device

Overview on supported devices and displays

Updated Jun 17, 2014 by olikr...@gmail.com

Supported Devices

Controller	Display/Size	Mode	Set Contrast	Page Size	Com	Device	C++ Constructor	Arduino	Arduino Due	Chipkit	AVR
SSD1325	NHD-27-12864, 128x64	BW	no	128 Bytes	SW SPI	`u8g_dev_ssd1325_nhd27oled_bw_sw_spi`	`U8GLIB_NHD27OLED_BW(sck, mosi, cs, a0 [, reset])`	verified	verified	impl.	impl.
					HW SPI	`u8g_dev_ssd1325_nhd27oled_bw_hw_spi`	`U8GLIB_NHD27OLED_BW(cs, a0 [, reset])`	verified	v1.14	n.a.	impl.
				256 Bytes	SW SPI	`u8g_dev_ssd1325_nhd27oled_2x_bw_sw_spi`	`U8GLIB_NHD27OLED_2X_BW(sck, mosi, cs, a0 [, reset])`	verified	verified	impl.	impl.
					HW SPI	`u8g_dev_ssd1325_nhd27oled_2x_bw_hw_spi`	`U8GLIB_NHD27OLED_2X_BW(cs, a0 [, reset])`	verified	v1.14	n.a.	impl.
		4L	yes	128 Bytes	SW SPI	`u8g_dev_ssd1325_nhd27oled_gr_sw_spi`	`U8GLIB_NHD27OLED_GR(sck, mosi, cs, a0 [, reset])`	verified	impl.	impl.	impl.
					HW SPI	`u8g_dev_ssd1325_nhd27oled_gr_hw_spi`	`U8GLIB_NHD27OLED_GR(cs, a0 [, reset])`	verified	v1.14	n.a.	impl.
				256 Bytes	SW SPI	`u8g_dev_ssd1325_nhd27oled_2x_gr_sw_spi`	`U8GLIB_NHD27OLED_2X_GR(sck, mosi, cs, a0 [, reset])`	verified	impl.	impl.	impl.
					HW SPI	`u8g_dev_ssd1325_nhd27oled_2x_gr_hw_spi`	`U8GLIB_NHD27OLED_2X_GR(cs, a0 [, reset])`	verified	v1.14	n.a.	impl.
SSD1327	96x96 OLED (Seeedstudio)	4L	no	96 Bytes	SW SPI	`u8g_dev_ssd1327_96x96_gr_sw_spi`	`U8GLIB_SSD1327_96X96_GR(sck, mosi, cs, a0 [, reset])`	impl.	impl.	impl.	impl.
					HW SPI	`u8g_dev_ssd1327_96x96_gr_hw_spi`	`U8GLIB_SSD1327_96X96_GR(cs, a0 [, reset])`	impl.	v1.14	n.a.	impl.
					I2C	`u8g_dev_ssd1327_96x96_gr_i2c`	`U8GLIB_SSD1327_96X96_GR(U8G_I2C_OPT_NONE)`	verified	n.a.	n.a.	verified
				192 Bytes	SW SPI	`u8g_dev_ssd1327_96x96_2x_gr_sw_spi`	`U8GLIB_SSD1327_96X96_2X_GR(sck, mosi, cs, a0 [, reset])`	impl.	impl.	impl.	impl.
					HW SPI	`u8g_dev_ssd1327_96x96_2x_gr_hw_spi`	`U8GLIB_SSD1327_96X96_2X_GR(cs, a0 [, reset])`	impl.	v1.14	n.a.	impl.
					I2C	`u8g_dev_ssd1327_96x96_2x_gr_i2c`	`U8GLIB_SSD1327_96X96_2X_GR(U8G_I2C_OPT_NONE)`	impl.	n.a.	n.a.	impl.
SSD1322	NHD-3.12-25664, 256x64	4L	yes	256 Bytes	SW SPI	`u8g_dev_ssd1322_nhd31oled_gr_sw_spi`	`U8GLIB_NHD31OLED_GR(sck, mosi, cs, a0 [, reset])`	impl.	impl.	impl.	impl.
					HW SPI	`u8g_dev_ssd1322_nhd31oled_gr_hw_spi`	`U8GLIB_NHD31OLED_GR(cs, a0 [, reset])`	impl.	v1.14	n.a.	impl.
SSD1306	128x64 (Adafruit)	BW	no	128 Bytes	SW SPI	`u8g_dev_ssd1306_128x64_sw_spi`	`U8GLIB_SSD1306_128X64(sck, mosi, cs, a0 [, reset])`	verified	impl.	impl.	impl.
					HW SPI	`u8g_dev_ssd1306_128x64_hw_spi`	`U8GLIB_SSD1306_128X64(cs, a0 [, reset])`	verified	v1.14	n.a.	impl.
					I2C	`u8g_dev_ssd1306_128x64_i2c`	`U8GLIB_SSD1306_128X64(U8G_I2C_OPT_NONE)`	verified	n.a.	n.a.	impl.
				256 Bytes	SW SPI	`u8g_dev_ssd1306_128x64_2x_sw_spi`	`U8GLIB_SSD1306_128X64_2X(sck, mosi, cs, a0 [, reset])`	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	`u8g_dev_ssd1306_128x64_2x_hw_spi`	`U8GLIB_SSD1306_128X64_2X(cs, a0 [, reset])`	impl. v1.13	v1.14	n.a.	impl. v1.13
					I2C	`u8g_dev_ssd1306_128x64_2x_i2c`	`U8GLIB_SSD1306_128X64_2X(U8G_I2C_OPT_NONE)`	impl. v1.13	n.a.	n.a.	impl. v1.13
	128x32	BW	no	128 Bytes	SW SPI	`u8g_dev_ssd1306_128x32_sw_spi`	`U8GLIB_SSD1306_128X32(sck, mosi, cs, a0 [, reset])`	impl.	impl.	impl.	impl.
					HW SPI	`u8g_dev_ssd1306_128x32_hw_spi`	`U8GLIB_SSD1306_128X32(cs, a0 [, reset])`	impl.	v1.14	n.a.	impl.
					I2C	`u8g_dev_ssd1306_128x32_i2c`	`U8GLIB_SSD1306_128X32(U8G_I2C_OPT_NONE)`	impl.	n.a.	n.a.	impl.
				256 Bytes	SW SPI	`u8g_dev_ssd1306_128x32_2x_sw_spi`	`U8GLIB_SSD1306_128X32_2X(sck, mosi, cs, a0 [, reset])`	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	`u8g_dev_ssd1306_128x32_2x_hw_spi`	`U8GLIB_SSD1306_128X32_2X(cs, a0 [, reset])`	impl. v1.13	v1.14	n.a.	impl. v1.13
					I2C	`u8g_dev_ssd1306_128x32_2x_i2c`	`U8GLIB_SSD1306_128X32_2X(U8G_I2C_OPT_NONE)`	impl. v1.13	n.a.	n.a.	impl. v1.13
SH1106	128x64	BW	no	128 Bytes	SW SPI	`u8g_dev_sh1106_128x64_sw_spi`	`U8GLIB_SH1106_128X64(sck, mosi, cs, a0 [, reset])`	verified	impl.	impl.	impl.
					HW SPI	`u8g_dev_sh1106_128x64_hw_spi`	`U8GLIB_SH1106_128X64(cs, a0 [, reset])`	verified	impl.	n.a.	impl.
					I2C	`u8g_dev_sh1106_128x64_i2c`	`U8GLIB_SH1106_128X64(U8G_I2C_OPT_NONE)`	verified	n.a.	n.a.	impl.
				256 Bytes	SW SPI	`u8g_dev_sh1106_128x64_2x_sw_spi`	`U8GLIB_SH1106_128X64_2X(sck, mosi, cs, a0 [, reset])`	impl.	impl.	impl.	impl.
					HW SPI	`u8g_dev_sh1106_128x64_2x_hw_spi`	`U8GLIB_SH1106_128X64_2X(cs, a0 [, reset])`	impl.	impl.	n.a.	impl.
					I2C	`u8g_dev_sh1106_128x64_2x_i2c`	`U8GLIB_SH1106_128X64_2X(U8G_I2C_OPT_NONE)`	impl.	n.a.	n.a.	impl.
SSD1309	128x64	BW	yes	128 Bytes	SW SPI	`u8g_dev_ssd1309_128x64_sw_spi`	`U8GLIB_SSD1309_128X64(sck, mosi, cs, a0 [, reset])`	impl. (v1.10)	impl. v1.11	impl. (v1.10)	impl. (v1.10)
					HW SPI	`u8g_dev_ssd1309_128x64_hw_spi`	`U8GLIB_SSD1309_128X64(cs, a0 [, reset])`	impl. (v1.10)	v1.14	n.a.	impl. (v1.10)
					I2C	`u8g_dev_ssd1309_128x64_i2c`	`U8GLIB_SSD1309_128X64(U8G_I2C_OPT_NONE)`	impl. (v1.10)	n.a.	n.a.	impl. (v1.10)
					SW	`u8g_dev_uc1701_dogs102_sw_spi`	`U8GLIB_DOGS102(sck, mosi, cs, a0 [, reset])`	verified	impl.	impl.	impl.

UC1701	DOGS102, 102x64	BW	impl. v1.09	102 Bytes	HW SPI	u8g_dev_uc1701_dogs102_hw_spi	U8GLIB_DOGS102(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
				204 Bytes	SW SPI	u8g_dev_uc1701_dogs102_2x_sw_spi	U8GLIB_DOGS102_2X(sck, mosi, cs, a0 [, reset])	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_uc1701_dogs102_2x_hw_spi	U8GLIB_DOGS102_2X(cs, a0 [, reset])	impl. v1.13	v1.14	n.a.	impl. v1.13
	MINI12864, 128x64	BW	impl. v1.09	128 Bytes	SW SPI	u8g_dev_uc1701_mini12864_sw_spi	U8GLIB_MINI12864(sck, mosi, cs, a0 [, reset])	impl.	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_uc1701_mini12864_hw_spi	U8GLIB_MINI12864(cs, a0 [, reset])	impl.	v1.14	n.a.	impl.
				256 Bytes	SW SPI	u8g_dev_uc1701_mini12864_2x_sw_spi	U8GLIB_MINI12864_2X(sck, mosi, cs, a0 [, reset])	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_uc1701_mini12864_2x_hw_spi	U8GLIB_MINI12864_2X(cs, a0 [, reset])	impl. v1.13	v1.14	n.a.	impl. v1.13
UC1601	c128032, 128x32	BW	impl. v1.13	128 Bytes	SW SPI	u8g_dev_uc1601_c128032_sw_spi	U8GLIB_UC1601_C128032(sck, mosi, cs, a0 [, reset])	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_uc1601_c128032_hw_spi	U8GLIB_UC1601_C128032(cs, a0 [, reset])	impl. v1.13	v1.14	n.a.	impl. v1.13
				256 Bytes	SW SPI	u8g_dev_uc1601_c128032_2x_sw_spi	U8GLIB_UC1601_C128032_2X(sck, mosi, cs, a0 [, reset])	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_uc1601_c128032_2x_hw_spi	U8GLIB_UC1601_C128032_2X(cs, a0 [, reset])	impl. v1.13	v1.14	n.a.	impl. v1.13
UC1608	240x64	BW	yes	240 Bytes	SW SPI	u8g_dev_uc1608_240x64_sw_spi	U8GLIB_UC1608_240X64(sck, mosi, cs, a0 [, reset])	impl. v1.15	impl. v1.15	impl. v1.15	impl. v1.15
					HW SPI	u8g_dev_uc1608_240x64_hw_spi	U8GLIB_UC1608_240X64(cs, a0 [, reset])	impl. v1.15	impl. v1.15	n.a.	impl. v1.15
				256 Bytes	SW SPI	u8g_dev_uc1608_240x64_2x_sw_spi	U8GLIB_UC1608_240X64_2X(sck, mosi, cs, a0 [, reset])	impl. v1.15	impl. v1.15	impl. v1.15	impl. v1.15
					HW SPI	u8g_dev_uc1608_240x64_2x_hw_spi	U8GLIB_UC1608_240X64_2X(cs, a0 [, reset])	impl. v1.15	impl. v1.15	n.a.	impl. v1.15
ST7565	DOGM128, 128x64	BW	impl. v1.09	128 Bytes	SW SPI	u8g_dev_st7565_dogm128_sw_spi	U8GLIB_DOGM128(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_st7565_dogm128_hw_spi	U8GLIB_DOGM128(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
				256 Bytes	SW SPI	u8g_dev_st7565_dogm128_2x_sw_spi	U8GLIB_DOGM128_2X(sck, mosi, cs, a0 [, reset])	verified	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_st7565_dogm128_2x_hw_spi	U8GLIB_DOGM128_2X(cs, a0 [, reset])	verified v1.13	v1.14	n.a.	impl. v1.13
	LM6059, 128x64, Adafruit	BW	impl. v1.09	128 Bytes	SW SPI	u8g_dev_st7565_lm6059_sw_spi	U8GLIB_LM6059(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_st7565_lm6059_hw_spi	U8GLIB_LM6059(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
				256 Bytes	SW SPI	u8g_dev_st7565_lm6059_2x_sw_spi	U8GLIB_LM6059_2X(sck, mosi, cs, a0 [, reset])	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_st7565_lm6059_2x_hw_spi	U8GLIB_LM6059_2X(cs, a0 [, reset])	impl. v1.13	v1.14	n.a.	impl. v1.13
	LM6063, 128x64	BW	impl. v1.09	128 Bytes	SW SPI	u8g_dev_st7565_lm6063_sw_spi	U8GLIB_LM6063(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_st7565_lm6063_hw_spi	U8GLIB_LM6063(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
				256 Bytes	SW SPI	u8g_dev_st7565_lm6063_2x_sw_spi	U8GLIB_LM6063_2X(sck, mosi, cs, a0 [, reset])	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_st7565_lm6063_2x_hw_spi	U8GLIB_LM6063_2X(cs, a0 [, reset])	impl. v1.13	v1.14	n.a.	impl. v1.13
	NHD C12864, 128x64	BW	impl. v1.09	128 Bytes	SW SPI	u8g_dev_st7565_nhd_c12864_sw_spi	U8GLIB_NHD_C12864(sck, mosi, cs, a0 [, reset])	verified, v1.08	impl. v1.11	impl. v1.08	impl. v1.08
					HW SPI	u8g_dev_st7565_nhd_c12864_hw_spi	U8GLIB_NHD_C12864(cs, a0 [, reset])	verified, v1.08	v1.14	n.a.	impl.
				256 Bytes	SW SPI	u8g_dev_st7565_nhd_c12864_2x_sw_spi	U8GLIB_NHD_C12864_2X(sck, mosi, cs, a0 [, reset])	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_st7565_nhd_c12864_2x_hw_spi	U8GLIB_NHD_C12864_2X(cs, a0 [, reset])	impl. v1.13	v1.14	n.a.	impl. v1.13
	Displaytech 64128n, 128x64	BW	impl. v1.09	128 Bytes	SW SPI	u8g_dev_st7565_64128n_sw_spi	U8GLIB_64128N(sck, mosi, cs, a0 [, reset])	verified, v1.09	impl. v1.11	impl. v1.09	impl. v1.09
					HW SPI	u8g_dev_st7565_64128n_hw_spi	U8GLIB_64128N(cs, a0 [, reset])	verified v1.09	v1.14	n.a.	impl. v1.09
				256 Bytes	SW SPI	u8g_dev_st7565_64128n_2x_sw_spi	U8GLIB_64128N_2X(sck, mosi, cs, a0 [, reset])	impl. v1.13	impl. v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_st7565_64128n_2x_hw_spi	U8GLIB_64128N_2X(cs, a0 [, reset])	impl. v1.13	v1.14	n.a.	impl. v1.13
	DOGM132, 132x32	BW	impl. v1.09	132 Bytes	SW SPI	u8g_dev_st7565_dogm132_sw_spi	U8GLIB_DOGM132(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_st7565_dogm132_hw_spi	U8GLIB_DOGM132(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
	NHD C12832, 128x32	BW	impl.	128 Bytes	SW SPI	u8g_dev_st7565_nhd_c12832_sw_spi	U8GLIB_NHD_C12832(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_st7565_nhd_c12832_hw_spi	U8GLIB_NHD_C12832(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
				128 Bytes	SW SPI	u8g_dev_st7920_128x64_sw_spi	U8GLIB_ST7920_128X64_1X(sck, mosi, cs [, reset]) U8GLIB_ST7920_128X64(sck, mosi, cs, a0 [, reset]) (OLD)	verified	impl. v1.11	impl.	impl.

					HW SPI	u8g_dev_st7920_128x64_hw_spi	U8GLIB_ST7920_128X64_1X(cs [, reset])
							U8GLIB_ST7920_128X64(cs, a0 [, reset]) (OLD)
								impl. v1.09	n.a.	n.a.	impl. v1.09

							U8GLIB_ST7920_128X64_1X(d0, d1, d2, d3, d4,				

ST7920	128x64, DFRobot	BW	n.a.		8 Bit	u8g_dev_st7920_128x64_8bit	d5, d6, d7, en, di, rw [, reset]) U8GLIB_ST7920_128X64(d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw [, reset]) (OLD)	impl.	impl. v1.11	n.a.	impl.
				512 Bytes	SW SPI	u8g_dev_st7920_128x64_4x_sw_spi	U8GLIB_ST7920_128X64_4X(sck, mosi, cs [, reset])	impl. v1.11	impl. v1.11	impl. v1.11	impl. v1.11
					HW SPI	u8g_dev_st7920_128x64_4x_hw_spi	U8GLIB_ST7920_128X64_4X(cs [, reset])				
					8 Bit	u8g_dev_st7920_128x64_4x_8bit	U8GLIB_ST7920_128X64_4X(d0, d1, d2, d3, d4, d5, d6, d7, en, di, rw [, reset])				
	192x32	BW	n.a.	192 Bytes	SW SPI	u8g_dev_st7920_192x32_sw_spi	U8GLIB_ST7920_192X32_1X(sck, mosi, cs [, reset]) U8GLIB_ST7920_192X32(sck, mosi, cs, a0 [, reset]) (OLD)	verified	impl. v1.11	impl.	verified

					HW SPI	u8g_dev_st7920_192x32_hw_spi	U8GLIB_ST7920_192X32_1X(cs [, reset])
							U8GLIB_ST7920_192X32(cs, a0 [, reset]) (OLD)
								impl. v1.09	n.a.	n.a.	impl. v1.09

					8 Bit	u8g_dev_st7920_192x32_8bit	U8GLIB_ST7920_192X32_1X(d0, d1, d2, d3, d4, d5, d6, d7, en, di, rw [, reset])
							U8GLIB_ST7920_192X32(d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw [, reset]) (OLD)
								verified	impl. v1.11	n.a.	impl.

				768 Bytes	SW SPI	u8g_dev_st7920_192x32_4x_sw_spi	U8GLIB_ST7920_192X32_4X(sck, mosi, cs [, reset])	impl. v1.11	impl. v1.11	impl. v1.11	impl. v1.11
					HW SPI	u8g_dev_st7920_192x32_4x_hw_spi	U8GLIB_ST7920_192X32_4X(cs [, reset])				
					8 Bit	u8g_dev_st7920_192x32_4x_8bit	U8GLIB_ST7920_192X32_4X(d0, d1, d2, d3, d4, d5, d6, d7, en, di, rw [, reset])				
	202x32	BW	n.a.	202 Bytes	SW SPI	u8g_dev_st7920_202x32_sw_spi	U8GLIB_ST7920_202X32_1X(sck, mosi, cs [, reset]) U8GLIB_ST7920_202X32(sck, mosi, cs, a0 [, reset]) (OLD)	impl. v1.08	impl. v1.11	impl. v1.08	impl. v1.08

					HW SPI	u8g_dev_st7920_202x32_hw_spi	U8GLIB_ST7920_202X32_1X(cs [, reset])
							U8GLIB_ST7920_202X32(cs, a0 [, reset]) (OLD)
								impl. v1.09	n.a.	n.a.	impl. v1.09

					8 Bit	u8g_dev_st7920_202x32_8bit	U8GLIB_ST7920_202X32_1X(d0, d1, d2, d3, d4, d5, d6, d7, en, di, rw [, reset])
							U8GLIB_ST7920_202X32(d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw [, reset]) (OLD)
								impl. v1.08	impl. v1.11	impl. v1.08	impl. v1.08

				808 Bytes	SW SPI	u8g_dev_st7920_202x32_4x_sw_spi	U8GLIB_ST7920_202X32_4X(sck, mosi, cs [, reset])	impl. v1.11	impl. v1.11	impl. v1.11	impl. v1.11
					HW SPI	u8g_dev_st7920_202x32_4x_hw_spi	U8GLIB_ST7920_202X32_4X(cs [, reset])				
					8 Bit	u8g_dev_st7920_202x32_4x_8bit	U8GLIB_ST7920_202X32_4X(d0, d1, d2, d3, d4, d5, d6, d7, en, di, rw [, reset])				
UC1610	DOGXL160, 160x104	BW	yes	160 Bytes	SW SPI	u8g_dev_uc1610_dogxl160_bw_sw_spi	U8GLIB_DOGXL160_BW(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_uc1610_dogxl160_bw_hw_spi	U8GLIB_DOGXL160_BW(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
				320 Bytes	SW SPI	u8g_dev_uc1610_dogxl160_2x_bw_sw_spi	U8GLIB_DOGXL160_2X_BW(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_uc1610_dogxl160_2x_bw_hw_spi	U8GLIB_DOGXL160_2X_BW(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
		4L	yes	160 Bytes	SW SPI	u8g_dev_uc1610_dogxl160_gr_sw_spi	U8GLIB_DOGXL160_GR(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_uc1610_dogxl160_gr_hw_spi	U8GLIB_DOGXL160_GR(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
				320 Bytes	SW SPI	u8g_dev_uc1610_dogxl160_2x_gr_sw_spi	U8GLIB_DOGXL160_2X_GR(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_uc1610_dogxl160_2x_gr_hw_spi	U8GLIB_DOGXL160_2X_GR(cs, a0 [, reset])	verified	v1.14	n.a.	impl.
LD7032	60x32	BW	no	64 Bytes	SW SPI	u8g_dev_ld7032_60x32_sw_spi	U8GLIB_LD7032_60x32(sck, mosi, cs, a0 [, reset])	verified v1.16	impl. v1.16	impl. v1.16	impl. v1.16
					HW SPI	u8g_dev_ld7032_60x32_hw_spi	U8GLIB_LD7032_60x32(cs, a0 [, reset])	impl. v1.16	impl. v1.16	impl. v1.16	impl. v1.16
PCD8544	84x48	BW	no	84 Bytes	SW SPI	u8g_dev_pcd8544_84x48_sw_spi	U8GLIB_PCD8544(sck, mosi, cs, a0 [, reset])	verified	impl. v1.11	impl.	impl.
					HW SPI	u8g_dev_pcd8544_84x48_hw_spi	U8GLIB_PCD8544(cs, a0 [, reset])	impl	v1.14	impl.	impl.
TLS8204	84x48	BW	no	84 Bytes	SW SPI	u8g_dev_tls8204_84x48_sw_spi	U8GLIB_TLS8204_84X48(sck, mosi, cs, a0 [, reset])	impl.	impl. v1.11	impl.	impl.
PCF8812	96x65	BW	no	96 Bytes	SW SPI	u8g_dev_pcf8812_96x65_sw_spi	U8GLIB_PCF8812(sck, mosi, cs, a0 [, reset])	verified	v1.14	impl.	impl.
					HW SPI	u8g_dev_pcf8812_96x65_hw_spi	U8GLIB_PCF8812(cs, a0 [, reset])	impl.	v1.14	impl.	impl.
KS0108	128x64	BW	no	128 Bytes	8Bit	u8g_dev_ks0108_128x64					
					8Bit Fast	u8g_dev_ks0108_128x64_fast	U8GLIB_KS0108_128(d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw [, reset])	verified	impl. v1.11	impl.	impl.
T6963	128x64	BW	no	256 Bytes	8Bit	u8g_dev_t6963_128x64_8bit	U8GLIB_T6963_128X64(d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd [, reset])	impl. v1.11	n.a.	n.a.	n.a.
	128x128	BW	no	256 Bytes	8Bit	u8g_dev_t6963_128x128_8bit	U8GLIB_T6963_128X128(d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd [, reset])	impl. v1.16	n.a.	n.a.	n.a.
	240x64	BW	no	480 Bytes	8Bit	u8g_dev_t6963_240x64_8bit	U8GLIB_T6963_240X64(d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd [, reset])	impl. v1.11	n.a.	n.a.	n.a.
	240x128	BW	no	480 Bytes	8Bit	u8g_dev_t6963_240x128_8bit	U8GLIB_T6963_240X128(d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd [, reset])	impl. v1.11	n.a.	n.a.	n.a.
SBN1661	122x32	BW	no	122 Bytes	8Bit Fast	u8g_dev_sbn1661_122x32	U8GLIB_SBN1661_122X32(d0, d1, d2, d3, d4, d5, d6, d7, cs1, cs2, di, rw [, reset])	verified	n.a.	impl.	n.a.

LC7981	160x80	BW	no	160 Bytes	8Bit	u8g_dev_lc7981_160x80_8bit	U8GLIB_LC7981_160X80(d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, di [, rw [, reset]])	verified	impl. v1.11	impl.	impl.
	240x64	BW	no	240 Bytes	8Bit	u8g_dev_lc7981_240x64_8bit	U8GLIB_LC7981_240X64(d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, di [, rw [, reset]])	impl.	impl. v1.11	impl.	impl.
	240x128	BW	no	240 Bytes	8Bit	u8g_dev_lc7981_240x128_8bit	U8GLIB_LC7981_240X128(d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, di [, rw [, reset]])	impl.	impl. v1.11	impl.	impl.
	320x64 (16 Bit Mode only)	BW	no	320 Bytes	8Bit	u8g_dev_lc7981_320x64_8bit	U8GLIB_LC7981_320X64(d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, di [, rw [, reset]])	impl. v1.09	impl. v1.11	impl. v1.09	impl. v1.09
SSD1351	128x128 (ILSoft)	332	yes	1024 Bytes	SW SPI	u8g_dev_ssd1351_128x128_332_sw_spi	U8GLIB_SSD1351_128X128_332(sck, mosi, cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_ssd1351_128x128_332_hw_spi	U8GLIB_SSD1351_128X128_332(cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
				4096 Bytes	SW SPI	u8g_dev_ssd1351_128x128_4X_332_sw_spi	U8GLIB_SSD1351_128X128_4X_332(sck, mosi, cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_ssd1351_128x128_4X_332_hw_spi	U8GLIB_SSD1351_128X128_4X_332(cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
		64K	yes	1024 Bytes	SW SPI	u8g_dev_ssd1351_128x128_hicolor_sw_spi	U8GLIB_SSD1351_128X128_HICOLOR(sck, mosi, cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_ssd1351_128x128_hicolor_hw_spi	U8GLIB_SSD1351_128X128_HICOLOR(cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
				4096 Bytes	SW SPI	u8g_dev_ssd1351_128x128_4X_hicolor_sw_spi	U8GLIB_SSD1351_128X128_4X_HICOLOR(sck, mosi, cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_ssd1351_128x128_4X_hicolor_hw_spi	U8GLIB_SSD1351_128X128_4X_HICOLOR(cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
	128x128 (Freetronics)	332	yes	1024 Bytes	SW SPI	u8g_dev_ssd1351_128x128gh_332_sw_spi	U8GLIB_SSD1351_128X128GH_332(sck, mosi, cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_ssd1351_128x128gh_332_hw_spi	U8GLIB_SSD1351_128X128GH_332(cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
				4096 Bytes	SW SPI	u8g_dev_ssd1351_128x128gh_4X_332_sw_spi	U8GLIB_SSD1351_128X128GH_4X_332(sck, mosi, cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_ssd1351_128x128gh_4X_332_hw_spi	U8GLIB_SSD1351_128X128GH_4X_332(cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
		64K	yes	1024 Bytes	SW SPI	u8g_dev_ssd1351_128x128gh_hicolor_sw_spi	U8GLIB_SSD1351_128X128GH_HICOLOR(sck, mosi, cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_ssd1351_128x128gh_hicolor_hw_spi	U8GLIB_SSD1351_128X128GH_HICOLOR(cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
				4096 Bytes	SW SPI	u8g_dev_ssd1351_128x128gh_4X_hicolor_sw_spi	U8GLIB_SSD1351_128X128GH_4X_HICOLOR(sck, mosi, cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
					HW SPI	u8g_dev_ssd1351_128x128gh_4X_hicolor_hw_spi	U8GLIB_SSD1351_128X128GH_4X_HICOLOR(cs, a0 [, reset])	verified v1.13	verified v1.13	impl. v1.13	impl. v1.13
HT1632	24x16	BW	yes	48 Bytes	SW SPI	u8g_dev_ht1632_24x16	U8GLIB_HT1632_24X16(wr/sck, data/mosi, cs)	impl. v1.13	n.a.	n.a.	n.a.
A2 Micro Printer	384x240	BW	n.a.	384 Bytes		u8g_dev_a2_micro_printer_384x240	n.a.	impl. v1.14	impl. v1.14	impl. v1.14	impl. v1.14
	192x120	BW	n.a.	192 Bytes		u8g_dev_a2_micro_printer_192x120_ds	n.a.	impl. v1.14	impl. v1.14	impl. v1.14	impl. v1.14
callback fn	Flipdisk, 28x14	BW	n.a.	52 Bytes		u8g_dev_flipdisc_2x7	U8GLIB_FLIPDISC_2X7()	impl.	impl. v1.11	impl.	impl.
virtual device	changeable	BW	n.a.	n.a.		u8g_dev_vs	U8GLIB_VS()	impl.	impl. v1.11	impl.	impl.
stdout	Console, 70x30	BW	n.a.	70 Bytes		u8g_dev_stdout		n.a.	n.a.	n.a.	n.a.
SDL	Desktop, 128x64	BW	no	128 Bytes		u8g_dev_sdl_1bit		n.a.	n.a.	n.a.	n.a.
		4L	no	128 Bytes		u8g_dev_sdl_2bit		n.a.	n.a.	n.a.	n.a.

- Currently unsupported
    - T6963 not supported by AVR, Chipkit (PIC32) and Arduino Due
    - HW SPI not supported by Chipkit (PIC32)
    - I2C not supported by Chipkit (PIC32) and Arduino Due
    - SBN1661/SED1520 not supported by AVR and Arduino Due
- Mode
    - BW: Black and white mode, one bit per pixel.
    - 4L: Four gray levels, two bits per pixel
    - 332: RGB mode with 3 bit for red and green and 2 bit for blue
    - 64K: Hi-Color RGB mode with 5 bit for red, 6 bit for green and 5 bit for blue
- Com
    - HW SPI: Use pin definitions from the Arduino variant files
    - SW SPI: Use internal software SPI communication
    - I2C: Use hardware I2C/TWI. The following options are supported:
        - `U8G_I2C_OPT_NONE`: Normal TWI operation.
        - `U8G_I2C_OPT_NO_ACK`: U8glib will ignore ACK (for those devices which do not send ACK, Available with v1.16)
    - 8Bit: 8 bit parallel communication with `digitalWrite`
    - 8Bit Fast: 8 bit parallel communication with `digitalPinToPort`
- Pin Names
    - **sck**: Clock
    - **mosi**: Serial data
    - **cs**, **cs1**, **cs2**: Chip select, datasheet names: CS, CS1, CS2 or CE
    - **a0**, **di**: Data/instruction register select, datasheet names: A0, DI, CD or RS
    - **rw**: Read/write line
    - **en**: Read/write enable

    - **reset**: Display reset

- Implementation Status
    - verified: Implemented and tested
    - impl.: Implemented
    - n.a.: Not available

- setContrast: For some devices, the contrast value can be assigned with the `setContrast` procedure.

- 16 Bit mode: U8glib can be put into 16 Bit mode. Remove comment from `#define U8G_16BIT 1` in u8g.h

- Other controller, which might be compatible (this information is not verified). A controller might be compatible with other controllers in the same line.
    - `ST7565`, `NT7534`, `SED1565`, `S1D15605`, `S6B0723`
    - `SBN1661`, `SED1520`: Write is triggered by CSx lines.
    - `SED1335`, `RA8835`, `S1D3305` (not supported by u8glib)
    - `NT7086`, `LC7981`, `HD61830` (?)
    - `SSD1306` (128x64), `SH1106` (132x64)

- LCD Modules with ST7920 Controller
    - Serial mode (PSB = LOW)
        - **sck**: Pin with label "E"
        - **mosi**: Pin with label "RW"
        - **cs**: Pin with label "RS".
        - Example: U8GLIB_ST7920_128X64_1X(sck, mosi, cs [, reset]) is equivalent to U8GLIB_ST7920_128X64_1X(E, RW, RS, RST) for many display modules.
    - Parallel mode (PSB = HIGH)
        - **en**: Pin with label "E"
        - **di**: Pin with label "RS"
        - **rw**: Pin with label "RW"

- LCD Modules with T6963 Controller
    - It is required to select the 8x8 font with u8glib.
        - If the display module has two font select pins, connect both to GND
        - If the display module has one font select pin, connect it to GND

Comment by sja...@gmail.com, Jan 25, 2013

Hi Oliver, I want to create dev_ssd1309_128x64 for Flylight FG12864AW display. I am not quite familiar with u8glib so I wold like to ask your help. Based on my research ssd1306 and ssd1309 have different "init sequence".

> Would it be enough just to correct u8g_dev_ssd1306_128x64.c file just by rewriting one of "init sequence" functions and using ssd1309 as ssd1306? I think that would be the most simple solution if that is enough. Write a new u8g_ssd1309_128x64.c file but probably it wouldn't be enough and I would need to integrate it to library somehow?

Thank you for you help.

Comment by project member olikr...@gmail.com, Jan 25, 2013

Hi If the ssd1309 only differs in the init sequence, then of course updating the init sequence should be enough. You can do this updated in the existing ssd1306 device. I can add it add ssd1309. You are also wellcome to contact me directly. Then it is probably easier to send you an updated version of u8glib

Oliver

Comment by simone...@gmail.com, Apr 17, 2013

I have a SED1565 (taken from a Nokia 7160) that has a resolution of 96x65. How do I make with work using the ST7565 code? Thanks.

Comment by project member olikr...@gmail.com, Apr 17, 2013

Maybe the SED1565 is not compatible to ST7565 (the information on this page is not validated regarding compatible chips). Perhaps try U8GLIB_PCF8812(sck, mosi, cs, a0 [, reset])

Comment by anton.ze...@tortenboxer.de, Jun 23, 2013

Hi, just want to tell that Nokia 3310 also works with HW_SPI. All I did: correct cabling for arduino HW SPI add to .....Arduino\libraries\U8glib\utility\u8g_dev_pcd8544_84x48.c at the last line: U8G_PB_DEV(u8g_dev_pcd8544_84x48_hw_spi , WIDTH, HEIGHT, PAGE_HEIGHT, u8g_dev_pcd8544_fn, U8G_COM_HW_SPI);

and in ......\Arduino\libraries\U8glib\U8glib.h next to U8GLIB_PCD8544 class U8GLIB_PCD8544_HW : public U8GLIB {

> public: U8GLIB_PCD8544_HW(uint8_t sck, uint8_t mosi, uint8_t cs, uint8_t a0, uint8_t reset = U8G_PIN_NONE) : U8GLIB(&u8g_dev_pcd8544_84x48_hw_spi, sck, mosi, cs, a0, reset) { }

};

and than I use U8GLIB_PCD8544_HW instead of U8GLIB_PCD8544

works straight, is a tick faster and saves 150 byte. Great lib, the only issue so far: 32KB flash is not the perfect choise for M2tklib and arduino. I have to reduce amount of different used GUI elements for not exceeding the 30KB limit. 3 KB just eaten by a font for big digits. but once you understand the syntax it is a hughe save of time.

Comment by project member olikr...@gmail.com, Jun 23, 2013

Thanks, i have added ~~issue 181~~ for this.

Comment by SkipDich...@gmail.com, Jul 14, 2013

Hi Oliver, these are great drivers, thank you very much!

When I was using HW (the u8g_dev_st7920_128x64_hw_spi) version on an AVR (AT90CAN128 @ 16MHz) device I had to set CPOL and CPHA, like.... SPCR = (1<<SPE) | (1<<MSTR)|(0<<SPR1)|(0<<SPR0)|(1<<CPOL)|(1<<CPHA);

This may help others. Kind Regards

Comment by project member olikr...@gmail.com, Jul 14, 2013

Hi Thanks for reporting this. Oliver

Comment by alexandr...@gmail.com, Jul 22, 2013

Hello,

this is to let you know that SSD1306 (128x64 from Adafruit) on Arduino Due with SW SPI works well.

U8GLIB_SSD1306_128X64(sck, mosi, cs, a0, reset)

Setting up the reset line seems to be required to get anything on the screen.

Thanks very much for these drivers.

Comment by project member olikr...@gmail.com, Jul 22, 2013

Thank you very much for the feedback

Comment by project member olikr...@gmail.com, Jul 22, 2013

BTW: HW SPI should be also available for the DUE

Comment by neiron....@gmail.com, Sep 15, 2013

Does it possible to add ILI9341 (SPI mode)? I found ILI9325, but it has small diferrencess. When i recive package then try run ILI9341 with ILI9325 devlib

Comment by project member olikr...@gmail.com, Sep 15, 2013

Full color controllers are quite complicated. Implementations is difficult without having the device in my lab. Even then, it will take a long time unless you have a good and fitting init procedure for it. Of course you can use u8glib, but you have to write your own device for it. I started to do the documentation for such a task: http://code.google.com/p/u8glib/wiki/devicedesign

Comment by avrbegin...@googlemail.com, Sep 18, 2013

I'm having trouble getting the adafruit 128x128 color OLED display with SSD1351 to work. I connected it to a teensy 2.0 (ATmega32u4):

    CS: PORTC, Bit 7 --> PN(2,7) A0: PORTB, Bit 7 --> PN(1,7) RESET: PORTD, Bit 7 --> PN(3,7)

using this call to u8g_InitSPI:

    (&u8g, &u8g_dev_ssd1351_128x128_332_hw_spi, PN(2,7), PN(1,7), PN(3,7), U8G_PIN_NONE, U8G_PIN_NONE);

the remaining code is exactly as in the hello world example, but the display stays black. The wiring was tested with an init routine from the adafruit arduino code for this display.

Comment by project member olikr...@gmail.com, Sep 18, 2013

The u8g_InitSPI is the software spi version. It expects 5 pins: uint8_t u8g_InitSPI(u8g_t **u8g, u8g_dev_t** dev, uint8_t sck, uint8_t mosi, uint8_t cs, uint8_t a0, uint8_t reset);

Probably you wanted to use HW SPI, which has only three pins as arguments: uint8_t u8g_InitHWSPI(u8g_t **u8g, u8g_dev_t** dev, uint8_t cs, uint8_t a0, uint8_t reset);

I suggest to replace u8g_InitSPI with u8g_InitHWSPI and remove the U8G_PIN_NONE.

Note: This combination has never been tested.

Comment by avrbegin...@googlemail.com, Sep 18, 2013

Thanks for your help, but no luck.

I also tried switching from SPI mode 0 to mode 3 (this is used in the adafruit code), but the display is black, no "Hello World!" to be seen. As the device list above shows the arduino version as verified, I'd expect the signal polarity to be valid as well. The SSD1351 datasheet, indicates that both mode 0 and mode 3 should be ok.

Comment by project member olikr...@gmail.com, Sep 18, 2013

> arduino version as verified yes, but u8g_InitSPI is the AVR variant. Did you try U8GLIB_SSD1351_128X128_332()? Additionally the Arduino version applies to the Arduino variant of u8glib. Additionally there is an issue with a GPIO line of the controller, which might be required to be set.

Comment by avrbegin...@googlemail.com, Sep 18, 2013

The table above lists U8GLIB_SSD1351_128X128_332() as a C++ constructor - how would I use that in plain C for an AVR? The adafruit code does not set the GPIO line, so this should be fine.

Comment by project member olikr...@gmail.com, Sep 18, 2013

Sure, but you did mention "Arduino"... What i wanted to say is this: HW SPI has not been tested with ATMEGA32U4 in the AVR variant of u8glib. My suggestion is to use SW SPI. If this works, then maybe try HW SPI again. That means you should have something like u8g_InitSPI(&u8g, &u8g_dev_ssd1351_128x128_332_hw_spi, PN(?,?), PN(?,?), PN(2,7), PN(1,7), PN(3,7)); where PN(?,?) are the SCK and MOSI pins.

Comment by avrbegin...@googlemail.com, Sep 18, 2013

Ah, I see. Adafruit have working arduino code for this display. It contains init and command/data writing routines which I ported to AVR. That's why I'm confident in my wiring.

Back to u8glib (AVR variant): Now I used u8g_InitSPI, but with &u8g_dev_ssd1351_128x128_332_sw_spi (you wrote *hw_spi) and have a blue "Hello, World!" on my display! Neat! Slow, though... Now I just need to find out how to use the HW SPI and/or how to speed things up.*

Comment by project member olikr...@gmail.com, Sep 18, 2013

Great. To change the color you need to use u8g_SetRGB(&u8g, r,g,b). This procedure is not yet mentioned in the reference.

Comment by avrbegin...@googlemail.com, Sep 18, 2013

Thanks! I already found that by searching through the source. HW SPI is much more important, as the AVR has a few more tasks...

Comment by avrbegin...@googlemail.com, Sep 18, 2013

OK, HW SPI works both in mode 0 and mode 3. I just had to change the pin numbers in u8g_com_atmega_hw_spi.c to the correct values.

Comment by project member olikr...@gmail.com, Sep 18, 2013

Very good, thank you for the information

Comment by drei...@shaw.ca, Oct 10, 2013

I managed to get a NHD-2.8-25664 with a SSD1322 controller working with the library. I displays all of the example objects OK. Still testing.

Comment by hpvide...@gmail.com, Oct 18, 2013

Thanks to this library, i can use my NHD-3.12-25664UCY2 screen,SSD1322 controller with Arduino.But i have small problem,it's slow.The problem is more visible, when is loading BITMAP. The screen is set to 4-wire SPI mode,by connecting MPU Interface Pin Selections,BS0 and BS1 to ground, and the (U8GLIB_NHD31OLED_GR u8g(13, 11, 10, 9) // SPI Com: SCK = 13, MOSI = 11, CS = 10, A0 = 9),is selected. Can somebody help me,to speed up this screen. Thanks in advance

Comment by project member olikr...@gmail.com, Oct 18, 2013

I i just notice, that the NHD31 is not listed in the device table. You could check U8GLIB_NHD31OLED_2X_GR. It should be faster.

Comment by hpvide...@gmail.com, Oct 18, 2013

Thank you for replay.The U8GLIB_NHD31OLED_2X_GR never work for this screen properly.In that mode, only half of the screen is working and half of the horizontal lines are active.But,you're right, it's faster.

Comment by project member olikr...@gmail.com, Oct 18, 2013

Thanks for the bugreport. I do not own this display, so i was not able to verify the output. I will review the code.

Comment by hpvide...@gmail.com, Oct 22, 2013

I will glad to provide a display to you for testing. Just let me know, where I need to send it.

Comment by project member olikr...@gmail.com, Oct 22, 2013

please contact me via e-mail (see source for address). I try to review/update the code. After that, maybe you can test the updated code.

Comment by anton.ze...@tortenboxer.de, Oct 25, 2013

I made a mechanical mistake and wonder if I can rotate a nokia 3310 via SW by 180 degree?

Comment by anton.ze...@tortenboxer.de, Nov 3, 2013

simple answer: u8g.setRot180(); I was not sure if it that simple. even with m2tklib but it is that simple great

Comment by project member olikr...@gmail.com, Nov 3, 2013

This is correct. Excause me for not answering your comment from 25 Oct. Seems that i missed reading the google.com notification on your comment.

Comment by lc.sting...@gmail.com, Jan 10, 2014

@ neiron....@gmail.com,

Have you try ILI9341 With ILI9325 ? Or Does anyone could help me (If this is possible) , I allready use U8G for another project and , would like to go further with TFT and this great Lib :) Thank you for answers and if this disturb this page (oliver you'll say it to me :)), make it by mail .

Comment by project member olikr...@gmail.com, Jan 10, 2014

There is another problem with full color support for U8glib: U8glib is optimized for monochrome displays. True color display procedures will be very slow with U8glib. I finally started a new lib for better support of true color displays: http://code.google.com/p/ucglib/ It will not be compatible with U8glib, but i intend to take over fonts and other drawing procedures from U8glib. The include file of Ucglib gives an idea about was has been done so far: http://code.google.com/p/ucglib/source/browse/csrc/ucg.h

SSD1351 is working. Coding on ILI9325 has started. For any other devices i probably need people who can do read datasheets, provide setup sequence for the display, do testing, etc...

Comment by rastaman...@mail.ru, Jun 13, 2014

How to connect ST7920 8 bit bus (AVR)   LCD Pin no cs1, cs2. Supported Devices written U8GLIB_ST7920_128X64 (d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw [, reset]) I am writing in AVR Studio 4.19

Comment by rastaman...@mail.ru, Jun 13, 2014

I am writing AVR Studio (Atmega32)

u8g_Init8Bit?(& u8g, & u8g_dev_st7920_128x64_4x_8bit, PN (0, 0), PN (0, 1), PN (0, 2), PN (0, 3), PN (0, 4), PN (0, 5), PN (0, 6), PN (0, 7), PN (2, 3), PN (3, 7), PN (2, 2), U8G_PIN_NONE);

get

- Error: too few arguments to function 'u8g_Init8Bit?'

Comment by project member olikr...@gmail.com, Jun 13, 2014

prototype is: uint8_t u8g_Init8Bit?(u8g_t **u8g, u8g_dev_t** dev, uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3, uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7,

uint8_t en, uint8_t cs1, uint8_t cs2, uint8_t di, uint8_t rw, uint8_t reset);

set CS2 to U8G_PIN_NONE

Comment by rastaman...@mail.ru, Jun 15, 2014

Thank you. Everything turned. 8-bit initialization(Atmega32,AVR Studio 4.19) : u8g_Init8Bit? (& u8g, & u8g_dev_st7920_128x64_8bit, PN (0, 0), PN (0, 1), PN (0, 2), PN (0, 3), PN (0, 4), PN (0, 5), PN (0, 6), PN (0, 7), PN (2, 3), U8G_PIN_NONE, U8G_PIN_NONE, PN (3, 7), PN (2, 2), U8G_PIN_NONE);

Comment by alidetu...@gmail.com, Sep 16, 2014

ssd1332怎么用这个库

Comment by alidetu...@gmail.com, Sep 16, 2014

ssd1332怎么用这个库

Comment by SuvenirU...@gmail.com, Sep 21, 2014

Controller RA8806 lcd 240x128 will work with u8glib ? It's compatible with T6963 240x128 ?

Comment by project member olikr...@gmail.com, Sep 21, 2014

As far as i remember, this controller is not compatible with the T6963 or any other controller supported by U8glib, so the display is unfortunately not supported.

Comment by SuvenirU...@gmail.com, Sep 21, 2014

Ok, thanks.

T6963 240x128 will work with arduino due ? In device list it's marked "n.a." But I try compilate without LCD on arduino due and no errors.

Comment by project member olikr...@gmail.com, Sep 21, 2014

I do not remeber exactly. At least it has not been tested. Additionally, the T6963 is a 5V controller, which probably can't be driven by the Due directly.

Comment by alidetu...@gmail.com, Sep 22, 2014

can Controller SSD1332 work with u8glib? It is multicolor like ssd1351.I use AVR Microcontroller.

Comment by project member olikr...@gmail.com, Sep 22, 2014

SSD1351 is handled by ucglib (https://code.google.com/p/ucglib/)

Comment by marccoll...@gmail.com, Sep 23, 2014

when an controller is supported that mean that all display who use it will work with u8glib?

because i would like to know if this display will work NHD-2.7-12864UCY3 http://www.newhavendisplay.com/nhd2712864ucy3-p-3621.html

Comment by project member olikr...@gmail.com, Sep 23, 2014

If a controller is supported, then the major tasks for the display has been done, but it does not mean that a specific display is supported. Yet there is a high probability.

The refered display is supported. Infact i have the same display in my lab and it is also part of my test equipment for u8glib.

Comment by alidetu...@gmail.com, Sep 23, 2014

How the ssd1332 can be used to u8glib.

Comment by alidetu...@gmail.com, Sep 23, 2014

Is CLKPR Necessary?I use atmega16a to control ,but have a error,CLKPR undeclared ? why and how to deal

Comment by christia...@gmail.com, Sep 30, 2014

Hello, at first - thank you very much for providing a powerful graphic-lib as this one. I have used it on an other hobby project some time ago. But at the moment I'm stucked with a different one: I've got several 128x128 color LCDs "on stock", all are equipped with a Samsung S6B33BG or S6B3306X11? in a parallel configuration. I'm trying at least for a month to get them to work (Arduino, beginner skills), but no progress so far, so I decided to write, maybe there is a chance to get at least one to work. It uses D0-7, DI, reset, CS, RD and WR. I tried to update a driver as well, but no success. Maybe someone got a display like these to work.

Thank you in advance!

Comment by project member olikr...@gmail.com, Sep 30, 2014

Did you check if it is compatible with one of the supported controllers? Compatible means, that the commands are identical. Also: Are you sure, that your setup is correct?

Comment by christia...@gmail.com, Sep 30, 2014

Yes, tried them all, my setup is at least quad-checked. No success. I also tried to adapt the T6963-driver with the initialization commands, but it seems that i've messed it up. Maybe there is a problem with the two RD and WR pins.

Comment by project member olikr...@gmail.com, Oct 1, 2014

well, i think first the datasheet of the S6B33BG needs to be studied. If it has an identical commandset to one of the existing controllers.

Comment by christia...@gmail.com, Oct 1, 2014

In the meantime, i did some research for compatibility on the net. It looks like the S6B **has an identical command set with the following displays: HDC1600, NT7571, NT75751. Info from http://wiki.s1mp3.org/LCMControllers This display is also used in "PIC18-LF4550-STK1" -> http://www.techtoys.com.hk/PIC_boards/PIC18-4455-STK1/pic18-4550-stk1.htm There is also sample code available. I try to get it working...**

Comment by project member olikr...@gmail.com, Oct 2, 2014

From what i have seen here: http://www.samsung.com/global/business/semiconductor/product/display-solution/detail?productId=5783 It seems to be RGB OLED display. However the S6B33B seems to a TFT display (according to the information from the techtoys.com.hk. So, what kind of display do you have exactly? In general, u8glib has been optimized for monochrome displays, so maybe this is better handled by ucglib.

Comment by christia...@gmail.com, Oct 15, 2014

In the end, i got it working! It is now a mix of Arduino and the TechToys? example. In fact, my display is a RGB-LCD. The Controller seems to be capable of driving different kinds of displays. If you want, i can send you my code (mess!) to implement it into your Lib.

Comment by adi.srin...@gmail.com, Dec 2 (3 days ago)

I think alidetu...@gmail.com and I have the same problem. Is CLKPR necessary? I'm using an ATMega16A.

Comment by project member olikr...@gmail.com, Dec 2 (3 days ago)

Discussed on "avr" wiki page

► Sign in to add a comment