



Micromega Corporation

Code Example uM-FPU64

Chip-on-Glass LCD Functions

Introduction

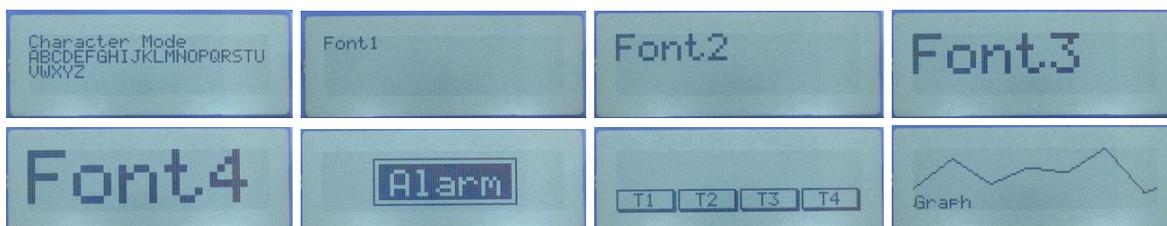
This code example provides FPU functions for driving a Newhaven Display 128x32 Chip-on-Glass (COG) LCD module (part# NHD-C12832A1Z). The 128x32 pixel black and white graphic display is interfaced to the FPU using an SPI interface. It has no built-in character mode or fonts.



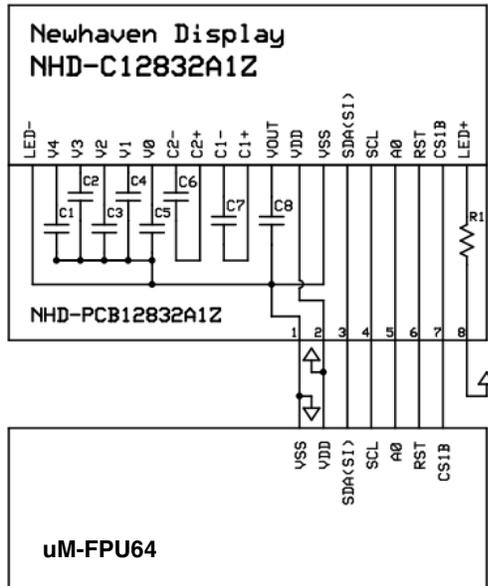
This code example provides FPU functions for displaying both graphic data and character data on the COG LCD. A display buffer of 128x32 bits is stored in RAM on the FPU. The drawing functions set or clear bits in the display buffer, not directly to the LCD. This allows for bit manipulation of the pixels and for multiple drawing operations to be done before updating the LCD. An update routine must be called to transfer the display buffer to the LCD. Font data is stored in Flash memory on the FPU. The font can be scaled up by the drawing routines to provide four font sizes. Using font size 1, the LCD can display test as 4 rows of 21 characters.

Sample Displays

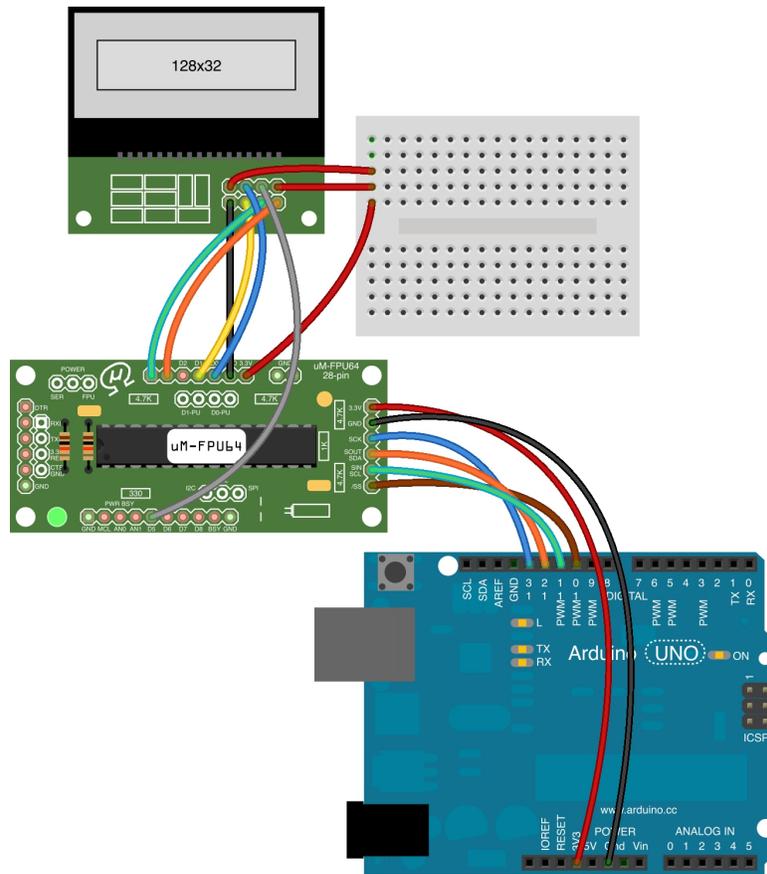
An Arduino test program is included with the code example. It produces the following sample displays.



Connecting the LCD to the uM-FPU64



uM-FPU64 28-pin Breakout Pins	NHD-PCB12832A1Z Prototype PCB Pins
D0	4 SCL
D1	3 SDA
D3	7 CS1B
D4	5 A0
D5	6 RST
GND	1 VSS
3.3V	2 VDD
3.3V	8 LED+



Made with Fritzing.org
(Fritzing file: *COG LCD Diagram.fzz*)

Menu Examples

FPU Functions

Arduino test file: *test_COG_LCD.ino*

FPU functions: *COG_LCD.fp4*

`COG_LCD_init`

Initializes the LCD display, clears the screen, and sets font size to 1 and pen type to `PEN_SET`.

`COG_LCD_test`

Shows various examples of calling the COG LCD functions. Sample displays are shown at the beginning of this document.

`COG_LCD_clear`

Clear the display buffer and updates the display.

`COG_LCD_drawClear`

Clears the display buffer.

`COG_LCD_penType(long)`

Sets the pen type.

<code>PEN_SET</code>	Set bits
<code>PEN_CLEAR</code>	Clear bits
<code>PEN_TOGGLE</code>	Toggle bits

`COG_LCD_fontSize(long)`

Sets the font size (1 to 4).

`COG_LCD_blackOnWhite`

Sets display for black bits on a white background.

`COG_LCD_whiteOnBlack`

Sets the display for white bits on a black background.

`COG_LCD_move(row, column)`

Moves the character position to the row and column specified.

`COG_LCD_print`

Displays the current string selection or the entire string buffer at the current character position.

`COG_LCD_drawLine(x1, y1, x2, y2)`

Draws a line in the display buffer from $x1,y1$ to $x2,y2$.

`COG_LCD_drawRect(x, y, width, height)`

Draws a rectangle of *width* and *height* at location x,y in the display buffer.

`COG_LCD_fillRect(x, y, width, height)`

Fills a rectangle of *width* and *height* at location x,y in the display buffer.

`COG_LCD_drawString(x, y)`

Draws the current string selection or the entire string buffer at location x,y in the display buffer.

`COG_LCD_writeChar(x, y, character)`

Draws a character at location x,y in the display buffer. It reads the font data stored in Flash a column at a time, and calls `COG_LCD_writeCharColumn` for each column of the character.

`COG_LCD_writeCharColumn(x, y, bitValue)`

It takes the bit value for the character column and expands it if the font size is greater than 1, then calls `COG_LCD_writeColumn` to store the *bitValue* to the display buffer.

`COG_LCD_writeColumn(x, bitValue, repeatCount)`

Stores the *bitValue* to the display buffer. If *repeatCount* is greater than zero, the *bitValue* is stored in sequential column of the display buffer.

`COG_LCD_update`

Updates the LCD by transferring the display buffer to the LCD.

`font_data`

A table of data defining the characters in the default font.

uM-FPU64 Features shown in Code Example

- accessing data in RAM using pointers
- accessing data in Flash memory using pointers
- interfacing COG LCD display using the DEVIO, SPI instruction

Further Information

See the Micromega website (www.micromegacorp.com) for additional information regarding the uM-FPU64 floating point coprocessor, including:

uM-FPU64 Datasheet

uM-FPU64 Instruction Set

uM-FPU64 IDE User Manual

uM-FPU64 IDE Compiler Manual

NHD-C12832A1Z-NSW-BBW-3V3 128x32 COG LCD datasheet, Newhaven Displays

NHD-PCB12832A1Z Prototyping PCB datasheet, Newhaven Displays