FocusLCDs.com
LCDs MADE SIMPLE®

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

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

Application Note FAN4212

RGB Interface with Arduino Due

This application note will discuss how to connect a 5.0” portrait mode display through the RGB interface with an Arduino Due.

RGB Interface with Arduino Due

The purpose of this application note is to review the RGB interface and discuss any requirements necessary for communicating with the microcontroller through this interface. The display communicates with the microcontroller over 45 pins through an RGB interface. An additional graphics controller is required to use this interface. The graphics controller chosen for this application is the SSD1963 LCD Controller Graphics card.

The display featured in this application note is a transmissive, 5.0” (67.56mmx122.35mm), portrait mode TFT display. The part number for this display is E50RG84885LWAM520-CA. This display has a built in controller IC ILI9806E. This display also has a capacitive touch feature. Features of this display are listed below.

TFT Features

Low Input Voltage: 3.3V
Display Colors: 65k/262k/16.7M colors
Interface: 3 wire SPI + 24-bit parallel RGB
CTP Interface: I2C
Display Mode: Transmissive/Normally Black

Backlight Voltage: 19.2V, 40mA
Viewing Angle: All View
TFT Controller IC: ILI9806E
CTP Controller IC: G911
CTP Touch Mode: 5-Point

Mechanical Information

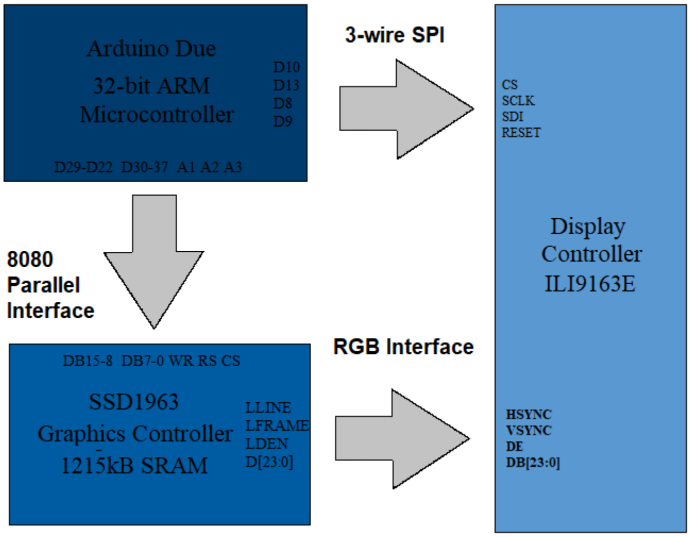
Item		Min	Typ.	Max	Unit	Note
Module Size	Height (H)		67.56		mm	-
	Vertical (V)		122.35		mm	-
	Depth (D)		4.03		mm	-

Requirements

This section will discuss the materials that are used in this application. Below is a list of the components used and their specified function.

Item	Description	Note
5.0” TFT Display	Part No: E50RG84885LWAM520-CA.	FocusLCDs.com
Arduino Duo	Arduino Due microcontroller, 32-bit ARM core processor, 54 pins.	Arduino
SSD1963 Controller	SSD1963 LCD Graphics Controller, 1215kB embedded SRAM for display frame buffer.	Solomon
FPC Connector	45-pin FPC connector. Converts 45-pin ribbon display connector to output pins.	
DC Power Supply	Voltage generator to provide 19.2V, 40mA to display backlight.	
Micro USB	Micro USB Connectors for Arduino serial programming interface.	

The display used in this application has an internal built-in display controller IC ILI9806E. This controller IC does not contain internal RAM and therefore an additional graphics controller SSD1963 is required to provide the RAM that supports the RGB interface. The ILI9806E controller IC that is built into the display provides the 3-wire serial interface to input the initialization commands for the RGB interface.



Once the initialization commands are sent via the 3-wire serial interface, the graphic controller will be used to send the RGB commands to the display. The SSD1963 graphics controller chip is used to communicate to the RGB interface on the display. The graphics controller provides the SRAM required to drive the display. This controller provides a 1215kB frame buffer to support the 24-bit graphics data to the display.

The graphics controller would not be required if the internal IC embedded in the display contains internal RAM. The specification sheet for the embedded display controller IC should be used to verify this information. The graphics controller chip is accessed after the SPI initialization of the display. The graphics controller chip communicates with the microcontroller through a 16-bit parallel 8080 MCU interface.

The microcontroller in this application is a 32-bit ARM core processor. This device communicates with the display over the serial interface to send the SPI initialization commands. Once completed, the microcontroller will then communicate to the graphics controller through 8-bit parallel initialization command and then 16-bit graphical data commands and functions.

Hardware Connections

A review of the connection ports and pins between each device will be specified in this section. Starting with the display and the connections with the graphics controller as well as the microcontroller. The display's 3 wire serial pins can be directly connected to the microcontroller. The RGB interface pins will be connected to the graphics controller. Below is a description of the pin connections on the display.

5.0" TFT Pin Assignment

Pin No.	TFT Pins	Description	Connection
1	XR	Resistive touch panel interface pins. Not connected for this display.	Leave open
2	YD		
3	XL		
4	YU		
5	GND	Ground	Ground
6	GND		
7	VCI	Supply voltage	3.3V
8	IOVCC	I/O supply voltage	
9	SDO	Serial data output pin	Not connected
10	SDI	Serial data input pin	Digital pin 8 on MCU
11	SCL	Serial clock pin	Digital pin 13 on MCU
12	CS	Chip select pin for the serial interface	Digital pin 10 on MCU
13	RESET	Reset pin	Digital pin 9 on MCU
14-37	DB23-DB0	24-bit parallel bi-directional data but for RGB interface	RGB data pins on GC (see datasheet)
38	DE	Data enable pin for RGB interface	LDEN pin on GC
39	DOTCLK	Dot clock signal for RGB interface	LSHIFT pin on GC
40	HSYNC	Line synchronizing signal for RGB interface	LLINE pin on GC
41	VSYNC	Frame synchronizing signal for RGB interface	LFRAME pin on GC
42	NC	Not connected	N/C
43	LEDK	Cathode pin of the backlight	Backlight ground
44	NC	Not connected	N/C
45	LEDA	Anode pin of the backlight	+19.2V, 40mA

GC: Graphics Controller (SSD1963), MCU: Microcontroller (Arduino Due), N/C: Not connected

The next hardware connection that will be reviewed is between the graphics controller chip and the microcontroller. The two devices are connected through a 16-bit 8080 parallel interface. The graphics controller will receive initialization commands from the microcontroller specifying its own requirements as well as commands that will be sent to the display over the RGB interface. The parallel connection between the microcontroller and the graphics controller are as follows.

Graphics Controller (SSD1963) Pin Assignment

Pin No.	SSD1963 Pins	Description	Connection
1	VCC	Voltage supply 3.3V	3.3V
2	GND	Ground	Logic ground
3	DB15	Data bus 15	Digital pin 29 on MCU
4	DB14	Data bus 14	Digital pin 28 on MCU
5	DB13	Data bus 13	Digital pin 27 on MCU
6	DB12	Data bus 12	Digital pin 26 on MCU
7	DB11	Data bus 11	Digital pin 25 on MCU
8	DB10	Data bus 10	Digital pin 24 on MCU
9	DB9	Data bus 9	Digital pin 23 on MCU
10	DB8	Data bus 8	Digital pin 22 on MCU
11	DB7	Data bus 7	Digital pin 30 on MCU
12	DB6	Data bus 6	Digital pin 31 on MCU
13	DB5	Data bus 5	Digital pin 32 on MCU
14	DB4	Data bus 4	Digital pin 33 on MCU
15	DB3	Data bus 3	Digital pin 34 on MCU
16	DB2	Data bus 2	Digital pin 35 on MCU
17	DB1	Data bus 1	Digital pin 36 on MCU
18	DB0	Data bus 0	Digital pin 37 on MCU
19	WR	Parallel interface write pin	Analog pin A1 on MCU
20	RS	Parallel interface register select pin	Analog pin A2 on MCU
21	RST	Parallel interface reset pin	3.3V
22	CS	Parallel interface chip select pin	Analog pin 3 on MCU
23	GND	Ground	Ground

GC: Graphics Controller (SSD1963), MCU: Microcontroller (Arduino Due), N/C: Not connected

The microcontroller provides the logic voltage and logic ground for both the display and the graphics controller. There are two reset pins total, one connected to the display and one for the graphics controller. It is important to only use the reset pin for the SPI initialization and set the graphics controller reset to 3.3V. If the graphics controller reset pin is toggled, the SPI initialization commands stored on the display will be reset as well.

The Arduino Due is connected to both the display and the graphics controller. First the microcontroller sends the SPI initialization commands to the display's embedded controller IC. Next it will communicate with the graphics controller over an 8-bit 8080 parallel interface to initialize the 16-bit parallel interface parameters. Once both devices are initialized, further commands can be sent through the graphics controller to the display to create an image.

Commands

The initialization commands required for the display and graphics controller to start will be reviewed in this section. The microcontroller will need to send two sets of initialization code over each interface. The pins will need to be calibrated to each of their specified interfaces, SPI and 8080 Parallel interfaces. The first set of initialization commands are sent to the display's SPI pins. This code will define the RGB interface parameters required to use this interface.

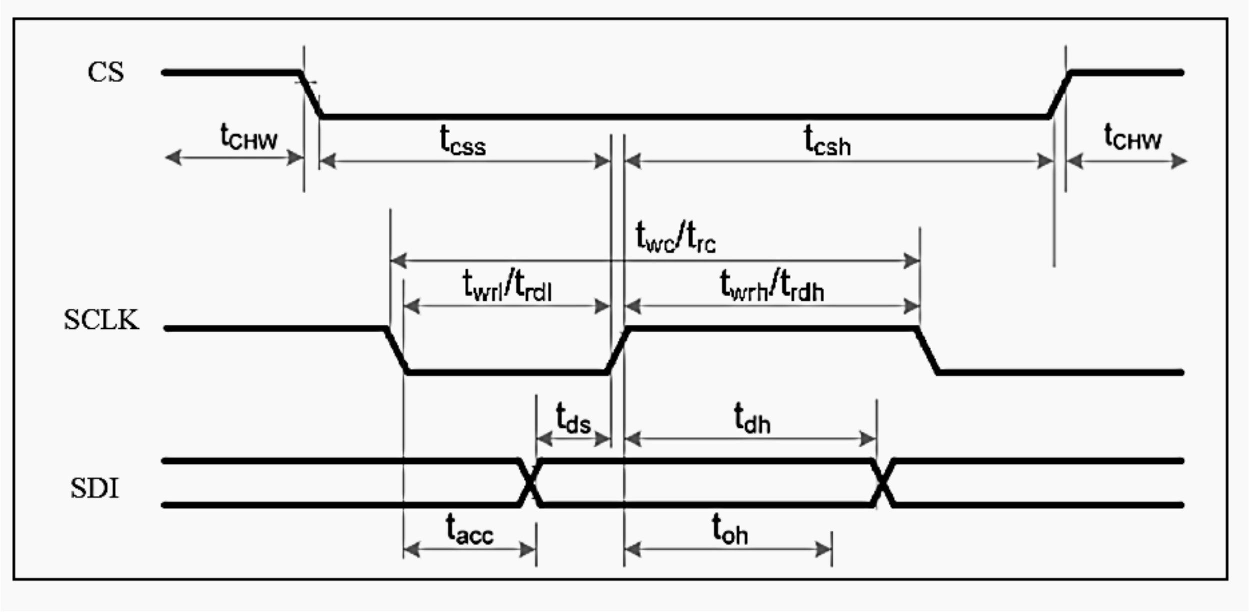
The initialization code that is sent to the displays SPI pins is as follows.

```
RESET_SPI=1;
delay(1);
RESET_SPI=0;
delay(10);
RESET_SPI=1;
delay(120);

write_command(0xE0);
```

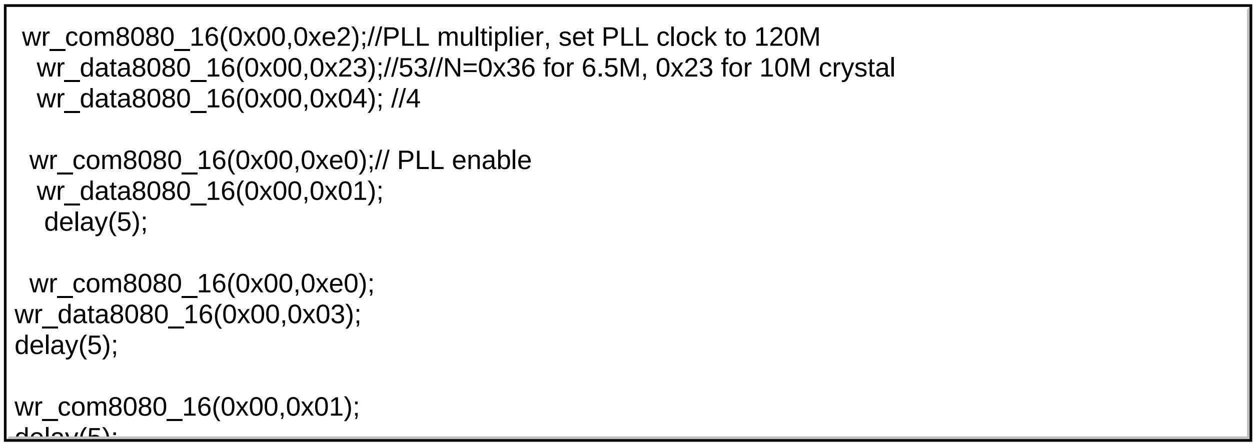
This code defines the RGB interface timing characteristics, the resolution of the display, the voltage settings and much more. The initialization commands and data specific to the display can be found in the data sheet of the embedded IC controller ILI9806E. Review the specification sheet for this controller for details on the commands and functions of this code.

To send the data and commands over the SPI interface, four peripheral pins will need to be declared depending on the microcontroller used. Once the data pins are setup the SPI interface can send the initialization code in the following sequence. This code is represented in the following timing diagram as a typical 3-wire SPI interface.



3-Wire Serial Interface Timing Diagram

The initialization code for the graphics controller is sent over a parallel 8080 interface. The sequence of commands is to first initialize the graphics controller and specify the display parameters, followed by the code that will be sent over the 16 data buses. The initialization commands for the graphics controller SSD1963 are detailed in the controller's specification sheet. Below is the code used to initialize the graphics controller.



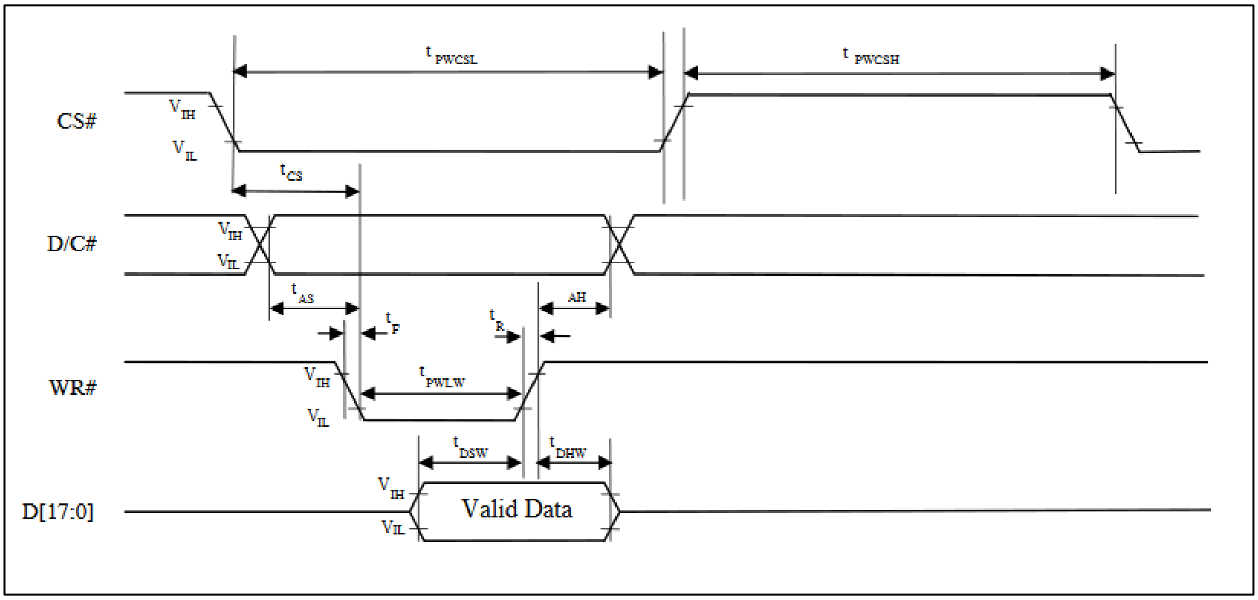
```
 wr_com8080_16(0x00,0xe2);//PLL multiplier, set PLL clock to 120M
   wr_data8080_16(0x00,0x23);//53//N=0x36 for 6.5M, 0x23 for 10M crystal
   wr_data8080_16(0x00,0x04); //4

  wr_com8080_16(0x00,0xe0);// PLL enable
   wr_data8080_16(0x00,0x01);
    delay(5);

  wr_com8080_16(0x00,0xe0);
wr_data8080_16(0x00,0x03);
delay(5);

wr_com8080_16(0x00,0x01);
delay(5);
```

The initialization code for the graphics controller defines the parameters of the display and the required timing characteristics. The specifications and timing definitions can be found in the spec sheet of the display. The commands that are sent in this section lets the graphics controller know the details specific to the display that we are using in this example. The data is sent over the 8080-parallel interface to the graphics controller. Below is the timing diagram of this parallel interface.



Parallel 8080-series Interface Timing Diagram

Conclusion

After the initialization commands for both the display and the graphics controller are sent, the display is now ready for communication. The graphical data or images will be sent to the graphics controller's RAM to be accessed by the display via the RGB interface. Communication to the display is fast and efficient for transmitting images that are to be displayed. For this example, an application was written to display bitmaps on the screen. Below is the output of this example.

DISCLAIMER

Buyers and others who are developing systems that incorporate Focus LCDs 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 Focus LCDs 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 Focus LCDs products, Designer will thoroughly test such applications and the functionality of such Focus LCDs products as used in such applications.