**Display - LCD12864 Shield**

**Modification history of the document**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date / Version** | **Status** | **Section** | **Type of modification** | **Author** |
| 29.03.2024 v0.1 | Draft | All | Created initial version of the document. | M. Teletin |

**Index of Contents**

1. Document Management 4

1.1. Purpose of the Document 4

1.2. Referenced Documents 4

1.3. Glossary 4

1.4. Contact Persons 4

2. Workshop description 5

2.1. LCD12864 Shield 5

2.1.1. Joystick 7

2.1.2. Pixel Coordinates 7

2.2. Integration of LCD display library 8

2.2.1. Configuration of display in STM Nucleo workspace 9

2.2.2. SPI configuration 11

2.2.3. GPIO pins configuration 11

3. Overview of the library 12

3.1. Image conversion 13

4. Exercises 15

5. Attachments 16

5.1. Reference 16

# Document Management

## Purpose of the Document

This document describes the activities that will be performed during a specific workshop organized by Preh Romania. It is a summary of the subjects / discussions addressed during the workshop, it also contains links to different useful resources that can be used for gathering additional information on the subjects or for gaining a deeper understanding of the subjects.

## Referenced Documents

|  |  |
| --- | --- |
| **Reference-ID** | **Name of the document, if necessary, incl. version and link** |
| [DS\_STM32F091RC] | [Datasheet STM32F091xB STM32F091xC](https://www.st.com/resource/en/datasheet/stm32f091rc.pdf) |
| [RM0091] | [Reference manual STM32F0x1/STM32F0x2/STM32F0x8 advanced Arm®-based 32-bit MCUs](https://www.st.com/resource/en/reference_manual/rm0091-stm32f0x1stm32f0x2stm32f0x8-advanced-armbased-32bit-mcus-stmicroelectronics.pdf) |
| [MB1136] | [MB1136-DEFAULT-C05 Board schematic](https://www.st.com/content/ccc/resource/technical/layouts_and_diagrams/schematic_pack/group2/5a/85/d6/9a/34/e2/47/1d/MB1136-DEFAULT-C05_Schematic/files/MB1136-DEFAULT-C05_Schematic.pdf/jcr:content/translations/en.MB1136-DEFAULT-C05_Schematic.pdf) |
| [UM1724] | [User manual STM32 Nucleo-64 boards (MB1136) (UM1724)](https://www.st.com/resource/en/user_manual/um1724-stm32-nucleo64-boards-mb1136-stmicroelectronics.pdf) |
| [UM1785] | [User Manual - Description of STM32F0 HAL and low-layer drivers](https://www.st.com/resource/en/user_manual/um1785-description-of-stm32f0-hal-and-lowlayer-drivers-stmicroelectronics.pdf) |

**Table 1: Referenced Documents**

## Glossary

|  |  |
| --- | --- |
| **Abbreviation / Term** | **Explanation** |
| GPIO | General Purpose Input Output |
| HAL | Hardware Abstraction Layer |
| HW | Hardware |
| LED | Light Emitting Diode |
| MCU | Microcontroller unit |
| UM | User Manual |
| VCC | Positive Supply Voltage |

**Table 2: Glossary**

## Contact Persons

|  |  |
| --- | --- |
| **Name** | **Phone No.** |
| Bogdan Lucian | lucian-viorel.bogdan@preh.ro |
| Iosub Alexandru | alexandru.iosub@preh.ro |
| Nicoara Ioan | ioan.nicoara@preh.ro |
| Smailovic Alex | alen.smailovic@preh.ro |
| Teletin Mihai | Mihai.Teletin@Preh.ro |

**Table 3: Contact Person**

# Workshop description

The goal of this workshop is to work with [development board STM32 Nucleo-64](https://www.st.com/en/evaluation-tools/nucleo-f091rc.html#documentation) and understanding and controlling LCD12864 Shield.

**Objectives:**

1. Integration of display library.
2. Understanding and configurating a serial communication bus(particularly SPI interface)for LCD12864 Shield .
3. Engage students in practical exercises to reinforce theoretical concepts.

## LCD12864 Shield

This framed LCD12864 Shield with LED backlight is compatible with most of **Arduino** controllers and supports. With 5 analog extention pins and 8 digital pins, the LCD12864 Shield also integrates a 5-key joystick for controlling addtional functions, making it an ideal module for prototyping and interactive projects.

* Power supply: 3.3V
* Pin used: D7, D8, D9, D10, D11, D13, A0
* Reset button
* 5 degree joystick (using Arduino Analog Pin 0)
* Backlit control (using Arduino Digital Pin 7)
* Extra 5 Analog pins & 8 Digital pins

A screen shot of a computer

Description automatically generated

A diagram of a computer scheme

Description automatically generated with medium confidence

### Joystick

A diagram of a circuit

Description automatically generated

The joystick is attached to analog pin AD0, and each joystick position alters the value of a voltage divider resistor, resulting in a different voltage value that must be read using an ADC pheriperal.

### Pixel Coordinates

A digital image consists of rows and columns of pixels. A pixel in such a picture can be identified by specifying the column and row in which it appears. In terms of coordinates, a pixel can be identified by a pair of integers representing the column and row numbers. For example, the pixel with coordinates (3,5) would be in column 3 and row 5. Traditionally, columns are numbered from left to right, beginning with zero. Most graphics systems, including those discussed in this chapter, number rows from top to bottom, beginning with zero. Some, notably OpenGL, number the rows from bottom to top.

A graph with numbers and a square

Description automatically generated

## Integration of LCD display library

Before starting the implementation, a new project must be created in STM32 Ice Cube IDE based on the previous workshops or the template project from the first workshop must be imported.

The guide uses different functions from the Hardware Abstraction Library provided by the manufacturer of the microcontroller, all the features of this library are described by [UM1785].



After extracting the archive, copy the display driver files (header files into the **Inc**lude folder and c files into the sources (**Src**) folder) to the relevant locations within the current project folder.

A screenshot of a computer

Description automatically generated

### Configuration of display in STM Nucleo workspace

LCD display is controlled through 6 pins from controller  
A diagram of a circuit

Description automatically generated

A computer chip with many different colored parts

Description automatically generated with medium confidence

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| uC Pins | Nucleo Pins | Mode | Lable | Description |
| PA5 | D13 | SPI CLK | \*auto labeled | Clock line for SPI |
| PA7 | D11 | SPI MOSI | \*auto labeled | Data line for SPI |
| PA8 | D7 | GPIO output | BL | Back light control |
| PA9 | D8 | GPIO output | SPIRST | Reset LCD |
| PB6 | D10 | GPIO output | SPICS | Chip Select |
| PC7 | D9 | GPIO output | SPICD | Command mode |

\*generator will auto-assign the labels

Labeling will generate some private defines in main.h header:

/\* Private defines -----------------------------------------------------------\*/

**#define** SPICD\_Pin GPIO\_PIN\_7

**#define** SPICD\_GPIO\_Port GPIOC

**#define** BL\_Pin GPIO\_PIN\_8

**#define** BL\_GPIO\_Port GPIOA

**#define** SPIRST\_Pin GPIO\_PIN\_9

**#define** SPIRST\_GPIO\_Port GPIOA

**#define** SPICS\_Pin GPIO\_PIN\_6

**#define** SPICS\_GPIO\_Port GPIOB

/\* USER CODE BEGIN Private defines \*/

### SPI configuration

Because communication with the display occurs on a single line, the STM board's MOSI and SPI must be configured as "Transmit Only Master." This means that in SPI mode, we are allowed to send data to the display but not read it.

LCD12864\_Shield is connected to D13 and D11, which means we should control the LCD via **SPI1**.

A screenshot of a computer

Description automatically generated

### GPIO pins configuration

The SPI peripheral does not control **Chip Select**; instead, the display driver uses pin **D10 (PB6),** which must be set as GPIO-output.

The **BL** pin controls the LCD's backlight; in Low mode, the backlight is turned off;

in High mode, the backlight is turned on. Backlight pin is connected to **D7 (PA8)**.

When **RST** is set to Low, the register settings are initialized (cleared). The reset operation is performed by the **D8 (PA9)** pin.

Command pin (**CD**) it determinate whether the data bits are display bits or command bits.  
CD – Hight indicated that data bits (D0 to D7) are display data.  
CD – Low indicates that data bits (D0 to D7) are control data.

A screenshot of a computer

Description automatically generated

# Overview of the library

This LCD driver provides a variety of screen interaction features, including writing text messages, drawing, printing images, and controlling the display.

// Initialization / Config Prototypes

**void** **st7565\_init**(**void**);

**void** **st7565\_set\_brightness**(uint8\_t val);

**void** **st7565\_fade\_out**(uint8\_t level);

**void** **st7565\_fade\_in**(uint8\_t level);

**void** **st7565\_backlight\_disable**(**void**);

**void** **st7565\_backlight\_enable**(**void**);

//Work with LCD Prototypes

**void** **st7565\_sendbyte**(uint8\_t c);

**void** **st7565\_clear\_screen**(**void**);

**void** **st7565\_clear\_buffer**(uint8\_t \*buffer);

**void** **st7565\_write\_buffer**(uint8\_t \*buffer);

//higher "draw"-prototypes

**void** **st7565\_drawstring**(uint8\_t \*buff, uint8\_t x, uint8\_t line, uint8\_t \*c);

**void** **st7565\_drawchar**(uint8\_t \*buff, uint8\_t x, uint8\_t line, uint8\_t c);

**void** **st7565\_setpixel**(uint8\_t \*buff, uint8\_t x, uint8\_t y, uint8\_t color);

**void** **st7565\_clearpixel**(uint8\_t \*buff, uint8\_t x, uint8\_t y);

**void** **st7565\_drawline**(uint8\_t \*buff, uint8\_t x0, uint8\_t y0, uint8\_t x1, uint8\_t y1,uint8\_t color);

**void** **st7565\_fillrect**(uint8\_t \*buff, uint8\_t x, uint8\_t y, uint8\_t w, uint8\_t h,uint8\_t color);

**void** **st7565\_drawrect**(uint8\_t \*buff, uint8\_t x, uint8\_t y, uint8\_t w, uint8\_t h,uint8\_t color);

**void** **st7565\_drawcircle**(uint8\_t \*buff, uint8\_t x0, uint8\_t y0, uint8\_t r, uint8\_t color);

**void** **st7565\_fillcircle**(uint8\_t \*buff, uint8\_t x0, uint8\_t y0, uint8\_t r, uint8\_t color);

//special "draw"-prototypes

**void** **st7565\_drawfallingbitmap**(uint8\_t \*buff, **const** uint8\_t \*bitmap, uint8\_t w, uint8\_t h);

**void** **st7565\_drawbitmap**(uint8\_t \*buff, uint8\_t x, uint8\_t y, **const** uint8\_t \*bitmap,uint8\_t w, uint8\_t h, uint8\_t color);

* If an operating system is utilized, it is important to remember that the display driver must be initialized before osKernel starts. The display init function uses a delay function, which disturbs the OS time.

// initialize LCD

st7565\_init();

st7565\_backlight\_enable();

HAL\_Delay(500);

...............................

/\* Init scheduler \*/

osKernelInitialize();

...............................

/\* Start scheduler \*/

osKernelStart();

## Image conversion



To display the content of a picture, it must be a bitmap file and the size of the display (128x64 pixels).  
  
Load the image into the LCDAssistant program (or use an online bitmap converter) and save the output to a .c/txt file.

A screenshot of a computer

Description automatically generated

It will create a 1024-byte array storing the image's content.

const unsigned char pData **[]** **=** **{**

0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x80**,** 0x80**,** 0x80**,**

0x80**,** 0xC0**,** 0x40**,** 0x40**,** 0x40**,** 0x40**,** 0x40**,** 0x20**,** 0x20**,** 0x20**,** 0x10**,** 0x10**,** 0x10**,** 0x10**,** 0x10**,** 0x10**,**

# Exercises

# Attachments

## Reference

1. https://wiki.dfrobot.com/LCD12864\_Shield\_SKU\_DFR0287#target\_3
2. https://github.com/Arduinolibrary/DFR0287\_LCD12864\_Shield\_for\_Arduino/raw/master/LCD12864%20Shield%20V1.1%20SCH.pdf
3. https://math.hws.edu/graphicsbook/c2/s1.html

https://en.radzio.dxp.pl/bitmap\_converter/