Updated 13/5/2009 Version 1.3a

omnima
Advanced lighting systems
Easy to use Seral/USB API
Programmable QVGA LCD Display
Click here to find out more ...

# Programmable
# NXP LPC2103 Solomon SSD1928
# LCD Display / Control Panel

**Technical overview**

## 1. Technical details

- NXP LPC2103  based board
- Solomon SSD1928 display controller
- 2 GPIO general purpose I/O ports / SPI or i2C interface
- 1x JTAG programming/debugging port
- SD card reader
- 256KB RAM
- Enclosure size: 100mm x 85mm
- RoHS / Leadfree

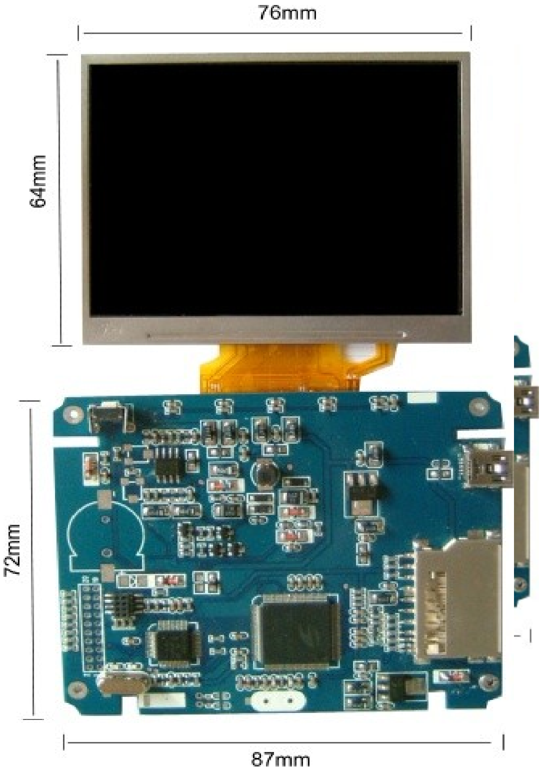
## 2. Main features

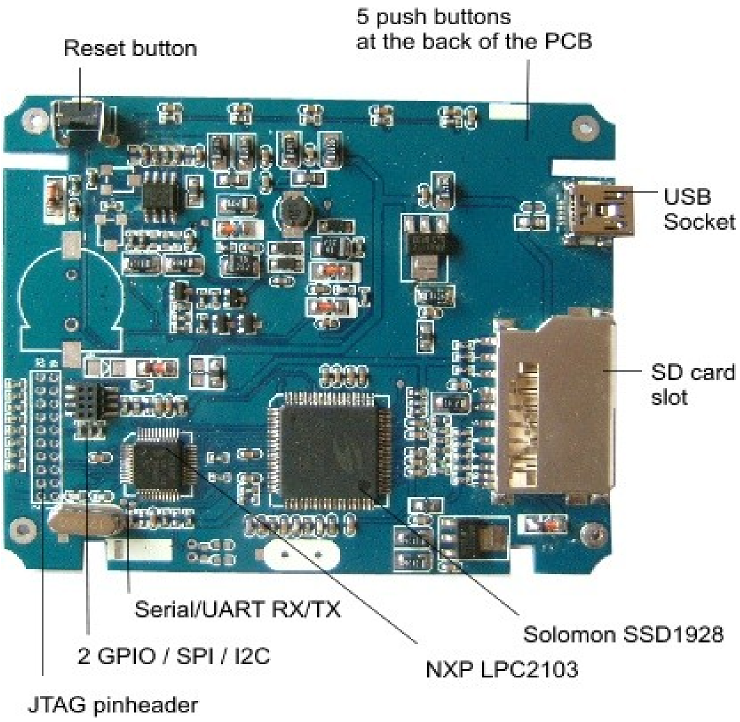
- Programmable LCD display
- USB port powered
- Solomon advanced graphics controller with JPEG decompression
- Expansion slot including 2 GPIOs for SPI / I2C connectivity
- Compact case with room for expansion PCBs
- Cost effective
- Simple to use serial/USB command-line interface
- Connects to all Windows and Linux computers

## 3. Package contents

- LCD display in plastic enclosure
- UK power supply adapter
- USB connectivity cable
- CD-Rom including Windows driver for the USB interface adapter

## 4. Board layout





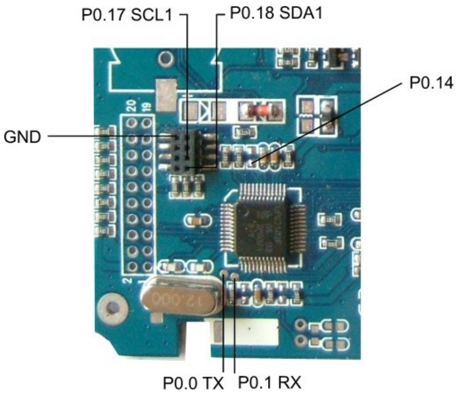
## CPU

CPU is an LPC2103 ARM7-based MCU. The CPU is clocked at 12MHz, so internally runs at 48MHz.

## Graphics controller

The SSD1928 is a fairly advanced display controller. It has a 2D graphics engine featuring panning, scrolling, rotation, line, rectangle and ellipse drawing. There are bit block transfers, and a hardware JPEG engine. The controller supports up to 320x240 in 32-bit colour.

## 5. Input/Output lines



| | |
|---|---|
| P0.17 | SPI SCL data line or GPIO pin |
| P0.18 | SPI SDA data line or GPIO pin |
| P0.14 | GPIO pin and ISP trigger pin |
| P0.0 TX | UART transmit pin or GPIO pin |
| P0.1 RX | UART receiver pin or GPIO pin |

## 6. JTAG interface

| | |
|---|---|
| 1: +3.3v supply * | 2: +3.3v supply |
| 3: 10k pull-up, ~TRST | 4: GND |
| 5: 10k pull-up, TDI | 6: GND |
| 7: 10k pull-up, TMS | 8: GND |
| 9: 10k pull-down, TCK | 10: GND |
| 11: 10k pull-down, RTCK | 12: GND |
| 13: 10k pull-up, TDO | 14: GND |
| 15: 10k pull-up, ~RST | 16: GND |

| | |
|---|---|
| 17: 10k pull-down,  n/c (DBGRQ) | 18: GND |
| 19: 10k pull-down, n/c ( DBGACK) | 20: GND |

* Specifically vref on the JTAG, but same as supply

This is a standard ARM JTAG pin-out and you can use a standard Macgraigor wiggler clone such as the one available from Jabs Place (see the reference section for contact info).

## 7. Power supply

The LCD is powered using a standard mini-USB connector and the DC power supply of +5V.

A UK adapter is included in the package.

## 8. Serial/USB command-line interface

Connection

- The PL2303 driver needs to be installed on computers running Windows. The driver is provided on the CD-Rom or USB dongle in the drivers folder.
- Linux kernel will usually be pre-configured to include the Prolific driver out of the box
- Serial connection configuration: 19200bps,8,N,1

Destination surfaces

MW - main window (320x240 fixed size, 16bit color)
FW - floating window (default size: 260x180, 16bit color)
SC - scratchpad (320x20, 16bit color - experimental)

Cursors

CR1 - cursor 1
CR2 - cursor 2

Built-in font

See the FontCharacters.txt file to the description and shape of characters included in this font.

Commands

- Commands are case sensitive
- Separators between parameters can be: (,) comma, (,) colon, (;) semicolon, ( ) space
- Parameters can be entered as decimal or hexadecimal numbers.
- Leading 0x is required for hexadecimal numbers i.e. use 0xff to specify 255 in decimal
- Commands should be followed by a <CR><LF> byte sequence

System commands

**#@Reboot**
Restarts the LCD

**#@EnterISP**
Restarts the LCD and enters the programming mode for 2 minutes. The LCD will restart automatically after 2 minutes. Use FlashMagic (http://www.flashmagictool.com/) to upload new firmware to the LCD over the same serial link.

System commands - continued

**#@Display On**
**#@Display Off**
Turns the display on/off

Window surfaces

There are four drawing surfaces provided.

Main window
The main window is fixed surface and is defined to be 320x240 pixels in 16bit color.
All 2D drawing routines can reference this surface as their source or destination as MW.

Floating window
The floating window overlaps the main window surface. It can be optionally shown or hidden and it's size and position changed. The main window is fixed surface and is defined to be 320x240 pixels in 16bit color. All 2D drawing routines can reference this surface as their source or destination as FW.

**#@FloatWin Open x,y,width,height**
[x,y - absolute location, width, height - specify the size of the window]
Example: #@FloatWin Open 10 10 200 200
Allocates new window (hides any existing floating window)

**#@FloatWin Show**
Enables and displays the floating window on the screen

**#@FloatWin Hide**
Hides the floating window

**#@FloatWin Move x,y**
Example1: #@FloatWin Move -10 -10
Example2: #@FloatWin Move 10 10
Moves the floating window in either positive or negative direction

Cursors

Cursors are intended to show small areas of text or graphics.
At present, only text and the characters from the buil-in font can be rendered and shown as cursors.
Cursors can be shown, hiden, moved and their blink ratio specified.
Cursors will overlap the main and floating windows when displayed on the screen.

**#@Cursor [CR1 or CR2] Open x,y,width,height**
Example: #@Cursor CR1 Open 50 50 320 20

Allocates new cursor (hides any previously displayed cursor)

**#@Cursor [CR1 or CR2] Show**
Example: #@Cursor CR1 Show
Enables and displays the cursor (CR1 or CR2) on the screen

**#@Cursor [CR1 or CR2] Hide**
Example: #@Cursor CR1 Hide
Hides the specified cursor (CR1 or CR2)

**#@Cursor [CR1 or CR2] Move x,y,totalTime,visibleTime**
Example: #@Cursor CR1 Move 10 10 100 60
Moves the specified cursor (CR1 or CR2) to the relative x,y location. The blink ratio for the cursor is specified by the last two parameters. The first blink parameter is the total display time. The second parameter specifies the proporation of total time during which the cursor is made visible.

**#@Cursor [CR1 or CR2] Text x,y,text**
#@Cursor CR1 Text 0 0 Hello!
Draws characters specified in this command on the cursor CR1 or CR2 surface

Drawing on surfaces

**#@FrColor Red,Green,Blue,Alpha**
Defines foreground color for subsequent drawing operations
Example: #@FrColor 255 255 255 0 - white color

**#@BkColor Red,Green,Blue,Alpha**
Defines background color for subsequent drawing operations
Example: #@BkColor 255 0 0 0 - red color

Sets the foreground and background drawing colour (Red,Green,Blue,Alpha)
Alpha is reserved for future use and the mode when the LCD is initialised in the 32bit drawing mode

**#@Text [MW or FW] Hor x,y,text**
Example: #@Text MW Hor 50 140 Hello World!
Draws text on (MW,FW) lacated a coordinates x,y and renders the provided text

**#@Arc [MW or FW] x,y,widht,height[,startAngle,endAngle]** - the arc start and end angle are optional
Example: #@Arc MW 150 20 5 5 0 360 - draws using frcolor

**#@Line MW x1,y1,x2,y2**
Example: #@Line MW 10 10 100 10 - draws using frcolor

**#@Rect [MW or FW] x1,y1,width,height**
Example: #@Rect MW 10 10 100 10 - draws using bkcolor, solid fill

**#@Blt [Source: MW or FW] x1,y1,width,height [Destination: MW or FW] x,y,[destWidth,destHeight]**
Copies images from source to destination, optionlly stratching the imge to fit the destinatio width and height
Example1: #@Blt MW 10 10 20 20 MW 50 50 50 50 - MW to MW copy, optional [50,50] provided to stretch the image
Example2: #@Blt FW 10 10 20 20 MW 50 50 - FW to MW copy, no stretch

**Response from the LCD**

**#@OKEYDN:{keynum}** - sent when key is pressed
**#@OKEYUP:{keynum}** - sent when key is released

keynum: 1-5

## 9. Development environment

For information on the development tools and further information please visit:

| | |
|---|---|
| EMB based JTAG cable | http://www.omnima.co.uk/forums/index.php?showtopic=96 |
| JTAG cable | http://jabsplace.co.uk/shop/index.php?main_page=product_info&cPath=18&products_id=117 |
| OpenOcd | http://openocd.berlios.de/web/ |
| GNU ARM C tools | http://www.gnuarm.com/ |
| NXP LPC2103 datasheet | http://www.nxp.com/acrobat/datasheets/LPC2101_02_03_2.pdf |
| SSD1928 datasheet | http://omnima.co.uk/docs/SSD1928_1.0.pdf |
| SSD192x API Gen | http://www.omnima.co.uk/forums/index.php?act=attach&type=post&id=66 |
| Sample C program | http://linux-adm5120.svn.sourceforge.net/viewvc/linux-adm5120/lpc2103/blink/ |