



Introduction to graphics and LCD technologies

NXP Product Line Microcontrollers
Business Line Standard ICs

Agenda

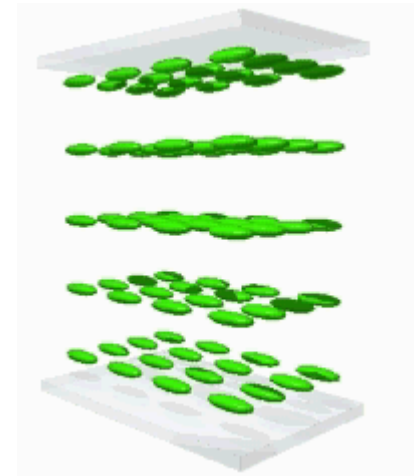
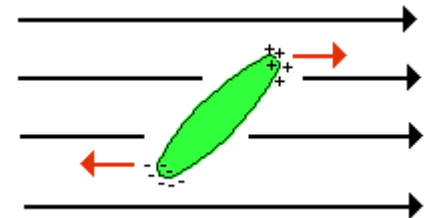
- ▶ Passive and active LCD technologies
 - How LCDs work, STN and TFT differences
 - How data is converted to colors on the LCD
- ▶ LCD signal interface and timing parameters
 - LCD signals and timing
 - Controlling the backlight
- ▶ Introduction to frame buffers with the LPC32x0 MCU
 - How graphics data is stored in memory
 - Color depth and lookup tables
- ▶ System considerations for LCD based systems
 - Mapping LCD data signals to the LCD controller signals
 - LCD data bandwidth
- ▶ Examples



Passive and active LCD technologies

How an LCD works

- ▶ An array of Liquid Crystal segments
 - When not in an electrical field, crystals are organized in a random pattern
 - When an electric field is applied, the crystals align to the field
 - The crystals themselves do not emit light, but 'gate' the amount of light that can pass through them
 - Crystals aligned perpendicular to a light source will prevent light from passing through them
- ▶ Each LCD segment is aligned with an electric field
- ▶ A light source (backlight) is needed to drive light through the aligned crystal field



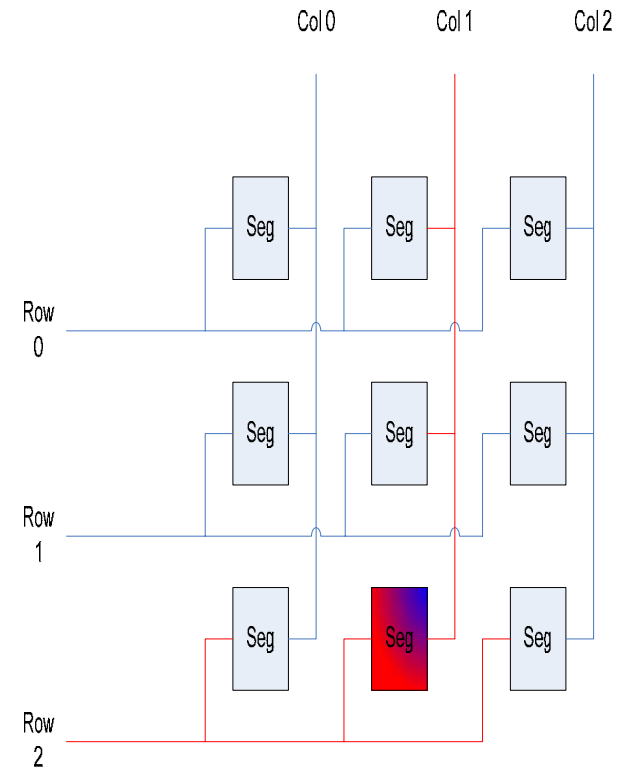
Courtesy of Sharp

Passive displays

▶ Passive LCD panels

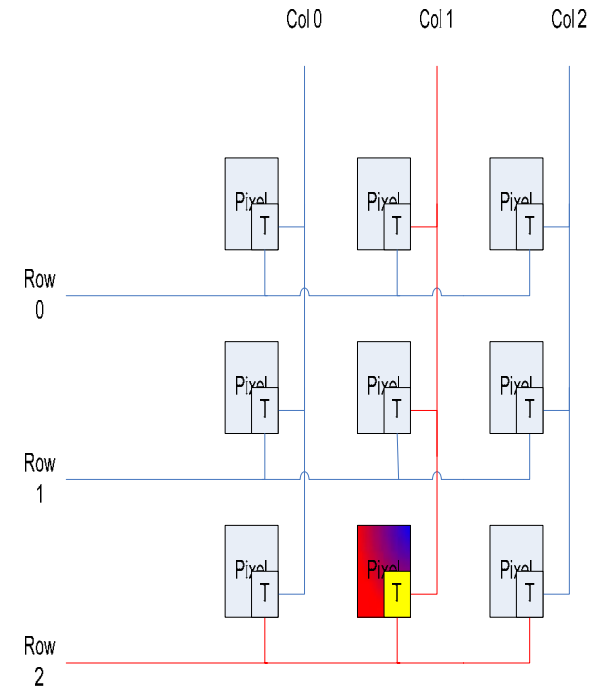
- Consists of a grid of row and columns electrical signals
- Columns and rows connect perpendicularly to every segment in the LCD
 - Columns and rows are multiplexed to many different segments
- An IC controls which column and row are selected to enable or disable the segment at the row/column intersection
- A small bias is applied to the row and column to generate a field at the intersection
 - No charge is stored at the segment
 - It may take multiple passes to correctly align the field to the desired value

▶ STN LCDs are passive displays



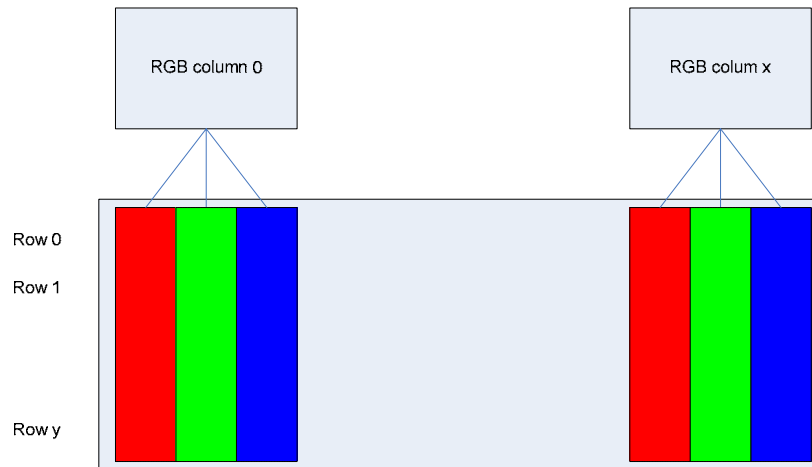
Active displays

- ▶ Active LCD panels
 - Consists of a grid of row and columns electrical signals
 - Columns and rows connect perpendicularly to a active device (transistor) for every segment in the LCD
 - Columns and rows are multiplexed to many different segments
 - An IC controls which column and row are selected to enable or disable the segment at the row/column intersection
 - The selected row and column enable the transistor
 - Charge is stored at the transistor
 - One pass will set the aligned state of the transistor (although it may still take a little time for all the crystals to align)
 - A stronger backlight is needed than a passive display
- ▶ TFT displays are active displays



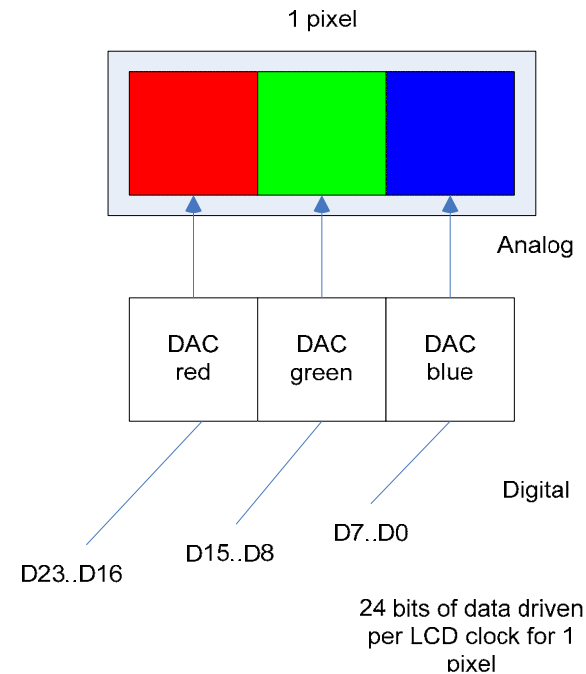
LCD panel Technologies – making colors

- ▶ Each LCD segment only gates reflected or generated light
 - Color filters allow generation of specific colors (RGB) at a segment
 - To generate a real world color, 3 segments are needed – these 3 segments individually pass light through a red, green, and blue filter to make a group of segments, or a RGB pixel
 - For a 320x240 RGB LCD display, there are actually $320 \times 3 = 960$ segments (columns) and 240 rows



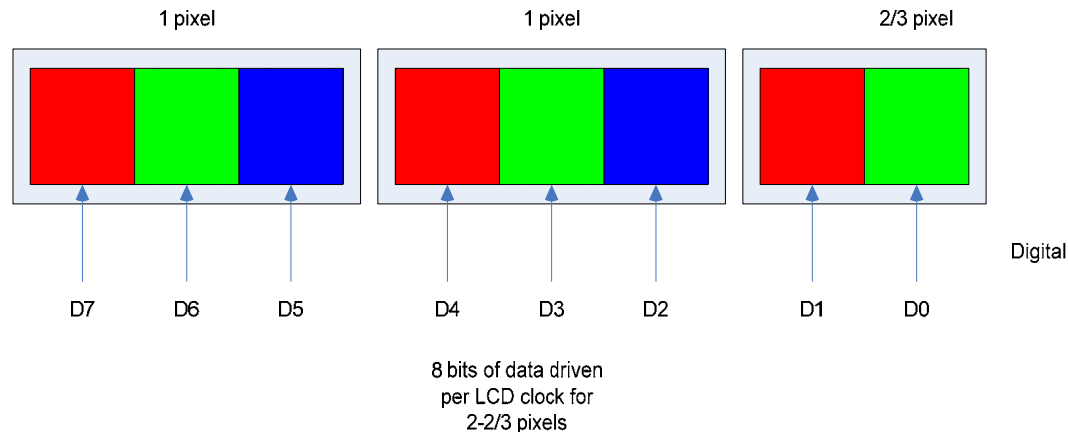
Generating color on an TFT display

- ▶ TFT displays can drive 3 segments (1 pixel) per clock with variable electric field strength
 - Supports many colors
 - Always 1 pixel per clock (3 segments of Red, green, and blue)
 - Color levels depend on the number of data lines on the LCD panel and number of LCD controller data output signals
 - May be 24 lines - 24bits per pixel (bpp)
 - 18bpp, 16bpp, 15bpp, 8bpp
 - Parallel data interface
 - 320 clocks require to place 320 pixels



Generating color on an STN display

- ▶ STN displays drive 1 or more segments per clock (full field strength on or off)
 - Can drive fractional pixels per clock
 - Serial interface
 - 120 clocks required to drive 320 pixels @ 2-2/3 pixels per clock (8 bit data bus)
 - 240 clocks required to drive 320 pixels @ 1-1/2 pixel per clock (4 bit data bus)
- ▶ Segments are alternated between on and off states to generate color depth
 - May take multiple refresh cycles to get the LCD color to a desired value (slow to respond due to a maximum of 1 digital state change per refresh cycle)
 - For example, a 50% duty cycle on a segment will give about 50% brightness



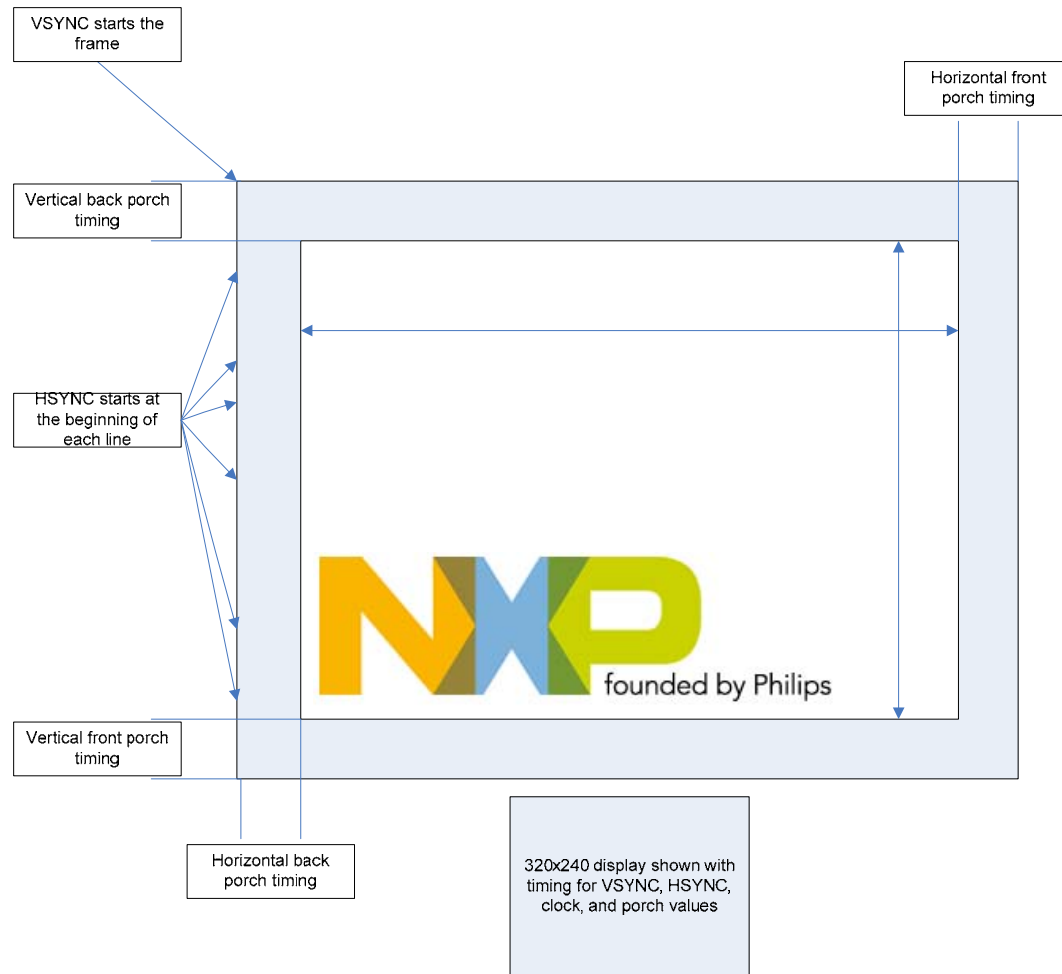


LCD signal interface and timing parameters

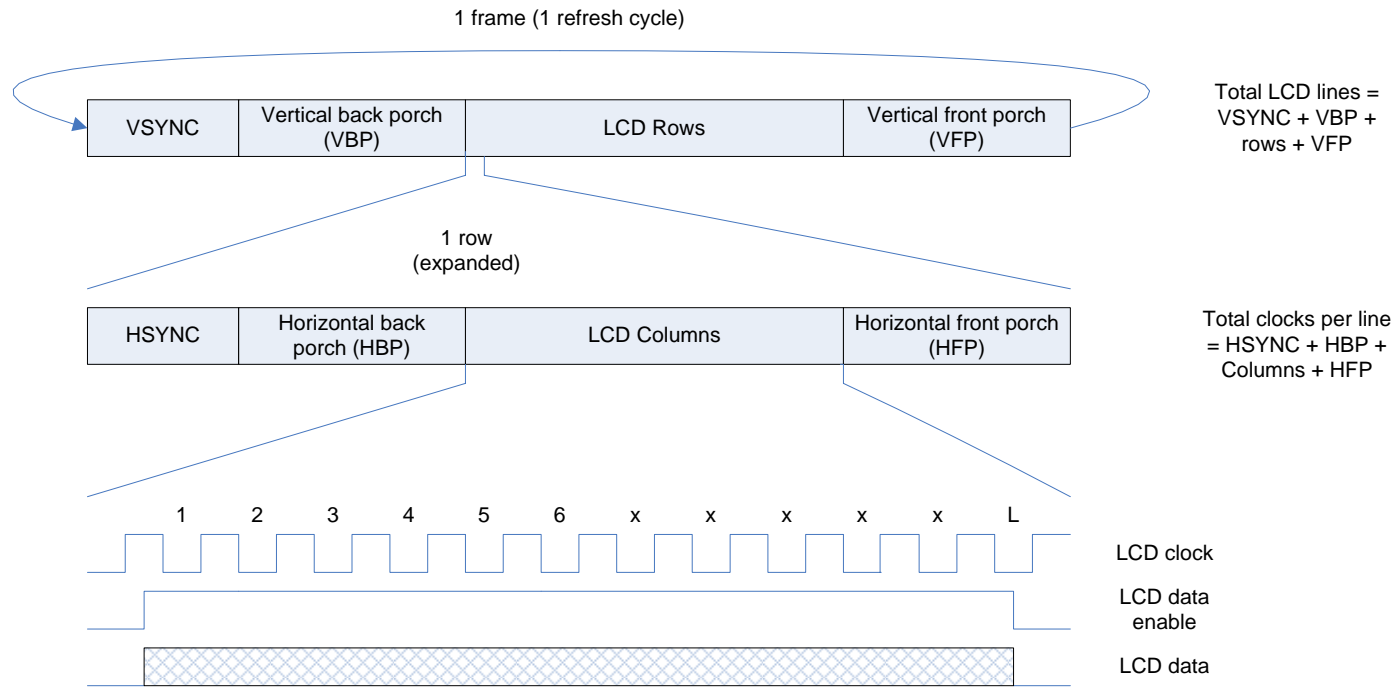
LCD signals and timing

- ▶ LCDs require the following basic timing signals:
 - VSYNC (Vertical Sync for TFT) or FP (Frame Pulse for STN)
 - Used to reset the LCD row pointer to top of the display
 - HSYNC (Horizontal sync for TFT) or LP (Line Pulse for STN)
 - Used to reset the LCD column pointer to the edge of the display
 - D0..dXX (1 or more data lines)
 - Data line function varies in STN and TFT modes and panel type
 - LCDCLK (LCD clock)
 - Used to panel control refresh rate
- ▶ Some panels may require additional timing signals:
 - STN panels usually require MDISP (AC bias)
 - Used for AC bias (to prevent panel damage)
 - LCDDATAENAB
 - Used to indicate valid data on the LCD data bus
- ▶ Other signals – some optional
 - LCD power, backlight power, touchscreen

LCD bus timing parameters



Frame and line timing parameters



$$\text{Total clocks per frame} = \text{Total LCD lines} * \text{Total clocks per line}$$

LQ043 timing (TFT) parameters example

7-1 Timing characteristics

Parameter		Symbol	Min.	Typ.	Max.	Unit	Remark
Clock	Frequency	1/Tc	7.83	9.00	9.26	MHz	
	Duty ratio	Th/T	40	50	60	%	
Data	Set up time	Tds	25	—	—	ns	
	Hold time	Tdh	25	—	—	ns	
Horizontal synchronizing	Period	TH	—	525	—	Clock	
	Pulse width	THp	—	41	—	Clock	
	Horizontal period	THd	—	480	—	Clock	
	Back porch	THb	—	2	—	Clock	
	Front porch	THf	—	2	—	Clock	
Vertical synchronizing	Period	TV	—	286	—	Line	
	Pulse width	TVp	—	10	—	Line	
	Vertical period	TVd	—	272	—	Line	
	Back porch	TVb	—	2	—	Line	
	Front porch	TVf	—	2	—	Line	

LCD software timing data structure

```
/* Sharp LQ043 display parameters */
```

```
const LCD_PARAM_T sharp_lq043 =
```

```
{
```

```
    2, /* Horizontal back porch */
```

```
    2, /* Horizontal front porch */
```

```
    41, /* HSYNC pulse width */
```

```
    480, /* Pixels per line */
```

```
    2, /* Vertical back porch */
```

```
    2, /* Vertical front porch */
```

```
    10, /* VSYNC pulse width */
```

```
    272, /* Lines per panel */
```

```
    0, /* Do not invert output enable */
```

```
    0, /* Do not invert panel clock */
```

```
    0, /* Do not invert HSYNC */
```

```
    0, /* Do not invert VSYNC */
```

```
    0, /* AC bias frequency (not used) */
```

```
    18, /* Bits per pixel */
```

```
    9000000, /* Optimal clock rate (Hz) */
```

```
    TFT /* LCD panel type */
```

```
};
```

```
/* LCD parameters can be abstracted to a common software structure that can easily configure a driver for the LCD panel! */
```

LQ043 timing parameters example

- ▶ Based on the previous parameters from the LQ043 datasheet
 - Vertical timings
 - VFP = 2 lines, VBP = 2 lines
 - VSYNC width = 10 lines
 - Vertical period = 272 lines (vertical resolution = 272 pixels)
 - Horizontal timings
 - HFP = 2 clocks, HBP = 2 clocks
 - HSYNC width = 41 clocks
 - Horizontal period = 480 clocks (horizontal resolution = 480 pixels)
 - Clock speed = 7.83MHz to 9.26MHz
- ▶ From this data,
 - A single line takes $(2 + 2 + 41 + 480)$ clocks = 525 clocks/line
 - A full frame takes $(2 + 2 + 10 + 272)$ lines = 286 lines/frame
 - A full frame in clocks = $286 * 525 = 150150$ clocks/frame
 - At 7.83MHz, the LCD would refresh at $7.83\text{M}/150.15\text{K} = 52.1\text{Hz}$
 - At 9.26MHz, the LCD would refresh at $9.26\text{M}/150.15\text{K} = 61.7\text{Hz}$

LCD backlights

- ▶ LCD backlights are not controlled through the LCD controller
 - Different backlight technology based on panel size
 - Large panels may use fluorescent lights
 - Smaller panels may use LEDs
 - Usually controlled by a constant current source
 - Digital signal for on/off control
 - PWMs for variable intensity
 - LCD diffuser keeps backlight brightness fairly uniform
- ▶ Most of an LCD's power usage is from the backlight
 - Small QVGA LED based panels may use 300mW or more

[NESW008B]

Parameter	Symbol	Min.	Typ.	Max.	Unit	Remark
Rated Voltage	V_{BL}	—	—	29.4	V	
Rated Current	I_L	—	18	20	mA	Ta=25°C



Introduction to frame buffers with the LPC32x0 MCU

Frame buffers

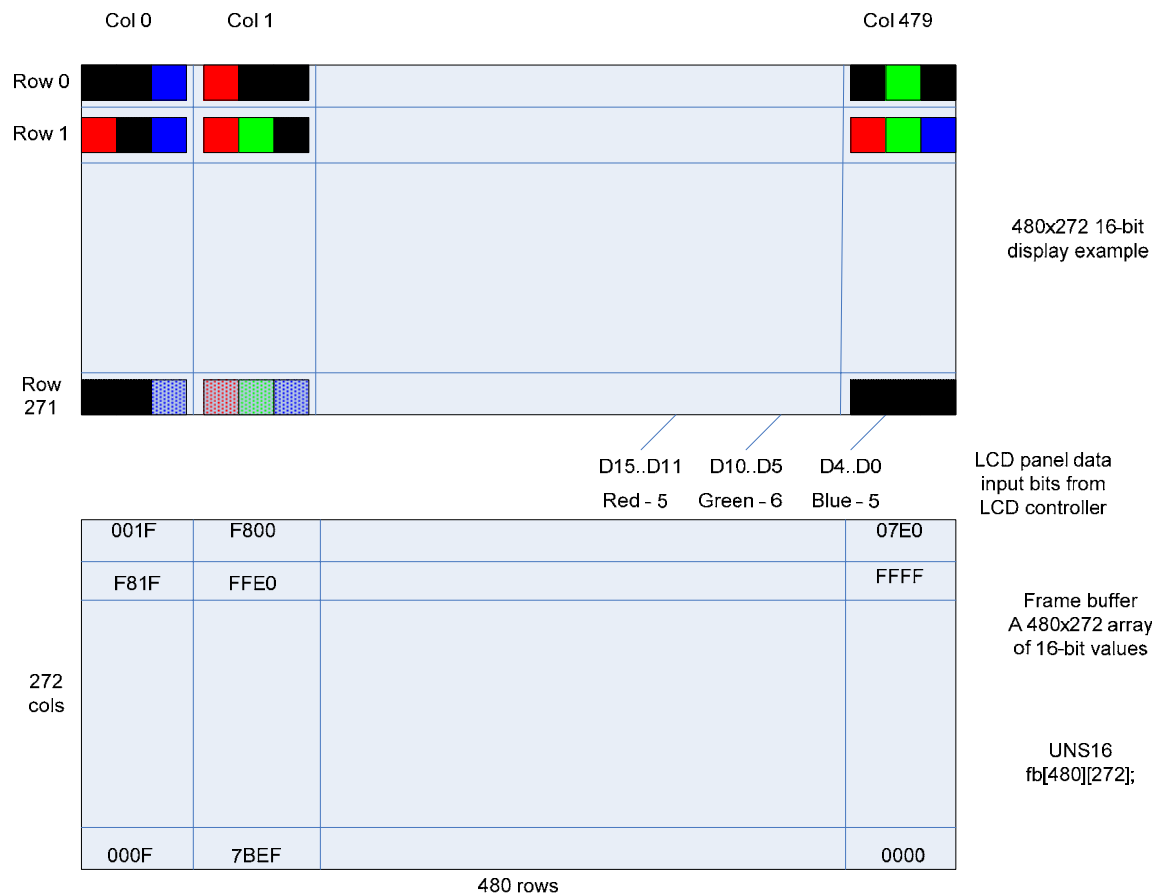
▶ What is a frame buffer?

- Memory allocated for data used to periodically refresh the display
 - Memory allocated to the frame buffer is usually shared with other system devices (CPU core, DMA, network, etc.)
 - Organized as an array of bits, bytes, half-words, or words, depending on the selected color depth and color bit organization
 - Pixel data may be packed bits (8 pixels/byte), bytes (up to 256 colors), 16-bit half-words (up to 64KColors), or 24-bit words (up to 16 million Colors)
- Buffer size is computed using (columns * rows * sizeof(pixel data))
 - Example : An 800x600 display @16bpp (half-word RGB565) = 800 columns*600 rows*2 bytes/pixel = 960000 bytes
 - 24-bit data is stored in a 32-bit field (tossing the high byte for each pixel)

Color patterns

- ▶ Digital systems usually stored data in RGB (red-green-blue) colorspace format
 - Red, green, and blue components are bitfields of a pixel's color value
 - Usually referred to as bits per pixel (bpp)
 - RGB332 → 8 bpp (Red 3, Green 3, Blue 2)
 - Organized as a byte in memory as (RRR GGG BB)
 - RGB555 → 16 bpp (Red 5, Green 5, Blue 5)
 - Organized as a half-word in memory (U RRRRRR GGGGGG BBBB)
 - RGB565 → 16 bpp (Red 5, Green 6, Blue 5)
 - Organized as a half-word in memory (RRRRR GGGGGG BBBB)
 - RGB888 → 24 bpp (Red 8, Green 8, Blue 8)
 - Organized as a word (32-bit) in memory
 - (UUUUUUUU RRRRRRRR GGGGGGGG BBBBBBBB) (U = unused)

Frame buffer data to LCD mapping example (16 bpp)

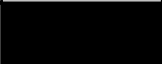

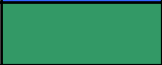







Color depth and lookup tables

- ▶ The frame buffer color depth and LCD color depth do not need to be the same
 - A lookup table can be used to map a small subset to values to LCD color values
 - For example, controller can be configured for 4bpp while the LCD interface is 16bpp
 - Allows a selection of 16 colors from a possible 64K colors
 - Colors can be dynamically changed by adjusting the lookup table
 - Allows a tradeoff for color depth and system bandwidth
 - 4bpp uses $\frac{1}{4}$ the bandwidth of 16bpp
 - Color depths of 8bpp or less require the use of a lookup table
 - 8bpp requires 256 lookup entries, 4bpp requires 16, etc.
 - Color index [0] uses 16-bit lookup table index [0] for the LCD
 - Colors can be mapped many different ways

Example lookup table for mapping 8-color RGB to RGB565

- ▶ This example table assumes that red, green, and blue can only be turned off or turned to full intensity
 - 3bpp colors = 8 colors → 16-bit RGB565 output

Frame buffer value	Lookup table value (16-bit RGB565 output)	System color	
R0, G0, B0 (0)	0x0000	black	
R0, G0, B1 (1)	0x001F	blue	
R0, G1, B0 (2)	0x07E0	green	
R0, G1, B1 (3)	0x07FF	cyan	
R1, G0, B0 (4)	0xF800	red	
R1, G0, B1 (5)	0xF81F	magenta	
R1, G1, B0 (6)	0xFFE0	yellow	
R1, G1, B1 (7)	0xFFFF	white	

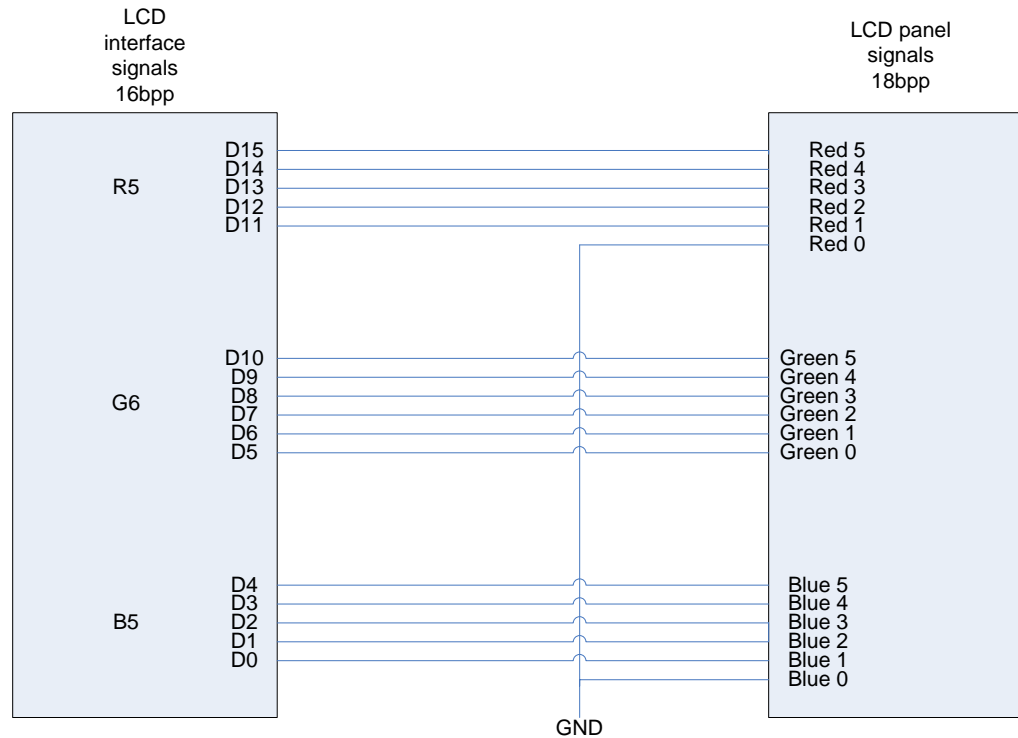


System considerations for LCD based systems

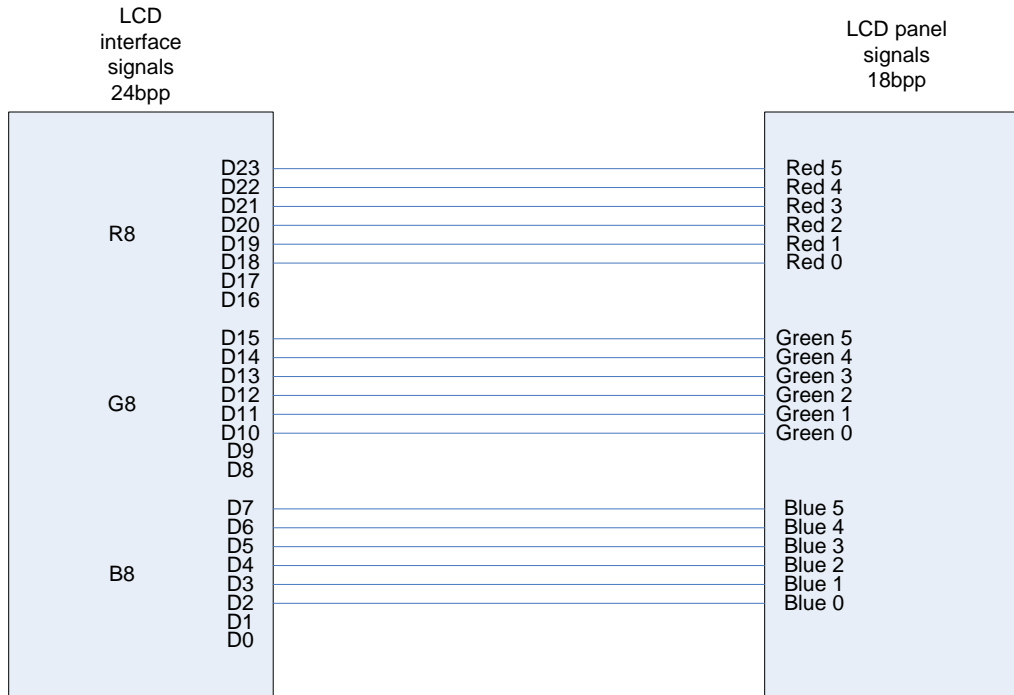
Mapping LCD data signals to the LCD controller signals

- ▶ LCD data signals don't always map directly to LCD interface signals (applies to TFT panels)
 - Example: The LCD controller only supports 16bpp and 24bpp output, but the LCD panel is 18bpp (256KColors)
- ▶ LCD signals in the datasheet usually use the Redx, Bluex, and Greenx naming convention, while the LCD interfaces uses Dx..D0
- ▶ Many options for connecting the LCD
 - Can ground unused LCD signals (usually the lower weighted bits) for interfaces with less signals than the LCD
 - Can keep LCD interface signals unused for interfaces with more signals than the LCD
 - Can also use non-standard color mappings
 - Such as RGB484

16-bit RGB565 interface to 18bpp LCD panel TFT example



24-bit RGB888 interface to 18bpp LCD panel TFT example



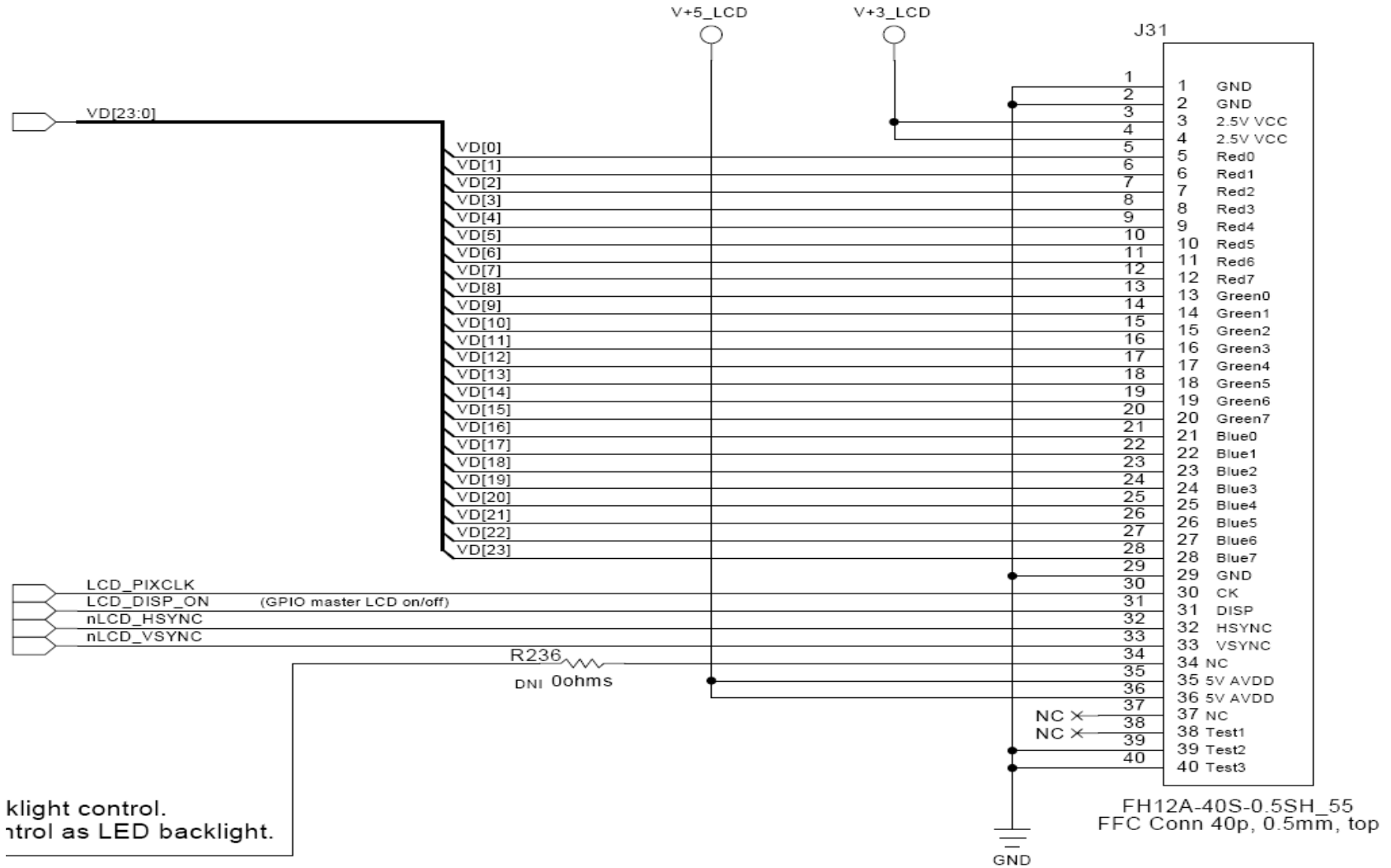
LCD data bandwidth

- ▶ LCDs need to be refreshed periodically from the frame buffer to keep their image
 - This refresh cycle usually occurs automatically in the background through DMA
 - In a shared memory system such as the LPC32x0, LCD bandwidth consumes a portion of the available system memory bandwidth
 - The refresh rate is derived from the pixel clock rate and panel parameters
 - Total memory bandwidth required to keep a panel refreshed is a product of the frame buffer size times the refresh rate
 - For a 640x480 pixel display using 16-bit color and being refreshed at 60Hz
 - Bytes/sec for LCD = $60 * 640 * 480 * 2 \text{ bytes/pixel} = 70.3125 \text{ MBytes/sec}$
 - Bandwidth used to keep the LCD refreshed will not be available for system use
 - Very large displays can slow down the CPU by delaying accesses to memory during refresh cycles
 - Performance impact can be minimized with features such as large CPU caches, layered (multiple) bus architecture, or effective use of internal memory



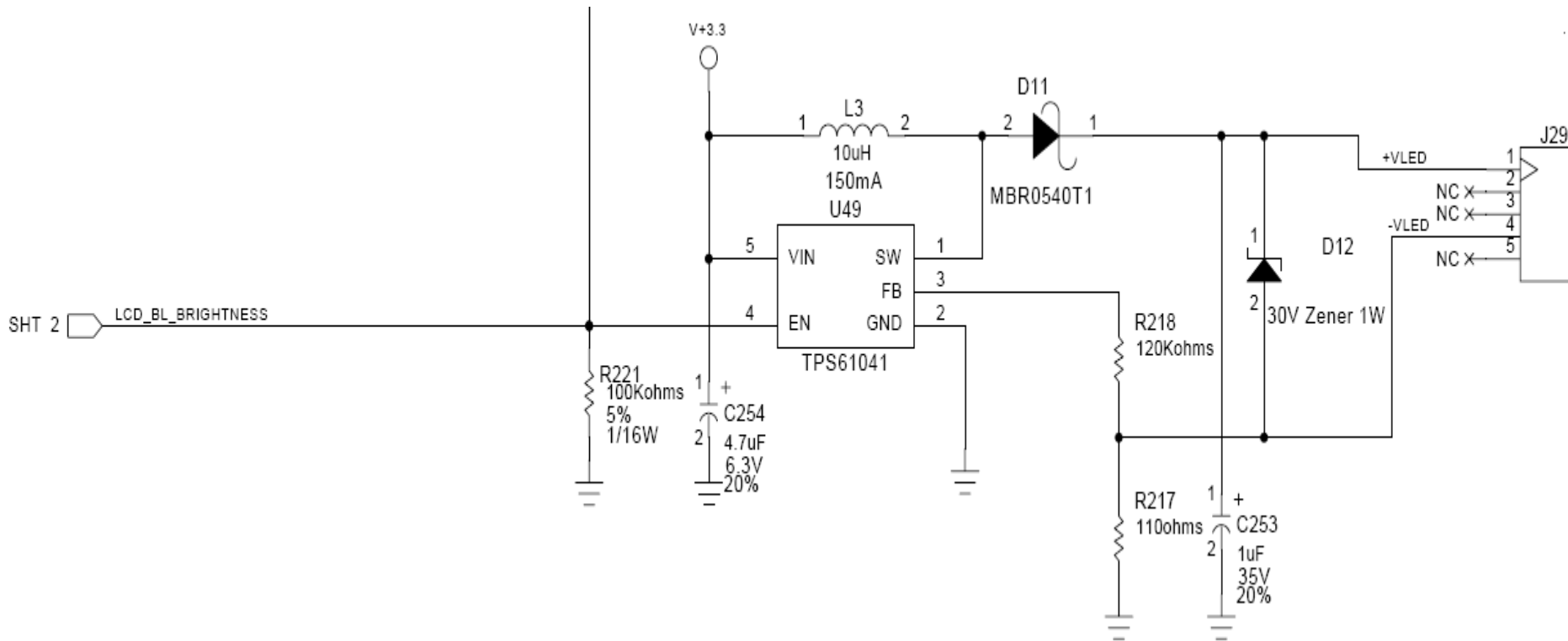
Examples

LQ043 TFT signal connections (24bpp)



LQ043 TFT PWM backlight power control

Variable brightness



Drawing a box on the LCD screen

- ▶ The following C code will draw a box on the display
 - The display size is 320x240 pixels and is RGB565 format
 - The box will be drawn from (100,50) to (220, 190)
 - The box top is red, the bottom is blue, and the sides green

```
void putpixel(int x, int y, unsigned short color) {  
    unsigned short *fb = fblog;  
    *(fb + (y * 240) + x) = color;  
}  
  
void drawbox(void) {  
    int x, y;  
    for (x = 40; x <= 200; x++) {putpixel(x, 30, 0xF800);putpixel(x, 290, 0x001F);}  
    for (y = 30; y <= 290; y++) {putpixel(40, y, 0x7E0); putpixel(200, y, 0x07E0);}  
}
```