STM32CubeG4 Nucleo demonstration firmware

Introduction

STM32Cube is an STMicroelectronics original initiative to significantly improve designer's productivity by reducing development effort, time and cost. STM32Cube covers the whole STM32 portfolio.

STM32Cube includes:

- A set of user-friendly software development tools to cover project development from the conception to the realization, among which:
  - STM32CubeMX, a graphical software configuration tool that allows the automatic generation of C initialization code using graphical wizards
  - STM32CubeIDE, an all-in-one development tool with IP configuration, code generation, code compilation, and debug features
  - STM32CubeProgrammer (STM32CubeProg), a programming tool available in graphical and command-line versions
  - STM32CubeMonitor-Power (STM32CubeMonPwr), a monitoring tool to measure and help in the optimization of the power consumption of the MCU
- STM32Cube MCU & MPU Packages, comprehensive embedded-software platforms specific to each microcontroller and microprocessor series (such as STM32CubeG4 for the STM32G4 Series), which include:
  - STM32Cube hardware abstraction layer (HAL), ensuring maximized portability across the STM32 portfolio
  - STM32Cube low-layer APIs, ensuring the best performance and footprints with a high degree of user control over the HW
  - A consistent set of middleware components such as FAT file system, RTOS, USB Device, and USB Power Delivery
  - All embedded software utilities with full sets of peripheral and applicative examples
- STM32Cube Expansion Packages, which contain embedded software components that complement the functionalities of the STM32Cube MCU & MPU Packages with:
  - Middleware extensions and applicative layers
  - Examples running on some specific STMicroelectronics development boards

The STM32CubeG4 Nucleo demonstration firmware is built around the STM32Cube hardware abstraction layer (HAL) and low-layer (LL) APIs, and board support package (BSP) components.

1 STM32CubeG4 main features

STM32CubeG4 gathers, in a single package, all the generic embedded software components, required to develop an application on STM32G4 microcontrollers. In line with the STM32Cube initiative, this set of components is highly portable, not only to the STM32G4 Series but also to other STM32 series.

STM32CubeG4 is fully compatible with the STM32CubeMX code generator that allows to generate initialization code.

The package includes a driver layer (HAL) proposing a set of abstraction services and a low-level hardware layer (LL) proposing a set of register-level functions, together with an extensive set of examples running on STMicroelectronics boards. HAL is available in open-source BSD license for user convenience.

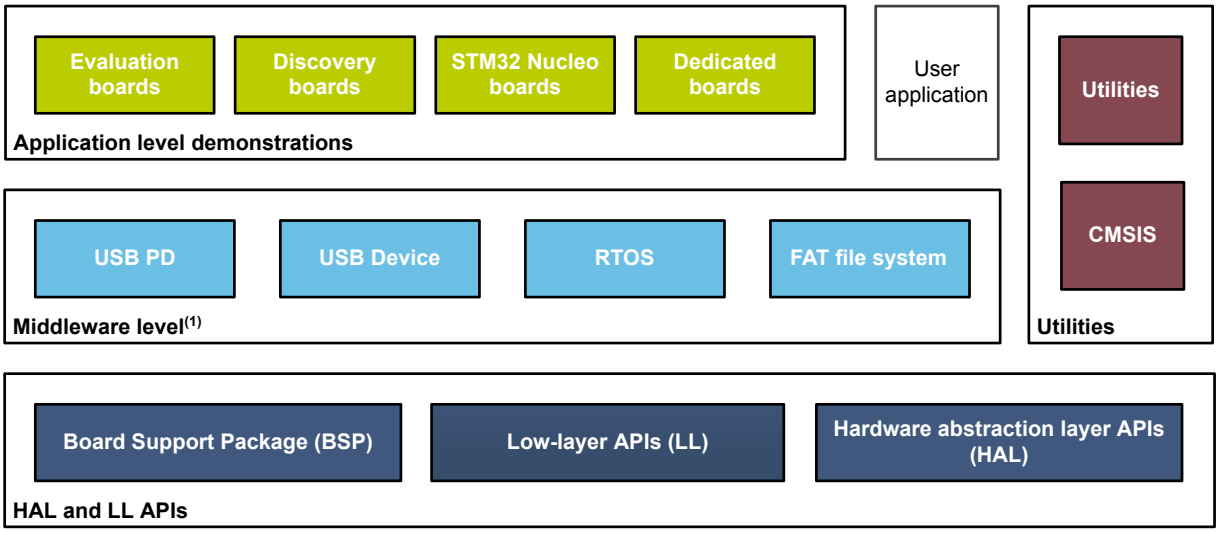
The STM32CubeG4 MCU Package also contains a set of middleware components with the corresponding examples. They come in free user-friendly license terms:

- FAT file system based on open source FatFS solution
- CMSIS-RTOS implementation with FreeRTOS™ open source solution
- USB PD Devices and Core libraries
- USB Device library

Several applications and demonstrations implementing all these middleware components are also provided in the STM32CubeG4 MCU Package.

The block diagram of STM32Cube is shown in Figure 1.

Figure 1. STM32CubeG4 firmware components



(1) The set of middleware components depends on the product Series.

The STM32G4 microcontrollers are based on the Arm® 32-bit Cortex®-M4 processor.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 Getting started with the demonstration

2.1 Hardware requirements

The hardware requirements to start the demonstration application are the following:

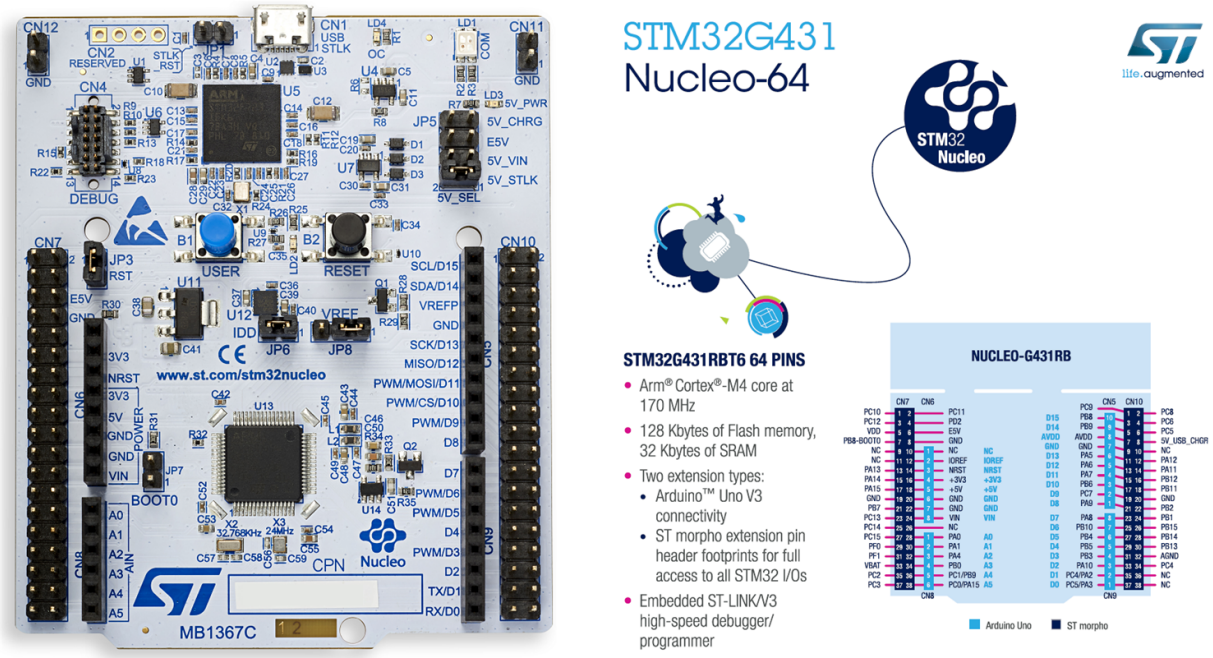
- NUCLEO-G431RB or NUCLEO-G474RE Nucleo board
- Adafruit® 1.8" TFT shield with joystick and microSD™ (reference ID: 802)
- One USB Type-A to Micro-B cable to power up the STM32 Nucleo board from the ST-LINK USB (USB connector CN1)
- One standard-capacity SD card (SDSC up to 4 Gbytes) or high-capacity SD card (HDSC up to 32 Gbytes)

2.1.1 NUCLEO-G431RB or NUCLEO-G474RE Nucleo board

The STM32 Nucleo board is a low-cost and easy-to-use development kit, to evaluate and start quickly some development with the Arm® 32-bit Cortex®-M STM32 microcontrollers. The demonstration runs on both the NUCLEO-G431RB and NUCLEO-G474RE Nucleo boards. Figure 2 shows the NUCLEO-G431RB and its description. Before installing and using the product, accept the Evaluation Product License Agreement available at www.st.com/epla.

For more information on the STM32 Nucleo boards visit www.st.com/stm32nucleo.

Figure 2. NUCLEO-G431RB Nucleo board



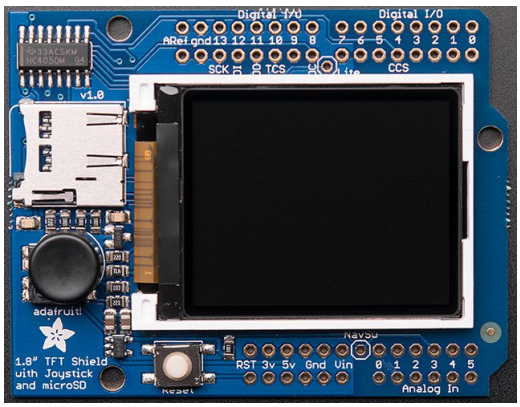
2.1.2 Adafruit® 1.8” TFT shield

The STM32 Nucleo board supports Arduino™ connectivity.

The compatible Adafruit® 1.8" TFT shield may be found on the Adafruit® website (reference ID: 802) with the following features (refer to Figure 3):

- one 1.8” 128×160 TFT color display
- one microSD™ card interface slot
- one 5-way joystick navigation switch (LEFT, RIGHT, UP, DOWN, and SEL)

Figure 3. **Adafruit® 1.8" TFT shield**



2.2 Hardware configuration

To start using the Adafruit® 1.8” TFT shield with the STM32 Nucleo board, follow the recommendations in Section 2.2.1 once all the hardware components are provisioned.

2.2.1 STM32 Nucleo board configuration

Check the positions of the jumpers on the STM32 Nucleo board as follows:

- JP1 OFF
- JP5 ON: PIN1 and PIN2 connected (ST-LINK configuration)
- JP6 (I_{DD}) ON

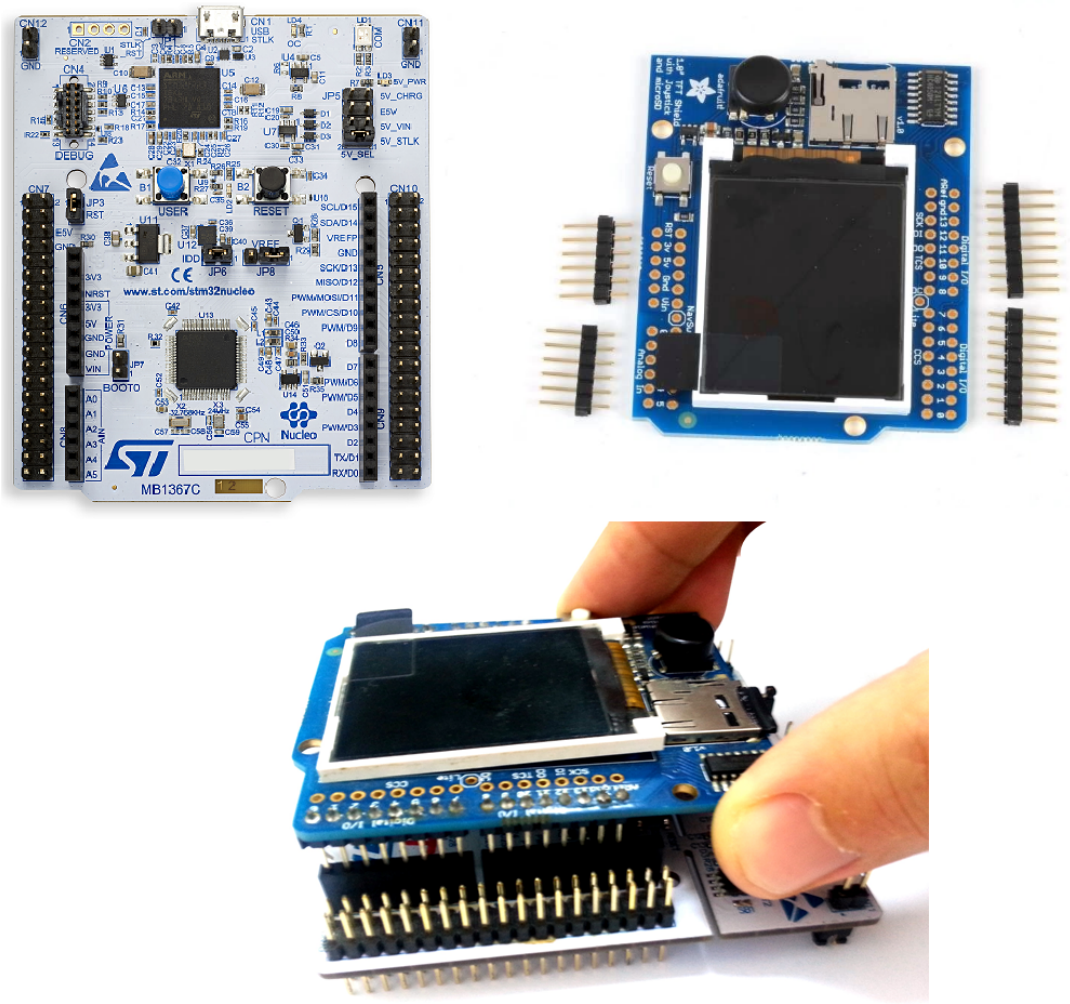
2.2.2 Assembling the Adafruit® shield

The Adafruit® TFT shield comes with all the surface parts soldered. The user can install the headers following the next steps:

1. Cut the breakaway header strip into sections, to fit the holes on the edge of the shield: two sections of six pins and two other sections of eight pins are needed.
2. To align the header strips for soldering, insert them (long pins down) into the headers of the STM32 Nucleo board using connectors CN5, CN6, CN8 and CN9.
3. Place the shield over the header strips, so that the short pins stick up through the holes.
4. Solder on each pin of the header onto the shield PCB to ensure good electrical contact.

The sequence of operations is illustrated in Figure 4.

Figure 4. **Assembling the Adafruit 1.8” TFT shield**

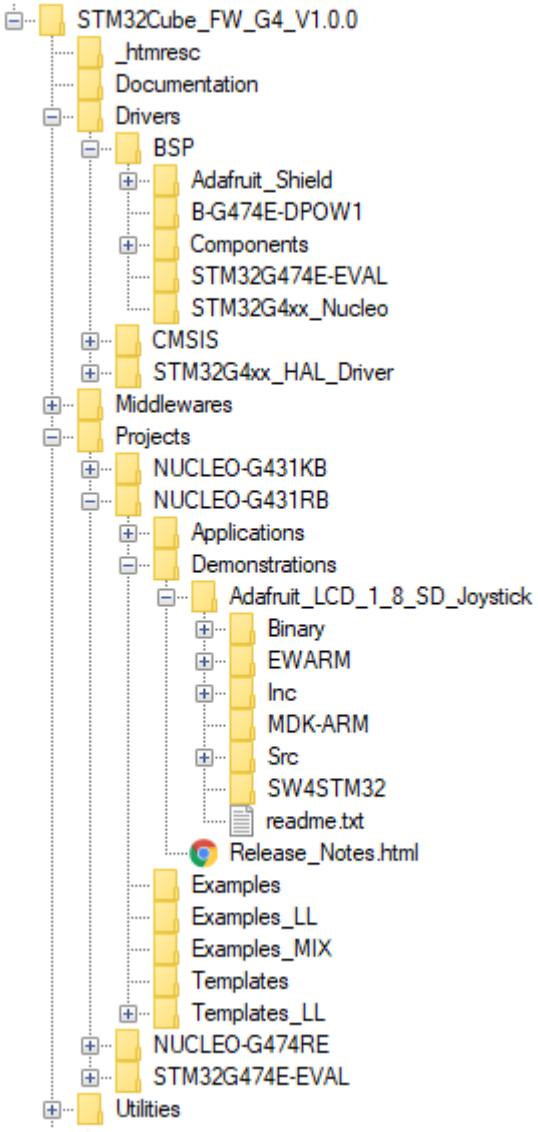


3 Demonstration firmware package

3.1 Demonstration repository

The demonstration `Adafruit_LCD_1_8_SD_Joystick` is provided within the STM32CubeG4 MCU Package, as shown in Figure 5.

Figure 5. Folder structure



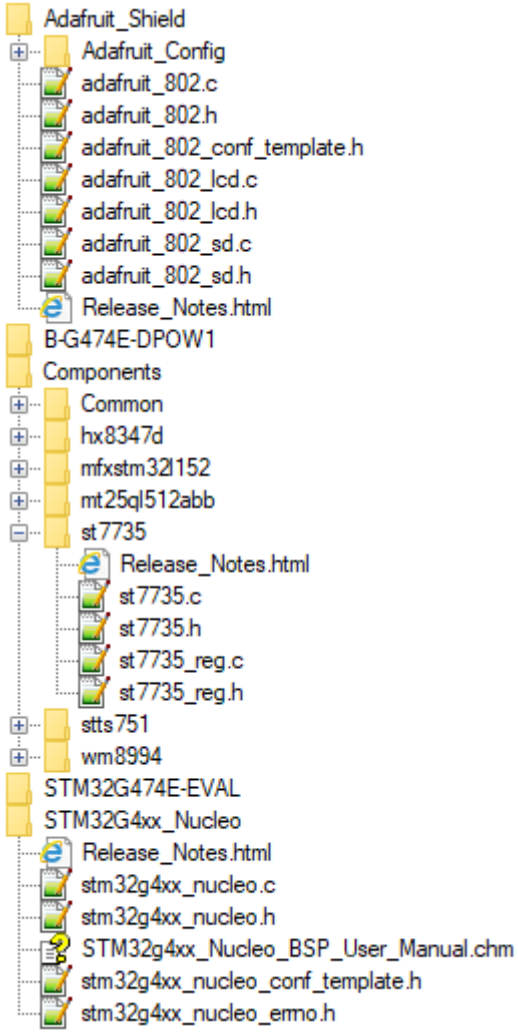
The demonstration sources are located in the project folder of the STM32CubeG4 MCU Package for the NUCLEO-G431RB and NUCLEO-G474RE Nucleo boards. The sources are divided into four groups, described as follows:

1. `Binary`: demonstration binary file in Hex format
2. `Inc`: contains the demonstration header files
3. `Src`: contains the demonstration source files
4. `Project settings`: a folder per toolchain containing the project settings and the linker files

3.2 STM32 Nucleo board BSP

Figure 6 shows the BSP architecture. For each board, a set of drivers for button, LED, and joystick is available within the `_nucleo.c` and `_nucleo.h` files, implementing the board capabilities and bus-link mechanism.

Figure 6. Nucleo BSP architecture



3.2.1 Joystick

The 5-way joystick on the shield is based on a resistor trick, to permit all switches to share one analog pin. Each movement of the joystick control connects a different resistor and results in a different voltage reading.

The ADC peripheral is configured within the `stm32g4xx_nucleo.c/stm32g4xx_nucleo.h` driver, to get analog voltage values through analog I/O pin 3. The `BSP_JOY_GetState()` function reads the analog pin and compares the result with five different ranges, to determine in which direction (if any) the stick has been moved: LEFT, RIGHT, UP, DOWN, or SEL.

3.2.2 LCD

The LCD available on the Adafruit® 1.8" TFT shield uses the 4-wire SPI to communicate with the STM32G431RBT6 or STM32G474RET6 chipset (digital I/O pins 13, 12, 11, and 10). It has its own pixel-addressable frame buffer to display text, shapes, lines, pixels, and others. The SPI peripheral is configured within the `stm32g4xx_nucleo.c/stm32g4xx_nucleo.h` driver, which contains also the SPI bus-link mechanism and I/O operations. The LCD is controlled by a dedicated BSP LCD driver (`stm32_adafruit_lcd.c/stm32_adafruit_lcd.h`), which uses the st7735 component that exports the LCD I/O operations needed for its process.

3.2.3 microSD™

The microSD™ slot available on the Adafruit® 1.8" TFT shield uses the 4-wire SPI to communicate with the STM32G431RBT6 or STM32G474RET6 chipset (digital I/O pins 13, 12, 11, and 10). The SPI peripheral is configured within the `stm32g4xx_nucleo.c/stm32g4xx_nucleo.h` driver, which contains also the SPI bus-link mechanism and I/O operations. The microSD™ is controlled by a dedicated BSP SD driver (`stm32_adafruit_sd.c/stm32_adafruit_sd.h`), which exports in a generic way the SD I/O operations needed for its process.

4 Functional description of the demonstration

This demonstration application shows how to use the STM32CubeG4 MCU Package with the NUCLEO-G431RB or NUCLEO-G474RE board, and the Adafruit® 1.8” TFT shield, by means of three examples related to the joystick, LCD, and microSD™ card.

Once started, the application checks the availability of the Adafruit® 1.8" TFT shield on top of the STM32 Nucleo board. This is done by reading the state of the I/O PB.00 pin (mapped to the joystick available on the shield). If the state of PB.00 is high, then the shield is available:

- If the Adafruit® 1.8" TFT shield is not available, LED2 blinks with a frequency of about 1 Hz. A first press on the user button makes LED2 blink with a frequency of about 5 Hz. At the second press, LED2 blinks with a frequency of about 10 Hz, and at the third press, back with a frequency of about 1 Hz again. This process operates as an infinite loop.
- If the Adafruit® 1.8" TFT shield is available, LED2 is turned ON, because it is sharing the same pin with the SPI CLK signal, used to communicate with the LCD and microSD™ available on the shield.

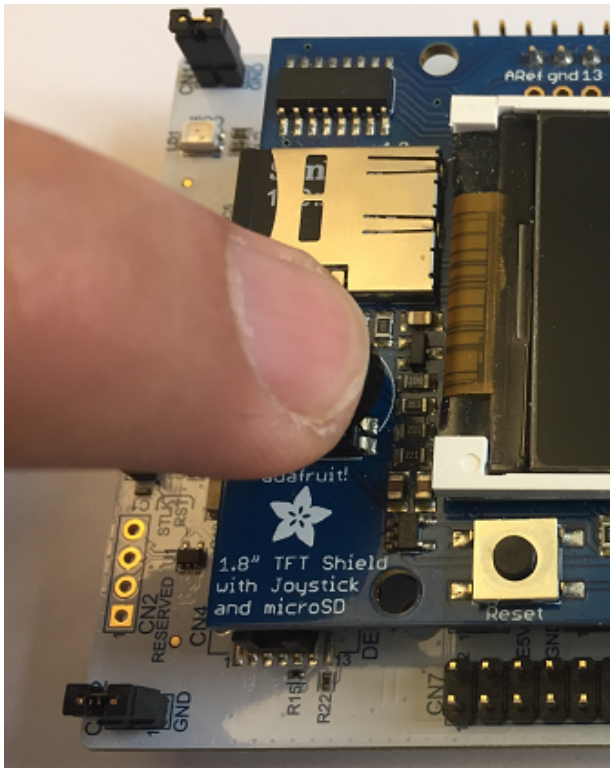
A menu is displayed on the Adafruit® 1.8" TFT describing the demonstration application, as shown in Figure 7.

Figure 7. Demonstration application menu on NUCLEO-G474RE

The user must follow the instructions below:

- Press the joystick SEL (refer to Figure 8) to start the first example related to the joystick:
  - Press the joystick DOWN, LEFT, RIGHT and UP to move the pointer "x" inside the displayed rectangle.
  - Press the joystick SEL to exit from the joystick example and display the description of the LCD example.
- Press the joystick SEL to start the second example related to the LCD:
  - See that different font sizes can be used to display text: 24, 20, and 16. Press the joystick SEL to get to the next page.
  - See that four shapes are drawn: a black hollow rectangle with red diagonals, a black filled rectangle, a blue hollow circle with red diameters, and a blue filled circle. Press the joystick SEL to get to the next page.
  - See that four shapes are drawn: a green hollow triangle, a red hollow ellipse, a green filled triangle, and a red filled ellipse. Press the joystick SEL to get to the next page.
  - See that a bitmap is displayed successively in the following LCD orientations: portrait, landscape, portrait rotated 180°, landscape rotated 180°, and back to portrait. No user action is required. The orientation is changed automatically after one second. Once the bitmap is back to its original position, press the joystick SEL to exit from the LCD example and display the description of the microSD™ card example.
- Press the joystick SEL to start the third and last example related to the microSD™ card:
  - First, make sure that a microSD™ card is inserted in the Adafruit® shield microSD™ holder.
  - See that the inserted microSD™ card is initialized, block erased, written, read, and tested (the written and read data must be identical). Once the test is over, the microSD™ card can be removed. Press the joystick SEL to exit from the microSD™ card example and get back to the description of the first joystick example.
- This described process is run in an infinite loop.

Figure 8. Reading the Adafruit shield joystick



The application manages some errors (refer to Figure 9) that can occur during the initialization of the joystick or microSD™ card.

Figure 9. Demonstration application error messages

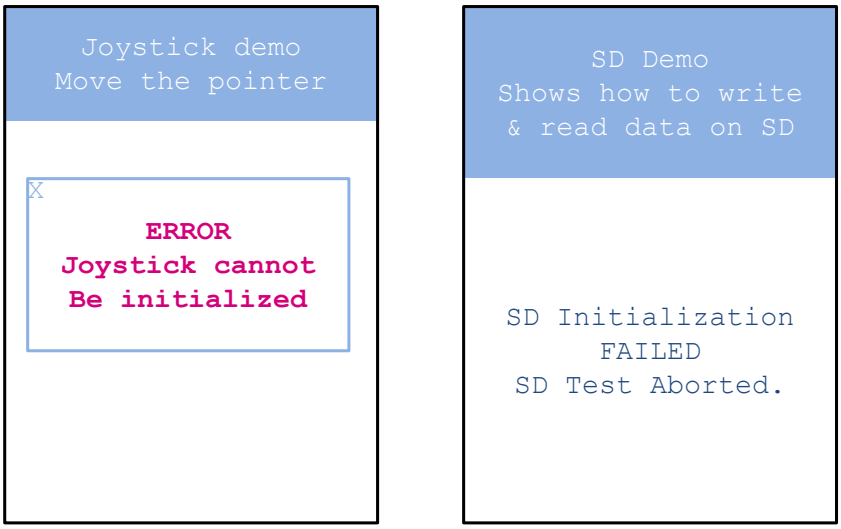
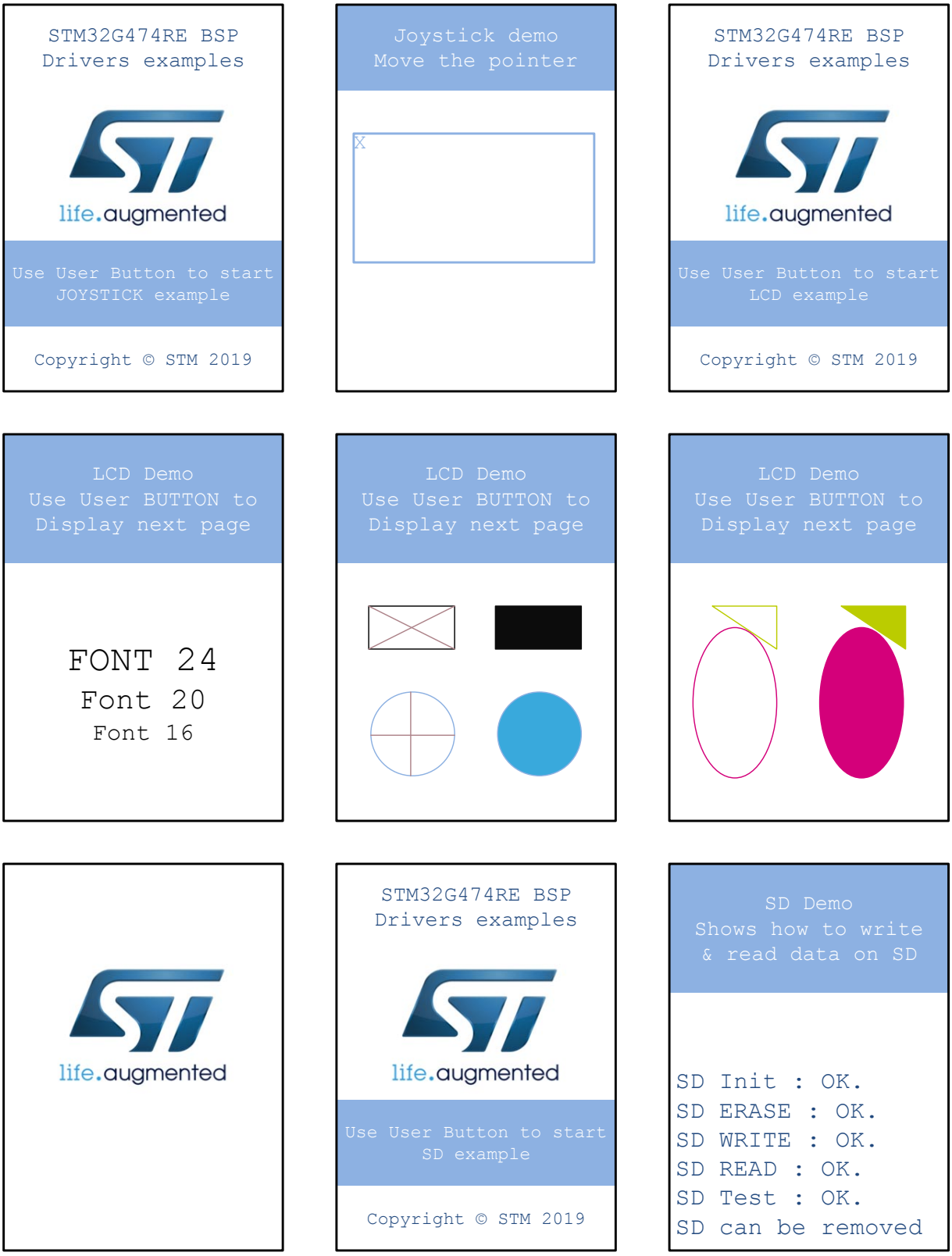


Figure 10 shows the demonstration running.

Figure 10. Demonstration running



4.1 Programming the firmware application

To program the STM32 Nucleo board with the demonstration application, proceed as follows:

1. Install the preferred Integrated Development Environment (IDE)
2. Install the STLINK-V3E driver available from the STMicroelectronics website
3. Choose one of the supported tool chains, such as IAR™, Keil®, or AC6, and follow the steps below:
   - Open the application folder:
     `Projects\SM32G431RB-Nucleo\Demonstrations\Adafruit_LCD_1_8_SD_Joystick`
     or
     `Projects\SM32G474RE-Nucleo\Demonstrations\Adafruit_LCD_1_8_SD_Joystick`
   - Chose the desired IDE project (EWARM for IAR™, MDK-ARM for Keil® or the SW4STM32 System Workbench for AC6)
   - Double click on the project file (for example `Project.eww` for EWARM)
   - Rebuild all files: go to [**Project**] and select [**Rebuild all**]
   - Load the project image: go to [**Project**] and select [**Debug**]
   - Run the program: go to [**Debug**] and select [**Go**]

The demonstration software as well as other software examples that allow the users to discover the STM32 microcontroller features, are available on STMicroelectronics website at www.st.com/stm32nucleo.

Revision history

Table 1. **Document revision history**

Date	Version	Changes
21-May-2019	1	Initial release.

Contents

List of tables

List of figures

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2019 STMicroelectronics – All rights reserved