AT91SAM9 LCD Controller

1. Scope

This application note explains how to connect the Atmel® ARM® Thumb®-based AT91SAM9 LCD Controller to an STN or TFT display. It includes display selection guidelines, describes the hardware and software configurations and gives performance results and software examples.

AT91 ARM Thumb Microcontrollers

Application Note

6300A–ATARM–09-Jul-07

2. Display Overview

2.1 TFT and STN Technologies

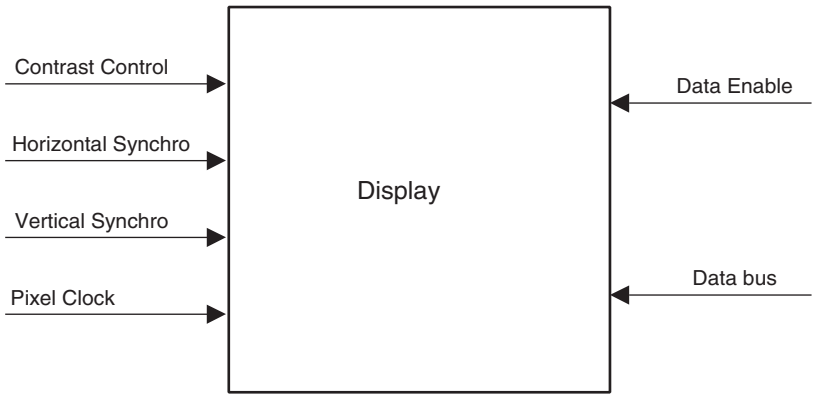
The most commonly used display technologies are TFT (Thin Film Transistor) and STN (Super Twisted Nematic).

STN displays use a passive matrix screen technology that has no active or controlling element inside the display cell. Pixels are controlled by energizing the appropriate row and column drive lines of the matrix from outside the display, resulting in a slow frame rate. STN screens have limited color range and viewing angle (~15 degrees max).

With TFT LCDs, each pixel is controlled by one to four transistors. Typically, one transistor is used for each of the RGB color channels. Because of this direct control technique, TFT screens are sometimes called Active-Matrix LCDs. TFT technology provides more accurate color control, allowing it to display more colors. TFT screens also offer a wider viewing angle range (30 to 70 degrees) than the other popular types of LCDs.

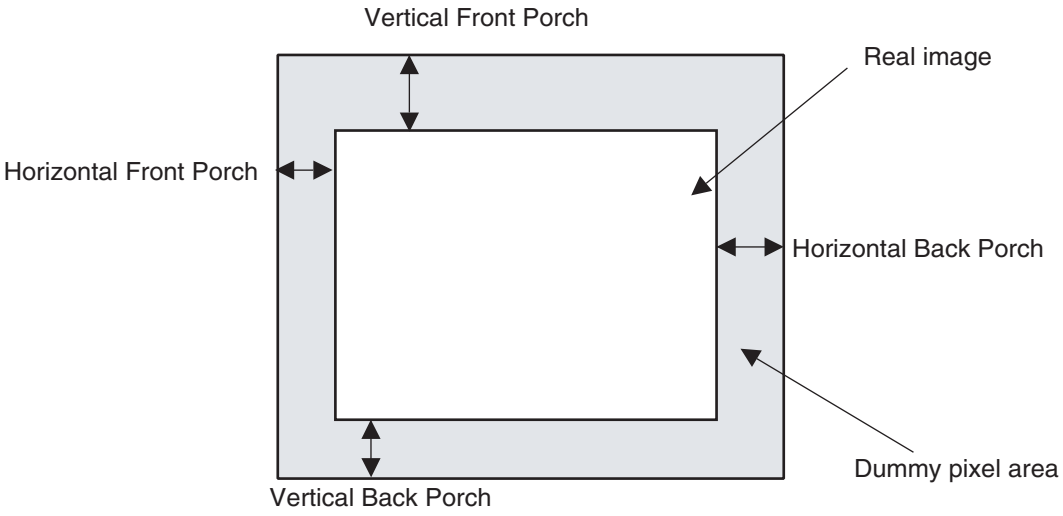
2.2 Interface

The typical interface of a display is based on analog and digital inputs. The digital lines are a data bus, a pixel clock, vertical and horizontal synchronization signals and a data enable line. The voltage input is generally used for contrast control. The LCD Controller drives all these interface lines. If more inputs are required by the display, it may be managed by PIOs (power control, AC bias on some STN displays, backlight control, etc.)

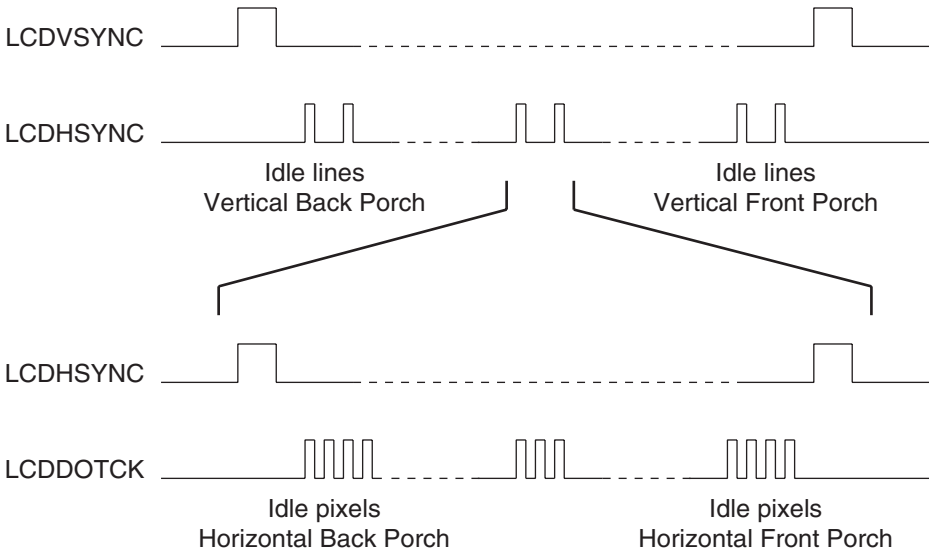


2.3 Blanking

For the internal synchronization mechanism, the display may need some dummy data at the beginning and/or end of a line, and at the beginning and/or end of a frame. This is called blanking.



The dummy pixels/lines are not part of the frame buffer; they are managed by the LCD Controller. Some delays must be introduced in vertical and horizontal timings to support it. These delays are often described as vertical/horizontal front/back porch delays in the display datasheets.



2.4 Definitions

The following definitions and clarifications are necessary when using the LCD Controller.

- Bits per pixel (Bpp): Corresponds to the number of bits that describe the pixel color. (i.e., in 16 bits per pixel mode, each pixel can have 65536 different colors).
- Interface Width (IfWidth): Corresponds to the number of I/O data lines (data bus width) used for the interface between the LCD controller and the display.
- Frame Rate: This parameter is given by the display datasheet (in Hz). This rate must be respected to ensure the correct behavior of the display, thus avoiding bad image quality (flickering).

3. Display Selection Guide

In order to verify the compatibility of a display with the AT91SAM9 LCD Controller, a list of items to check is given below.

3.1 Technology Type

The LCD controller supports TFT RGB (up to 16M colors), color STN (up to 4096 colors) and Monochrome STN (up to 16 gray shades) displays. Other technologies may also be used if their interface is compatible.

3.2 Screen Size

The screen size is fully programmable. The maximum size supported is 2048 x 2048. Most of standard display resolution levels are compatible: VGA (640x480), QVGA (320 x 240), etc.

In STN mode, single and double scan modes are supported. For a double scan mode, the maximum size for each panel is 1024 x 2048.

3.3 Hardware Interface

The interface must be 3.3V compliant.

All the display lines should be present in the lines description list. If the display features more lines (i.e, gate command), the LCD Controller is not able to manage them. They could be managed by the PIO Controller or other peripherals (PWM, TC, etc.).

The polarity is programmable on:

LCDDOTCK: dot clock signal

LCDDEN: data enable signal

LCDVSYNC: vertical synchro signal

LCDHSYNC: horizontal synchro signal

Refer to Section 6.1 for more details on data bus connection in TFT mode.

3.4 Timings

The display size (LxH) and the frame rate are given by the display specifications.

The LCD Controller clock (LCDDOTCK) verifies the following formula:

STN Mono:

$$LCDDOTCKfreq = \frac{FrameRate \times DisplaySize}{IfWidth}$$

STN Color:

$$LCDDOTCKfreq = \frac{FrameRate \times DisplaySize \times 3}{IfWidth}$$

TFT:

$$LCDDOTCKfreq = FrameRate \times DisplaySize$$

According to the AT91SAM9 features, the maximum LCDDOTCK frequency is equal to the master clock (in bypass mode).

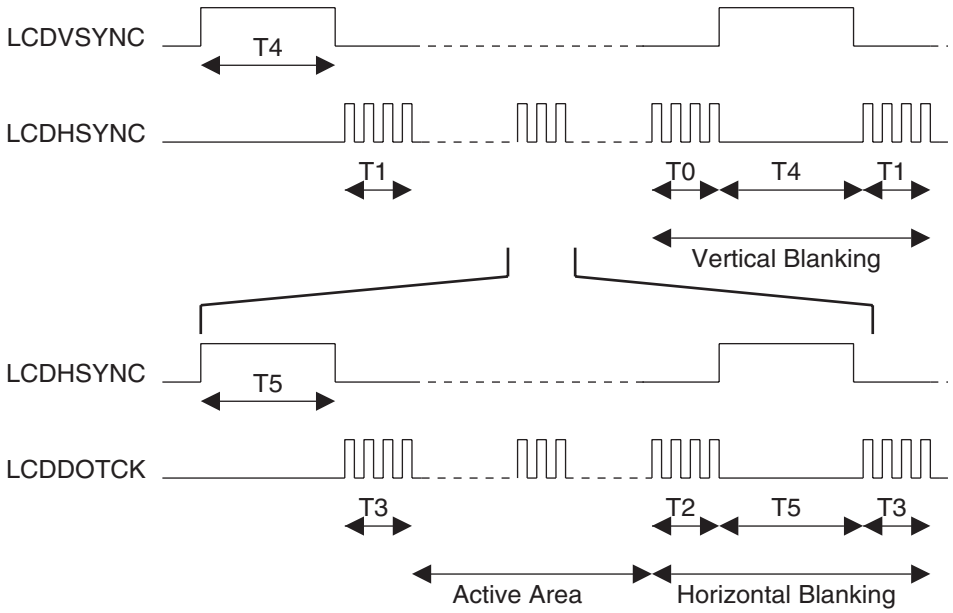
The signal timings provided by the display specification must respect the timings supported by the LCD Controller.

- Vertical pulse width from 1 to 64 lines.
- Horizontal pulse width from 1 to 64 dot clock cycles.

- Vertical Front Porch from 1 to 256 lines.
- Vertical Back Porch from 1 to 256 lines.

- Horizontal Front Porch from 1 to 2048 dot clock cycles.
- Horizontal Back Porch from 1 to 256 dot clock cycles.

Parameter	Min	Max	Unit	Symbol
Vertical Front Porch	1	256	Line	T0
Vertical Back Porch	1	256	Line	T1
Horizontal Front Porch	1	2048	LCDDOTCK cycle	T2
Horizontal Back porch	1	256	LCDDOTCK cycle	T3
Vertical Pulse Width	1	64	Line	T4
Horizontal Pulse Width	1	64	LCDDOTCK cycles	T5



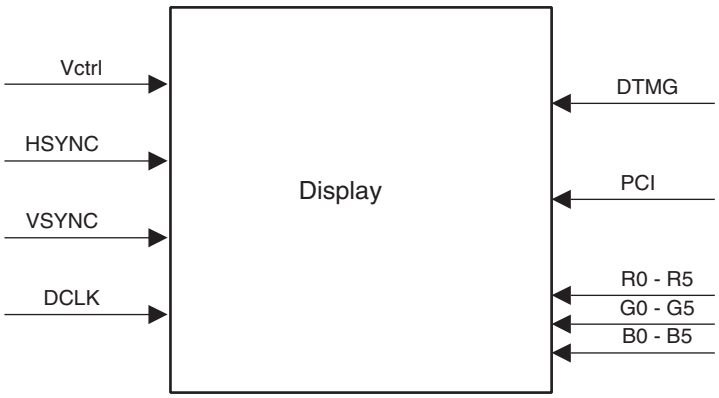
4. TFT Display Selection Guide Example

4.1 Display Description

An example of a display specification (TX09D71VM1CCA on AT91SAM9261-EK):

- TFT QVGA
- RGB 262K colors
- VCC [3.0 to 3.6V]
- Frame rate: 60 Hz
- Vertical Pulse Width : typ = 1 (in lines)
- Vertical Back Porch (VBP) : typ = 4 (in lines)
- Vertical Front Porch (VFP) : typ = 2 (in lines)
- Horizontal Pulse Width : typ = 5 (in pixels)
- Horizontal Back Porch (HBP): typ = 17 (in pixels)
- Horizontal Front Porch (HFP): typ = 11 (in pixels)

Interface:



4.2 Compatibility

To determine whether the display in the example is compatible, the following points should be evaluated:

4.2.1 Technology

It is a TFT display, thus the technology is supported.

4.2.2 Size

QVGA is 320 x 240, thus the size is acceptable.

4.2.3 Interface

VCC = 3.3V

Data bus 18 bits, 6 bits per color.

VSYNC corresponds to LCDVSYNC

HSYNC to LCDHSYNC

DCLK to LCDDOTCK

DTMG to LCDDEN

Add a passive filter between LCDCC and Vctrl.

Use a PIO for PCI control.

This interface is compatible.

4.2.4 Timings

Frame rate 60 Hz:

LCDDOTCKfreq = 60 x 320 x 240 = 4.608 MHz < LCDDOTCKfreq max.

The delays are all between 1 and 17. The LCD controller is able to manage delays between 1 and 256: all the blanking delays are in a compatible range.

The timings are compatible.

4.2.5 Conclusion

The display given in the example is compatible with the AT91SAM9261 LCD Controller.

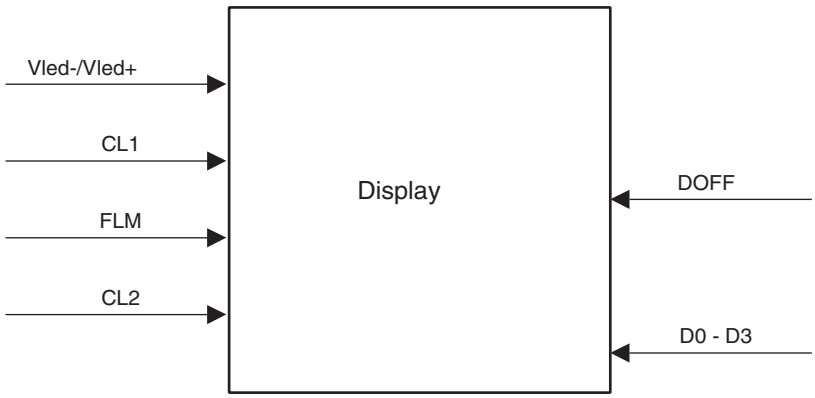
5. STN Display Selection Guide Example

5.1 Display Description

An example of a display specification (SP06Q002):

- STN QVGA
- 16 gray shades
- VCC [3.0 to 3.6V]
- Frame rate: 75 Hz
- Horizontal Pulse Width : typ = 1 (in pixels)
- Horizontal Back Porch (HBP): typ = 1 (in pixels)
- Horizontal Front Porch (HFP): typ = 1 (in pixels)

Interface:



5.2 Compatibility

To determine whether the display in the example is compatible, the following points should be evaluated:

5.2.1 Technology

It is a STN display, thus the technology is supported.

5.2.2 Size

QVGA is 320 x 240, thus the size is acceptable.

5.2.3 Interface

VCC = 3.3V

Data bus 4 bits.

FLM corresponds to LCDVSYNC

CL1 to LCDHSYNC

CL2 to LCDDOTCK

Add a passive filter between LCDCC and Vled-/leld+

Use a PIO for DOFF control.

This interface is compatible.

5.2.4 Timings

Frame rate 75 Hz:

LCDDOTCKfreq = 75 x 320 x 240 / 4 = 1.44MHz < LCDDOTCKfreq max.

The delays are all equal to 1. The LCD controller is able to manage delays between 1 and 256: all the blanking delays are in a compatible range. In STN mode, vertical timings should be set to 0.

The timings are compatible.

5.2.5 Conclusion

The display given in the example is compatible with the AT91SAM9 LCD Controller.

6. Configuration

6.1 TFT Hardware Connection

The AT91SAM9 LCD Controller is based on Blue-Green-Red (BGR) standard while most graphical layers use RGB color map.

As an example, the memory map of a pixel in 16-bit mode in the frame buffer is:

I	B4	B3	B2	B1	B0	G4	G3	G2	G1	G0	R4	R3	R2	R1	R0

where:

– I is Intensity Bit
– R[4:0], G[4:0] and B[4:0] are respectively red, green and blue video components.

Most graphical layers use RGB 5-6-5:

R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

This leads to a swap between blue and red components. There are two solutions to correct, via software or via hardware.

6.1.1 Software Swap

The swap of blue and red components by software is CPU consuming.

This is done using a piece of code. An example of the code follows, in C for an inline function:

```
__inline unsigned int swap_it(swap){
  unsigned int tmpR,tmpB;
  tmpB = swap <<10;//extract Blue
  tmpR = swap >>10;//extract Red
  swap = swap & 0x83E0; //keep only Green and I Bit
  return swap |= tmpB|tmpR;/returns the swapped color
}
```

In ASM, once compiled with armcc:

```
mov      r0,r5
mov      r3,r0,lsl #10
mov      r2,r0,asr #10
ldr      r1,0x000005fc ; = #0x000083e0
and      r0,r0,r1
orr      r1,r3,r2
orr      r1,r1,r0
```

ARMulate gives 8 CPU cycles for this sequence.

It must be done for each pixel of the frame, i.e., 320*240 = 76,800 times for each frame. Therefore, 76,800*8 = 614,400 CPU cycles per frame.

This means 614,400 * 25 = 15,360,000 CPU cycles per second for a 25 Hz refresh rate.

Compared to the 200,000,000 CPU cycles per second when running at 200 MHz, this gives 7.7% of the CPU time wasted to swap the LCD framebuffer.

For use the software solution, the LCD device is to be connected as follows to the PIOB lines.

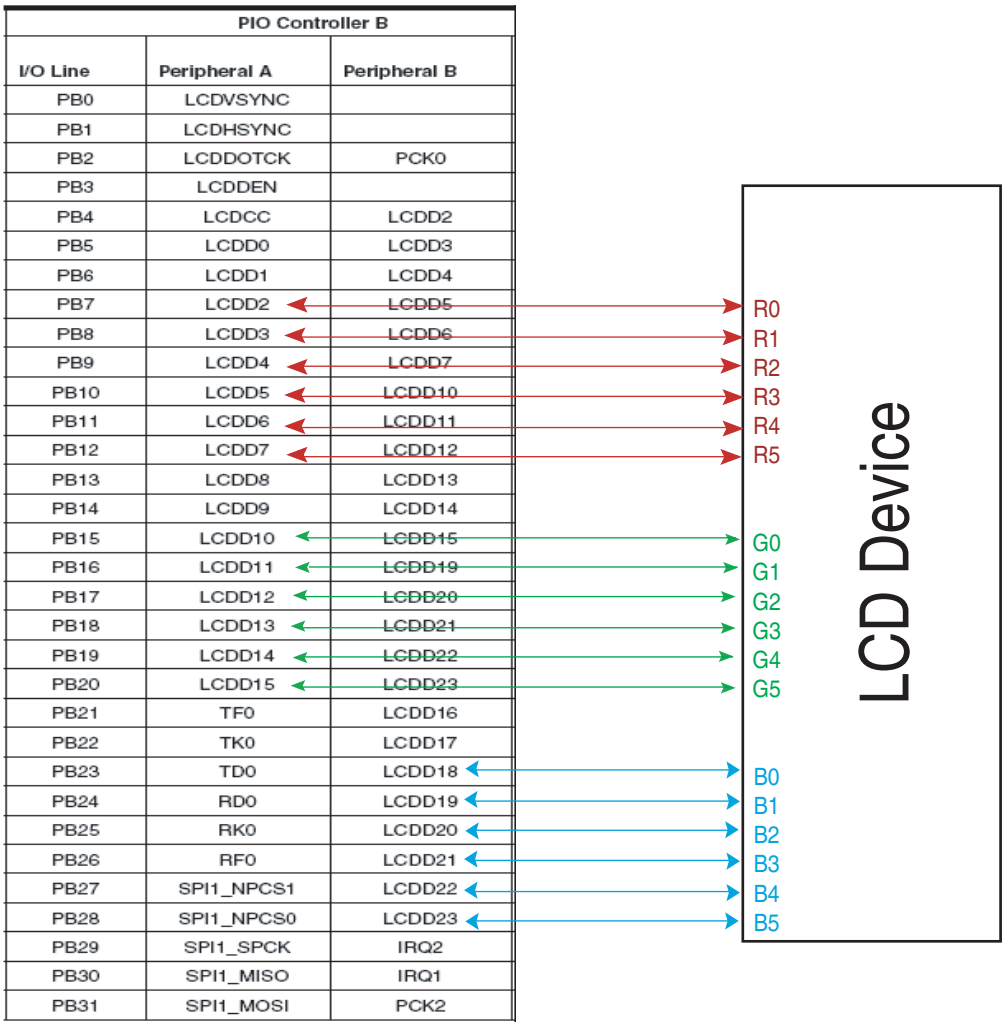
6.1.1.1 18-bit TFT Display

In this mode, the LCD Controller can be programmed in 16-bit or 24-bit mode, as the LCD Controller does not offer 18-bit mode.

In 16-bit mode, some framebuffer memory is saved. The number of colors available is 65,536.

In 24-bit mode, as only 18 bits are wired to the LCD device, the number of colors available is 262,144.

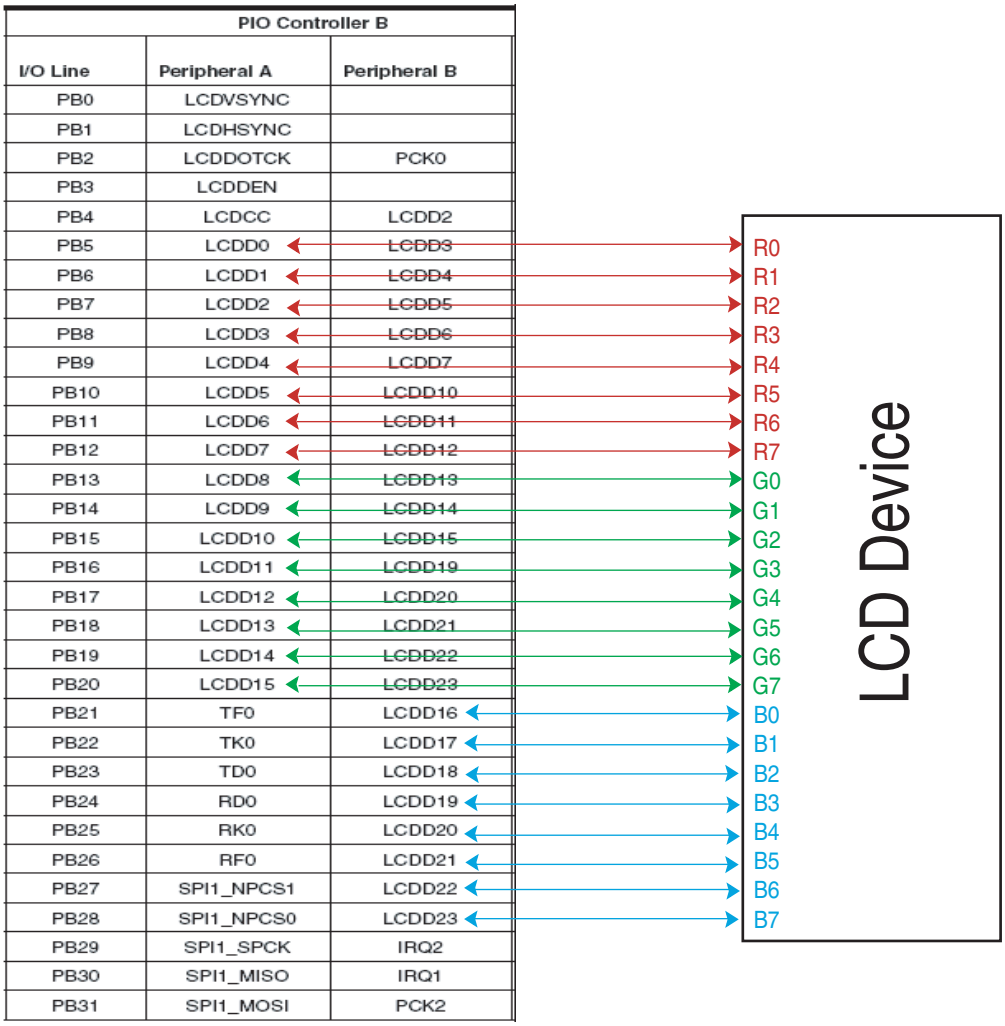
Figure 6-1. 18-bit TFT Display Hardware Connection (Software Swap) in 24-bit Mode



PIO Controller B		
I/O Line	Peripheral A	Peripheral B
PB0	LCDVSYNC	
PB1	LCDHSYNC	
PB2	LCDDOTCK	PCK0
PB3	LCDDEN	
PB4	LCDCC	LCDD2
PB5	LCDD0	LCDD3
PB6	LCDD1	LCDD4
PB7	LCDD2	~~LCDD5~~
PB8	LCDD3	~~LCDD6~~
PB9	LCDD4	~~LCDD7~~
PB10	LCDD5	~~LCDD10~~
PB11	LCDD6	~~LCDD11~~
PB12	LCDD7	~~LCDD12~~
PB13	LCDD8	LCDD13
PB14	LCDD9	LCDD14
PB15	LCDD10	~~LCDD15~~
PB16	LCDD11	~~LCDD19~~
PB17	LCDD12	~~LCDD20~~
PB18	LCDD13	~~LCDD21~~
PB19	LCDD14	~~LCDD22~~
PB20	LCDD15	~~LCDD23~~
PB21	TF0	LCDD16
PB22	TK0	LCDD17
PB23	TD0	LCDD18
PB24	RD0	LCDD19
PB25	RK0	LCDD20
PB26	RF0	LCDD21
PB27	SPI1_NPCS1	LCDD22
PB28	SPI1_NPCS0	LCDD23
PB29	SPI1_SPCK	IRQ2
PB30	SPI1_MISO	IRQ1
PB31	SPI1_MOSI	PCK2

6.1.1.2 24-bit TFT Display

Figure 6-2. 24-bit TFT Display Hardware Connection (Software Swap)



PIO Controller B		
I/O Line	Peripheral A	Peripheral B
PB0	LCDVSYNC	
PB1	LCDHSYNC	
PB2	LCDDOTCK	PCK0
PB3	LCDDEN	
PB4	LCDCC	LCDD2
PB5	LCDD0	~~LCDD3~~
PB6	LCDD1	~~LCDD4~~
PB7	LCDD2	~~LCDD5~~
PB8	LCDD3	~~LCDD6~~
PB9	LCDD4	~~LCDD7~~
PB10	LCDD5	~~LCDD10~~
PB11	LCDD6	~~LCDD11~~
PB12	LCDD7	~~LCDD12~~
PB13	LCDD8	~~LCDD13~~
PB14	LCDD9	~~LCDD14~~
PB15	LCDD10	~~LCDD15~~
PB16	LCDD11	~~LCDD19~~
PB17	LCDD12	~~LCDD20~~
PB18	LCDD13	~~LCDD21~~
PB19	LCDD14	~~LCDD22~~
PB20	LCDD15	~~LCDD23~~
PB21	TF0	LCDD16
PB22	TK0	LCDD17
PB23	TD0	LCDD18
PB24	RD0	LCDD19
PB25	RK0	LCDD20
PB26	RF0	LCDD21
PB27	SPI1_NPCS1	LCDD22
PB28	SPI1_NPCS0	LCDD23
PB29	SPI1_SPCK	IRQ2
PB30	SPI1_MISO	IRQ1
PB31	SPI1_MOSI	PCK2

6.1.2 Hardware Swap

The CPU time for swapping can be reduced by the hardware solution for swapping of the red and blue signals. The LCD device is connected as follows to the PIOB lines.

6.1.2.1 18-bit TFT Display

In this mode, the LCD Controller can be programmed in 16-bit or 24-bit mode.

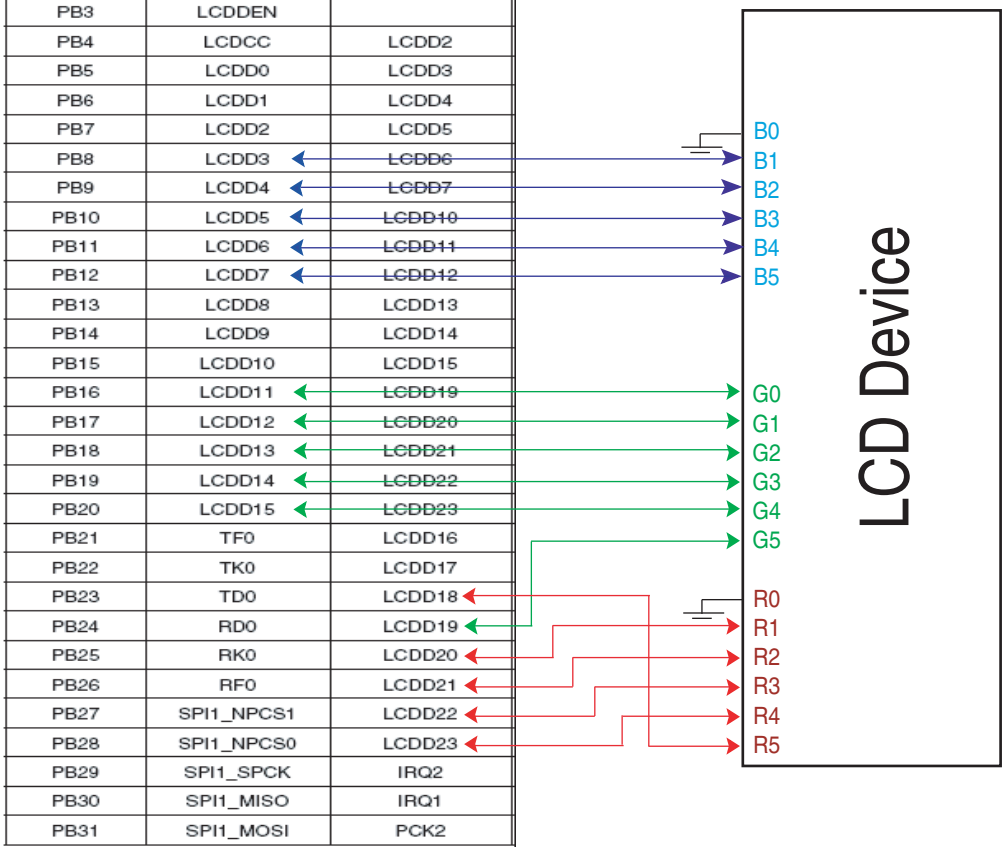
In 16-bit mode, some framebuffer memory is saved. The following table explains how to connect a 18-bit TFT display in 16-bit mode, using the RGB 6-5-6 format via the 1-5-5-5 format of the LCD Controller. The number of colors available is 65,536.

Memory Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RGB 5-6-5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
LCD Controller BGR Format	I	B4	B3	B2	B1	B0	G4	G3	G2	G1	G0	R4	R3	R2	R1	R0
LCD Controller Interface	D18	D23	D22	D21	D20	D19	D15	D14	D13	D12	D11	D7	D6	D5	D4	D3
Display Interface	R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1

The 5 bits for Red and Blue are connected on the MSB pins of the display. B0 and R0 are connected to GND. LCDD10 and LCDD2 are not used.

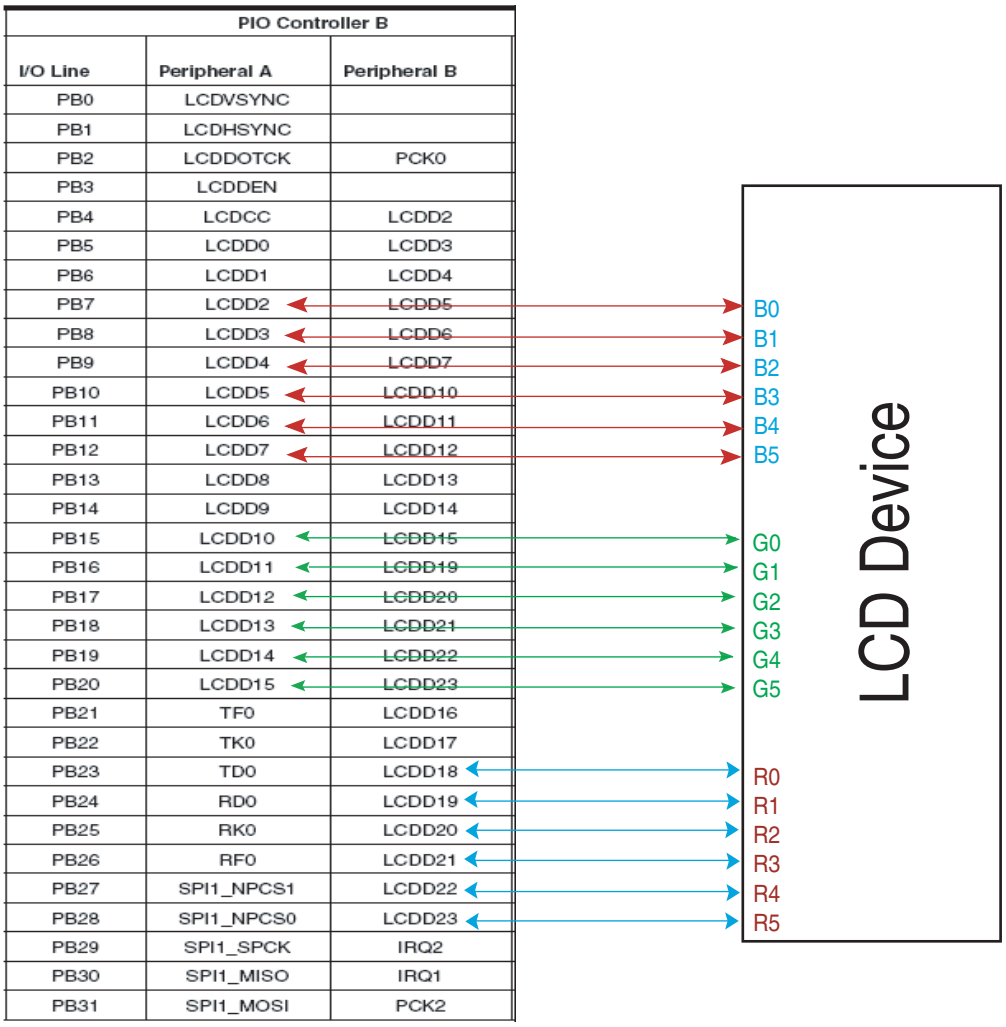
Figure 6-3. 18-bit TFT Display Hardware Connection in 16-bit Mode: RGB 5-6-5 Format

PIO Controller B		
I/O Line	Peripheral A	Peripheral B
PB0	LCDVSYNC	
PB1	LCDHSYNC	
PB2	LCDDOTCK	PCK0
PB3	LCDDEN	
PB4	LCDCC	LCDD2
PB5	LCDD0	LCDD3
PB6	LCDD1	LCDD4
PB7	LCDD2	LCDD5
PB8	LCDD3	~~LCDD6~~
PB9	LCDD4	~~LCDD7~~
PB10	LCDD5	~~LCDD10~~
PB11	LCDD6	~~LCDD11~~
PB12	LCDD7	~~LCDD12~~
PB13	LCDD8	LCDD13
PB14	LCDD9	LCDD14
PB15	LCDD10	LCDD15
PB16	LCDD11	~~LCDD19~~
PB17	LCDD12	~~LCDD20~~
PB18	LCDD13	~~LCDD21~~
PB19	LCDD14	~~LCDD22~~
PB20	LCDD15	~~LCDD23~~
PB21	TF0	LCDD16
PB22	TK0	LCDD17
PB23	TD0	LCDD18
PB24	RD0	LCDD19
PB25	RK0	LCDD20
PB26	RF0	LCDD21
PB27	SPI1_NPCS1	LCDD22
PB28	SPI1_NPCS0	LCDD23
PB29	SPI1_SPCK	IRQ2
PB30	SPI1_MISO	IRQ1
PB31	SPI1_MOSI	PCK2



In 24-bit mode, as only 18 bits are wired to the LCD device, the number of colors available is 262,144.

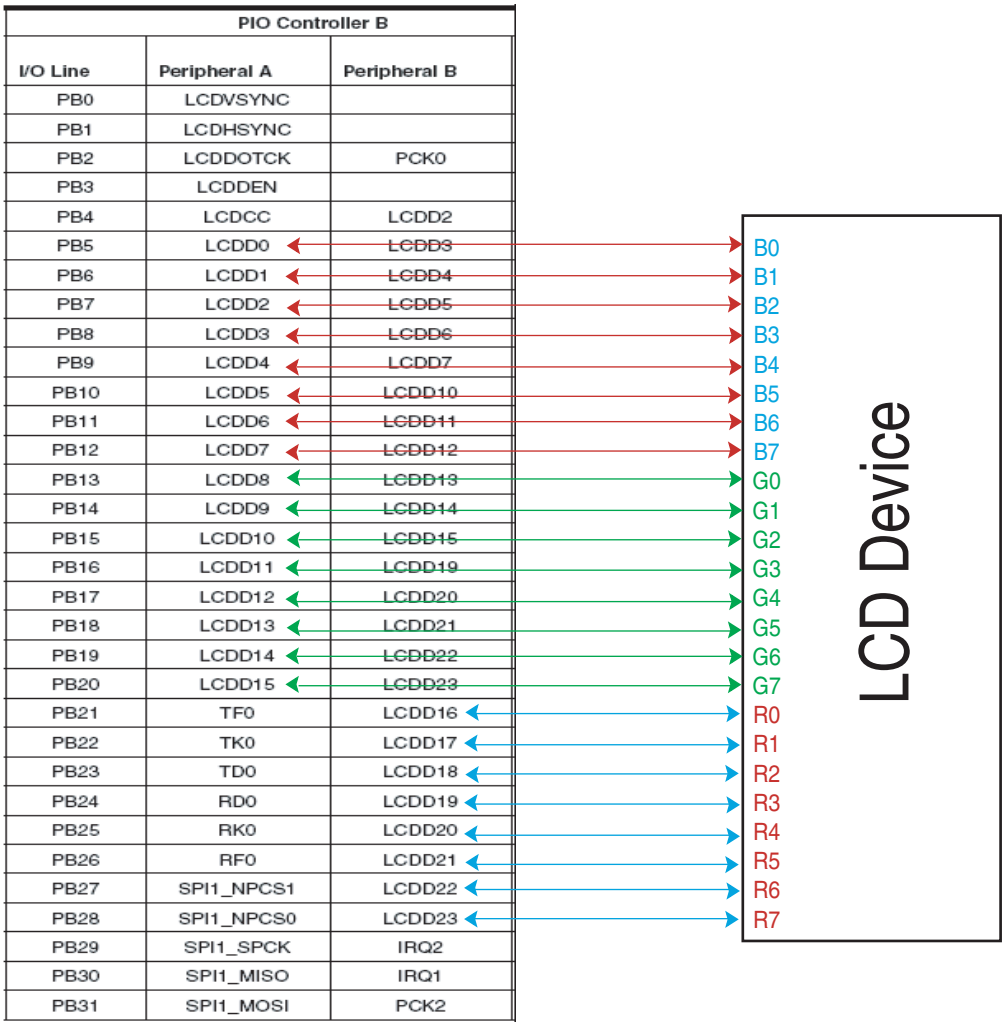
Figure 6-4. 18-bit TFT Display Hardware Connection in 24-bit Mode



PIO Controller B		
I/O Line	Peripheral A	Peripheral B
PB0	LCDVSYNC	
PB1	LCDHSYNC	
PB2	LCDDOTCK	PCK0
PB3	LCDDEN	
PB4	LCDCC	LCDD2
PB5	LCDD0	LCDD3
PB6	LCDD1	LCDD4
PB7	LCDD2	~~LCDD5~~
PB8	LCDD3	~~LCDD6~~
PB9	LCDD4	~~LCDD7~~
PB10	LCDD5	~~LCDD10~~
PB11	LCDD6	~~LCDD11~~
PB12	LCDD7	~~LCDD12~~
PB13	LCDD8	LCDD13
PB14	LCDD9	LCDD14
PB15	LCDD10	~~LCDD15~~
PB16	LCDD11	~~LCDD19~~
PB17	LCDD12	~~LCDD20~~
PB18	LCDD13	~~LCDD21~~
PB19	LCDD14	~~LCDD22~~
PB20	LCDD15	~~LCDD23~~
PB21	TF0	LCDD16
PB22	TK0	LCDD17
PB23	TD0	LCDD18
PB24	RD0	LCDD19
PB25	RK0	LCDD20
PB26	RF0	LCDD21
PB27	SPI1_NPCS1	LCDD22
PB28	SPI1_NPCS0	LCDD23
PB29	SPI1_SPCK	IRQ2
PB30	SPI1_MISO	IRQ1
PB31	SPI1_MOSI	PCK2

6.1.2.2 *24-bit TFT Display*

Figure 6-5. 24-bit TFT Display Hardware Connection in 24-bit Mode



PIO Controller B		
I/O Line	Peripheral A	Peripheral B
PB0	LCDVSYNC	
PB1	LCDHSYNC	
PB2	LCDDOTCK	PCK0
PB3	LCDDEN	
PB4	LCDCC	LCDD2
PB5	LCDD0	~~LCDD3~~
PB6	LCDD1	~~LCDD4~~
PB7	LCDD2	~~LCDD5~~
PB8	LCDD3	~~LCDD6~~
PB9	LCDD4	~~LCDD7~~
PB10	LCDD5	~~LCDD10~~
PB11	LCDD6	~~LCDD11~~
PB12	LCDD7	~~LCDD12~~
PB13	LCDD8	~~LCDD13~~
PB14	LCDD9	~~LCDD14~~
PB15	LCDD10	~~LCDD15~~
PB16	LCDD11	~~LCDD19~~
PB17	LCDD12	~~LCDD20~~
PB18	LCDD13	~~LCDD21~~
PB19	LCDD14	~~LCDD22~~
PB20	LCDD15	~~LCDD23~~
PB21	TF0	LCDD16
PB22	TK0	LCDD17
PB23	TD0	LCDD18
PB24	RD0	LCDD19
PB25	RK0	LCDD20
PB26	RF0	LCDD21
PB27	SPI1_NPCS1	LCDD22
PB28	SPI1_NPCS0	LCDD23
PB29	SPI1_SPCK	IRQ2
PB30	SPI1_MISO	IRQ1
PB31	SPI1_MOSI	PCK2

6.2 STN Hardware Connection

STN display data bus is not based on RGB or BGR bus. It is based on data bus D0 to Dn.

Refer to the AT91SAM9 datasheet for more details.

6.3 Software

6.3.1 Initialization Sequence for AT91SAM9261 device example

6.3.1.1 *Power Management*

The HCLK bit of the PMC_SCER register must be set before using the LCD Controller. It enables the LCD Controller clock. No specific peripheral clock is needed.

6.3.1.2 I/O Lines

The pins used for interfacing the LCD Controller are multiplexed with PIO lines. The programmer must first program the PIO Controller to assign the pins to their peripheral function. If some I/O lines of the LCD Controller are not used by the application, they can be used for other purposes by the PIO Controller.

6.3.1.3 Turn off the LCD Controller

To be sure that initialization is correct, first disable the LCD Controller and DMA by clearing the LCD_PWR and the DMAEN bits in the PWRCON and the DMACON registers, respectively.

6.3.1.4 Palette

If the number of colors is less than or equal to 256, the palette LUT must be initialized.

6.3.1.5 LCDCON1

The LCDDOTCK is configured in the LCDCON1 register. If the LCDDOTCK frequency is equal to the LCDC Clock frequency, the BYPASS bit must be set. If not, the CLKVAL field must be programmed as described below.

Calculate the theoretical Dot Clock:

STN Mono:

$$LCDDOTCKfreq = \frac{FrameRate \times DisplaySize}{IfWidth}$$

STN Color:

$$LCDDOTCKfreq = \frac{FrameRate \times DisplaySize \times 3}{IfWidth}$$

TFT:

$$LCDDOTCKfreq = FrameRate \times DisplaySize$$

Calculate the corresponding theoretical value for CLKVAL:

$$CLKVAL_{theo} = \frac{LCDCClockFreq}{(2 \times LCDDOTCK_{theo}) - 1}$$

Round down the $CLKVAL_{theo}$ value to find the real CLKVAL value.

Check the LCDDOT frequency:

$$LCDDOTCK = \frac{LCDCClock}{2 \times (CLKVAL + 1)}$$

Example:

A TFT QVGA display has the following features: 320 x 240, 60 Hz, 262K colors, 18-bit data bus. LCDC Clock is 100 MHz.

$LCDDOTCK_{theo}$ = 320 x 240 x 60 = 4.608 MHz

$CLKVAL_{theo}$ = 100 / (2 x 4.608) - 1 = 9.85

CLKVAL = 9

LCDDOTCK = 100 / (2 x (9+1)) = 5 MHz.

An STN QVGA display has the following features: 320 x 240, 75 Hz, 4-bit data bus. LCDC Clock is 100 MHz.

$LCDDOTCK_{theo}$ = 320 x 240 x 75 / 4 = 1.44 MHz

$CLKVAL_{theo}$ = 100 / (2 x 1.44) - 1 = 34.7

CLKVAL = 34

LCDDOTCK = 100 / (2 x (9+1)) = 1.43MHz.

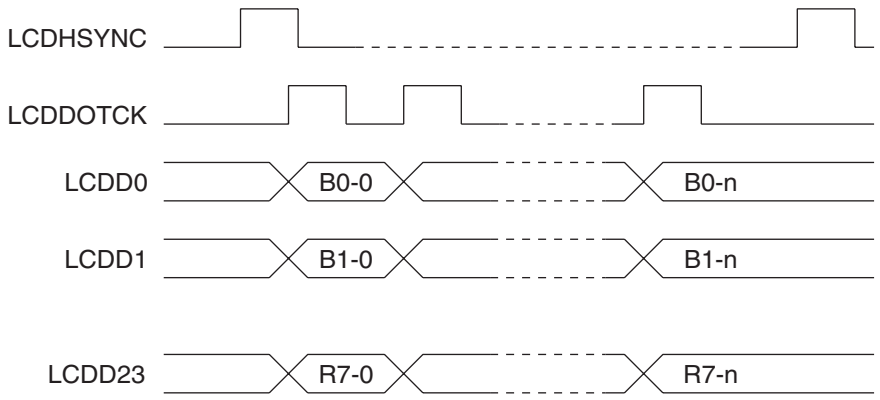
6.3.1.6 *LCDCON2 and LCDFRMCFG for AT91SAM9261*

The display resolution is set in the LCDFRMCFG register.

The LINEVAL field corresponds to the number of lines of the display - 1.

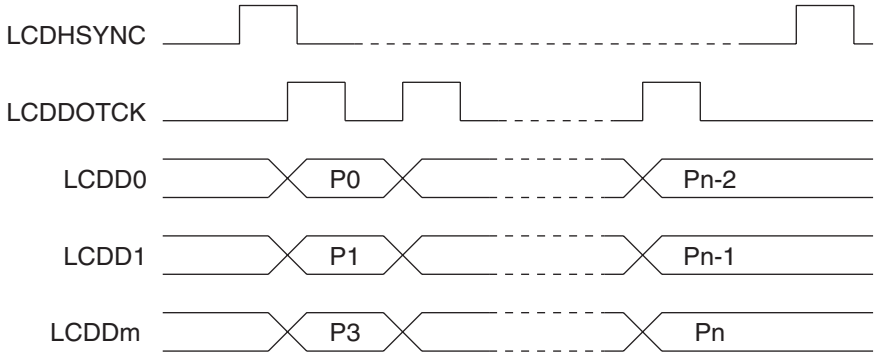
Note that the HOZVAL field does NOT always correspond to the number of columns:

In TFT mode, $HOZVAL = ColumnNumber - 1$



Horizontal size is n+1. The LCD Controller needs n+1 LCDDOTCK cycles to send a line. HOZVAL field must be set with the value n.

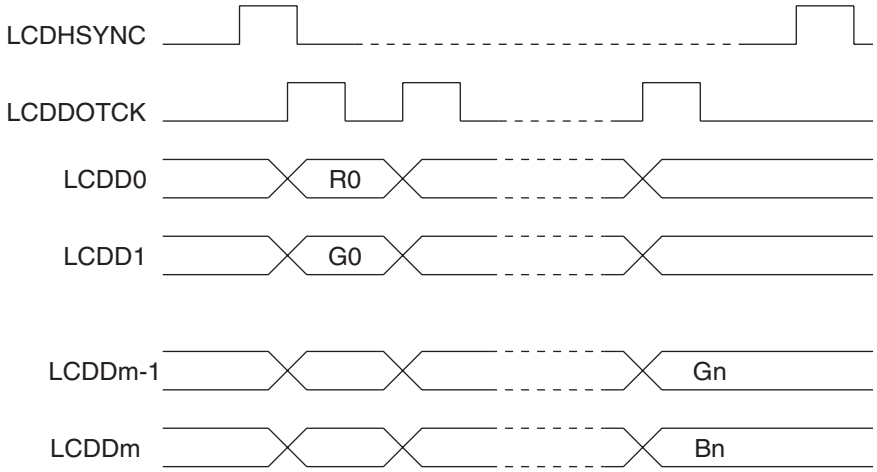
In STN monochrome mode, $HOZVAL = \frac{ColumnNumber}{Ifwidth} - 1$



Horizontal size is n+1. Interface width is m+1. The LCD Controller needs (n+1) / (m+1) LCD-DOTCK cycles to send a line.

HOZVAL field must be set with the value $\frac{n+1}{m+1} - 1$.

In STN color mode, $HOZVAL = \frac{ColumnNumber \times 3}{Ifwidth} - 1$



Horizontal size is n+1. Interface width is m+1. The LCD Controller needs 3 x (n+1) / (m+1) LCD-DOTCK cycles to send a line.

HOZVAL field must be set with the value $\frac{(n+1) \times 3}{m+1} - 1$.

The other display parameters are configured in the LCDCON2 register:

Type, Scan mode, Interface width, Pixel size, etc.

6.3.1.7 LCDCON2 and LCDFRMCFG for Other AT91SAM9 Products

The display resolution is set in the LCDFRMCFG register.

The LINEVAL field corresponds to the number of lines of the display - 1.

The LINESIZE field corresponds to the number of columns of the display - 1.

Horizontal size is n+1. The LCD Controller needs n+1 LCDDOTCK cycles to send a line. LINESIZE field must be set with the value n.

The other display parameters are configured in the LCDCON2 register:

Type, Scan mode, Interface width, Pixel size, etc.

6.3.1.8 *Timings*

The LCDHSYNC and LCDVSYNC signal pulse width are configurable through the VPW and HPW fields in the LCDTIM1 and LCDTIM2 registers. It is defined in number of LCDDOTCK cycles.

6.3.1.9 *Blanking*

Write VFP, VBP, HFP, HBP according to the data given in the display datasheet.

6.3.1.10 *Interrupt*

The LCD Controller interrupts are configured as the other AT91 peripherals through the Enable/Disable/Mask/Status/Clear registers.

6.3.1.11 *Contrast*

The contrast PWM Signal is programmable from 0 to FF in the CVAL field of the CONTRAST_VAL register. Polarity and frequency are also programmable in the PS and POL fields of the CONTRAST_CTR register. To enable/disable it, set or clear the ENA bit.

6.3.1.12 *DMA Base Address*

In TFT and STN Single Scan mode, the frame buffer address is stored in the DMABADDR1 register. In STN Double Scan mode, the second frame buffer address is stored in the DMABADDR2 register.

If the frame buffer is located in the SRAM memory, use the address base 0x0030 000.

If the frame buffer is located in the SDRAM, it is recommended to store it in a different bank than the application in order to optimize the SDRAM access. Refer to the Atmel application note"Using the SDRAM on AT91SAM9 Microcontrollers", ref. 6256, for more details.

6.3.1.13 *DMA Frame Configuration*

Two parameters are needed to configure the DMA: the burst length and the frame size.

BRSTLN = burst length - 1

FRMSIZE = DisplaySize x Bpp / 32

Note that the Bpp parameter to use is the one used for the LCD Controller configuration. If it is an 18-bit display, the configuration is 16 bits or 24 bits. In this formula, Bpp is 16 or 24, not 18.

6.3.1.14 *FIFO*

The FIFOTH field must be programmed with:

FIFOTH = 512 - (2 x DMA_BURST_LENGTH + 3)

where:

DMA_BURST_LENGTH is defined in the DMAFRMCFG register

6.3.1.15 *DMA Enable*

Reset and enable the DMA by setting the DMARST and DMAEN bits in the DMACON register. It must be done in 2 steps.

6.3.1.16 *LCD Enable*

Enable the LCD Controller by setting the LCD_PWR bit in the PWRCON register. At this moment, the LCD Controller is running and is sending frames to the displays via the LCD Controller interface. Depending on the display interface, other IOs may be configured to enable the display, or for backlight activation.

6.3.2 Software Examples

The AT91SAM9261-BasicLCD-IAR_4_41 Project configures:

A monochrome STN display, 320 x 240, 16 gray shades

A Color STN display, 320 x 240, 4096 colors

A TFT display, 320 x 240, 16M colors

7. Performances

7.1 Matrix Configuration

In order to set the maximum priority for the LCD Controller (display frame rate is more important than application performance), it is recommended to set the slave where the frame buffer is stored with the following configuration: LCD as a Fixed default Master.

7.2 Bus Bandwidth

7.2.1 Application in SDRAM, Frame Buffer in SDRAM

The Matrix has to share the SDRAM access between the ARM Core and the LCD Controller. The effect on the application performance depends on the LCD throughput. The throughput is the product of the frame rate multiplied by the display size and the number of bits per pixel.

Section 7.2.5 gives details on the impact on the performances of different LCD configurations.

7.2.2 Application in SRAM, Frame Buffer in SDRAM

Thanks to the multi-layer feature of the matrix, transfers can be done in parallel between SRAM and the ARM Core for the application, and between SDRAM and the LCD Controller for the frame buffer. Thus, application performances are not affected by the LCD.

7.2.3 Application in SDRAM, Frame Buffer in SRAM

Thanks to the multi-layer feature of the matrix, transfer can be done in parallel between SDRAM and the ARM Core for the application, and between SRAM and the LCD Controller for the frame buffer. Thus, application performances are not affected by the LCD.

7.2.4 Application in SRAM, Frame Buffer in SRAM

The matrix has to share the SRAM access between the ARM Core and the LCD Controller. The effect on the application performance depends on the LCD throughput. The throughput is the product of the frame rate multiplied by the display size and the number of bits per pixel.

Section 7.2.5 gives details on the impact on the performances of different LCD configurations

7.2.5 Bus Bandwidth Summary

Some performance measurements have been done on a AT91SAM9261-EK board. The processor clock is set to 200 MHz, the master clock is set to 100 MHz. The TCM, the MMU and the Caches are not activated. The applied algorithm is Dhrystone 2.1. The Matrix configuration is the default configuration: No Default Master. LCD Configuration: TFT 240 x 320, 16-bit, 70 Hz.

Dhrystone application executes calculation, read and write operations in the memory.

The results below illustrate the memory bandwidth availability, in %, compared to the same configuration with LCD controller off.

	Appli: SRAM Buffer: SRAM	**Appli: SDRAM Buffer: SDRAM**	**Appli: SDRAM Buffer: SRAM**	**Appli: SRAM Buffer: SDRAM**
Bandwidth in % used for framebuffer transfer	2%	2%	0%	0%

The results when application and buffer are not in the same memory are explained by the matrix properties. Transfers are done in parallel from SDRAM to LCD and between ARM926™ and SRAM or from SRAM to LCD and between ARM926 and SDRAM.

Comments: the frame buffer is not refreshed. This example can be seen as a picture display test. For the movie reader application, a refresh rate of the frame buffer at 25 frames per second should be taken into account, increasing the total used bandwidth.

Table 7-1. Frame Buffer and Application in SDRAM

	320 x 240 50Hz	**320 x 240 70Hz**	**640 x 480 50Hz**	**640 x 480 70Hz**
24-bit	2%	4%	15%	23%
16-bit	1%	2%	9%	15%

At 70 Hz, 640 x 480 size, 24-bit, the throughput is 16.1Million words/s. With a Master clock at 100 MHz, the performance loss is about 23%.

In order to estimate performance loss for other LCD configurations with high throughput, we can consider that the performance loss is linear with the throughput.

8. Revision History

Revision History

Doc. Rev	Comments	Change Request Ref.
6300B	Added new Section 6.3.1.7 ”LCDCON2 and LCDFRMCFG for Other AT91SAM9 Products” on page 18.	4510
6300A	First issue.	

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com
www.atmel.com/AT91SAM

Technical Support
AT91SAM Support

Sales Contacts
www.atmel.com/contacts/

Literature Requests
www.atmel.com/literature