# Application Note

## Interfacing a Color LCD Module to the EP72xx and EP73xx

### Introduction

As the world of PDAs and other hand-held devices evolves, more and more of these products need the support of color displays. Today, the use of color displays constitutes only a small market share. However, the desire for color support is growing dramatically. This application note is created to support this trend.

This application note describes how the LCD Controller integrated into the Cirrus Logic EP72xx embedded processor can be used to drive a color LCD module. This application note will first describe in detail how the EP72xx's LCD Controller can be used to support color displays. Next, it will provide an example of an application using a Sharp LM057QCTT03 ¼ VGA Color LCD module. Finally, it will provide a list of other color LCD modules that are known to be compatible with LCD Controller in the EP72xx.

### EP72xx LCD Controller Description

The LCD Controller provides all the necessary control signals to interface directly to a single-panel, multiplexed LCD module. The EP72xx uses the Universal Memory Architecture (UMA) for storing the video frame buffer. It shares the main memory bus with the core processor (i.e., the ARM720T). The total frame buffer size can be programmed up to 128 kbytes. The panel size is programmable and can basically support any panel size available, including the support for full VGA sizes. Any width (line length) from 32 to 1024 in 16-pixel increments can be programmed. The total number of lines (rows) supported is solely determined by the total frame buffer size programmed, divided by the panel width and color depth programmed.

The controller can also be programmed to provide 1-, 2-, or 4-bits-per-pixel color depth. The use of these bits is up to the application. They can be used to support a monochrome, gray-scale, or color display.

As an example: If a 1/2 VGA display is used, and 4 bpp is desired, the Video Buffer Size field in the LCD Control register (LCDCON) will have to be programmed to equal 640 x 240 x 4 = 614,400 bits = 76,800 bytes. The Line Length field in the register will have to be programmed to equal 640, and the color depth field will have to be programmed to equal 4. These three settings will result in the support for the 240 lines.

To support the various possible colors and gray-scale levels, the LCD Controller has two 32-bit palette registers. These palette registers are broken down into eight addressable nibbles each. This makes a total of sixteen nibbles. The nibbles are addressed by the data in the frame buffer. When the LCD Controller is configured to support 4 bpp, each four bits of data in the frame buffer is used to represent one pixel. Each of these nibbles addresses one of the sixteen nibbles in the two palette registers.

For example, if a nibble in the frame buffer contained the decimal value of ten, it would point to the tenth nibble in the two palette registers. This addressing scheme is used to map the data value in the frame buffer to the actual gray-scale level that will be supplied to the display interface. When 4 bpp mode is configured, all sixteen nibbles in the palettes registers are used for the mapping. This of course, is due to the fact that four bits can provide sixteen different values. When 2 bpp mode is configured, only the lowest addressable four nibbles are used. When in 1 bpp mode, only the lowest two nibbles are used.

Each palette register nibble can be programmed with a value from 0 to 15. These sixteen different values correspond to 16 different color depth levels. When the value programmed into each nibble matches its nibble

CIRRUS LOGIC®
P.O. Box 17847, Austin, Texas 78760
(512) 445 7222 FAX: (512) 445 7581
http://www.cirrus.com

address, the frame buffer data corresponds exactly with the gray-scale levels. This is the usual approach. It is called Direct Bit Mapping.

When the value programmed into any of the nibbles does not match its nibble address, then interesting patterns can be generated on the display. For example, if it is desirable to toggle the display's image between normal and reverse video, this can easily be achieved by simply inverting the values in the nibbles of the palette registers. This is much quicker and easier than having to invert all of the data in the frame buffer.

The LCD Controller also contains a nine-word deep FIFO. It is used as an intermediate storage buffer for the frame data. An integrated DMA controller is used to fetch display data from the frame buffer memory, and refill the FIFO. Thus, once the LCD Controller is configured and the frame buffer data is stored, the EP72xx can continue executing other tasks without having to service the LCD Controller.

*Note: This DMA controller is dedicated to servicing the LCD Controller. It cannot be used for any other purpose.*

## How Gray-Scaling Works

When a product has a LCD module that is providing different gray-scale levels, it isn't actually using a display that can be programmed to drive each pixel with a different level of intensity. The displays actually only have the ability to either fully turn "on" each pixel or turn it "off". One of the poor characteristics of LCDs in general is their response time. The response time can be defined as the amount of time it takes for a pixel to change from being turned "on" to "off", and vice versa. Typical response times are in the hundreds of milliseconds. The technique used for gray scaling takes advantage of this weakness. A modulating technique is used to drive each pixel. This technique drives each pixel a percentage of the time over a fixed time period.

Within the LCD Controller is a sixteen-cycle counter. It is used to create a sixteen-cycle period. When the LCD Controller needs to drive a pixel, it reads the value of the palette nibble pointed to by the frame buffer data. The value is used to determine how often within a sixteen-cycle period, the pixel needs to be turned "on". For example, if the value is 4, then the pixel would be turned "on" once every four clock cycles. This equates to 4/16 of the available sixteen-cycle period. By doing this, the naked eye thinks the pixel was actually being driven at ¼ the maximum brightness level.

In actuality, the nibble value does not correlate one-to-one with the number of times in the period that the pixel is turned "on". This is due to non-linear characteristics of the Response time. The EP72xx data sheets include a table that contains the actual gray-scale value mappings. The table is called the "Gray-scale Value to Color Mapping". It can be found in the Palette Register section.

## Color Support

When used with a so-called monochrome LCD module, it provides support for up to 15 different gray-scale levels.

*Note: Fifteen is the correct number, not 16. This is due to the fact that the middle two gray-scale levels are identical in contrast, and thus only provide one level instead of two.*

Color displays have three sub-pixels per pixel. Each pixel is made-up of three sub-color pixels (Red, Green, and Blue). The same technique described above for gray-scaling is applied to color displays. Hence, each sub-color pixel can be modulated to provide the perceived affect of supporting 15 different shades of its color. Therefore, one pixel can be driven with one of fifteen possible shades of red, green, and blue.

The LCD Controller can support up to 15 x 15 x 15 = 3,375 different colors. This is based upon the assumption that the LCD Controller has been programmed to provide 4 bpp. Now each nibble in the frame buffer represents one of the sub-pixels in the display. When programmed to provide 2 bpp, $2^2 \times 2^2 \times 2^2 = 64$ different colors can be supported. When programmed to provide 1 bpp, $2^1 \times 2^1 \times 2^1 = 8$ different colors can be supported.o support a ¼ VGA color display, the actual number of pixels triples due to the three sub-pixels. This equates to 320 (x3) x 240 = 230,400 pixels. If programmed to 4 bpp to support the maximum number of colors, the frame buffer size becomes 230,400 x 4 = 921,600 bits, or 115,200 bytes in size. This falls within the maximum size constraints of the frame buffer (i.e., 128 kbytes) mentioned above.

Another characteristic of displays is their Refresh Rate. The Refresh Rate is a number that states how frequently the entire frame of data can be rewritten to the display. If the data is written too slowly, the display's Response time will affect the quality of its appearance. Too fast, and the display's Response time will not be able to keep up with the changing pixel drive states. Most displays today recommend rates around 70 – 80 Hz.

The LCD Controller has a fixed-clock source, which is used to generate the frequency that the pixels are written to the display. When the EP72xx's PLL is used to create its clocks, the LCD Controller is provided with a 36.864 MHz clock. When the EP72xx is supplied a 13 MHz external clock, the LCD Controller is supplied with a 13 MHz clock. These fixed clock sources place limits on the maximum rate that the pixels can be written to the display. Thus there is a limit on the maximum Refresh Rate.

The EP72xx device has a field in the LCD Control Register (LCDCON) called the "Pixel Prescale". The Pixel Prescale field is a 6-bit field that sets the pixel rate prescale. When the EP72xx PLL is used to create its clocks, the pixel rate is derived from Equation 1.

Equation 1. Pixel Rate (MHz) = 36.864/(Pixel prescale + 1)

The Pixel Prescale value can be expressed in terms of the LCD size by Equation 2.

Equation 2. Pixel Prescale = (36864000/(Refresh Rate x Total pixels in display)) – 1

The value should be rounded down to the nearest whole number and zero is illegal and will result in no pixel clock.

For a ¼ VGA color display, the maximum Refresh Rate can be calculated by using the equations below. Rearranging Equation 2 to solve for the maximum Refresh Rate results in Equation 3.

Equation 3. Refresh Rate = 36864000/ (Total pixels in display) x (Pixel Prescale + 1)

Since, the minimum Pixel Prescale value is 1, we use this value in Equation 3 to calculate the maximum Refresh Rate, as shown in Equation 4.

Equation 4. Maximum Refresh Rate = 36864000/((320 (x3) x 240) x (1 + 1)) = 80 Hz

The LCD Controller can provide a maximum Refresh Rate of 80 Hz, for a ¼ VGA color LCD display. This is perfect for most displays.

If color is desired, the maximum recommended display size is a ¼ VGA color LCD module. This is solely due to their minimum required Refresh Rate. If a 1/2 VGA color display size is desired, the below restrictions must be taken into consideration:

1). The LCD Controller can only support a 320 x 480 configuration, not a 640 x 240. If a color display with a 640 x 240 configuration were chosen, it would require 640 x 3 = 1920 pixels/line, which is larger than the maximum programmable line width of 1024.

2). The LCD Controller can only provide a maximum Refresh Rate of 40 Hz (per Equation 3). This is too slow for most displays.

If a monochrome display is chosen, the LCD Controller can support a full VGA display with a Refresh Rate of 60 Hz; along with support for up to 2 bpp, or 4 gray-scale levels.

*Note: The Refresh Rate is completely independent of the number of bits per pixel selected.*

## The LCD Module Interface Port

The LCD module interface port built into the EP72xx device, contains the following signals:

DD[0:3]

FRM

CL1

CL2

M

DD[0:3] are the four data lines. When the LCD Controller writes out this port, it presents four pixels at a time using these lines. Each data line is either HIGH or LOW. Thus each pixel value is either HIGH or LOW.

FRM is the frame sync signal. It toggles HIGH after all of the pixel data for a frame has been completely written out the interface. It is used by the display to force it to reset its line (row) counter back to zero. Thus, the display will start driving the next nibble of data to the first line of the display.

CL1 is the line strobe signal. It toggles HIGH after all of the pixel data for a line has been written out the interface.

CL2 is the pixel data clock. It is used by the display to clock in each nibble of pixel data. Its period is ¼ the actual pixel rate. When CL1 toggles, CL2 stays LOW. Thus, CL2 LOW time is doubled when CL1 toggles at the end of each line.

M is the AC Bias signal. This signal is used by the display to tell it when the drive voltage to the display should be reversed. If used, it becomes active HIGH for a programmed quantity of CL2 cycles during each CL1 cycle. This is done periodically to minimize any DC voltage bias that may build-up across the display. DC voltage build-up is undesirable, since it can damage the display. The value for M is based upon the exact display being used. Therefore, the value must be obtained from the display's datasheet.

## Example Color LCD MOdule Interface

The schematic diagram of Figure 1 depicts one example solution of interfacing the EP72xx LCD Controller to a Sharp LM057QCTT03 ¼ VGA Color LCD module. This display has an 8-bit data interface. The sole purpose of this logic is to convert the 4-bit interface into an 8-bit. It creates an 8-bit interface out of two 4-bit nibbles. This logic has no affect on the programming of the LCD Controller registers. The LCD Controller will provide the same Refresh Rate and pixel color depth.

The left side of the schematic has all the input signals from the LCD Controller. The right side has all the output signals that connect directly to the display.

Since the data provided to the display is made up of two sets of data that is output from the EP72xx, the clock supporting this 8-bit data word, must be half the rate as the original. This means that CL2 from the EP72xx must be halved. This is accomplished by use of the D flip-flop configured to toggle its output every time its clock toggles HIGH. By using EP72xx_CL2 as the input clock, the output toggles at ½ the clock rate, and thus becomes the desired LCD_CL2.

EP72xx_CL1 is routed directly to the display. It is also used to reset the D flip-flop so that the signal LCD_CL2 starts off in the LOW State.

The '174 register is used to store the lower half of the 8-bit data word. When the upper nibble is available, both nibbles are provided to the display together.

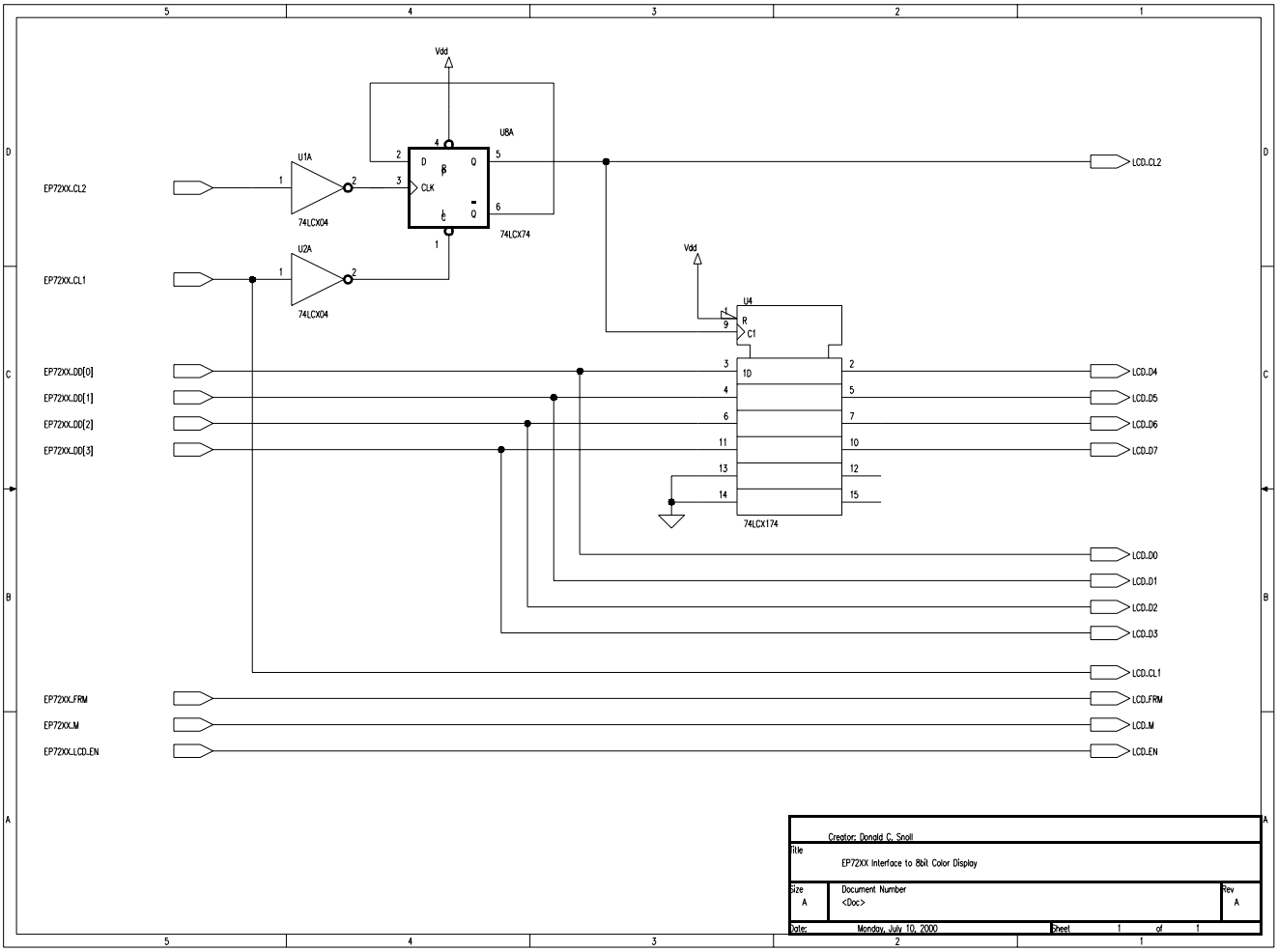


**Figure 1. EP72xx Interface to 8-Bit Color Display**

## *Known Compatible LCD Modules*

Table A gives a list of LCD Modules that are known to be compatible with LCD Controller integrated into the EP72xx device.

*Note: The list in Table A is based solely upon work that was done to create this Application Note. Many other display modules are compatible as well, including display sizes up to full VGA. However, if a full VGA display module is desired, the color depth will be restricted due to the limits on the maximum frame buffer size and the maximum pixel rate (i.e., Refresh Rate) supported.*

**Table A. Compatible LCD Modules**

| LCD Vendor and Model # | LCD Characteristics | Availability |
|---|---|---|
| Sharp LM057QC1T01 | 8-bit, 6.1", 320 x 240, Color display, no touchscreen | Now |
| Sharp LM057QCTT03 | 8-bit, 6.1", 320 x 240, Color Transmissive, CCTF backlight, Touchscreen | Now |
| Sharp LM038QC1R10 | 8-bit, 3.8", 320 x 240, Color Reflective, no backlight, no touchscreen | Now |
| Sharp LM038QC1S10 | 8-bit, 3.8", 320 x 240, Color Tranflective, CCFT backlight, no touchscreen | Now |

**Table A. Compatible LCD Modules**

| | | |
|---|---|---|
| Sharp LM038QC1TS10 | 8-bit, 3.8", 320 x 240, Color Transflective, CCFT backlight, touchscreen | Now |
| Sharp LM038QB1R10 | 4-bit, 3.8", 320 x 240, Monochrome reflective, no backlight, no touchscreen | Now |
| Sharp LM038QB1S10 | 4-bit, 3.8", 320 x 240, Monochrome transflective, EL backlight, no touchscreen | Now |
| Sharp LM038QBTS10 | 4-bit, 3.8", 320 x 240, Monochrome transflective, EL backlight, touchscreen | Now |
| ALPS LRHDD1014A | 4-bit, 6.8", 640 x 240, Monochrome display, Touchscreen | Now |
| ALPS LRHDD1013A | 4-bit, 6.8", 640 x 240, Monochrome display, Touchscreen | Now |
| ALPS LRH7U504XA | 4-bit, 2.9", 320 x 240, Monochrome display, Touchscreen | Now |
| ALPS LFH8P402XA | 4-bit, 2.8", 320 x 240, Monochrome display, Touchscreen | Now |

# • Notes •


Maverick™
from
CIRRUS LOGIC