



FocusLCDs.com
LCDs MADE SIMPLE®

Ph. 480-503-4295 | NOPP@FocusLCD.com

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

Application Note FAN4212

3 Wire SPI + Parallel RGB Interface

A display interface that is common for TFT displays is the combination of the 3-wire Serial Peripheral Interface (SPI) and the 16/18/24-bit RGB parallel interface. The two interfaces are used in a sequence to communicate data to the display. This note will discuss how to set up a display using the 3-wire SPI +24-bit RGB parallel interface.

3 Wire SPI + Parallel RGB Interface

The 3-wire Serial Peripheral Interface (SPI) and the 16/18/24-bit RGB parallel interface are used in a sequence to communicate data to the display. The SPI interface is first used to initialize the display parameters through command registers. Once initialized, the display can receive data over the RGB parallel interface at a higher speed. This data can be sent in segments of 16, 18 or 24-bit lengths depending on what is specified by the SPI initialization commands.

The display used in this application note is a 3" TFT with 854x480 pixels. This display has high resolution and high color which requires a faster interface so the pixel clock can maintain a 60Hz frame rate. The part number for this display is E30RA-FW400-N. Details of this display are described below.



Item	Specification	Unit
Part No.	E30RA-FW400-N	-
Type	TFT	-
Resolution	480x854	dots
Colors	16.7M	-
Size	42.04 x 73.67 x 2.17	mm
Interface	3SPI+16/18/24bit RGB	-
Voltage	3.3	V
Controller	ST7701S	-
Backlight	400	nits
Mode	Transmissive	-

Connecting the Display

The display is connected to a graphics controller over a 24 parallel data bus and the display's signaling pins. A graphics controller is used in addition to a microcontroller to provide the high-speed clock signal required to support this resolution and color depth. The connections to the graphics controller in the following table can be changed to connect to a single microcontroller, if the device has RGB parallel dedicated pins. The minimum requirement of RAM reserved for the frame buffer is calculated as (480 pixels x 854 pixels x 24bits/pp) 1.23 MB which will need to be provided by the microcontroller, graphics controller or external memory.

There are 45 pins that are required to be connected to the graphics controller. This includes the 3-wire SPI interface pins and the 24-bit parallel RGB data pins. This display is operated at a low voltage of 3.3V which can be provided by the microcontroller. The display has an additional feature of IPS view, and a bright backlight operated at 19.2V. The backlight voltage will need to be provided externally to the display. Below are the pin functions and connections between the display and the controller.

NO.	Pin	Description	Connection
1	XR	Resistive touch panel terminals	Not connected
2	YD		
3	XL		
4	YU		
5	GND	System ground	Ground
6	GND		
7	VCI	Supply voltage for the display	3.3V
8	IOVCC	Supply voltage for the I/O pins	1.8-3.3V
9	SDO	Serial interface output pin	Not connected
10	SDI	Serial data input pin	Digital pin of microcontroller
11	SCL	Serial interface clock	Digital pin of microcontroller or system clock terminal @2MHz
12	CS	Chip select signal for the serial interface	Digital pin on microcontroller
13	RESET	Reset signal of the display	Reset pin or digital pin on microcontroller
14-37	DB23-DB0	24-bit parallel bidirectional data bus for the RGB interface.	Connected to 24 bidirectional digital pins on the graphics controller
38	DE	Data enable signal for the RGB parallel interface	Connect to DE pin on graphics controller
39	PCLK	Clock signal for the pixels of the display	Graphics controller DCLK pin set to 29MHz
40	HSYNC	Horizontal sync signal. Synchronizes each line of pixels.	Hsync pin on graphics controller
41	VSYNC	Vertical sync signal. Synchronizes the frame total of the display.	Vsync pin on graphics controller

42	NC	Not connected	Not connected
43	LEDK	Cathode pin of the backlight	External backlight power GND
44	NC	Not connected	Not connected
45	LEDA	Anode pin of the backlight	External backlight voltage 19.2V, 20mA

Setting Interface Clock Signals

The serial interface clock is run at a lower frequency of 2Mhz. This frequency was chosen to maintain signal integrity and can be changed to any value between 1-15Mhz for this controller. At higher frequencies, noise can become a factor and create errors when sending commands and data. Since the SPI interface is only used to send 8-bit commands to initialize the display, it does not require a high frequency clock. This makes it possible for a basic microcontroller to provide these signals without the use of the graphics controller.

The DCLK signal is determined by the resolution, porch settings of the HSYNC and VSYNC signals and the frame rate of the display. The frame rate is standard at 60Hz, which is the minimum frequency to maintain an image. The timing parameters of the HSYNC and VSYNC signals are specified by the controller. Below are the timing parameters specified in the ST7701S datasheet and the displays datasheet.

Parameter	Description	Min.	Typ.	Max.	Unit
Horizontal Back Porch	Start position of the horizontal active area.	2	50	255	DCLK
Horizontal Front Porch	Start position of the HSYNC pulse signal	2	10	-	DCLK
Horizontal Pulse Width	Number of DCLKs in one HSYNC pulse signal	2	2	255	DCLK
Horizontal Total	Total number of DCLKs in horizontal line	2	542	-	DCLK
Vertical Back Porch	Start position of the vertical active area	2	30	254	line
Vertical Front Porch	Start position on the VSYNC pulse signal	2	10	-	line
Vertical Pulse Width	Number of lines in the VSYNC signal	2	2	254	line
Vertical Total	Total number of HSYNC pulses in one frame	2	896	-	line

The total HSYNC period for this display is calculated as the total time in DCLKs for one line, including the time where the pixels are not active. In this example, this value is calculated to be $480+50+10+2=542$ DCLKs. The total VSYNC period for one frame of pixels to be displayed is calculated in the number of lines. The total time it takes for one frame of pixels is $854+30+10+2=896$ lines.

The dot clock (DCLK) signal is determined by these two totals and the frame rate in order to maintain the image at 60Hz. The calculation of the DCLK signal is as follows: $542 \times 896 \times 60 = 29\text{Mhz}$. This is a high clock speed for the RGB interface. The maximum clock speed for the RGB parallel interface with this controller is 33MHz. Certain steps should be taken to reduce noise on this signal to ensure reliability of the data transmitted.

Initializing the Display Through SPI

The 3-wire SPI interface is used to send 8-bit commands and data to initialize the display. Each of these commands can be found in the specification sheet for the display's embedded controller IC ST7701S. These commands specify the selection of color format, resolution, timing parameters and power control. There are a few ways to send these commands to the display.

Most microcontrollers offer dedicated SPI interface pins for the SPI protocol. There are many code repositories available for the SPI interface so there are many options for sending these commands and data. In this application, the SCL clock signal is set at 2MHz. This lower frequency was chosen over a maximum frequency of 15MHz to ensure proper signal timing and to avoid noise. Since the SPI interface is used for a brief initialization period, high speed is not required.

Another way is to use any digital pins available and "bit-bang" the information. This is when the clock and data pins are signaled on and off in the right sequence to communicate the data. This is done when there are not specific SPI pins available. This is common with the 8051 microcontrollers where most ARM and AVR microcontrollers have an SPI interface. This option has the benefit of control over whether the data is set at the rising or falling edge of the clock signal. This can be different for each display that uses the SPI interface for initialization, so it is important to confirm in the data sheet.

Below are SPI initialization commands for the 24-bit parallel RGB interface. These commands set many functions of the display such as: timing constraints for the RGB interface, color depth, resolution, and voltage control.

Initializing the Graphics Controller

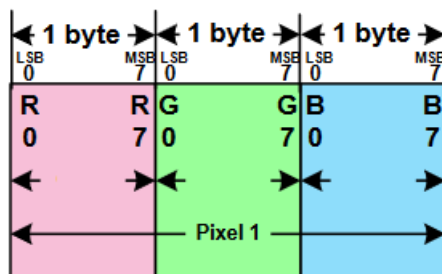
This application uses a graphics controller to increase the clock speed and provide memory. If the microcontroller is providing the RGB function to the display, then this step can be skipped. The graphics controller used in this application is SSD1963. This graphics controller provides the memory for the frame buffer of the display. The DCLK signal is also generated by this controller to an RGB PLL setting of 29MHz.

The RGB interface pins are connected to the graphics controller in parallel. This application uses the 24-bit interface so all 24 data bus pins must be connected to the controller. The HSYNC, VSYNC, DE and DCLK pins are also connected to this controller. The graphics controller is controlled by commands over a 16-bit parallel MCU interface. These pins will need to be connected to the microcontroller to send the initialization commands as well as the data that will be projected on the display.

The initialization of the graphics controller will specify the desired DCLK speed that is needed for the RGB interface. These commands are sent by the microcontroller in sequences of 8-bits. Because the commands are 8-bits, it is possible to send the data over an 8-bit parallel interface. The commands that are used to initialize the graphics controller are reviewed in the data sheet of the IC SSD1963. Below are the commands that were used to initialize the graphics controller.

Using the RGB Interface

The RGB interface can now be used to transmit data to the display. The serial interface no longer needs to be used after initialization is complete. Now the 24 RGB data pins will send the 24-bits of RGB data per clock cycle. The data in this example is processed at 24bpp. This means that each pixel will have 24-bits of information for the color depth and that the display is capable of 24 million different color variations.



A brief example of how to display colors on the display will be reviewed. Displaying colors onto the screen requires sending 24-bit color data to each of the visible pixels in the display area. This can be done by writing to each location in RAM where every pixel is stored and telling it which color you have selected. This application can be used to portray bitmap pictures through the same process.

Interface	Cycle	D[23]	D[22]	D[21]	D[20]	D[19]	D[18]	D[17]	D[16]	D[15]	D[14]	D[13]	D[12]	D[11]	D[10]	D[9]	D[8]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
24 bits	1 st	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
18 bits	1 st							R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
16 bits (565 format)	1 st									R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1
16 bits	1 st									R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0
	2 nd									B7	B6	B5	B4	B3	B2	B1	B0	R7	R6	R5	R4	R3	R2	R1	R0
	3 rd									G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
12 bits	1 st													R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4
	2 nd													G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
9 bits	1 st																R5	R4	R3	R2	R1	R0	G5	G4	G3
	2 nd																G2	G1	G0	B5	B4	B3	B2	B1	B0
8 bits	1 st																	R7	R6	R5	R4	R3	R2	R1	R0
	2 nd																	G7	G6	G5	G4	G3	G2	G1	G0
	3 rd																	B7	B6	B5	B4	B3	B2	B1	B0

Below is the example used to write to RAM stored on the SSD1963 controller. This process is done using the command “memory write” (0x2c) of the graphics controller for each of the pixels on the display. The colors can be sent in 6-digit, 24-bit sequences and decoded or as three 8-bit sequences representing one for each of the red, green, and blue components of the color. Below the color will be sent as components of the RGB color in three steps on eight bits to the graphics controller. The image above shows how the 24-bit data is sent to the graphics controller in sequences of 16-bits.

DISCLAIMER

Buyers and others who are developing systems that incorporate FocusLCDs products (collectively, “Designers”) understand and agree that Designers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Designers have full and exclusive responsibility to assure the safety of Designers’ applications and compliance of their applications (and of all FocusLCDs products used in or for Designers’ applications) with all applicable regulations, laws and other applicable requirements.

Designer represents that, with respect to their applications, Designer has all the necessary expertise to create and implement safeguards that:

- (1) anticipate dangerous consequences of failures
- (2) monitor failures and their consequences, and

(3) lessen the likelihood of failures that might cause harm and take appropriate actions.

Designer agrees that prior to using or distributing any applications that include FocusLCDs products, Designer will thoroughly test such applications and the functionality of such FocusLCDs products as used in such applications.