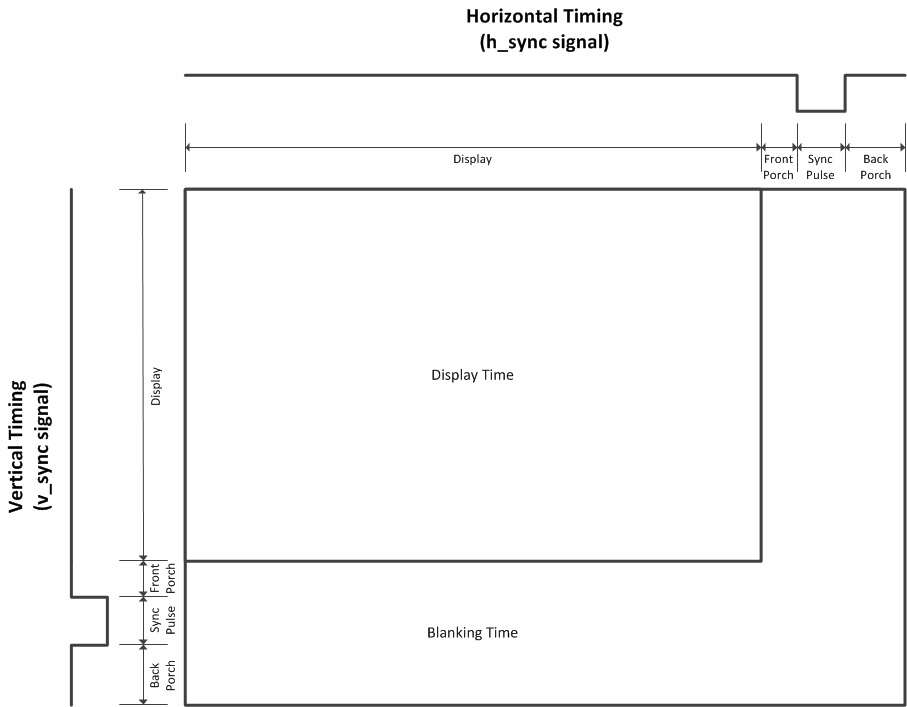
For 640\*480 @60Hz we have:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Horizontal** | **Vertical** |
| Active Pixels | 640 | 480 |
| Front Porch | 16 | 10 |
| Sync Width | 96 | 2 |
| Back Porch | 48 | 33 |
| Total Blanking | 160 | 45 |
| Total Pixels | 800 | 525 |
| Sync Polarity | negative | negative |

The blanking interval has three parts: **front porch**, **sync**, and **back porch**. The front porch occurs before the sync signal, the back porch after.

Therefore, including blanking, we have a total of 800x525 pixels.

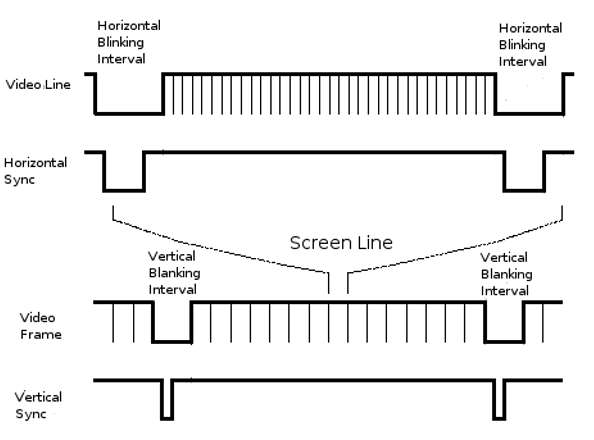
We can rearrange the pixel arrangement to:



The refresh rate is 60 Hz, so the total number of pixels per second is:

800 x 525 x 60 = 25,200,000

Therefore, we want a **pixel clock** of 25.2 MHz.



HSYNC and VSYNC signals are a train of squared pulses of +5V (+3.3V serves too) whereas RGB signals take values in a continuous (analog) voltage range from +0V (absolutely dark) to +0.7V (maximum brightness).

# HDMI

HDMI PCB Considerations

* Use no more than two vias per trace and avoid via stubs.
* Match the differential pair impedance to the impedance of the connector and cable assembly (100 ohm ±10%).
* Minimize inter-pair and intra-pair skew to meet the TMDS signal skew requirement.
* Avoid routing a differential pair over a gap in the underneath plane.
* Use standard high speed PCB design practices.
* Use level shifters to meet electrical compliance at both TX and RX.
* Use robust cables, such as Cat2 cable for HDMI 2.0

Table

Description automatically generated

See this file: <https://www.ti.com/lit/pdf/slla414#:~:text=Inter%2Dpair%20skew%20is%20used,do%20not%20need%20to%20match>.

Audio on HDMI

Graphical user interface, application

Description automatically generated

Any device with HDMI must support the minimum standard of uncompressed stereo [LPCM audio](https://www.the-home-cinema-guide.com/blu-ray-audio-codecs-explained.html) (Other formats are optional).

For digital audio, if an HDMI device has audio, it is required to implement the baseline format: stereo (uncompressed) PCM.

The HDMI specification allows for 8-channels of compressed and uncompressed audio at 1-bit, 16-bit, 20-bit and 24-bit – at sample rates of 32kHz, 44.1 kHz, 48 kHz, 88.2 kHz, 96 kHz, 176.4 kHz and 192 kHz (Supporting 32 channels of audio, started in HDMI 2.0).

Interesting sites:

<https://scolton.blogspot.com/>

# BMP

Bitmap file format

Bitmap file header is:

|  |  |  |
| --- | --- | --- |
| Offset HEX/DEC | Size (Bytes) | Purpose |
| 00/0 | 2 | The header field used to identify the BMP and DIB file is 0x42 0x4D in hexadecimal, same as BM in ASCII. It can following possible values.\* BM – Windows 3.1x, 95, NT, … etc. \* BA – OS/2 struct bitmap array \* CI – OS/2 struct color icon \* CP – OS/2 const color pointer \* IC – OS/2 struct icon \* PT – OS/2 pointer |
| 02/2 | 4 | The size of the BMP file in bytes |
| 06/6 | 2 | Reserved; actual value depends on the application that creates the image |
| 08/8 | 2 | Reserved; actual value depends on the application that creates the image |
| 0A/10 | 4 | The offset, i.e. starting address, of the byte where the bitmap image data (pixel array) can be found. |

DIB Header or info header:

|  |  |  |
| --- | --- | --- |
| Offset HEX/DEC | Size (Bytes) | Purpose |
| 0E | 4 | Size of InfoHeader =40 |
| 12 | 4 | Horizontal width of bitmap in pixels |
| 16 | 4 | Vertical height of bitmap in pixels |
| 1A | 2 | Number of Planes (=1) |
| 1C | 2 | Bits per Pixel used to store palette entry information. This also identifies in an indirect way the number of possible colors. Possible values are:  1 = monochrome palette. NumColors = 1  4 = 4bit palletized. NumColors = 16  8 = 8bit palletized. NumColors = 256  16 = 16bit RGB. NumColors = 65536  24 = 24bit RGB. NumColors = 16M |
| 1E | 4 | Type of Compression  0 = BI\_RGB no compression  1 = BI\_RLE8 8bit RLE encoding  2 = BI\_RLE4 4bit RLE encoding |
| 22 | 4 | (compressed) Size of Image  It is valid to set this =0 if Compression = 0 |
| 26 | 4 | horizontal resolution: Pixels/meter |
| 2A | 4 | vertical resolution: Pixels/meter |
| 2E | 4 | Number of actually used colors. For a 8-bit / pixel bitmap this will be 100h or 256. |
| 32 | 4 | Number of important colors  0 = all |

Color table:

present only if Info.BitsPerPixel less than 8    
colors should be ordered by importance:

|  |  |  |
| --- | --- | --- |
| Offset HEX/DEC | Size (Bytes) | Purpose |
| 36 | 1 | Red intensity |
| 37 | 1 | Green intensity |
| 38 | 1 | Blue intensity |
| 39 | 1 | unused (=0) |