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

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

Application Note FAN4205

Storing 65k Color Bitmaps in Flash Memory: 1.8" TFT and ST Nucleo-L476RG

This application note discusses how to setup a 1.8" TFT display with the ST Nucleo-L476RG microcontroller. An example of loading bitmap images stored in flash memory onto the LCD can be found at the end of this application note. A further demonstration of this display's features can be found here.

Storing 65k Color Bitmaps in Flash Memory: 1.8" TFT and ST Nucleo-L476RG

The display used in this project is a 1.8" TFT with 128x160 pixels of resolution. The microcontroller used is the Nucleo-L476RG from ST. The TFT display will be interfaced with the microcontroller via a 4-wire serial connection and programmed using the Arduino IDE platform.

In just a few steps the TFT can be wired and programmed to display up to 65K colors and 128x160 pixels of resolution. This display is a good option for storing 16-bit color bitmaps as they will take up less space in the flash memory. Various wiring and interface options are available, from 3-4 wire serial, to 16/18-bit RGB and 8/9/16/18-bit MCU parallel. The 4-wire serial interface is fast when interfaced with the ST microcontroller, however for additional data transmissions speed the parallel options are available. Additional features of this display are below. As always, check out the data sheet for the specs of this specific display. (datasheet)

- 45 pins, 34.0x47.0 mm
- 128(RGB)x160 pixels, TN
- ILI9163 controller
- White LED Backlight
- Transmissive/Normally White
- Serial and Parallel interfaces
- 3.3V (TYP)

What You'll Need

1. First you will need an IDE to program the display. This ST microcontroller lets you use a range of IDE's so it is up to personal preference. I am going to use the Arduino IDE, any IDE that supports C++ and is compatible with the ST microcontroller will work. There is a list of options on ST's website. We will come back to this after the hardware is setup.

2. Below is a list of the physical materials you will need to setup the project.

QTY	Description	Note
1	E17RG11216LW6M300-N 1.8" TFT display	Focus LCDs
1	ST Nucleo-L476RG microcontroller with USB Cable	ST
1	45-pin FPC connector board 0.5 mm pitch	
1	Male-to-Male Jumper Wires	
1	55Ω resistor	
1	Soldering Iron	
1	Solder	
1	Solderless breadboard	

Wiring

There are 45 pins on the ribbon cable connected to the display. You will need to connect a 45-pin FPC connector board to convert the ribbon cable to usable pin outs to connect to the microcontroller.

Figure 1: 1.8" TFT with FPC cable

There are only a few connections that need to be made between the display for the 4-wire serial interface. The unused parallel data pins will be pinned to GND.

Consult the datasheet for a detailed explanation of each pin assignment and their functions. The 4-wire serial data pins are connected to the ST specific serial inputs for the "Hardware SPI" programming option. While any pins can be used, their location must be defined in the "Software SPI" programming option.

Pin Assignments

Pin definitions and connection points are described in the table below. We will use the 4-wire serial interface for this example to save data pins on the microcontroller. A more in-depth description of each of the pins can be found on the datasheet. All unused pins are connected to ground.

Pin No.	Pin Name	Description	Connection
1	GND	Ground	Ground
2	LEDK	Backlight cathode	GND
3-5	LEDA	Backlight anode (3.3V, 60mA)	55Ω resistor → 3.3V
6	IM0	Parallel interface selection	GND
7	IM1	Parallel interface selection	GND
8	IM2	Parallel or serial selection, serial = high	3.3V
9	RCM1	RGB or MCU	GND
10	SPI4W	3/4 wire serial selection, 4-wire = high	3.3V
11-12	VCC	Supply voltage (3.3V)	3.3V
13	SDI	SDI (MISO)	D12
14	RD	Read enable	3.3V
15	DC/SCL	SCL in serial interface (SCLK)	D13
16	WR/RS	D/CX in 4-wire serial interface	D9
17	CS	Chip select	D10
18	Reset	Reset pin	D8
19-35	DB17-DB1	Data pins for parallel interface	GND
36	DB0	SDA pin for serial interface (MOSI)	D11
37	PCLK	Clock for RGB	GND
38	DE	Data enable	GND
39	HSYNC	HSYNC for RGB	GND
40	VSYNC	VSYNC for RGB	GND
41-44		Not connected	
45	GND	Ground	GND

The Nucleo-L476RG has dedicated serial input pins specific to the board. The pin locations can be seen below and are described in the table for how they are connected to the display. These and other hardware pin definitions can be verified on the board.

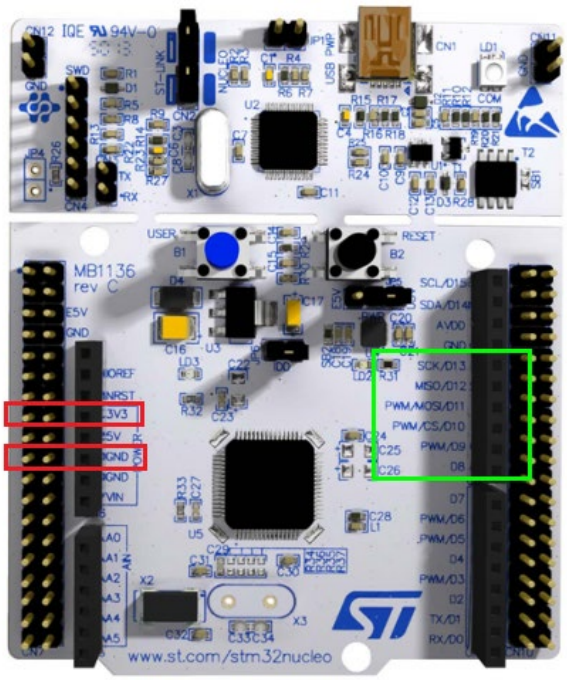


Figure 2: ST Nucleo-L476RG 4-wire Serial Hardware Pin Definitions

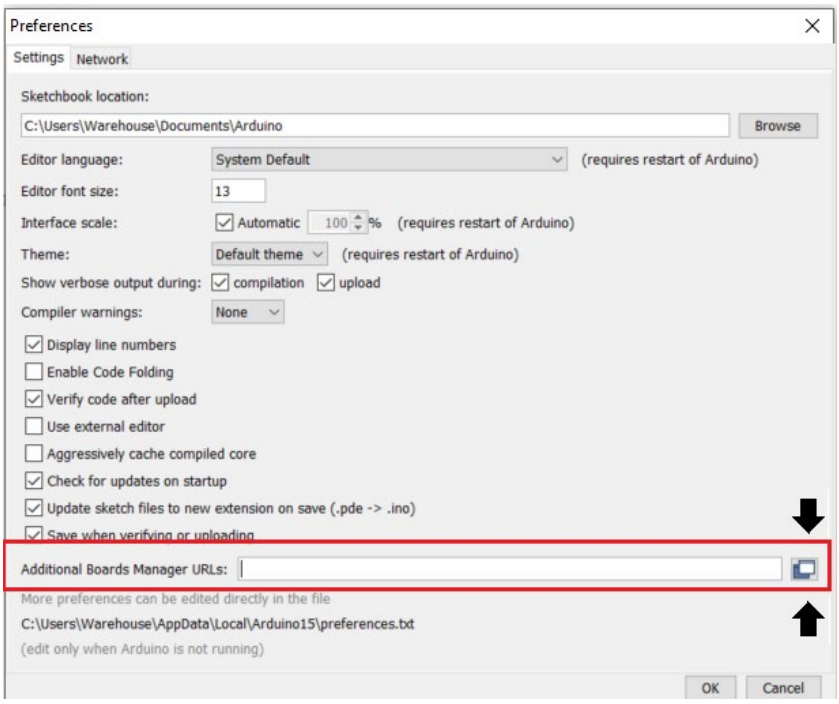
After the screen is connected, and the TI microcontroller is plugged into the computer you will see the white LED backlight come on. That is a good sign that things are connected correctly.

Programming the ST Nucleo-L476RG

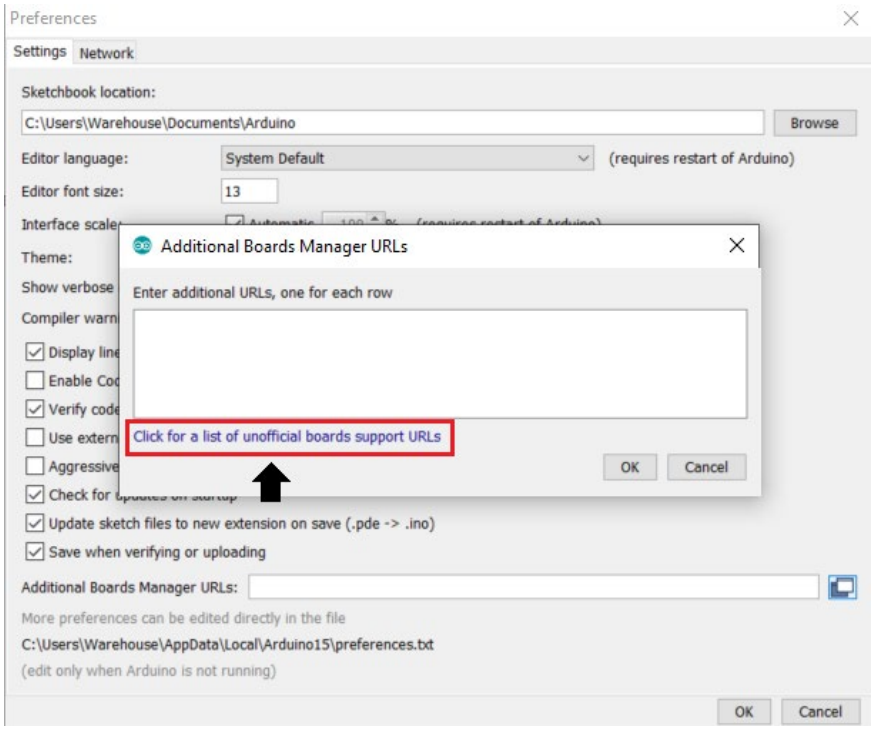
Now it is time to program the microcontroller. For this example, I used the Arduino IDE. There are alternative IDE's that you can use to program the display. I chose Arduino IDE from preference and the accessibility of a variety of examples for TFT's . I'll briefly explain how to add the Nucleo-L476RG ST board to the Arduino IDE if you choose to go this route. If you choose a different IDE then you can skip to part 2.

1.) First open the Arduino IDE and go to File → Preferences

The Preferences window should appear as below. Near the bottom of the menu there is an option for "Additional Boards Manager URL's.



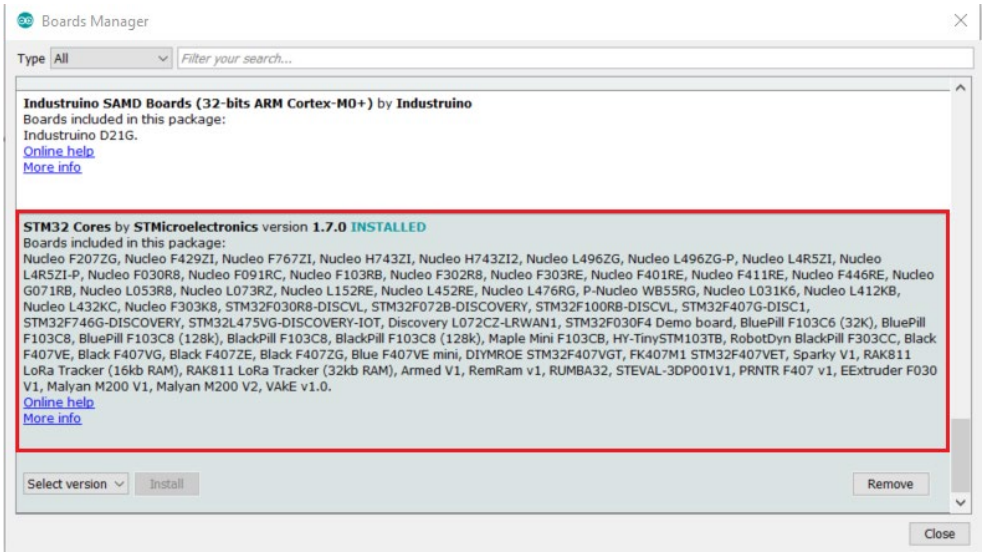
Click on the options for the Additional Boards Manager and the option for the list of “unofficial boards URLs”. It will bring up a list of the devices compatible with the Arduino IDE.



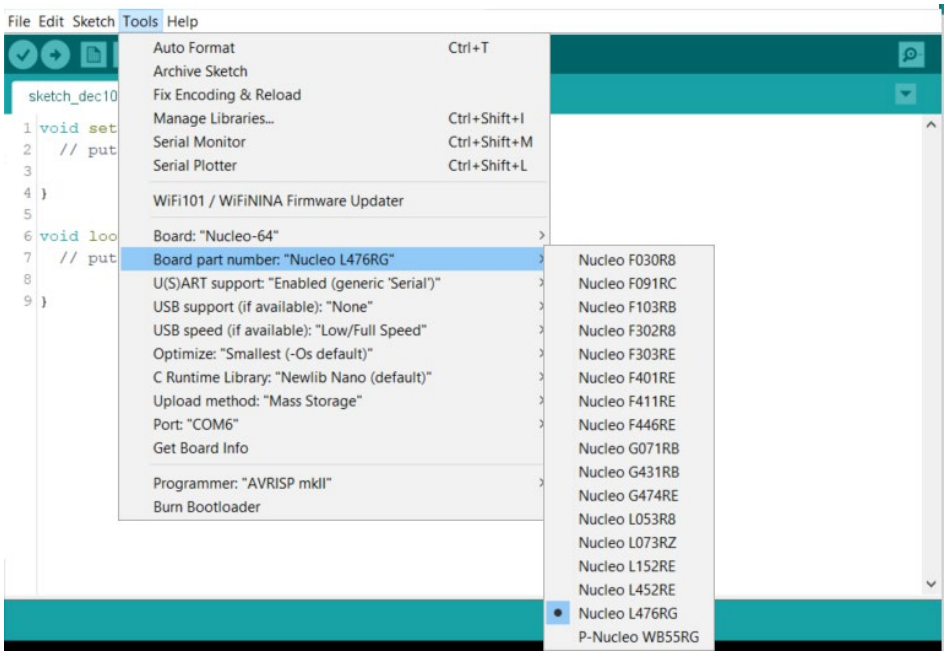
Copy and paste the STM32 core URL into the additional boards manager section in the Arduino IDE. The URL is as follows:

"https://raw.githubusercontent.com/stm32duino/BoardManagerFiles/master/STM32/package_stm_index.json"

Press Ok and then restart the IDE. Next you can verify that these boards are installed by going to Tools → Board Manager and scroll to the bottom to see if the STM32 Core based boards are added to the IDE.

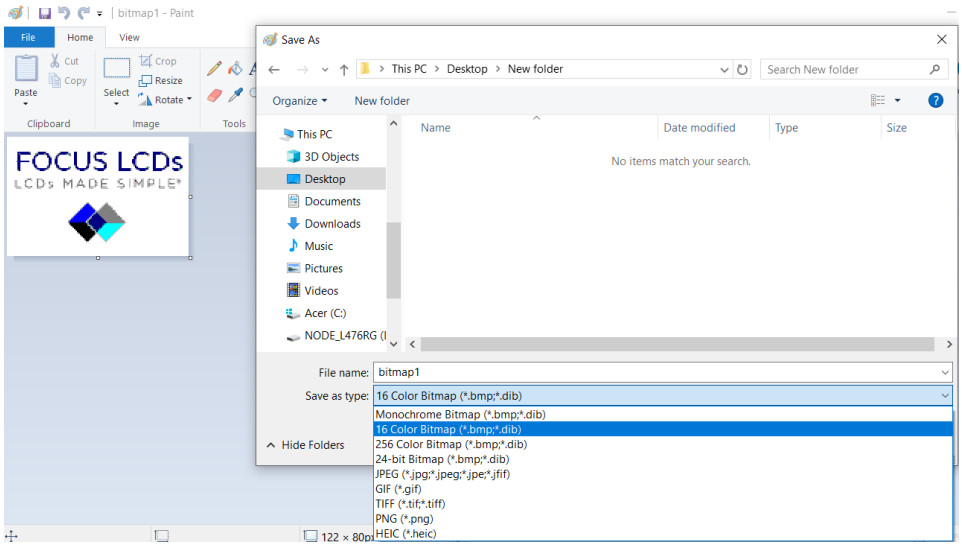


After this is installed go to Tools and choose the Nucleo-64 boards and below choose the Nucleo-L476RG. The settings are chosen below.

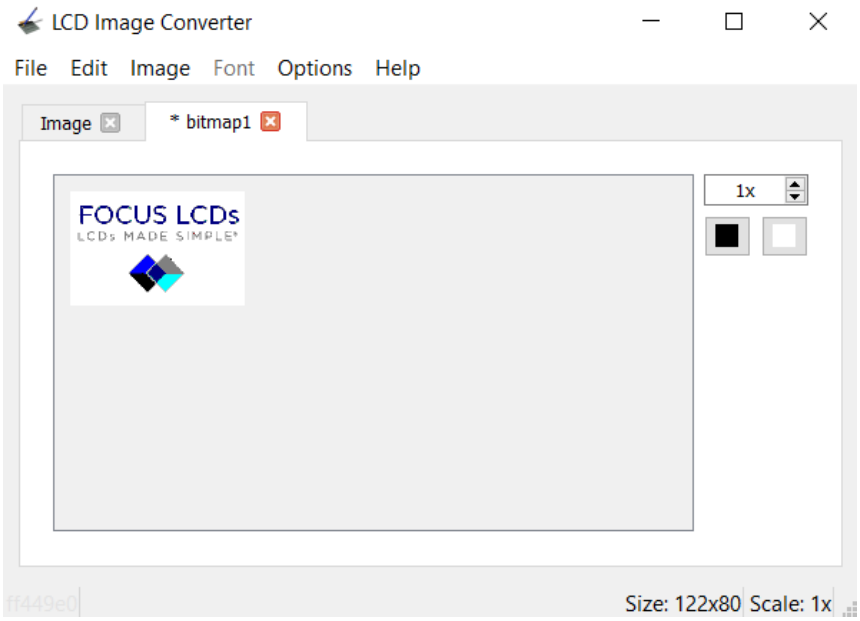


2.) Now that the board is available on the Arduino IDE it is time to program the display driver that in the TFT. I have prewritten files that specify the register setting and definitions for the ILI9163 LCD driver. You can download the example from Github. This library is reliant on the Adafruit GFX library so you will need to download and add these files to the Arduino IDE as well.

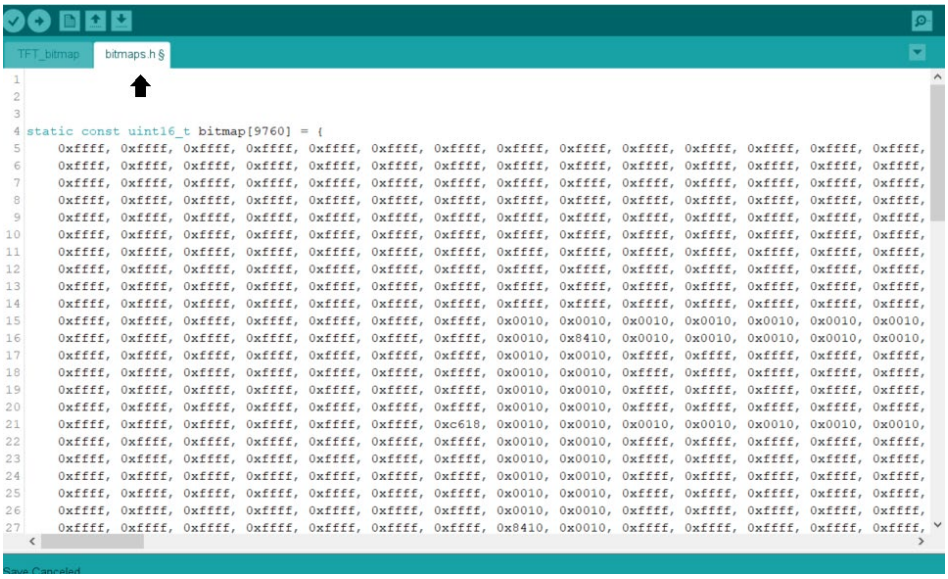
The example is to draw customs bitmaps on the display. To create the bitmap we will need an image that fits into a 128x160 pixel range. The bitmap in the example was created on Paint and saved as a 16-bit .bmp file because the display supports up to 65K colors.



Then we will convert the image into a readable Hexadecimal file. I used the LCD Image Converter application to do this. You can download this application and upload the bitmap image of your choice.

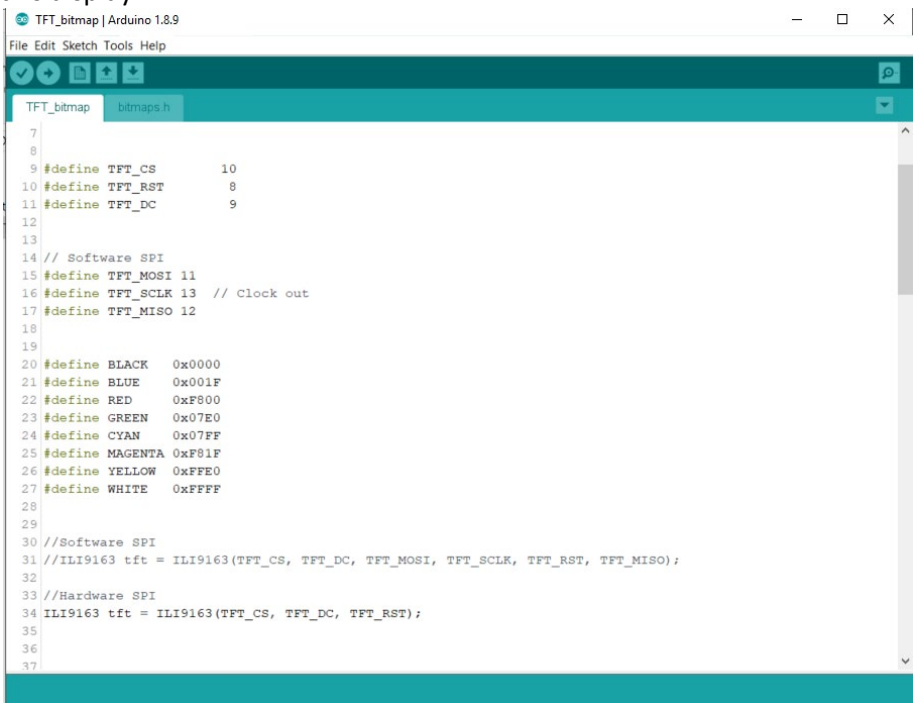


Verify that the conversion options are set to R5G6B5 color, Little-Endian and a block size of 16-bits for the 16-bit color image.



Then choose the convert option and the LCD Image Converter will create a .c file with the image information in hexadecimal. Since the image is only 128x160 pixels and up to 65K colors the file size is small. Open the file in a notepad or editor and then copy and paste the image information into the Arduino IDE .h file named “Bitmaps.h”.

3.) Now in the main program you can compile and run the code to upload the bitmap image. As you can see the bitmap only takes up a small amount of the microcontrollers flash memory storage. Through the same process additional bitmaps can be added as well as other functions that demonstrate the abilities of this display.



Hardware and software definitions are available. The wiring is set up so that the 4-wire SPI pins are at the correct location for the hardware definition. The hardware option is much faster than the software and is the recommended option. The example in the image above is using the software declaration of the data pins connected to the ST board. Verify that these values are correct as shown above.

Summary

This 1.8” TFT is a good option for displaying 16-bit 65K color images. This is compatible with most microcontrollers as it saves on-board memory. This is beneficial for storing bitmaps on flash memory because the screen is small and the 65K color bitmap image won’t take all the on-board storage. This display also has a version with a resistive touch screen. This would be a good option for a digital push button. This is discussed in Focus LCDs' application note FAN4206.

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.