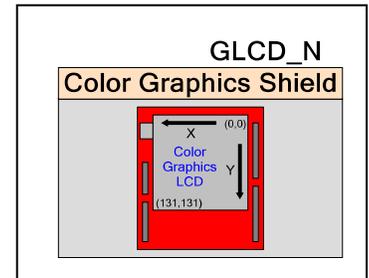


# Color Graphics Shield

0.3

## Features

- Supports Sparkfun's and other compatible Nokia 6100 color graphics shields.
- Compatible with Philips PCF8833 and Epson S1D15F20 controller (Not tested with Epson controller at this time.)
- Graphics Functions include Pixel, DrawLine, DrawCircle, DrawRect, PrintChar, and PrintString.
- Compatible with both PSoC 4 and PSoC 5LP
- Compatible with PSoC 4's SCB and UDB based SPI components.
- Uses 3-wire SPI plus one line for LCD reset.



## General Description

The Color Graphics Shield component was created to support the popular Sparkfun Color Graphics Arduino Shield that was based on the Nokia 6100 color LCD. It interfaces to any PSoC 4 or PSoC 5LP family controller. It is intended for the low cost development boards that are hardware compatible with Arduino Shields, such as the Cypress PSoC 4 Pioneer board (CY8CKIT-042) and the PSoC 5LP based FreeSoc Explorer board.

## Input/Output Connections

This section describes the various input and output connections for this component. All pins are hidden in the schematic but are present in the Design Wide Resource page.

### SCLK\* – Digital Output

This SCLK output carries the Serial Clock signal. It provides the master (PSoC) synchronization clock output to the slave (LCD Controller) device on the bus. This clock runs at about 6 MHz.

### MOSI\* – Digital Output

The MOSI (Master Out Slave In) is the signal used to transfer data from the SPI Master (PSoC) to the slave (LCD Controller). All display data and display commands are sent via this signal.

## SS – Digital Output

This signal goes low when data and commands are sent to the graphics LCD controller. In systems that contain more than one device, it allows a single SPI master to control multiple slaves.

## Reset – Analog Input

This signal is driven low after power up or reset to put the display controller into a know state before communication begins between the PSoC and the display controller.

### Connecting the PSoC to the display:

There are three basic configurations, two for PSoC 4 and one for PSoC 5LP. Tables 1 and 2 below show the connections between the PSoC and the display for these configurations.

**Table 1 Cypress PSoC 4 Pioneer Cy8CKIT-042 board connections**

Display Signal		PSoC 4 Pioneer Board CY8CKIT-042	Description
SPI_Type (UDB)	SPI_Type (SBC)		
SCLK	sclk_m	P0[6]	SPI Clock signal
MOSI	mosi_m	P3[0]	SPI Master Out Slave In
SS	ss3_m	P3[6]	SPI Slave Select
Reset	Reset	P2[6]	Display Reset
N/A	ss0_m, ss1_m, ss2_m	Don't care	Leave these signals disconnected.
N/A	miso_m	Don't care	Leave this signal unconnected.

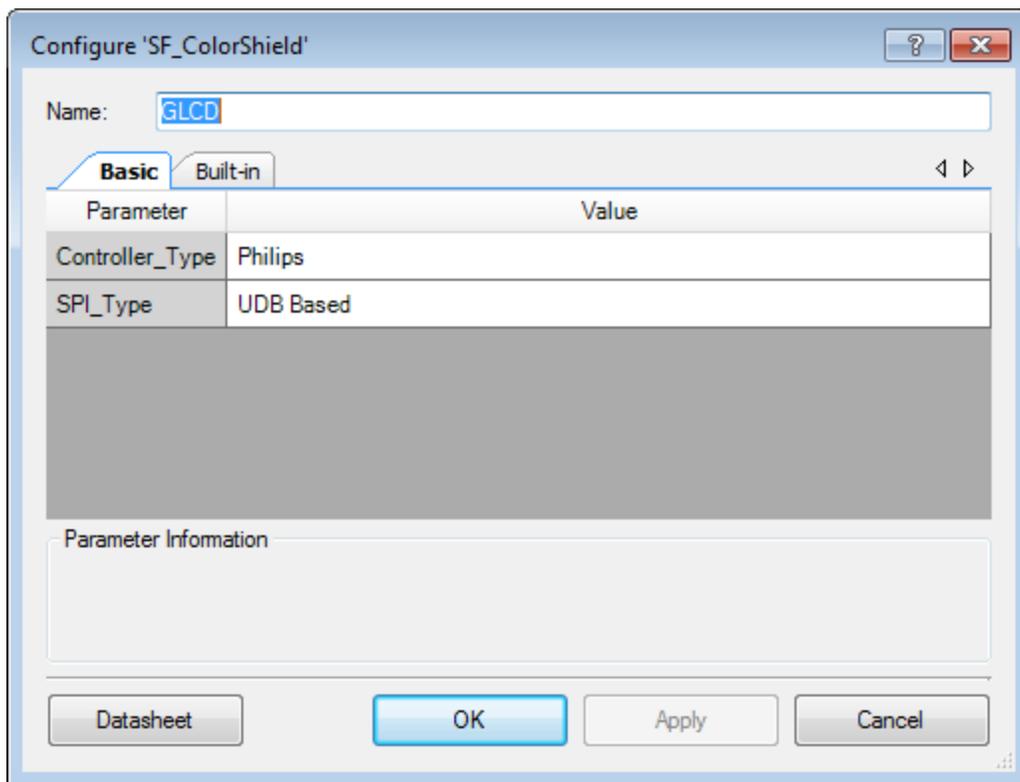
**Table 2 FreeSoC Explorer Board ( PSoC 5LP ) connections**

Display Signal Name	PSoC 5LP FreeSoC Explorer Board	Description
SCLK	P1[4]	SPI Clock signal
MOSI	P5[3]	SPI Master Out Slave In
SS	P3[6]	SPI Slave Select
Reset	P5[1]	Display Reset

## Parameters and Setup

Drag a Color Graphics Shield component onto your design and double-click it to open the Configure dialog.

**Figure 1 Configure Color Graphics Shield Dialog**



The Color Graphics Shield component provides the following parameters.

## Controller\_Type

The Nokia 6100 color LCD may contain one of two different display controllers, the Philips PCF8833 or Epson S1D15G10. This parameter allows the user to select the correct firmware for the display controller found on the user's display.

- **Philips**
- Epson

## SPI\_Type

This parameter is only valid when using a PSoC 4 device. It allows the user to select either a SCB or USB SPI master, depending on which resource is available.

- **UDB Based**
- SCB

## Resources

This component uses 2 UDBs when configured for a PSoC 5LP or a PSoC 4 in UDB mode. When configured for PSoC 4, in SCB mode, it consumes a single SCB block.

## Display Coordinate Orientation:

The Graphics display has a simple coordinate system with (0,0) in the upper left and (131,131) in the lower right. Most of these displays only show a matrix of 128x128 so the first and last 2 rows and columns are not visible. See the figure 2 below for coordinate system orientation.

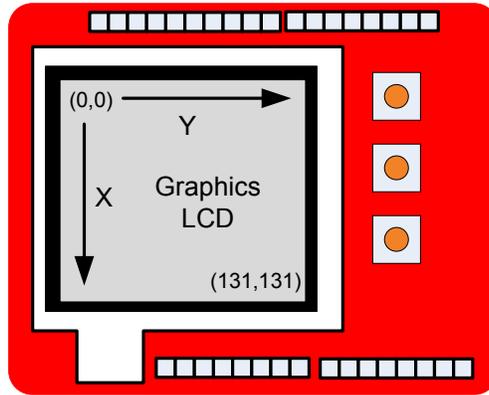


Figure 2 Display coordinate orientation of standard display

## Application Programming Interface

Application Programming Interface (API) routines allow you to control and configure the component using software. The following table list and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name "GLCD\_1" to the first instance of a component in a given design. You can rename the instance to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is "GLCD".

Function	Description
void GLCD_Start()	Starts the component and initializes the LCD.
void GLCD_Stop()	Clears the display and turns it off.
void GLCD_Clear()	Clears the display with a given color.
void GLCD_Contrast()	Used to set display contrast, default is 60.
void GLCD_Pixel()	Set a pixel at given location with a given color.

<b>Function</b>	<b>Description</b>
void GLCD_Draw_Circle()	Draws a circle at a given location, radius and color.
void GLCD_DrawLine()	Draws a line on the display from one point to another with a given color.
void GLCD_DrawRect()	Draw a filled or unfilled rectangle of a given color, defined by two points.
void GLCD_PrintChar()	Print a single character on the display at a given location and color.
void GLCD_PrintString()	Print a string of characters at a given location with the specified color.

## **void GLCD\_Start(void)**

- Description:** Resets and initializes the LCD. By default it clears the display to black and sets the contrast to 60.
- Parameters:** None
- Return Value:** None
- Side Effects:** None

## **void GLCD\_Stop(void)**

- Description:** Clears the display and turns off the LCD. This command does not affect the backlight.
- Parameters:** None
- Return Value:** None
- Side Effects:** None

## **void GLCD\_Clear(int32 color)**

- Description:** This command clears the display with a single given color.
- Parameters:** (int32) color: Color used to clear the display.
- Return Value:** None
- Side Effects:** All information on display will be lost.

## **void GLCD\_Contrast(int32 contrast)**

- Description:** Sets the display contrast with the given value. This command is usually used when the user has visual feedback while making this adjustment. The default value is 60.
- Parameters:** (int32) contrast: Contrast value.
- Return Value:** None
- Side Effects:** If this adjustment is set too high or low, the display will appear blank.

## **void GLCD\_Pixel(int32, x, int32 y, int32 color)**

- Description:** Set the pixel at location (x,y) with the given color.
- Parameters:** (int32) x: X display location  
(int32) y: Y display location.  
(int32) color: Color to set the pixel.
- Return Value:** None
- Side Effects:** None

**void GLCD\_DrawCircle(int32 x, int32 y, int32 radius, int32 color)**

**Description:** Draws a circle at the given (x,y) location with the given radius and color.

**Parameters:** (int32) (x,y): Center of circle.  
(int32) radius: Radius of the circle.  
(int32) color: Color to draw circle.

**Return Value:** None

**Side Effects:** None

**void GLCD\_DrawLine(int32 x0, int32 y0, int32 x1, int32 y1, int32 color)**

**Description:** Draws a line on the display from (x0,y0) to (x1,y1) with the given color.

**Parameters:** (int32) (x0,y0): Line endpoint one.  
(int32) (x1,y1): Line endpoint two.  
(int32) color: Line color.

**Return Value:** None

**Side Effects:** None

**void GLCD\_DrawRect(int32 x0, int32 y0, int32 x1, int32 y1, int32 fill, int32 color)**

**Description:** This function draws a rectangle defined by opposite corners (x0,y0) and (x1,y1). The rectangle can be either filled or empty.

**Parameters:** (int32) (x0,y0): Rectangle corner one.  
(int32) (x1,y1): Rectangle corner two.  
(int32) fill: If non-zero the rectangle is filled, if zero, rectangle will not be filled.  
(int32) color: Color to draw rectangle and optionally fill it.

**Return Value:** None

**Side Effects:** None

**void GLCD\_PrintChar(uint8 c, int32 x, int32 y, int32 fColor, int32 bColor)**

**Description:** This function prints a single character on the display with a given foreground and background color.

**Parameters:** (uint8) c: Character to be printed.  
(int32) (x,y): Upper left hand corner of character to be displayed.  
(int32) fColor: Actual color character will be displayed.  
(int32) bColor: Color of background when character is displayed.

**Return Value:** None

**Side Effects:** None

## void GLCD\_PrintString(char \*pString, int32 x, int32 y, int32 fColor, int32 bColor))

- Description:** This function prints a null terminated string on the display with a given foreground and background color.
- Parameters:** (char \*) pString: Null terminated string to be printed.  
 (int32) (x,y): Upper left hand corner of string to be displayed.  
 (int32) fColor: Actual color character will be displayed.  
 (int32) bColor: Color of background when character will be displayed
- Return Value:** None
- Side Effects:** None

## Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
0.3		Initial document

© Cypress Semiconductor Corporation, 2013. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

