AN13295

How to Port U8g2 to LPC55(S)6x and Support a Mono OLED Display

Rev. 0 — 30 June 2021

Application Note

Contents

1 Introduction to LPC55(S)6x and U8g2

The LPC5500 is an Arm® Cortex®-M33 based microcontroller for embedded applications. These devices include:

- up to 320 kB of on-chip SRAM
- up to 640 kB on-chip flash
- high-speed and full-speed USB host and device interface with crystal-less operation for full-speed
- five general-purpose timers
- one SCTimer/PWM
- one RTC/alarm timer
- one 24-bit Multi-Rate Timer (MRT)
- one Windowed Watchdog Timer (WWDT)
- eight flexible serial communication peripherals (each of which can be a USART, SPI, I^{2}C, or I^{2}S interface)
- one 16-bit 1.0 Msps ADC
- temperature sensor

The Arm Cortex®- M33 provides a security foundation, offering isolation to protect valuable IP and data with TrustZone® technology.

U8g2 is a monochrome graphics library for embedded devices. U8g2 supports monochrome OLEDs and LCDs, which include the following controllers: SSD1305, SSD1306, SSD1309, SSD1322, SSD1325, SSD1327, SSD1329, SSD1606, SSD1607, SH1106, SH1107, SH1108, SH1122, T6963, RA8835, LC7981, PCD8544, PCF8812, HX1230, UC1601, UC1604, UC1608, UC1610, UC1611, UC1701, ST7565, ST7567, ST7588, ST75256, NT7534, IST3020, ST7920, LD7032, KS0108, SED1520, SBN1661, IL3820, MAX7219. See here for a full list.

U8g2 also includes U8x8 libraries:

- U8g2
  - Includes all graphics procedures (line/box/circle draw).
  - Supports many fonts. (Almost) no restriction on the font height.
  - Requires some memory in the microcontroller to render the display.
- U8x8
  - Text output only (character) device.
  - Only fonts allowed with fit into an 8x8 pixel grid.
  - Writes directly to the display. No buffer in the microcontroller required.

End user should check the LICENSE of U8g2: https://github.com/olikraus/u8g2/blob/master/LICENSE

The U8g2lib code is licensed under the terms of the new-bsd license (two-clause **bsd** license).

This application note is based on LPC55(S)6x MCU and describes how to port U8g2 to support a 128×64 mono OLED screen. The OLED driver IC is SSD1306.

2 Introduction to OLED panel

There are many 128 × 64 resolution OLED panel modules on the market and they are very easy to purchase. When searching for **OLED 0.96 SSD1306** on Taobao or Amazon, you can get OLED screen modules information as shown in Figure 1.

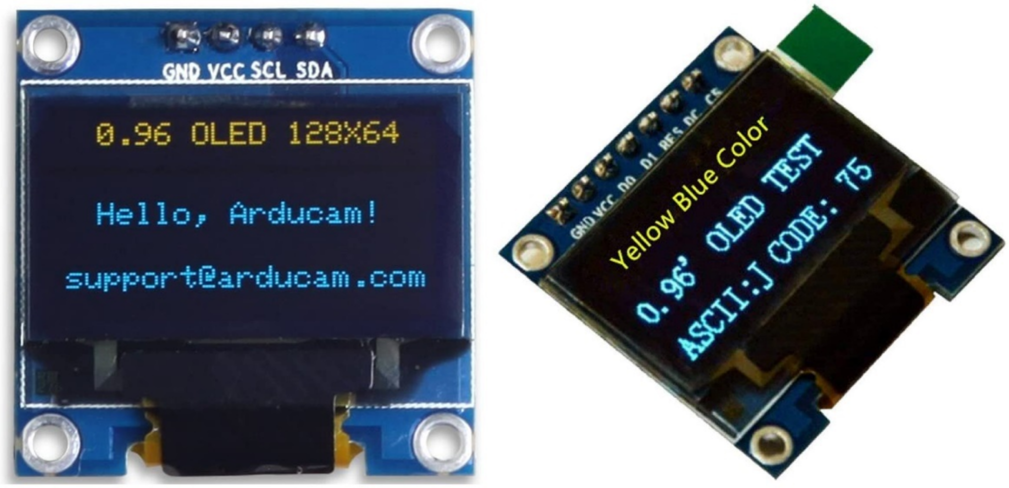

Figure 1. 0.96' 128 × 64 SSD1306 OLED panel modules

Table 1. SSD1306 MCU control protocol configuration pins

SSD1306 pin name	I^2C interface	6800-parallel interface (8 bit)	8080-parallel interface (8 bit)	4-wire serial interface	3-wire serial interface
BS0	0[1]	0	0	0	1[2]
BS1	1	0	1	0	0
BS2	0	1	1	0	0

1. 0 is connected to V_{SS}
2. 1 is connected to V_{DD}

Usually, the interface for MCU to drive SSD1306 is I^2C, 3-wire, or 4-wire SPI. Developer must adjust the module or the voltage level status on the BS0, BS1, and BS2 pins on the OLED screen according to their actual interface requirements to configure the drive interface of the OLED screen, as shown in Table 1.

For specific timing control protocol, see SSD1306.

3 LPC55S69-EVK connecting with OLED screen

Developer can use the Flexcomm function of LPC55S69 to communicate with the OLED screen.

- If the OLED screen is an I^2C interface, configure Flexcomm as I^2C.
- If the OLED screen is an SPI interface, configure Flexcomm as SPI.

On the LPC55S69-EVK evaluation board, use **Flexcomm4** as the I^2C interface to connect to the I^2C OLED screen and use the High-Speed SPI interface to connect to the SPI OLED module.

3.1 LPC55S69-EVK evaluation board

Figure 2 shows the LPC55S69-EVK, the official evaluation board for the LPC55(S)6x series.

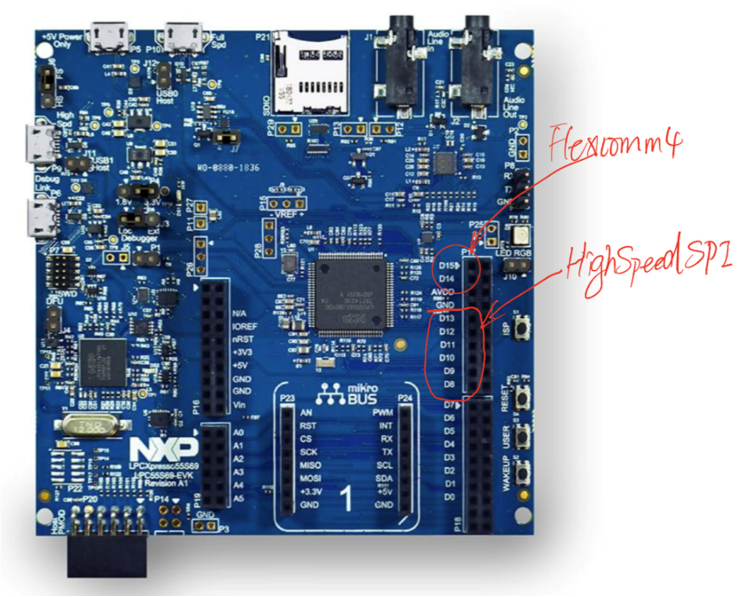

Figure 2. LPC55S69-EVK evaluation board

Figure 2 marks:

- the position of Flexcomm4, with the silkscreen of D15 and D14 on P17 connector
- the position of SPI, with the silkscreen of D13, D12, D11, and D10 on P17 connector

User can also use **AVDD** and **GND** connectors on P17 to power up OLED module.

3.2 LPC55S69-EVK evaluation board connected OLED module with I^{2}C interface

Figure 3 shows the I^{2}C interface OLED module connected with LPC55S69-EVK.

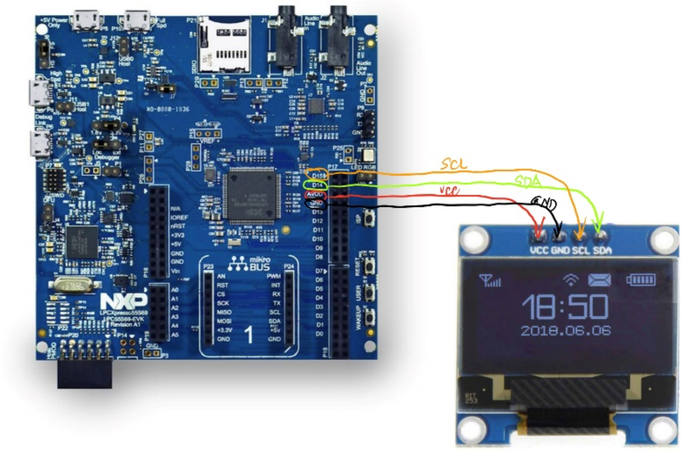

Figure 3. LPC55S69-EVK connect with OLED module through I^{2}C interface

3.3 LPC55S69-EVK evaluation board connected OLED module with SPI interface

Figure 4 shows the SPI interface OLED module connected with LPC55S69-EVK.

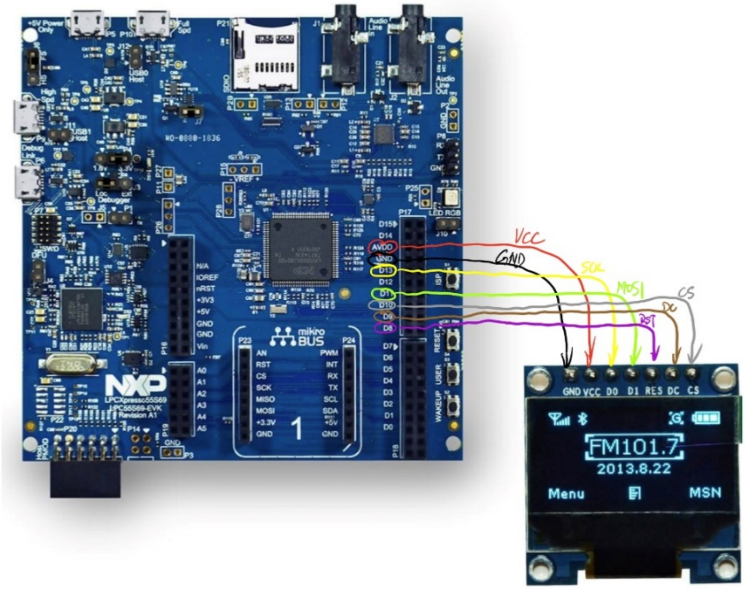

Figure 4. LPC55S69-EVK connected with OLED module through SPI interface

4 Introduction to U8g2 usage and port

4.1 How to use U8g2

U8g2 can be used as C-library with any microcontroller. The setup sequence looks as below:

```
u8g2_t u8g2; // a structure which will contain all the data for one display
...
u8g2_Setup_ssd1306_i2c_128x64_noname_2(&u8g2, U8G2_R0, u8x8_byte_sw_i2c,
u8x8_gpio_and_delay_lpc11u3x);  // init u8g2 structure
u8g2_InitDisplay(&u8g2); // send init sequence to the display, display is in sleep mode after this,
u8g2_SetPowerSave(&u8g2, 0); // wake up display
```

All available setup procedures are listed in the rest of the document. Each setup procedure requires four arguments:

1. U8g2: Pointer to an empty u8g2 structure, as shown in Example
2. Rotation: Rotation procedure, as shown in Table 2
3. Byte communication procedure: Either existing procedure or a custom procedure for the target controller
4. Low-level delay and GPIO procedure: A custom procedure

Table 2. U8g2 initialize API second parameter

Rotation/Mirror	Description
U8G2_R0	No rotation, landscape
U8G2_R1	90 degrees clockwise rotation

Table continues on the next page...

Table 2. U8g2 initialize API second parameter (continued)

Rotation/Mirror	Description
`U8G2_R2`	180 degrees clockwise rotation
`U8G2_R3`	270 degrees clockwise rotation
`U8G2_MIRROR`	No rotation, landscape, display content is mirrored (v2.6.x)

Table 3. U8g2 initialize API third parameter

Byte procedure	Description
`u8x8_byte_4wire_sw_spi`	Standard 8-bit SPI communication with four pins: SCK, MOSI, DC, CS
`u8x8_byte_3wire_sw_spi`	9-bit communication with three pins: SCK, MOSI, CS
`u8x8_byte_8bit_6800mode`	Parallel interface, 6800 format
`u8x8_byte_8bit_8080mode`	Parallel interface, 8080 format
`u8x8_byte_sw_i2c`	Two wires, I^{2}C communication
`u8x8_byte_ks0108`	Special interface for KS0108 controller

The target display must support this interface. Do not use the SPI interface with a display which does not support SPI.

4.2 How to use SDK API to support U8g2

Developer may use hardware I^{2}C, SPI, or GPIO simulate I^{2}C, SPI to drive OLED. Therefore, in terms of using SDK API to support U8g2, we provide software to support all kinds of driver situation.

Users must care about the macro definitions in the `_oled_ssd1306.c` and *.h* driver.

According to the needs of your own driving method, modify the following macro definition to **1**:

```
#define SSD1306_USE_I2C_GPIO                0u
#define SSD1306_USE_I2C_HW                  0u
#define SSD1306_USE_SPI_GPIO                0u
#define SSD1306_USE_SPI_HW                  0u
```

If developers have their own hardware platform, they need consider the IO pins function configuration in `driver_oled_ssd1306.h`. To choose hardware I^{2}C or SPI, consider the Flexcomm port setting and baud rate setting in `driver_oled_ssd1306.h`.

- Initialize API for GPIO simulate I^{2}C

```
uint8_t u8x8_gpio_and_delay_lpc55(u8x8_t *u8x8, uint8_t msg, uint8_t arguing, void *arg_ptr)
```

- Initialize API for GPIO simulate SPI

```
uint8_t u8x8_gpio_and_delay_lpc55(u8x8_t *u8x8, uint8_t msg, uint8_t arg_int, void *arg_ptr)
```

- Initialize API for Hardware I^{2}C

```
uint8_t u8x8_byte_hw_i2c_lpc55(u8x8_t *u8x8, uint8_t msg, uint8_t arg_int, void *arg_ptr)
uint8_t u8x8_gpio_and_delay_lpc55(u8x8_t *u8x8, uint8_t msg, uint8_t arg_int, void
*arg_ptr)
```

- Initialize API for Hardware SPI:

```
uint8_t u8x8_byte_4wire_hw_spi_lpc55(u8x8_t *u8x8, uint8_t msg, uint8_t arg_int, void *arg_ptr)
uint8_t u8x8_gpio_and_delay_lpc55(u8x8_t *u8x8, uint8_t msg, uint8_t arg_int, void
*arg_ptr)
```

`u8x8_gpio_and_delay_lpc55()` is the basic API, including GPIO initialization function and necessary delay functions.

5 KEIL, IAR, and MCUXpresso debugging U8g2

Before downloading the project, connect the LPC55S69-EVK with PC through USB debug port (P6).

5.1 KEIL MDK environment debug

KEIL MDK project folder is located at `lpc55s69_evk_u8g2_mdk`.

Open the project and click

to compile the project. Once the compile succeeds, press LOAD

to download code to the evaluation board.

5.2 IAR environment debug

IAR project folder is located at `lpc55s69_evk_u8g2_iar`.

Open the project and click **F7** to compile the project. Once the compile succeeds, press **F8** to download code to the evaluation board.

5.3 MCUXpresso environment debug

MCUXpresso environment need user import the *lpc55s69_evk_u8g2_mcux.zip* to project workspace first.

Once importing the project, press Build

to compile the project.

Once the compile succeeds, press

to download code to the evaluation board.

Figure 5 shows the real effect on EVK.

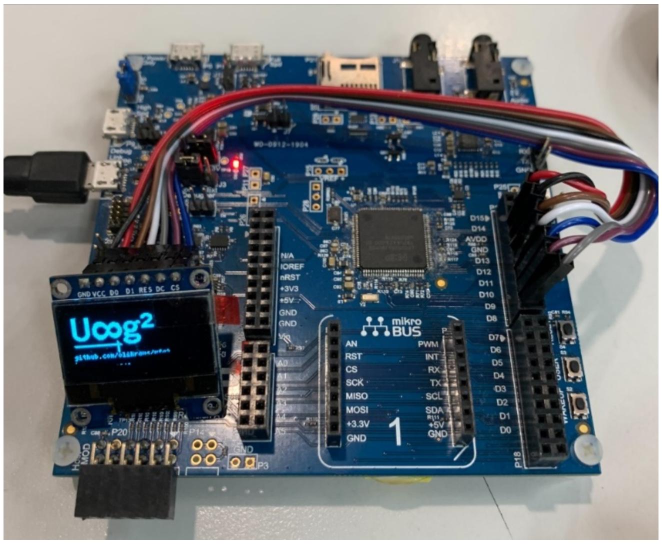

Figure 5. U8g2 display effect on LPC55S69 platform

6 Conclusion

U8g2 is a monochrome display UI library, suitable for embedded platforms. With 150 MHz core frequency, high-speed SPI, and large RAM, LPC55(S)xx series can have a better display phenomenon.

7 Reference

1. *LPC55S6x/LPC55S2x/LPC552x User manual* (document UM11126)
2. *U8g2 Setup Guide and Reference Manual*

8 Revision history

Rev.	Date	Description
0	30 June 2021	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 30 June 2021
Document identifier: AN13295