GLCD Library for Pinguino32X version 1.7-32X

The port for pinguino, was based on the original version with some modifications based on versions 1.1 and 2.0 of GLCD library for Arduino.

This is a list of functions supported by the library

GLCD.Init(invert) initialize the library for normal or inverted drawing. If invert is false, drawing sets pixels, if true pixels are cleared when drawn (see also SetInverted method).

GLCD.GotoXY(x,y) locate the graphic cursor at positions x and y, 0,0 is upper left corner.

GLCD.ClearScreenX() clear the LCD screen

GLCD.ClearScreen(color) clear the LCD screen with color (BLACK/WHITE)

Graphic Drawing Functions (color WHITE clears pixels, BLACK sets pixels)

GLCD.DrawCircle(x, y, radius, color) draw circle with center at x,y

GLCD.DrawLine(x1,y1,x2,y2,color) draw line from x1,y1 to x2,y2

GLCD.DrawVertLine(x, y, length, color) draw vertical line

GLCD.DrawHoriLine(x, y, length, color) draw horizontal line

GLCD.DrawRect(x, y, width, height, color) draw rectangle

GLCD.DrawRoundRect(x, y, width, height, radius, color) as above with rounded edges

GLCD.FillRect(x, y, width, height, color) draw filled rectangle

GLCD.InvertRect(x, y, width, height) invert pixels within given rectangle

GLCD.SetInverted(invert) set drawing mode to inverted

GLCD.SetDot(x, y, color); draw a dot in the given color at the given location

GLCD.DrawBitmap(bitmap, x, y, color); draw the bitmap at the given x,y position

Font Functions
GLCD.SelectFont(font, color) select font, defaults color to black if not specified

GLCD.PutChar(character) print given character to screen at current cursor location

GLCD.Puts(string) print given string to screen at current cursor location

GLCD.Puts_P(string) print string from program memory to screen at current cursor location

GLCD.PrintNumber(number) print the integer value of the given number at current cursor location

GLCD.PrintFloat(number, precision) print the decimal value of the given number at current cursor location

GLCD.CursorTo(x, y); // 0 based coordinates for fixed width fonts (i.e. the supplied system font)

GLCD Panel Pinouts Wiring and Configuration							
Pinguino32X PINS Minimum/UBW32	Function	Pinouts				Comments	
		A	B	C	D		
5V	+5 volts	1(+)	!2!(-)	!2!(-)	!2!(-)		
Gnd	GND	2(-)	!1!(+)	!1!(+)	!1!(+)		
external	Contrast in	3	3	3	3	Wiper of contrast tripot	
40/D0	D0	4	7	7	7	Data bits (1byte) ↔ PIC32 PORTD[1]	
43/D1	D1	5	8	8	8		
44/D2	D2	6	9	9	9		
45/D3	D3	7	10	10	10		
48/D4	D4	8	11	11	11		
49/D5	D5	9	12	12	12		
50/D6	D6	10	13	13	13		
51/D7	D7	11	14	14	14		
33/A5	CSEL1	12	15	16	15	Chip 1 select	
34/A14	CSEL2	13	16	15	18	Chip 2 select	
Reset	Reset	14	17	17	16	Connect to VPP/MCLR (between 10K and reset button)	
31/A3	RW	15	5	5	5	Read/write	
30/A2	DI (aka RS)	16	4	4	4	Data/Instruction	
32/A4	EN	17	6	6	6	Enable	
external	Contrast out	18	18	18	17	10k or 20k preset	
external	Backlight +5	19	19	19	19	100 to 330 ohm resistor to +5v	
GND	Backlight GND	20	20	20	20		

1- PORTD using RD0~RD7 pins as GLCD 8bits data port.

Pinout A panels:

- HDM64GS12L-4
- Crystalfontz CFAG12864B (tested by biomed)
- Sparkfun LCD-00710CM (tested by biomed)
- NKC Electronics LCD-0022 (tested by NKC Electronics)

Pinout B panels:

- HDM64GS12L-5
- Lumex LCM-S12864GSF (tested by jowan)
- Futurlec BLUE128X64LCD (tested by tyggerjai)
- AZ Displays AGM1264F (tested by santy)
- Displaytech 64128A BC (tested by Udo Klein)
- Adafruit GLCD (Leave RESET pin disconnected or you may experience upload problems) (tested by Things)
- DataVision DG12864-88 (tested by wglover)
- Topway LM12864LDW (tested by zandaa)
- Satistronics RT12864J-1 (tested by doublet) (tested with pinguino32X by anunakin)
- Digitron SG12864J4 (also appears to need RESET disconnected for uploads)

Pinout C panels:

- Shenzhen Jinghua Displays Co Ltd. JM12864 (tested by macpod)
  - Vee (pin 3) should be left disconnected. The pot on the display controls contrast
  - Backlight LED may already have resistors added.

Pinout D panels:

- TECH12864g (tested by anunakin) (tested with pinguino32X by anunakin)

User Defines on GLCD library

We can use some user defines on .pde files:

#define USEFLOATS()
This gives us
GLCD.PrintFloat(number,decimal_part) print the decimal value of the given number at current cursor location.

Use of float function uses a lot of flash memory.

#define SLOWDISPLAY()
This changes code to use a bit slow delays for Enable/RS/RW pins, with it you can use slow displays. If your ks0108 display not working this can help you.

We got about 13 FPS with glcd_sample.pde using SLOWDISPLAY() and 17FPS without it.

Library Project Files

tools/pic32mx/include/pinguino/
- ks0108.c
- ks0108.h

lib/
- ks0108.pdl32

sketch folder (where samples goes)
examples/glcd/
- GLCD_Clock.pde
- glcd_sample.pde
- Arial14.h
- ArialBold14.h
- Corsiva12.h
- ks0108.c
- Pinguino.h
- SystemFont5x7.h
- VerdanaBold28.h
- VerdanaNumbers.h
- Winks.h

doc/
- README.pdf
- README.odt

Convert a font to compatible header file
- utils/GLCDFontCreator2.zip

Convert a 1bit bitmap image to compliant header file
- utils/Processing Bitmap 1bit Converter.zip

オープンソースコード いきかた!