Image viewer based on the Adafruit® TFT shield with STM8 Nucleo-64 boards

Introduction

The NUCLEO-8S208RB (built around the STM8S208RBT6 device) and the NUCLEO-8L152R8 (built around the STM8L152R8T6 device) are boards that allow the evaluation of the main features of all the STM8S Series and STM8L Series microcontrollers.

This application note provides a short description on how to use the Adafruit 1.8" TFT shield with STM8 Nucleo-64 boards.

Once the microcontroller (STM8S208RBT6 or STM8L152R8T6 in this example) has been powered up through a USB cable connected to the host PC, an availability check of the Adafruit 1.8" TFT is done. After the check, a menu is displayed to select the image scrolling mode (manual or automatic) using the shield joystick. As a result, all .bmp images saved in the microSD card root are displayed in the desired mode.

Table 1. Applicable products

Type	Part number
Evaluation boards	NUCLEO-8S208RB
	NUCLEO-8L152R8

Reference documents

- *STM8 Nucleo-64 boards* data brief (DB3591)
- *STM8L152R8T6 Nucleo-64 board* user manual(UM2351)
- *STM8S208RBT6 Nucleo-64 board* user manual (UM2364)

AN5182 - Rev 1 - June 2018
For further information contact your local STMicroelectronics sales office.

www.st.com

1 Getting started with the demonstration

1.1 Hardware requirements

The hardware requirements to start the demonstration application are:

- STM8 Nucleo-64 booard (NUCLEO-8S208RB or NUCLEO-8L152R8)
- Adafruit 1.8" TFT shield with joystick and microSD (reference ID: 802)
- One "USB type A to Micro-B" cable to power up the board from the USB ST-LINK (USB connector CN6)
- A standard capacity SD card (SDSC) with a capacity up to 32 Gbytes.

1.1.1 STM8 Nucleo-64 boards

The STM8 Nucleo-64 boards are a low-cost and easy-to-use development kit that helps to quickly evaluate and start some development with STM8S Series and STM8L Series microcontrollers.

Before installing and using the product, accept the evaluation product license agreement available at www.st.com/epla.

For more information on the STM8 Nucleo-64 boards go to www.st.com/stm8nucleo.
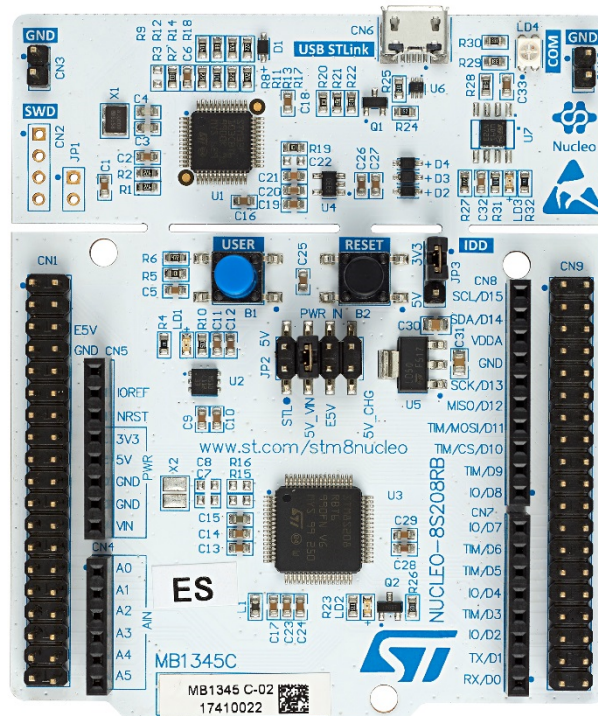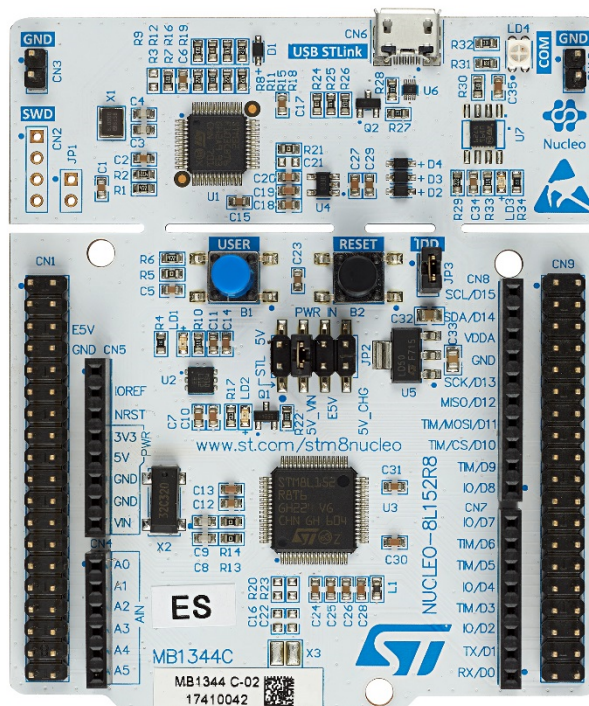
**Figure 1. NUCLEO-8S208RB board**

Figure 2. NUCLEO-8L152R8 board

1.1.2 Adafruit TFT shield

The STM8 Nucleo-64 board supports Arduino connectivity.

This Adafruit 1.8" TFT shield may be found on the Adafruit website (reference ID 802) with the following features:

- One 1.8" TFT display with 128x160 color pixels
- One microSD card interface slot
- One 5-way joystick navigation switch (left, right, up, down, select).

Figure 3. Adafruit 1.8" TFT shield
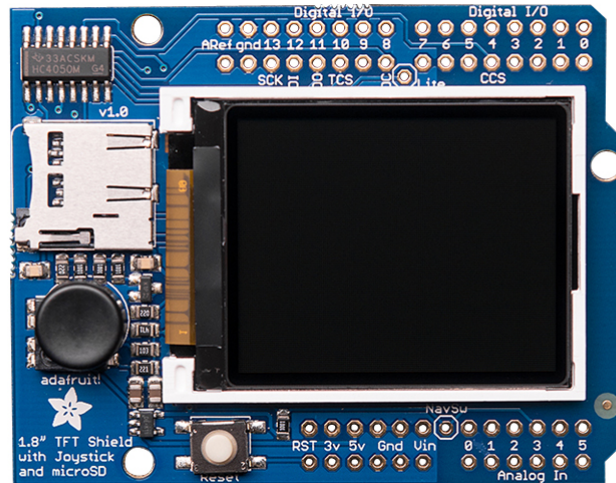
Note: *This shield is just an example of Arduino shield usage; more details are available on Adafruit website.*

1.2 Hardware configuration

Other than gathering the required hardware, some recommendations must be followed before start using the Adafruit 1.8" TFT shield with the STM8 Nucleo-64 board. This section specifies the recommended boards configurations and how to assemble the Adafruit shield.

1.2.1 STM8 Nucleo-64 boards configuration

For the NUCLEO-8S208RB board, check jumpers' positions as follows:

- JP1 OFF
- JP2 (PWR) on position (1-2)
- JP3 (I_{DD})on position (1-2) or (2-3)

For the NUCLEO-8L152R8 board, check jumpers' positions as follows:

- JP1 OFF
- JP2 (PWR) on position (1-2)
- JP3 (I_{DD}) ON.

1.2.2 Assembling the Adafruit shield

The Adafruit TFT shield comes with all surface mount parts presoldered. User can install the headers following the next steps:

1. Cut the breakaway header strip into sections. In order to fit the holes on the edge of the shield, two sections of 6 pins and two other sections of 8 pins are needed.
2. To align the header strips for soldering, insert them (long pins down) into the headers of the STM8 Nucleo-64 board using the 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 is shown in the figure below:

Figure 4. **Assembling the Adafruit 1.8" TFT shield**
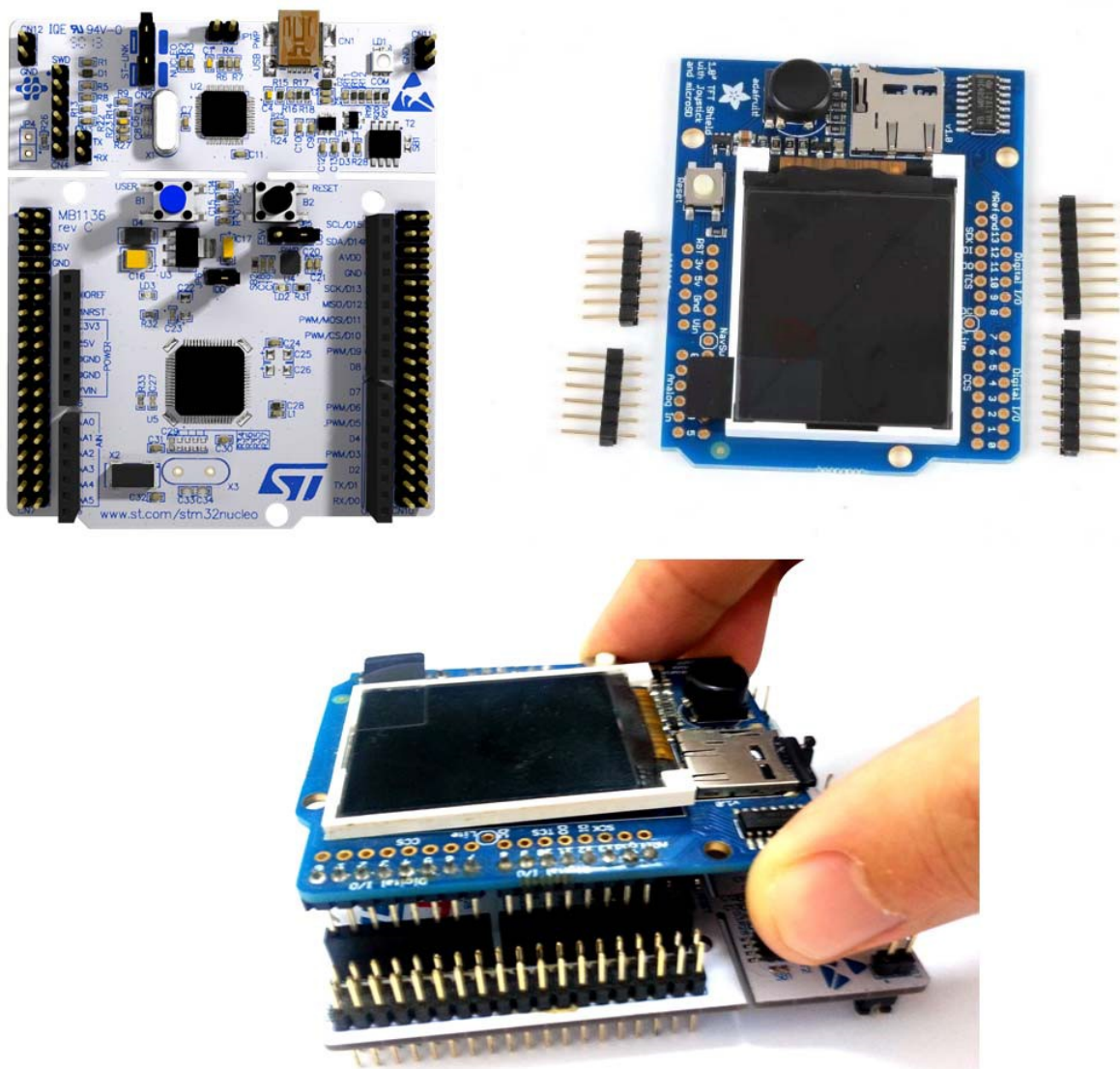
2 Example firmware package

2.1 Example repository

The examples sources are located in the projects folder of the dedicated STM8S and STM8L boards firmware packages. The sources are divided into four groups described as follows:

- Media: contains BMP_128x160 image files
- Inc:contains the example header files
- Src: contains the example source files
- Project settings: a folder per toolchain containing the project settings.

2.2 Nucleo boards drivers

For each board, a set of button, LED and joystick drivers is available within the *stm8s_nucleo.c/.h* files (for NUCLEO-8S208RB board) and *stm8l_nucleo.c/.h* files (for NUCLEO-8L152R8 board), implementing the board capabilities and the bus link mechanism.

2.2.1 Joystick

The 5-way joystick on the shield is based on a resistor trick to permit all the 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 *stm8s_nucleo.c/.h* driver or *stm8l_nucleo.c/.h* driver respectively, in order to get analog voltage values through the 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, select).

2.2.2 LCD

The LCD available on the Adafruit 1.8" TFT shield uses 4-wire SPI to communicate with the STM8S Series or STM8L Series device (Digital I/O pins 13, 11, 10 and 8) and has its own pixel-addressable frame buffer to display text, shapes, lines, pixels, etc.

The SPI peripheral is configured within the *stm8s_nucleo.c/.h* driver or *stm8l_nucleo.c/.h* driver for each board respectively. Each driver also contains the SPI bus link mechanism and the IO operations.

The LCD is controlled by a dedicated BSP LCD driver *stm8_adafruit_lcd.c/.h* that exports in a generic way the LCD IO operations needed for its process.

2.2.3 MicroSD

The microSD slot available on the Adafruit 1.8" TFT shield uses 4-wire SPI to communicate with the STM8S Series or STM8L Series chip (Digital I/O pins 13, 12, 11 and 4).

The SPI peripheral is configured within the *stm8s_nucleo.c/.h* driver or *stm8l_nucleo.c/.h* driver respectively. The driver also contains the SPI bus link mechanism and the IO operations.

The microSD is controlled by a dedicated BSP SD *driver stm8_adafruit_usd.c/.h* which exports the SD IO operations needed for its process in a generic way.

3 Example functional description

This example shows how to use the STM8S/L Series firmware package with the STM8 Nucelo-64 boards and the Adafruit 1.8" TFT shield to display a 128x160 pixel full color bitmap from a microSD card using the FatFS file system.

To start with this example, the user has to copy the provided 128x160 pixel bitmap pictures, available within the FW package under "\Media" folder, to the root directory of a FAT formatted microSD card and insert the microSD card into the Adafruit shield microSD holder.

Note that the microSD card can have a storage capacity of up to 4 Gbytes (SDSC) and that the bitmap images must have the properties detailed in the table below.

Table 2. Bitmap image properties

	Dimensions	128 x 160
	Width	128 pixels
	Height	160 pixels
	Bit depth	16
	Item type	BMP file
	Name	Must not exceed 11 characters (including the bpm extension)

Once started, the application checks the availability of the Adafruit 1.8" TFT shield on top of the STM8 Nucleo-64 boards. This is done by reading the state of IO PB.02 pin (mapped to the joystick available on the shield). If the state of PB.02 is high, then the shield is available.

If the Adafruit 1.8" TFT shield is not available, the USER LED toggles with a frequency equal to ~1 Hz.

A second press on the user button lets USER_LED toggling with a second frequency equal to ~5 Hz.

The third press, changes USER_LED toggling frequency to ~10 Hz.

The described process is done in an infinite loop.

If the Adafruit 1.8" TFT shield is available, USER_LED is turned ON, because it is sharing the same pin with the SPI CLK signal used to communicate with the LCD and the microSD available on the shield.

A menu is displayed on the Adafruit 1.8" TFT describing the demonstration application, as shown in the figure below.
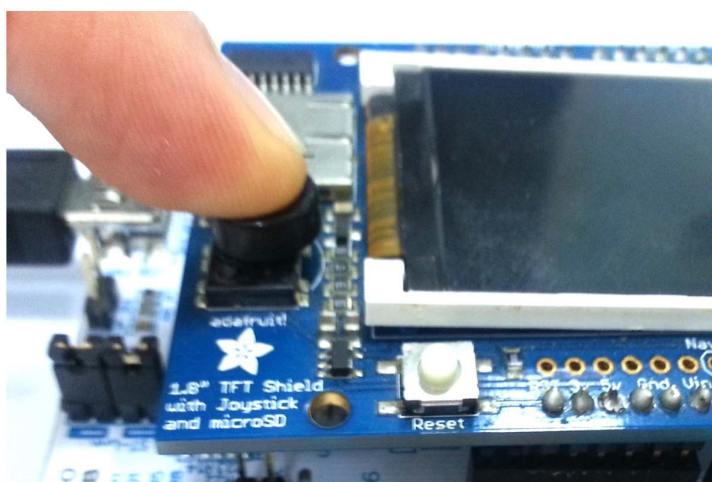
Figure 5. Demonstration application menu
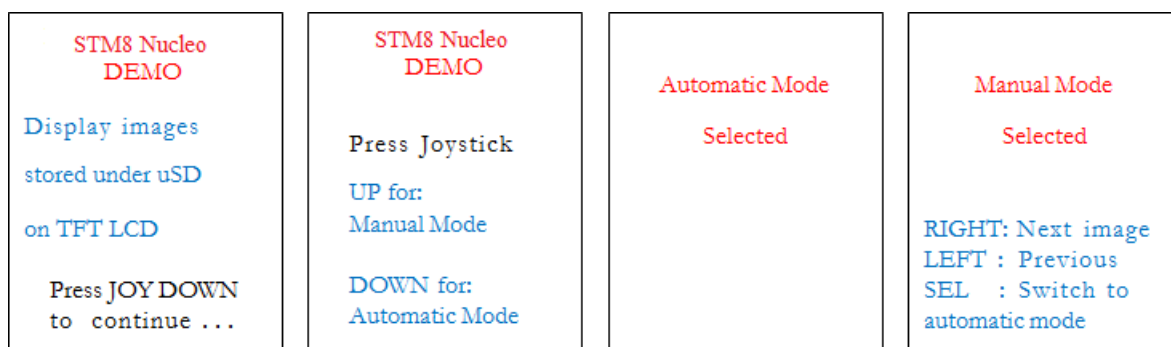
The user has to follow the instructions below:

1. Press the joystick DOWN to continue the display of the menu (see figure below)

**Figure 6. Reading the Adafruit shield joystick**

2. Choose one of the available display modes (manual or automatic) using the joystick button:
   – **Automatic mode:** selected by pressing the joystick DOWN.
     The bitmap images available on the microSD card are displayed sequentially in a forever loop.
   – **Manual mode:** selected by pressing the joystick UP.
     The bitmap images available on the microSD card are displayed by pressing joystick RIGHT to display next images, or Joystick LEFT to display the previous one.
     A long pressing (~1 s) goes to the joystick SEL, which switches the display mode from manual to automatic.

Note that the application manages some errors (see the figure below), that can occur when accessing the microSD card to load the bitmap images:

- If the microSD card is not FAT formatted, a message is displayed on TFT.
  In this case, format the microSD card and put the bmp files available within the FW package under \Media folder into its root directory.
- If the content of the microSD card is other than a bitmap file, a message is displayed on TFT mentioning that the file is not supported.
  The user has to ensure that the files, available under the microSD card root directory, are respecting the described bitmap properties.

Open directory fails.	SD Card is not FAT formatted. Please Format the microSD card.	Cannot mount FatFs on Drive.	File type is not supported.
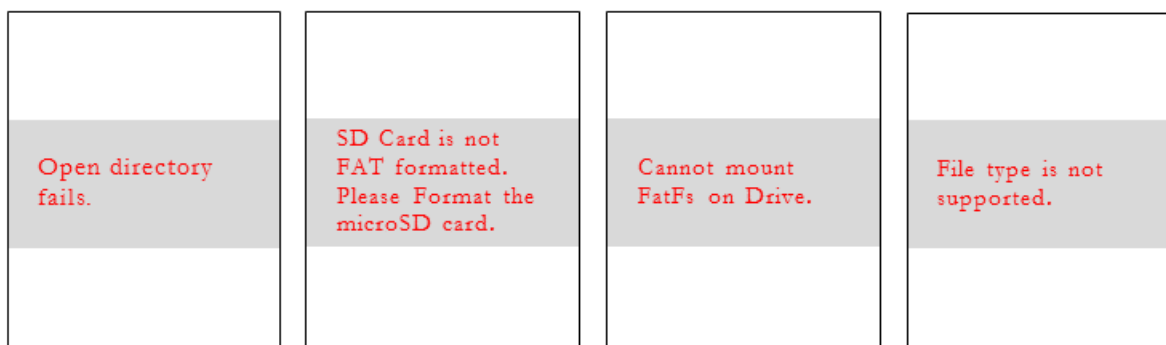
The figure below show an example of the application running.

Figure 8. Example of application running on STM8 Nucleo-64 board
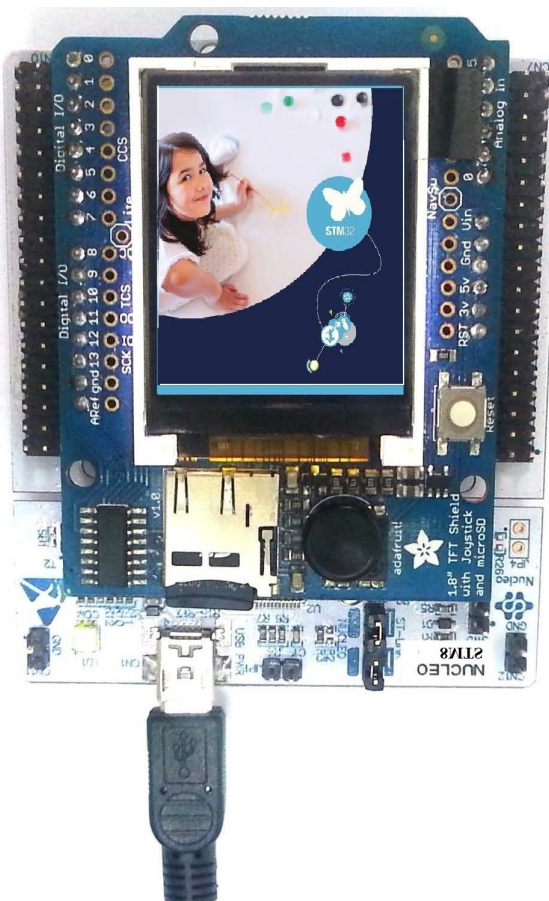
3.1 Programming firmware application

Toprogram the STM8 Nucleo-64 board with the example, proceed as follows:

- Install the preferred integrated development environment (IDE).
- Install the ST-LINK/V2.1 driver available on the ST website.
- Choose one of the supported tool chains (IAR® / STVD-COSMIC) and follow the steps below:

1. Open the application folder: Project\Image_Viewer
2. Chose the desired IDE project (EWARM for IAR, STVD-COSMIC)
3. Double click on the project file (for example Project.eww for EWARM)
4. Rebuild all files: go to **Project** and select **Rebuild all**
5. Load the project image: go to **Project** and select **Debug**
6. Run the program: go to **Debug** and select **Go**.

The demonstration software as well as other software examples that allow the user to discover the STM8 microcontrollers features are available on ST website at www.st.com/stm8.

4 Frequenty asked questions

1. **How can this application be used to display the user's own images?**

   By using any image editing tool and croping the image to no larger than 160 pixels high and 128 pixels wide.

   The image must be saved as a 16-bit color BMP format file.

2. **Can more bitmap files be displayed?**

   Yes. More pictures can be displayed. The user has to copy the images under the microSD root directory and modify the define value of MAX_BMP_FILES constant to the desired number of files.

   The _FS_LOCK value has to be fine tuned by defining the number of files that can be opened simultaneously. This is done under the "ffconf.h" the FatFs configuration file.

3. **Can the bitmap files be put other than under the root directory of the microSD?**

   If the bitmap files are stored in a folder other than the root directory, they cannot be accessed by the demonstration application. The "File type not supported" error message is displayed on the LCD. To make it work the user has to add the new directory path within f_open() and f_opendir() FatFs APIs calls' under the fatfs_storage.c file.

Revision history

Table 3. Document revision history

Date	Version	Changes
29-Jun-2018	1	Initial release.

Contents

List of tables

List of figures

IMPORTANT NOTICE – PLEASE READ CAREFULLY