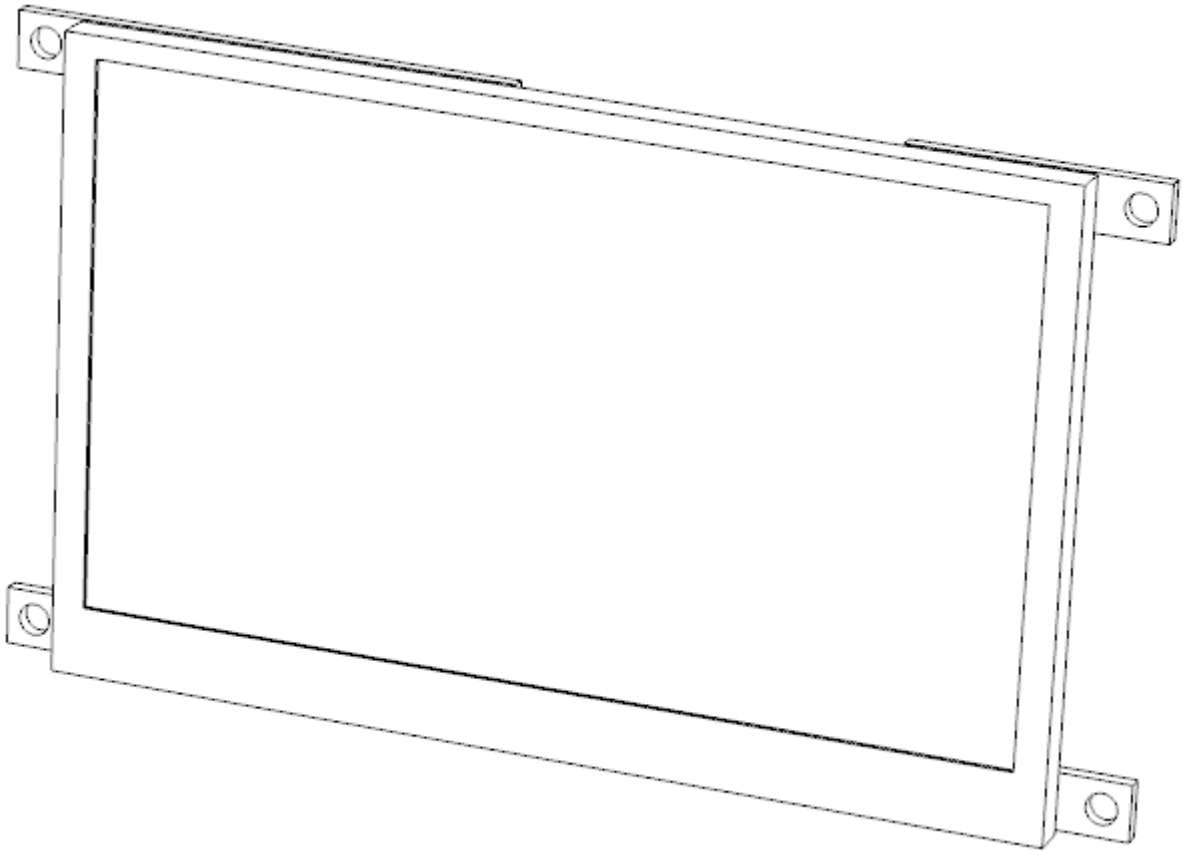
SPI043 Datasheet



Very preliminary. Guaranteed to change.

Version 0.0

www.lcdinfo.com

Table of Contents

1. Introduction

Overview

The SPI043 is a controller board designed to be used with a 4.3” TFT LCD panel from Sharp. The controller board provides framebuffer functionality and offers an SPI interface for data transfer. The controller board integrates voltage regulation and backlight driving circuitry.

It is also possible to use the display as an input user interface with the optional touchscreen.

Features

- 4.3" 480x272 pixel resolution color TFT LCD
- Optional touchscreen
- High speed SPI interface
- 16 bit colors (RGB565)
- 8 MB framebuffer memory
- 5 V power supply voltage
- 3.3 V logic input voltage (typical)

2. Physical Characteristics

Item	Specifications	Unit
Display resolution	480 x 272	dot
Screen size	10.9 (4.3” type) diagonal	cm
Active area	95.04(H) x 53.856(V)	mm
Pixel pitch	0.198 x 0.198	mm
Unit outline dimensions	120(W) x 67.2(H) x ___(D)	mm
Weight	TBD	g
Backlight type	LED	

3. Electrical Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage [1)]	V_{CC}	4.8	5.0	5.2	V
Current Consumption without backlight [2)]			70		mA
Backlight Current Consumption full brightness			110		mA
Input voltage (Low) [3)]	V_{IL}			0.8	V
Input voltage (High) [3)]	V_{IH}	2.0			V
Input voltage [3)]	V_{IN}	-0.5		+3.8	V

Note:

1. Sharp LCD panel specification
2. Including LCD panel
3. MOSI, SCK, D/C, SSEL

4. SPI043 board

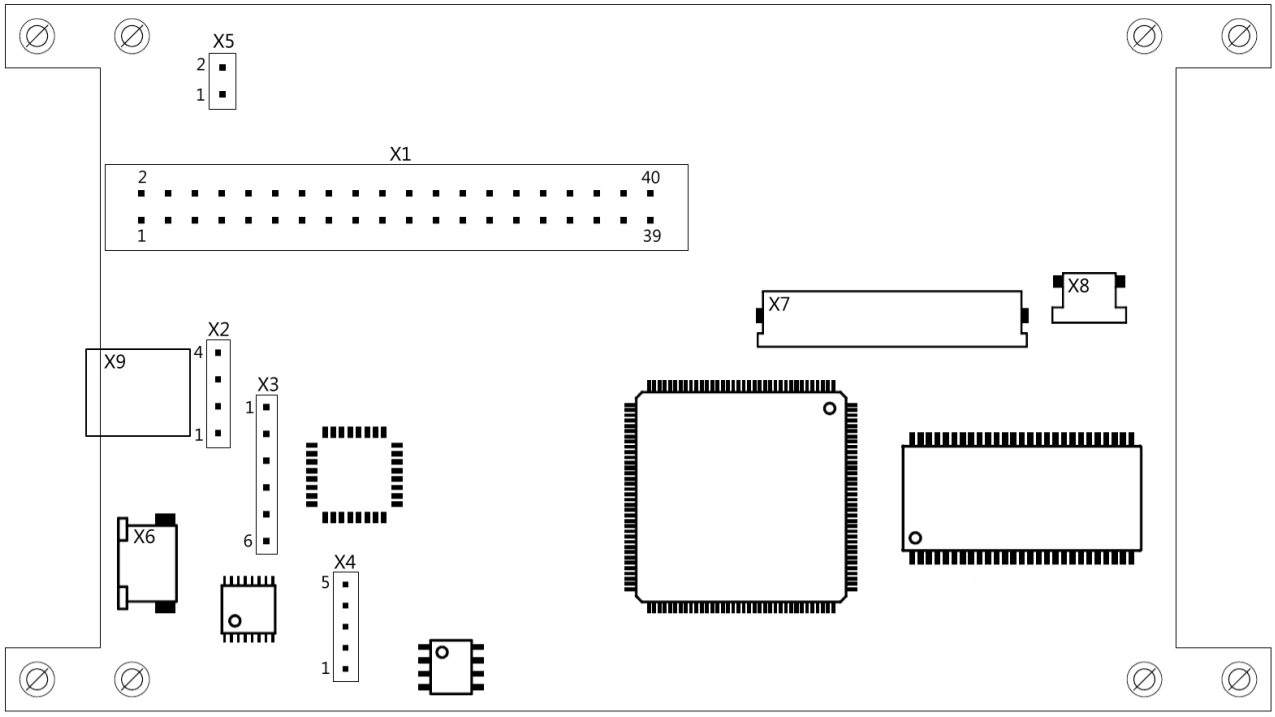


Fig. 1 Connectors

5. Connectors

Interface connector – X1

2.54 mm 20 x 2

Pin	Name	Function	Pin	Name	Function
1	+5V	Power supply	2	+5V	Power supply
3	+5V	Power supply	4	GND	GND(0V)
5	GND	GND(0V)	6	TXD	
7	RXD		8	GND	
9			10	GND	
11			12		
13			14	GND	
15			16		
17			18	GND	
19			20		
21			22	GND	
23			24		
25			26	GND	
27			28	GND	
29	MOSI	Serial data input	30	GND	
31	MISO	Serial data output	32	GND	
33	SCK	Serial clock input	34	GND	
35	D/C	Data / Command input	36	GND	
37	SSEL	Slave select input	38	GND	
39			40	GND	

Same pinout is also used in the optional 40 pin 0.5 mm FFC connector that can be used instead of the 2.54 mm pin header.

Currently only SPI interface is supported but there are enough pins connected to implement up to 16 bit parallel interface in the future if needed (which should be quite simple and only require removing/bypassing the SPI de-serializer block from the design and directly exposing the parallel bus with a future firmware update).

Touchscreen controller connector – X3

2.54 mm 6 x 1

For directly accessing the touchscreen controller. Preprocesssed touchscreen data can be also accessed through the UART serial interface (TXD, RXD) from the X1 connector.

Pin	Name	Function
1	GND	
2	PENIRQ	
3	SSEL	
4	SCK	
5	MISO	
6	MOSI	

Power connector – X5

2.54 mm 2 x 1

Additional power connector. Can be left unconnected if power is supplied using X1.

Pin	Name	Function
1	+5V	
2	GND	

Display connector - X7

Backlight connector - X8

Touchscreen connector - X6

4 way 1 mm pitch FFC/FPC connector with bottom contacts.

Pin	Name	Function
1	X1	
2	Y2	
3	X2	
4	Y1	

USB connector - X9

Only used for firmware updates.

Pin	Name	Function
1	VBUS	+5 VDC
2	D-	Data -
3	D+	Data +
4	NC	Unconnected
5	GND	Ground

6. Getting started

LCD panel mounting

The SPI043 controller board is designed to be mechanically compatible with the Sharp LQ043T3DX02 and compatible LCD panels. These LCD panels do not provide built-in mounting tabs but the SPI043 mounting holes can be used to mount the assembly including the controller board and the LCD panel. The LCD panel can be folded against the bottom side of the controller board and attached using suitable adhesive. Even if not attaching the LCD panel to the controller board some insulation should be used between the LCD panel metal frame and the circuit board to prevent a short circuit between them.

Connecting the display module

The controller board and display can be connected with just one cable using the connector X1. At higher serial interface bit rates more attention should be placed on signal integrity of the cable and all the ground connections near the serial interface signals should be connected.

7. Serial interface

Frame format

16 bits per frame

Maximum bit rate

Currently tested serial bit rate is the NXP LPC214x maximum of 30 MHz. The controller should be able to handle bit rates much higher than that as long as the signals remain clean.

Serial signals

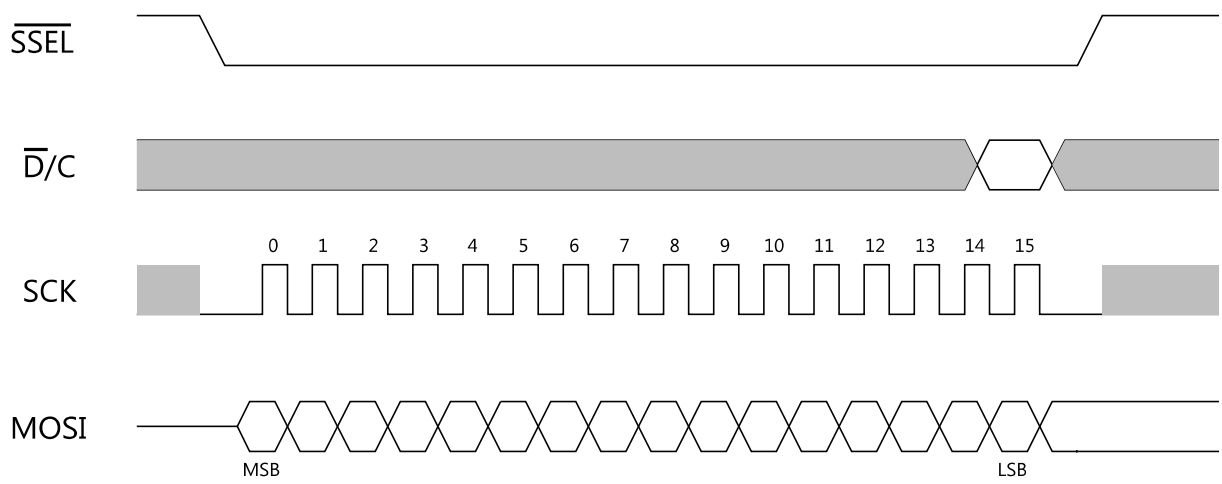


Fig. 2 Serial bus protocol – transmission of one frame

The controller begins receiving the 16 bit serial frame when the SSEL signal goes LOW. After the 16th clock edge the de-serialized frame is passed forward to the input FIFO first and from there to the controller pipeline for further processing. The D/C signal is also sampled on the 16th clock edge. If SSEL goes HIGH before the 16th bit is received the bits received are discarded and the controller again waits for the SSEL to go LOW to begin receiving the next 16 bit frame.

If the D/C signal is HIGH the frame is processed as a command and if D/C signal is LOW the frame is stored to the framebuffer RAM to the current address.

It is also possible to write multiple 16 bit frames sequentially without pulsing SSEL line HIGH between each frame.

Data is sampled on the rising edge of SCK.

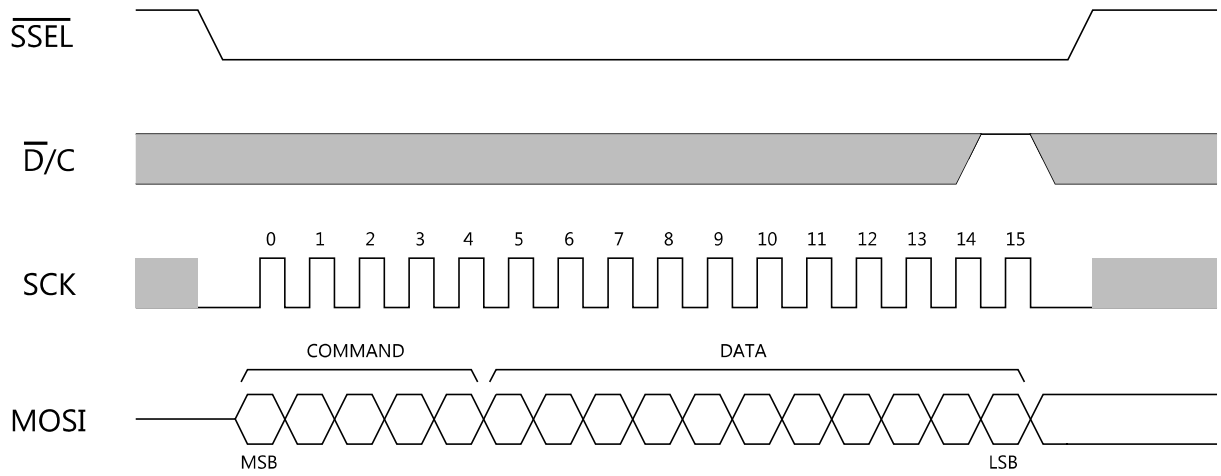


Fig. 3 Command Write

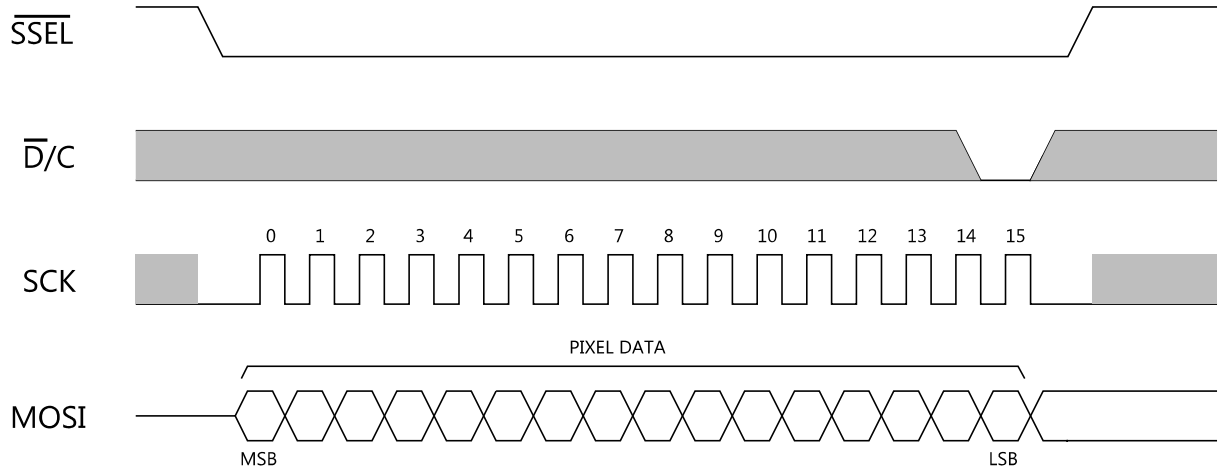


Fig. 4 Pixel Write

AC characteristics

Symbol	Parameter		Min.	Typ.	Max.	Unit

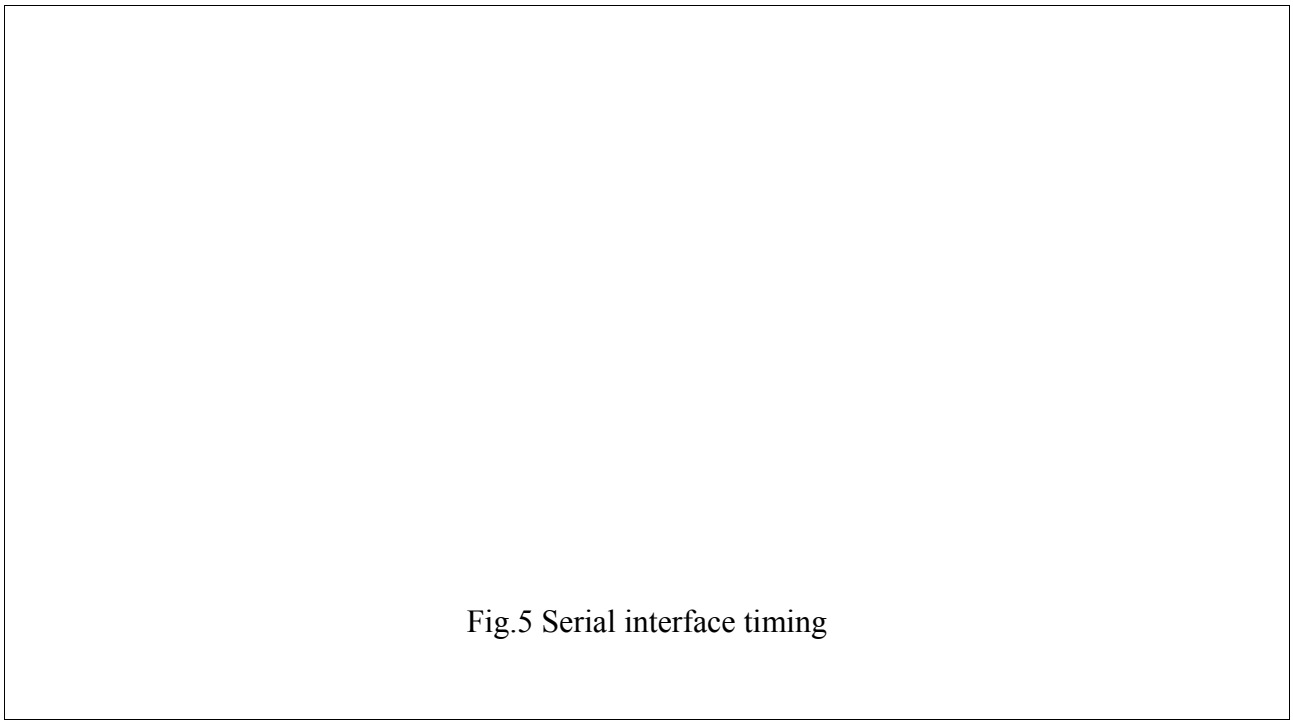


Fig.5 Serial interface timing

8. Commands

Introduction

A command consists of 5 bits of command identifier and 11 bits of data. The controller interprets the currently written frame as a command when the D/C signal is high.

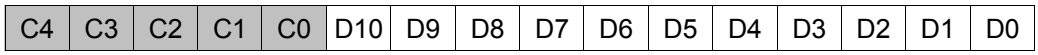


Fig. 6 Command format

Commands

Table 1

Command	C4	C3	C2	C1	C0		Description
SETADDRLO	0	0	0	0	1	0x01	Set RAM address – LSB 11 bits
SETADDRHI	0	0	0	1	0	0x02	Set RAM address – MSB 9 bits [1)]
SETFRAMEADDRLO	0	0	1	1	0	0x06	Set visible frame address – LSB 11 bits
SETFRAMEADDRHI	0	0	1	1	1	0x07	Set visible frame address – MSB 9 bits [1)]
SETWINDOWX1							
SETWINDOWY1							
SETWINDOWX2							
SETWINDOWY2							

Note:

1. Write LSB part of address first, set the 2 unused MSB bits of data part to 0

SETADDRLO

Sets the LSB 11 bits of the framebuffer address. SETADDRLO needs to be written before SETADDRHI.

SETADDRHI

Sets the MSB 9 bits of the framebuffer address. The unused 2 MSB bits of the data part should be set to 0. SETADDRLO needs to be written before SETADDRHI. Writing SETADDRHI triggers the controller to process the full 20 bit long address combined of the 11 bit LSB part and the 9 bit MSB part.

SETFRAMEADDRLO

Sets the LSB 11 bits of the visible frame start address. SETFRAMEADDRLO needs to be written before SETFRAMEADDRHI.

SETFRAMEADDRHI

Sets the MSB 9 bits of the visible frame start address. The unused 2 MSB bits of the data part should be set to 0. SETFRAMEADDRLO needs to be written before SETFRAMEADDRHI. Writing SETFRAMEADDRHI triggers the controller to process the full 20 bit long address combined of the 11 bit LSB part and the 9 bit MSB part.

The actual frame start address change after the SETFRAMEADDRHI command is synchronized with the next display VSYNC in the controller.

9. Framebuffer memory

Framebuffer memory addressing

The controller has 8 MB of memory available for storing the image data. The memory size of 4096 x 256 x 16 bits makes for a total of 1048576 different addresses, 16 bits of data per address. One frame requires 480 x 272 x 2 = 261120 bytes of memory. This means that it is possible to store 32 separate full frames of image data to the controller memory. Aligning the start address for each frame to the 512 byte boundaries results in a bit more efficient memory bandwidth use.

Images can be first drawn to a non visible memory area and then the frame start address can be changed to show the image.

It is also possible to implement hardware accelerated vertical scrolling by first filling the controller memory with video data and then just changing the frame start address to scroll the image.

The framebuffer address is automatically incremented after each written pixel.

Pixel format

The SPI043 controller uses the RGB565 pixel format. One pixel consists of 16 bits of data – 5 bits red, 6 bits green and 5 bits blue. This results in 65536 different colors.

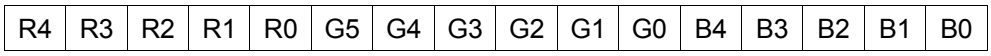


Fig. 7 Pixel format

10. Appendix A – Dimensional drawings