FocusLCDs.com
LCDs MADE SIMPLE®
Ph. 480-503-4295 | NOPP@FocusLCD.com

TFT | CHARACTER | UWVD | FSC | SEGMENT | CUSTOM | REPLACEMENT

LCD Resources:

Graphics Controllers for High Speed Display Interfaces

Graphics Controllers for High Speed Display Interfaces

This resource will discuss the options for graphics controllers for interfacing a display with a high-speed interface. The high-speed interfaces can often benefit from a graphics controller in addition to the central controller to make signal timing more manageable. These interfaces have more demanding requirements on a microcontroller, such as a fast oscillator for the input clocking signal, data signals and synchronization signals. Some examples of high-speed display interfaces would be RGB DPI, LVDS and MIPI DSI. If a microcontroller does not meet the requirements for the display interface, an additional graphics controller chip can be implemented to provide the timing signals and memory to the display.

In addition to controlling the signals and providing memory for the displays frame buffer, an external graphics controller offers a different interface for programming the device. The graphics controller will convert the input signal from the programming controller to match the interface of the display. This can make the connection between graphics controller and the logic controller less intensive. The connection interface will often be offered with lower clocking speeds and smaller bit-width data sizes (SPI, I2C, MCU). This lessens the demand on the microcontroller that would otherwise need to provide these signals, memory, and pin connections.

Three high speed display interfaces will be reviewed in this application. These interfaces are the 24-bit RGB display parallel interface, the MIPI DSI interface and the LVDS interface. An example of the RGB parallel interface will be highlighted to specify requirements of high-speed interfaces and how this becomes an issue for some microcontrollers. Graphics controllers will be offered as an alternative for controllers that do not meet the requirements of the interfaces discussed.

The RGB Interface

To elaborate, consider an 800x480 TFT display with a 24-bit RGB parallel interface. This size display would have a typical clock frequency requirement of about 31MHz. This is how often the 24-bits of data will be transmitted to the display. Clock frequency is calculated as follows:

$$Pixel\ Clock = Horizontal\ Area\ x\ Vertical\ Area\ x\ Frame\ Rate\ x\ (1 + \%Blanking)$$

The clock would need to be provided from the microcontroller as an input signal to the display. As well as a horizontal sync parameter for every row of pixels (including the timing porch parameters) and a vertical sync for every frame at a refresh rate of about 60Hz. Depending on the display there could also be a “data enable” signal, triggered for each set of active data. For reference the ATmega AVR controller common in the Arduino Uno has a typical frequency of 8-16MHz and therefor would be incompatible for the RGB parallel interface.

The minimum memory needed to display one frame of pixels is known as the frame buffer. The frame buffer is calculated by the resolution and the color depth used for the display. The color depth is how many bits are transmitted per pixel, commonly abbreviated as bpp. In continuation with the example, suppose the display is set to use all 24 data lines to send 24-bits of RGB data per pixel. This is known as RGB888 (8bits R, 8bits B, 8bit G) and has a color range of 16.7M colors. The calculation of the frame buffer would be as follows:

$$Frame\ Buffer = 800\ pixels\ x\ 480\ pixels\ x\ 24bpp = 1.15MB$$

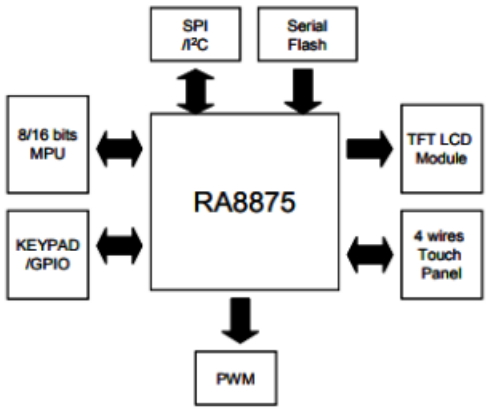
This is the amount of memory needed for a full frame of pixels. The frame buffer is usually stored in SRAM and for reference the Arduino AVR mentioned previously only provides 2kB of SRAM. It is important to note that the displays built in controller IC will sometimes offer internal SRAM provided for the frame buffer. It is necessary to check the specification sheet of the IC of the display to verify if this memory is offered.

To simplify the incompatibility, AVR controllers transmit data in 8-bit increments while the 24-bit RGB DPI requires 24-bits of RGB data to be transmitted per clock sequence. Alternative approaches for the RGB parallel interface would be to use a more powerful controller or to add a graphics controller chip.

RGB Interface Controllers

To determine which graphics controller to choose, we need to assess which requirements need to be fulfilled. One controller that is commonly used is the RA8875. This controller is programmed through a 4-wire SPI interface to send the RGB data to the display. This controller would reduce the amount of pin connections and is programmable through a lower frequency interface such as SPI.

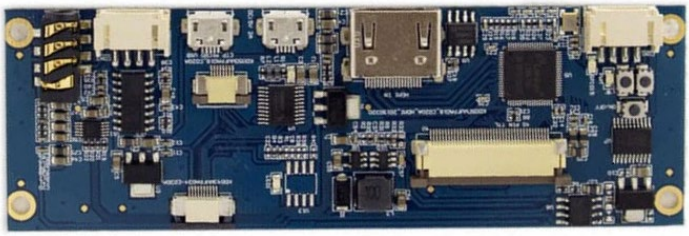
This chip *does* offer a PLL clock frequency multiplier; however, it is not capable of producing a pixel clock frequency that supports 24-bpp at a resolution of 800x480 pixels. In addition, this chip offers 768kB of RAM for the frame buffer which is *insufficient* for the current example. This controller would be better suited for a display with lower resolution or a display that is set to a lower color depth such as RGB565 (16-bpp).



Another graphics controller for the RGB interface is the SSD1963 chip. This controller communicates with the microcontroller over an 8080 or 6800 MCU interface. This controller supports 24-bpp color depth at a resolution up to 864x480 pixels. This controller has a 1215kB frame buffer which provides enough memory for the example stated previously.

This controller would be a good choice for a display of this resolution, color depth and communication interface. This would support an AVR if the 8-bit MCU communication interface is chosen. If a faster interface, such as the 16/18/24-bit MCU interface, is used to communicate with this controller, the AVR would be insufficient. An ARM microcontroller is often a better option to consider even with the addition of a graphics controlling chip.

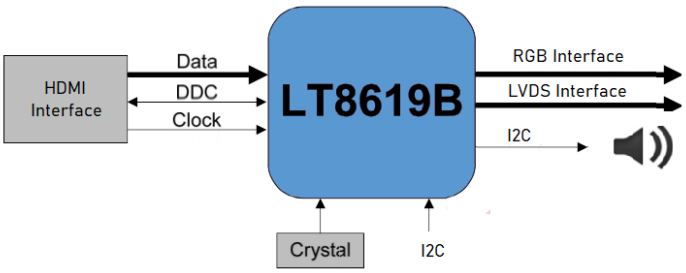
A different option for the RGB parallel interface is to use the LT8619b chip which communicates with the displays over an HDMI interface. This chip supports up to 24-bit color depth and up to 4k resolution. This controller is offered by Focus LCDs on the HDMI module boards available on our website, FocusLCDs.com. These chips are compatible with two interfaces, RGB parallel and LVDS. This would be a good choice if using a single board computer or a processor with a HDMI connection port.



FocusLCDs.com
LCDs MADE SIMPLE®

LVDS Interface Controllers

This interface has low voltage differential signaling for high speed data transmission. The LVDS interface is a common connection type for laptop computers. The LVDS interface transmits data at a rate of 100Mb/s and requires a special functioning chip to generate these clocking signals. Most single board computers or devices with a LVDS port available will have a graphics controller capable of transmitting LVDS data.

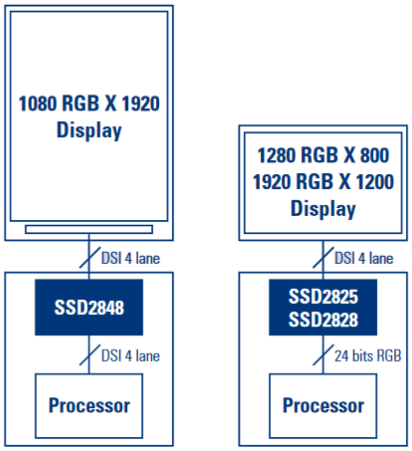


A common LVDS driver is the LT8619b controller made by Lontium. This controller offers the transmission of LVDS data through the HDMI interface. This makes connection to the CPU simple through a single cable. This controller supports the LVDS communication up to 1080 pixels to maintain a refresh rate of 60Hz. This controller would be incompatible with resolutions greater than 1080p. Additionally, this board offers an audio output. There are similar graphics controllers from Lontium in this series that offer different input interfaces.

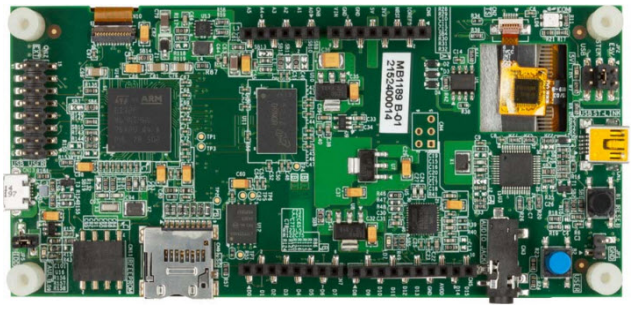
Another LVDS controller is RTD2660 from Realtek which outputs both RGB and LVDS timing signals. This controller can communicate the signals required for the LVDS interface over an HDMI or VGA input. This controller supports resolutions of 1024x600 and 800x480 and the firmware can be programmed and changed for common development boards. Similar Realtek driver boards offer LVDS control with and without audio and at different resolutions.

MIPI Interface Controllers

The MIPI display serial interface requires high-speed differential signals to be provided at a rate of 1Gb/s depending on resolution and number of data lanes used. The MIPI interface is commonly used for cellphone displays. One graphics controller for the MIPI interface is the SSD2828 controller. This controller has a 4-lane MIPI interface and can provide the display with the high speed and low power signals needed. This controller also offers 27.6Mb of memory for the frame buffer which supports displays with resolution up to 1200x1920 pixels. Similar Solomon chips of the same series offer different supported resolutions. This controller converts a 24-bit RGB interface into the 4-lane MIPI interface.



More advanced microcontroller boards will include a MIPI DSI graphics controller with pins specifically for transmitting data through this interface. Since this interface is high speed and data heavy, these microcontroller with the MIPI DSI capabilities will often be more costly. Some examples of these microcontrollers and development boards are from the STM 32-bit microcontroller series.



The STM32F469/479 microcontroller has an integrated MIPI controller and an ARM Cortex-M4 processor. This would be a sufficient microcontroller for this interface and no external graphics controller would be required. If a MIPI integrated microcontroller is used, it is important to verify that the controller has enough memory for the framebuffer. The addition of external RAM memory to support these controllers can reduce the data transmission speed which can make them incompatible for the MIPI interface.

Conclusion

High speed interfaces are desirable for applications with high resolution and high-speed data transfer. It often becomes necessary to support the microcontroller with a chip specifically designed for this task. This makes programming and timing the interface easier through a lower frequency interface to indicate

the specifications of the display. This note offers only a few examples of high-speed interface graphics controllers. There are many graphics controllers available that can meet the demands of these interfaces.

The most significant attributes to look for when choosing a graphics controller will be memory, speed, and pin requirements. The memory requirements are a minimum set by the frame buffer and color depth chosen. The speed of data transmission is set by the interface used and the resolution of the display to maintain a refresh rate of 60Hz. The pin requirements are defined by the display, interface and availability provided by the central controller for the application. Choosing a "conversion" interface (RGB to SPI, LVDS to MCU) between the graphics controller and the high-speed interface of the display is reliant on the capabilities of the microcontroller or processor being used to send the commands to the graphics controller.

Additionally, it is always important to check the capabilities of the internal IC of the display. Some displays will offer internal memory for the frame buffer or an internal oscillator to meet these requirements. These features will be described in the specification sheet of the displays controller which are available on FocusLCDs.com for each of the displays available.