



life.augmented

STM32H7R-S workshop

Graphical UI

Advanced Graphics UI

- Advanced graphics UI demand:
 - **Substantial ROM** for housing an extensive collection of graphical assets (bitmaps, fonts)
 - **Sufficient RAM** to hold the frame buffer (size determined by the screen resolution, color depth)
 - **High-speed bus interface** to ensure high frame rates
 - **Powerful computation capabilities** for graphics rendering
- STM32H7R/S offers:
 - Choose external ROM / RAM fitting your UI requirements
 - High-speed external memory interfaces with speeds up to 200 MHz DTR
 - ARM Cortex-M7 running @600 MHz, bolstered by graphic accelerators and enablers

For the graphics demo on your board assets size is around 45.8 MBytes

Configuration :
screen resolution 800x480
16 bits color depth with double frame buffer.

Frame buffer size: $800 \times 480 \times 16 \text{ bits} \times 2$: 1500 KBytes

High Performance Graphics on STM32H7R/S

Graphical accelerator :

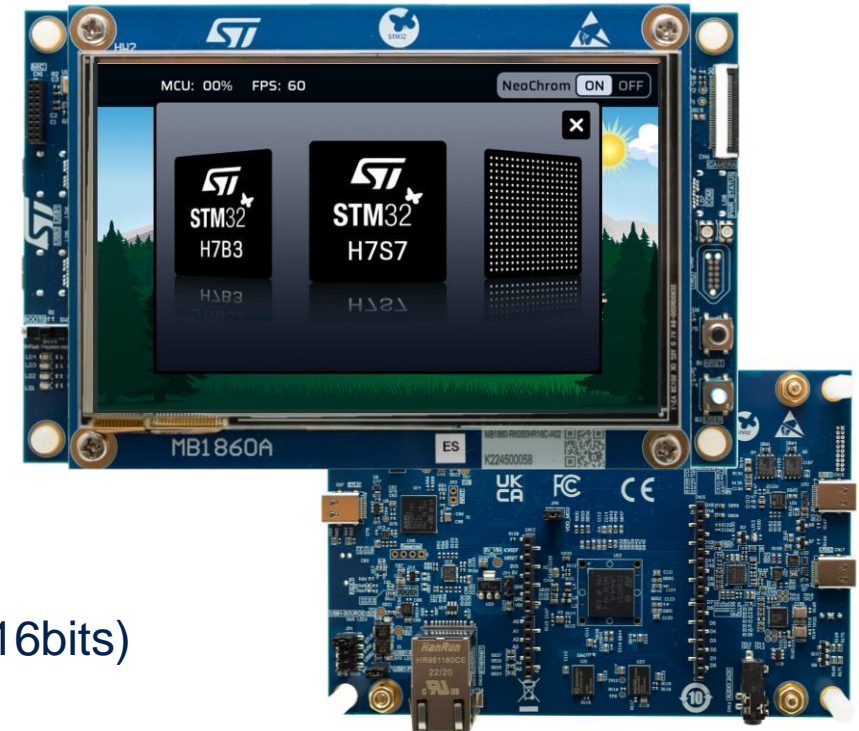
- Chrom-ART™ (DMA2D)
- **NeoChrom GPU™** (GPU2D)

Graphic enablers

- JPEG CODEC for accelerating MJPEG Videos
- Chrom-GRC™ (GFXMMU)

Supported display interface:

- Parallel (FMC) interface displays supporting up to WSVGA (1024×600 16bits)
- **Parallel LCD-TFT controller** (LTDC) supporting up to SVGA resolution (800 × 600 24bits)
Format : RGB-TFT 16/18/24bits and parallel 8080 display interface



STM32 hardware embedded graphics HW acceleration

NeoChrom GPU (GPU2D)

Offloads the CPU from graphics tasks

Lower memory consumption

Higher GUI performance – smooth and richer graphics effects:

- Realizing 3D-like graphics on STM32 microcontroller

→ The technology behind

- Simple Drawing
- 2D Copy
- Alpha blending
- Color format conversion
- **Advanced Drawing**
- **Scaling, Rotation**
- **Perspective correct texture mapping**
- Image format compression

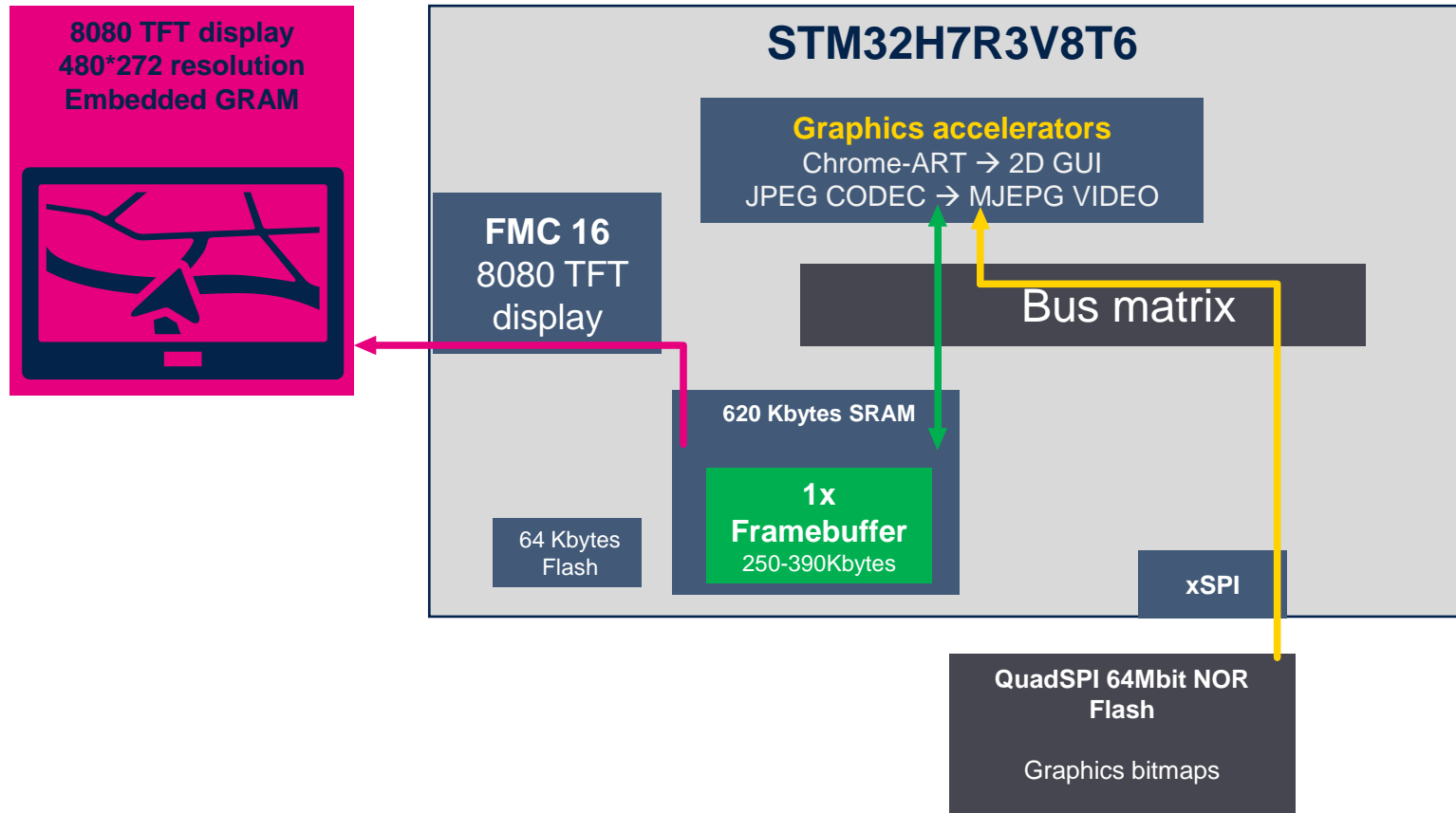


STM32H7R3: Simple Purpose GUI

100K BoM Estimations (MCU + memory)

- STM32H7R3V8T6: 2.4 \$
- QuadSPI 64Mbit: 0.6 \$

Total Estimations: 3 \$



UI requirements

1. 480*272 16 or 24bpp.
2. 2D graphics
3. Background, few buttons, text
4. Display sensor inputs
5. Control xy machine settings
6. 5 screens, simple screen transitions

Other requirements

- Ethernet for PoE and communication
- CAN-FC for machine/industrial communication

Microcontroller & Memory

CPN: STM32H7R3V8T6
Package: LQFP100
External flash: QuadSPI NOR
External ram: N/A

STM32H7R7: High performance GUI

100K BoM Estimations (MCU + memory)

- STM32H7R7L8: 3.7 \$
- Octal SPI 256 Mbit: 1.5 \$
- 32bit SDRAM 64Mbit: 1 \$

Total Estimations: 6.2 \$

UI requirements

1. 800*480 16 bpp. 60FPS
2. 2 & 2.5D graphics
3. Background, buttons, sliders, etc.
4. Scaling of icons on a slider wheel
5. MJPEG video fullscreen
6. Display and control product via UI

Other requirements

- Memory scalability to accomodate different markets.
- OTA FW update requiring dual bank ext memory
- Ethernet*

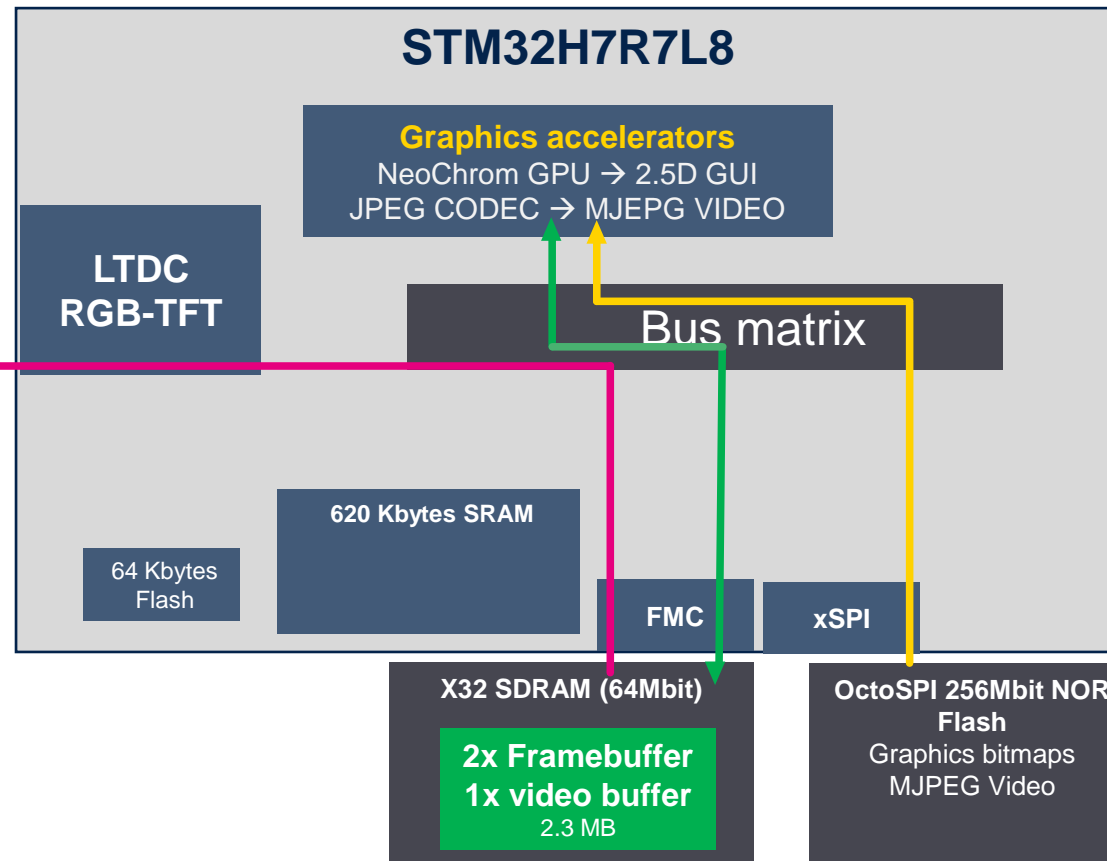
Microcontroller & Memory

CPN: STM32H7R7L8

Package: TFBGA225

External flash: OctoSPI Flash

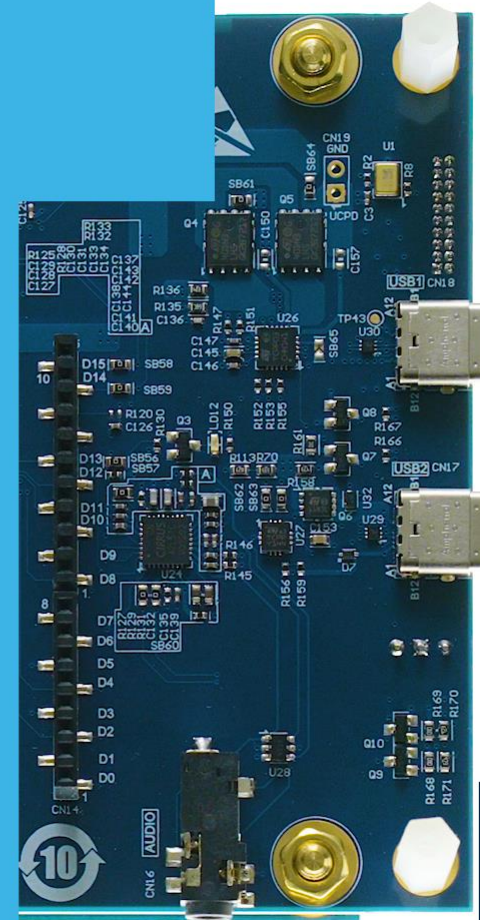
External ram: SDRAM



- Step 1 : NeoChrom callback to get the assets from external flash
- Step 2 : The image is processed by Neochrom and stored into dedicated memory in External RAM
- Step 3 : The framebuffer is transferred to the display

STM32H7S8-DK Demo

Run MPU-Like
GUI applications
on STM32H7RS





TouchGFX sub-demo overview

MICROCONTROLLER, DISCO BOARD & DEMO INFORMATION

MCU: STM32H7S7L8 Running at 600MHz (Cortex-M7)

Internal flash: 64kB Bootflash

Internal SRAM: 620kB

New MCU Features

High speed memory interfaces (200MHz), NeoChrom GPU, 2xUSB-C w. UCPD, I3C, Advanced Security and much more

Board information

External flash: OctoSPI NOR Flash

External RAM: 16-bit Serial PSRAM

LCD: 5" RGB-TFT IPS display, with 800 x 480 resolution and 16M colors

Demo:

TouchGFX based UI Demos running in 16bpp, double framebuffers in external memory

SVG DEMO



SVG demo

Cortex-M7 at 600Hz enable
smooth vectorial graphic

TouchGFX sub-demo overview





TouchGFX sub-demo overview

NeoChrom **ON**

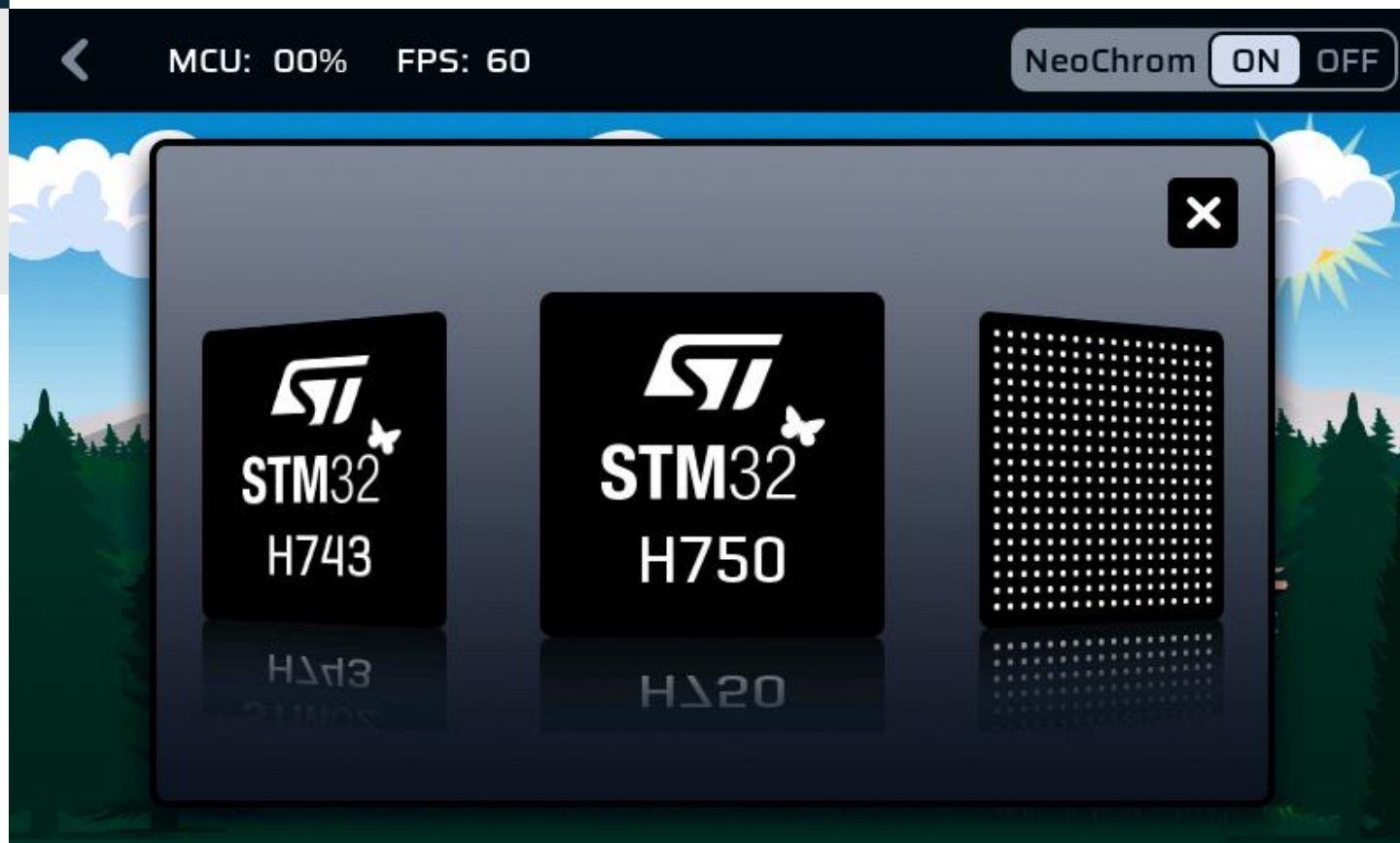
CPU: ~14%
FPS: 60

NeoChrom **OFF**

CPU: ~80%
FPS: 19

Rousset Factory

Bitmap scaling
Bitmap rotations
Perspective correct texture
mapping





MJPEG video
480*272 MJPEG video

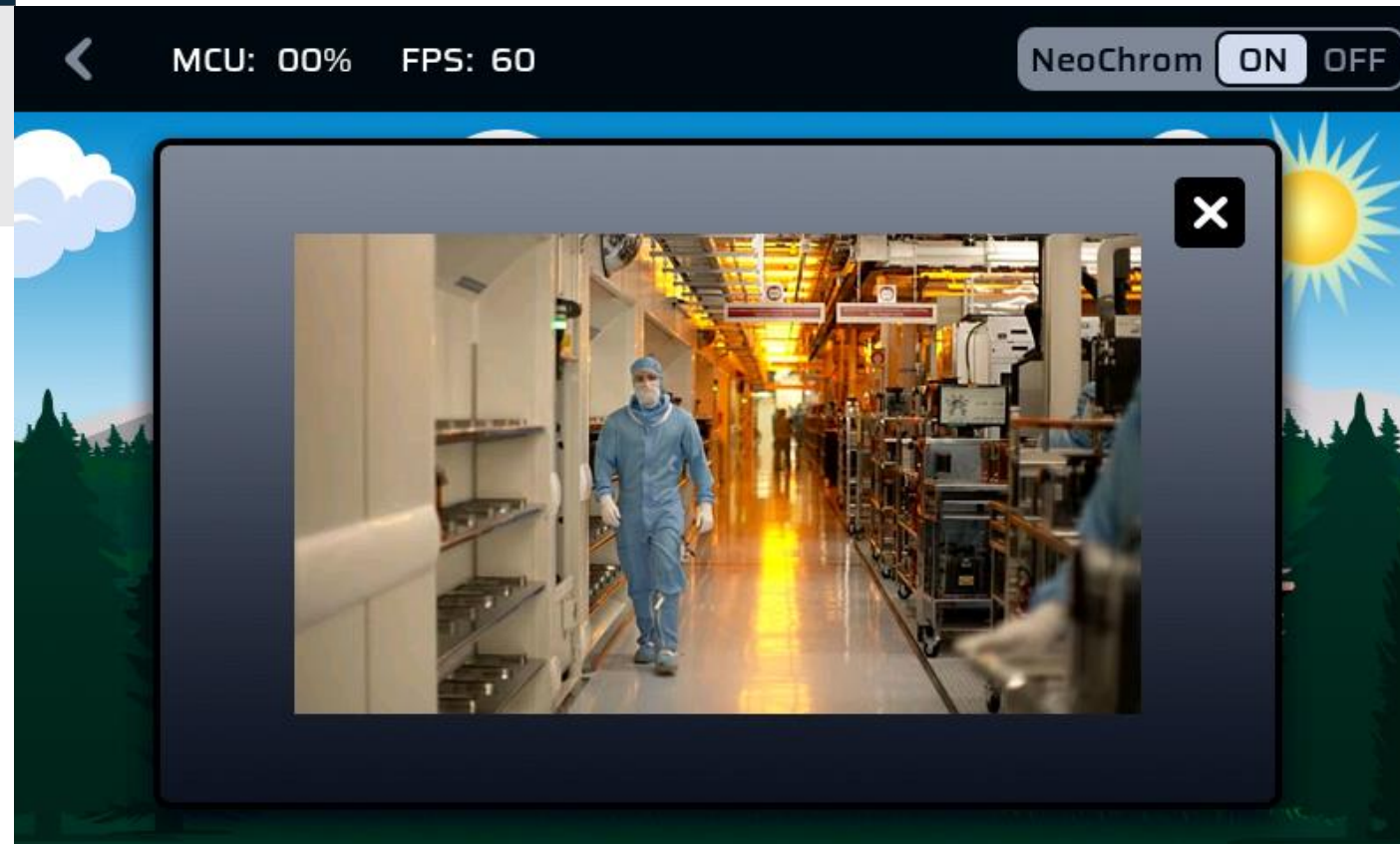
TouchGFX sub-demo overview

NeoChrom **ON**

CPU: 30%
FPS: 60

NeoChrom **OFF**

CPU: ~80%
FPS: 30



TRANSITIONS DEMO



Screen Transition

Full screen bitmap animation
(spin out)
Bitmap rotation, scaling, move,
spin, etc.
Menu overlay

TouchGFX sub-demo overview

NeoChrom **ON**

CPU: ~5%
FPS: ~35-60

NeoChrom **OFF**

CPU: ~77%
FPS: 25



COMPASS DEMO



Compass

Large bitmap rotation

TouchGFX sub-demo overview

NeoChrom **ON**

CPU: ~5%
FPS: ~45-60

NeoChrom **OFF**

CPU: ~85%
FPS: 20



E-BIKE DEMO



E-Bike

Anti-aliased drawing (graph)
Alpha blending
Gauge with rotating needle

TouchGFX sub-demo overview

NeoChrom **ON**

CPU: ~40%
FPS: 60

NeoChrom **OFF**

CPU: ~60%
FPS: 30



Free GUI Solution for STM32H7R/S

Utilize the X-Cube-TouchGFX Free software tool for developing amazing graphical user interfaces.

- High performance
- Low memory footprint
- WYSIWYG PC Tool with PC Simulator
- +40 widgets
- Free stock-images
- Multi language
- Raster and Vector graphics

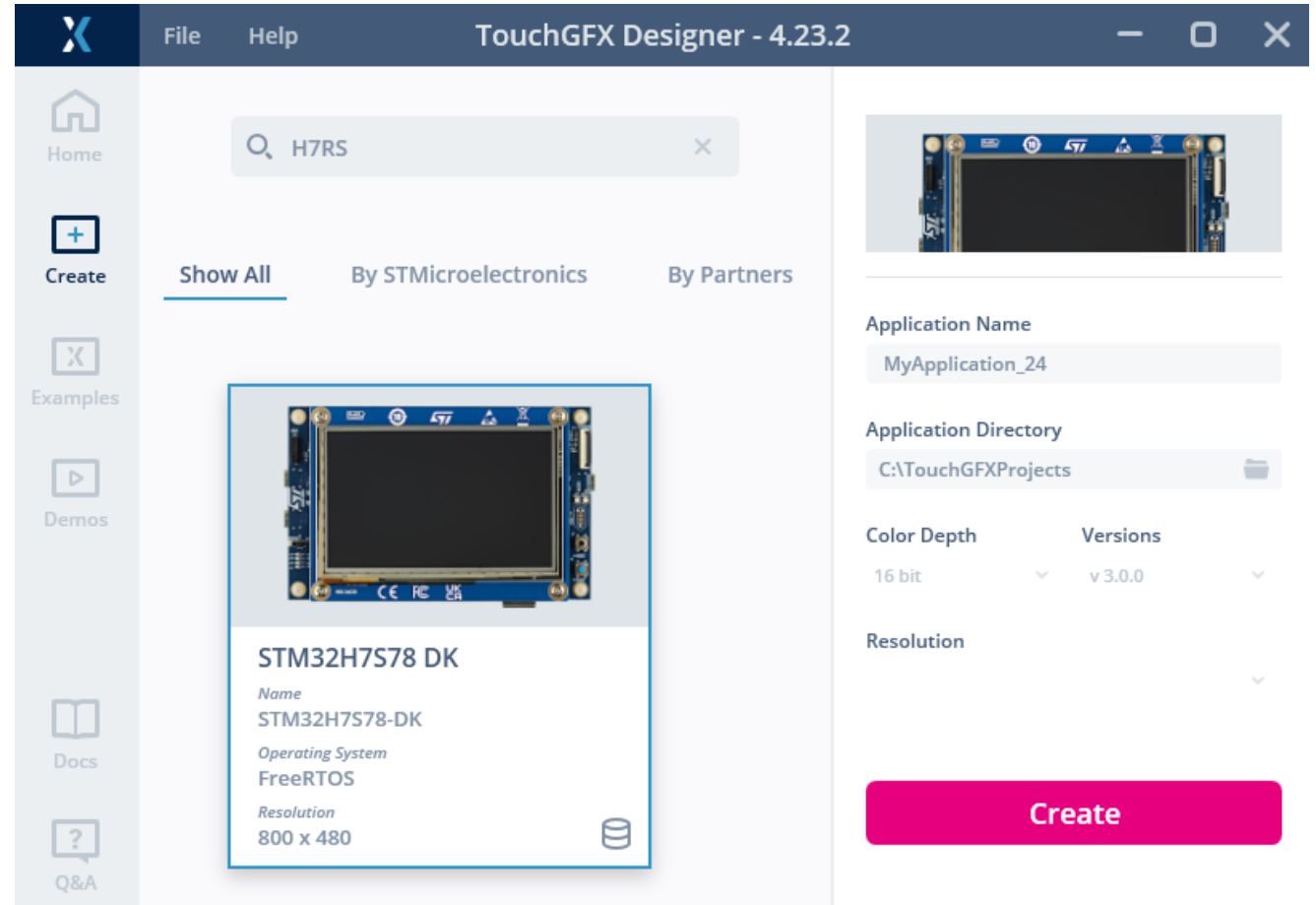


TouchGFX











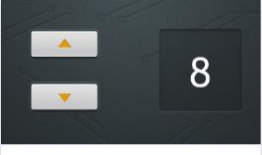


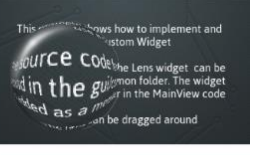

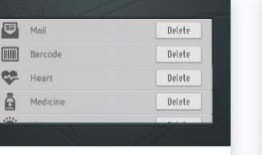
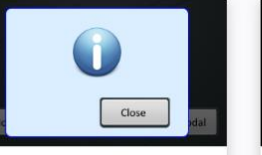
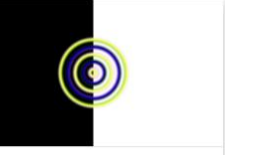
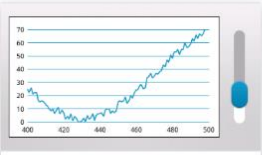






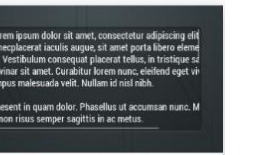
life.augmented

TouchGFX Board Setup for STM32H7S78-DK

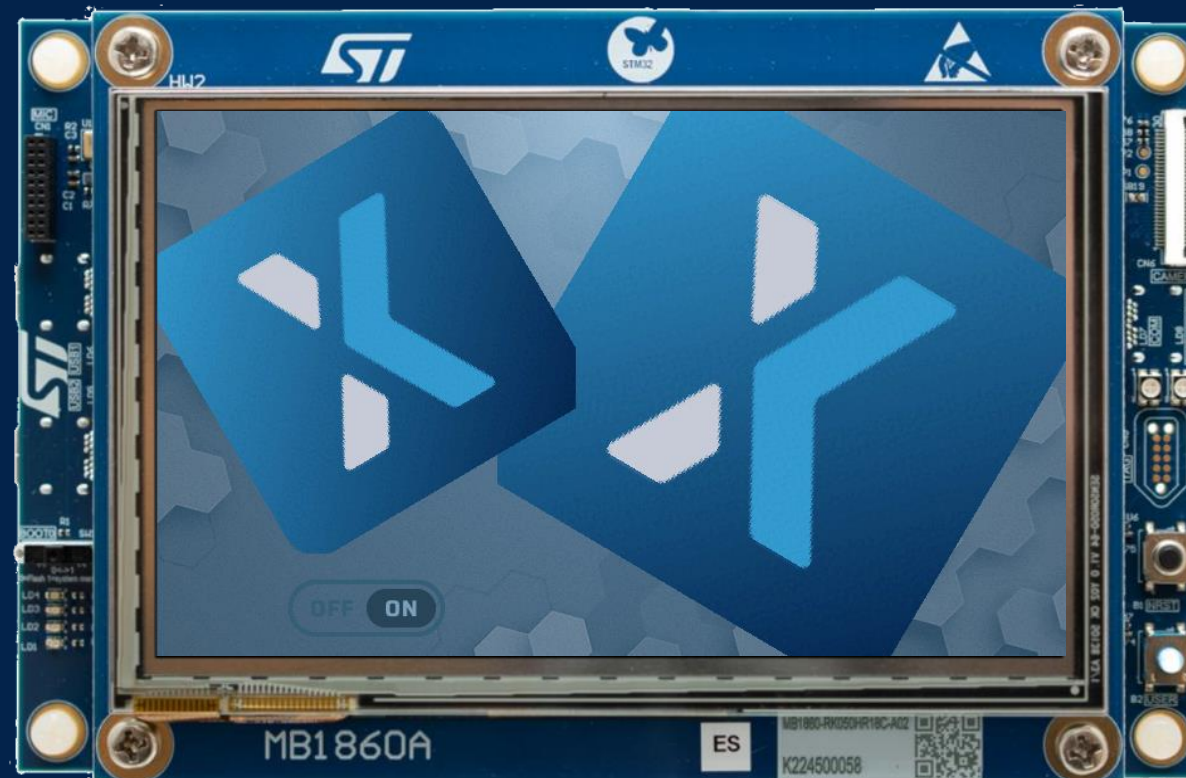


Available now !

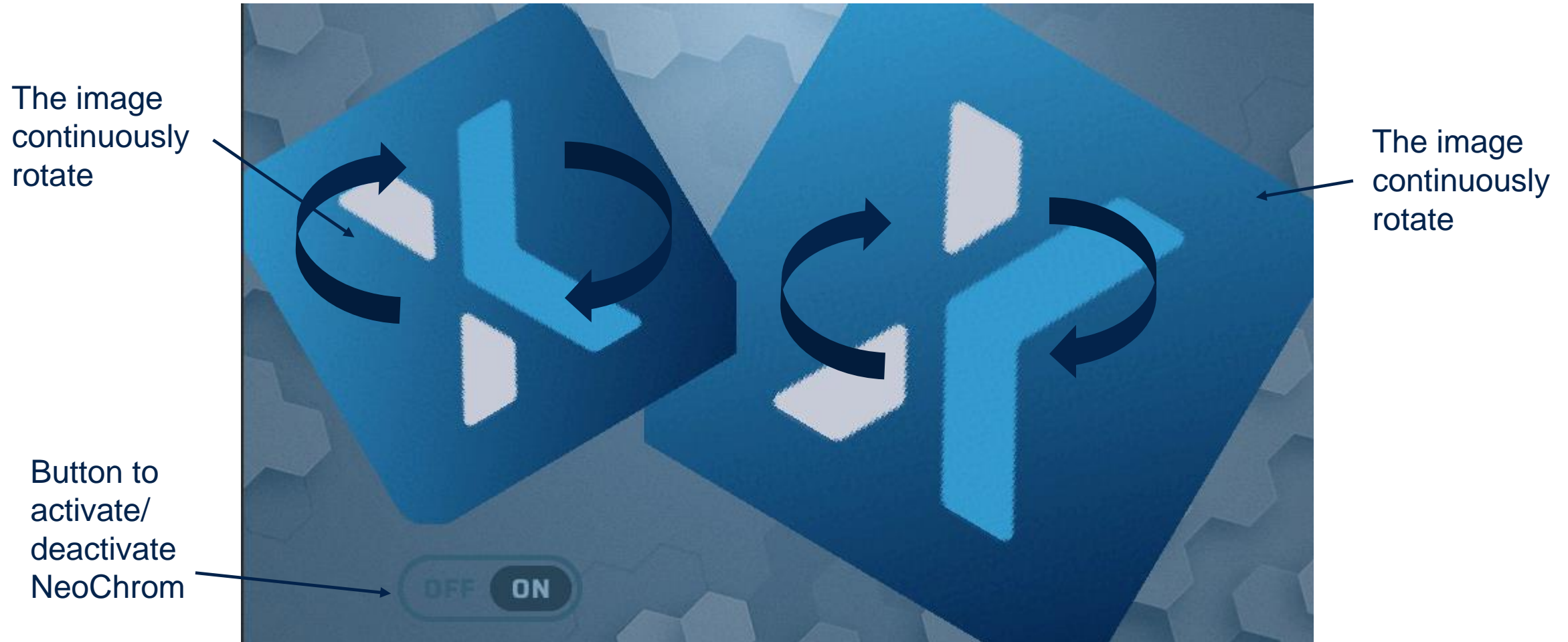
Many code example and demo supported STM32H7S78-DK

 <p>TouchGFX Demo 2 Name TouchGFXDemo2 Resolution 800 x 480</p>	 <p>Animation Texture Mapper... Name AnimationTextureMapperExample Resolution 480 x 272, ...</p>	 <p>Arabic Text Example Name ArabicTextExample Resolution 480 x 272, ...</p>	 <p>Dice Animation Name DiceAnimation Resolution 800 x 480</p>	 <p>HVAC IoT Demo Name HVAC-IOT-Demo Resolution 800 x 480</p>	 <p>Knight Hits Zombie Game Name KnightHitsZombie Resolution 800 x 480</p>	 <p>Ninja Vs Robot Name NinjaVsRobot Resolution 800 x 480</p>	 <p>Screen Transitions Name ScreenTransitions Resolution 800 x 480</p>
 <p>Button Example Name ButtonExample Resolution 480 x 272, ...</p>	 <p>Clock Example Name ClockExample Resolution 480 x 272, ...</p>	 <p>Custom Trigger Action Exa... Name CustomTriggerActionExample Resolution 480 x 272, ...</p>	 <p>Custom Widget Example Name CustomWidgetExample Resolution 480 x 272, ...</p>	 <p>Line and Circle Example Name LineAndCircleExample Resolution 480 x 272, ...</p>	 <p>ListLayout Example Name ListLayout Resolution 480 x 272, ...</p>	 <p>ModalWindow Example Name ModalWindowExample Resolution 480 x 272, ...</p>	 <p>Pixel Data Example Name PixelDataExample Resolution 480 x 272, ...</p>
 <p>Dynamic Graph Example Name DynamicGraphExample Resolution 480 x 272, ...</p>	 <p>Flex Button Example Name FlexButtonExample Resolution 480 x 272, ...</p>	 <p>Gauge Example Name GaugeExample Resolution 480 x 272, ...</p>	 <p>Keyboard Example Name KeyboardExample Resolution 480 x 272, ...</p>	 <p>Progress Indicator Example Name ProgressIndicatorExample Resolution 480 x 272, ...</p>	 <p>RadioButton Example Name RadioButtonExample Resolution 480 x 272, ...</p>	 <p>Scale and Zoom Example Name ScaleZoomExample Resolution 480 x 272, ...</p>	 <p>Scrollable Container Example Name ScrollableContainer Resolution 480 x 272, ...</p>

STM32H7R/S TouchGFX Demo with STM32H7S78-DK



Create a TouchGFX UI which demonstrate NeoChrom benefit



Start a project for STM32H7S78-DK

Home

1

+

Create

Examples

Demos

Docs

Q&A

File

Help

TouchGFX Designer - 4.24.0

2

Q

H7S


X

Show All

By STMicroelectronics

By Partners

3



STM32H7S78 DK

Name


STM32H7S78-DK

Operating System

FreeRTOS

Resolution

800 x 480



Application Name

STM32H7RS_WS_test

Application Directory

C:\TouchGFXProjects

Color Depth

16 bit

Versions

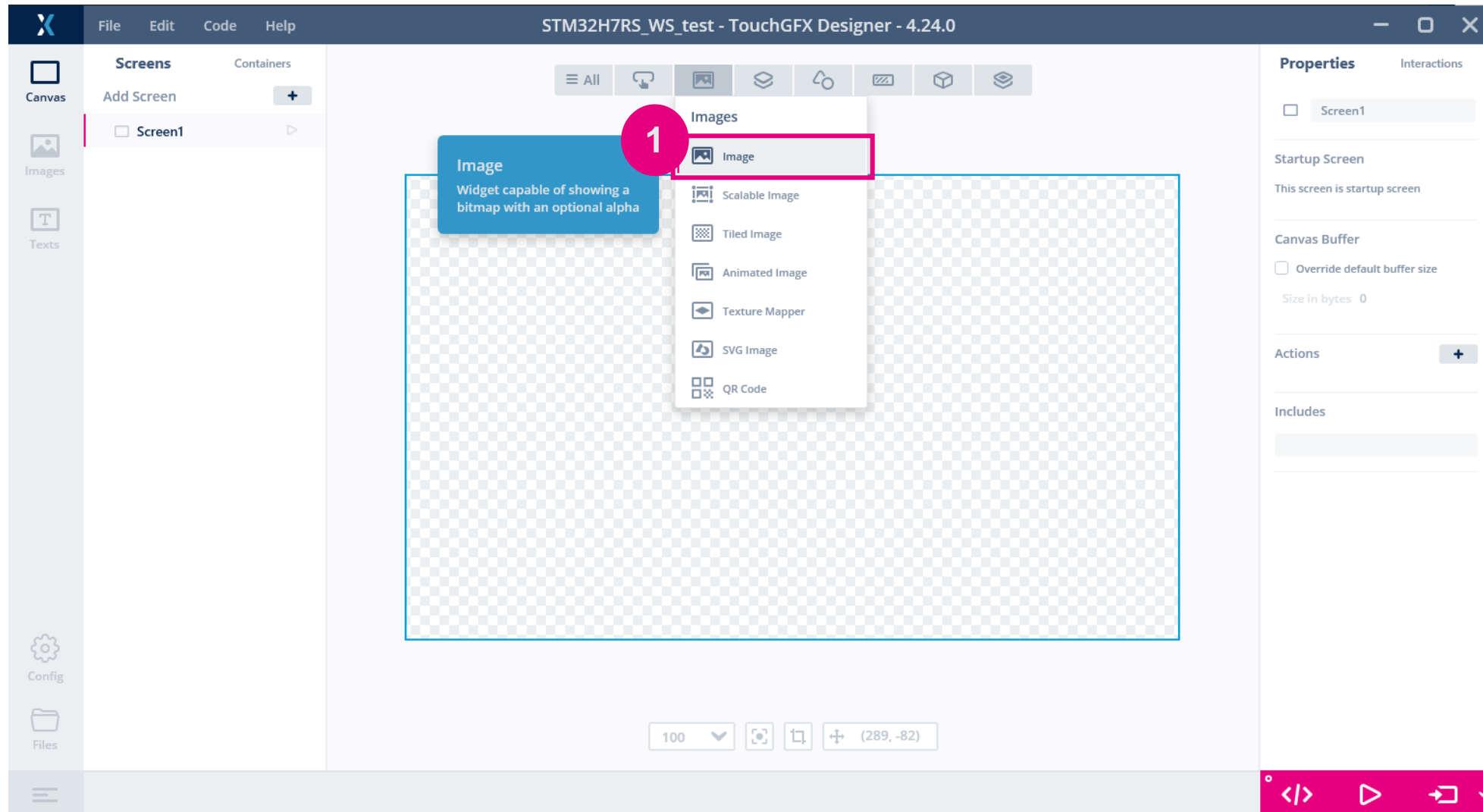
v 3.0.1

Resolution

4

Create

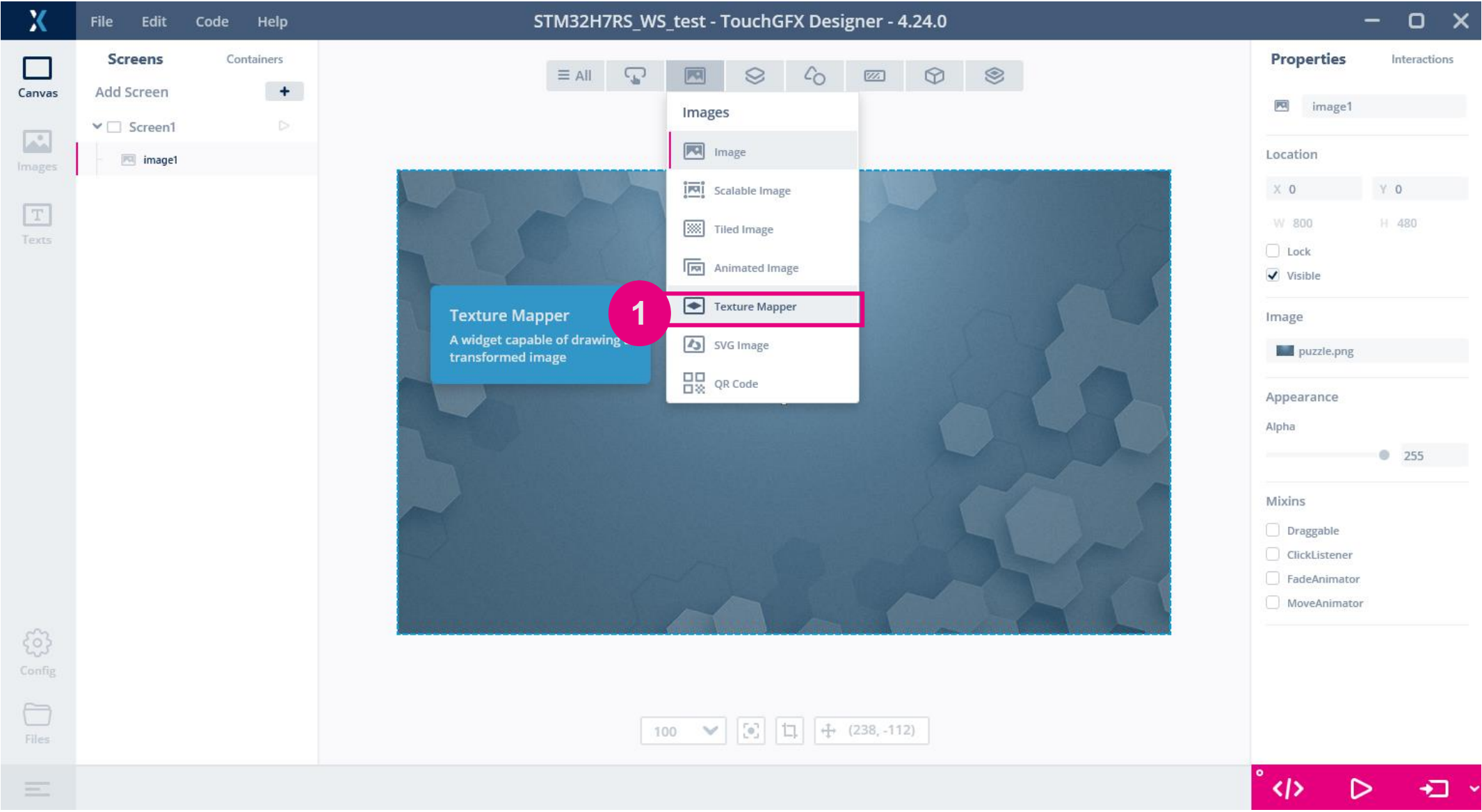
Insert an image



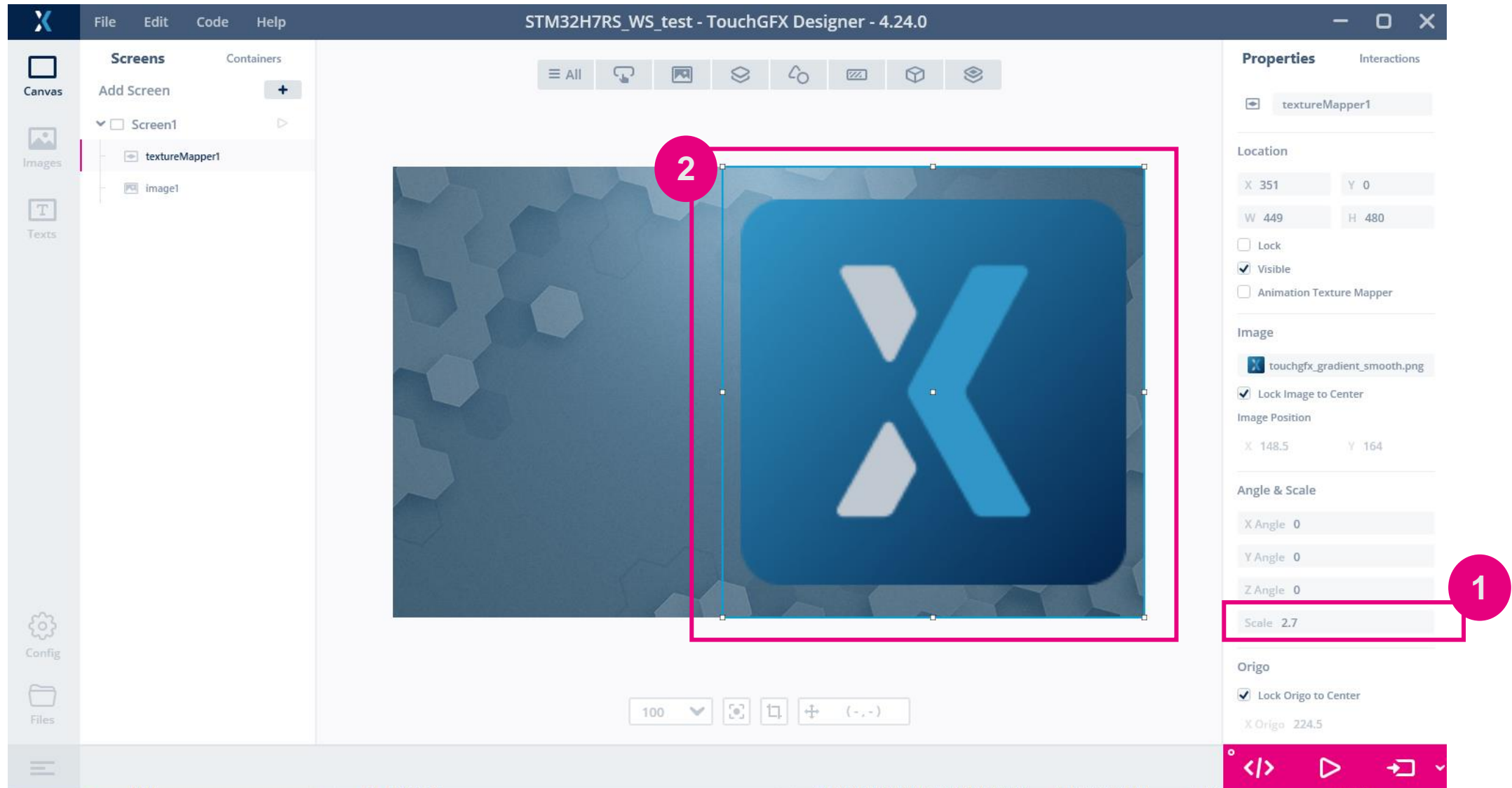
Chose the background image



Insert a texture mapper

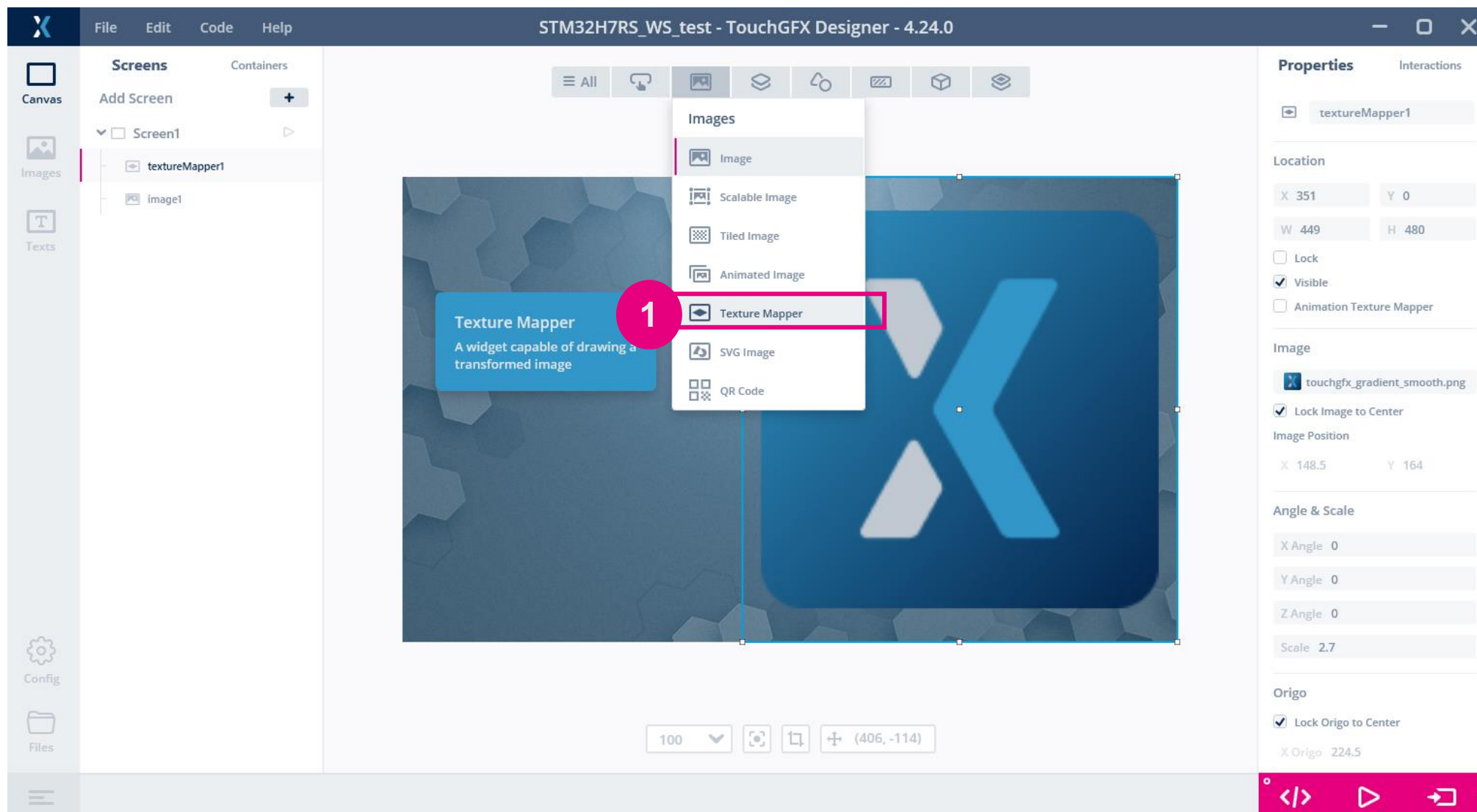


Increase the size and update location

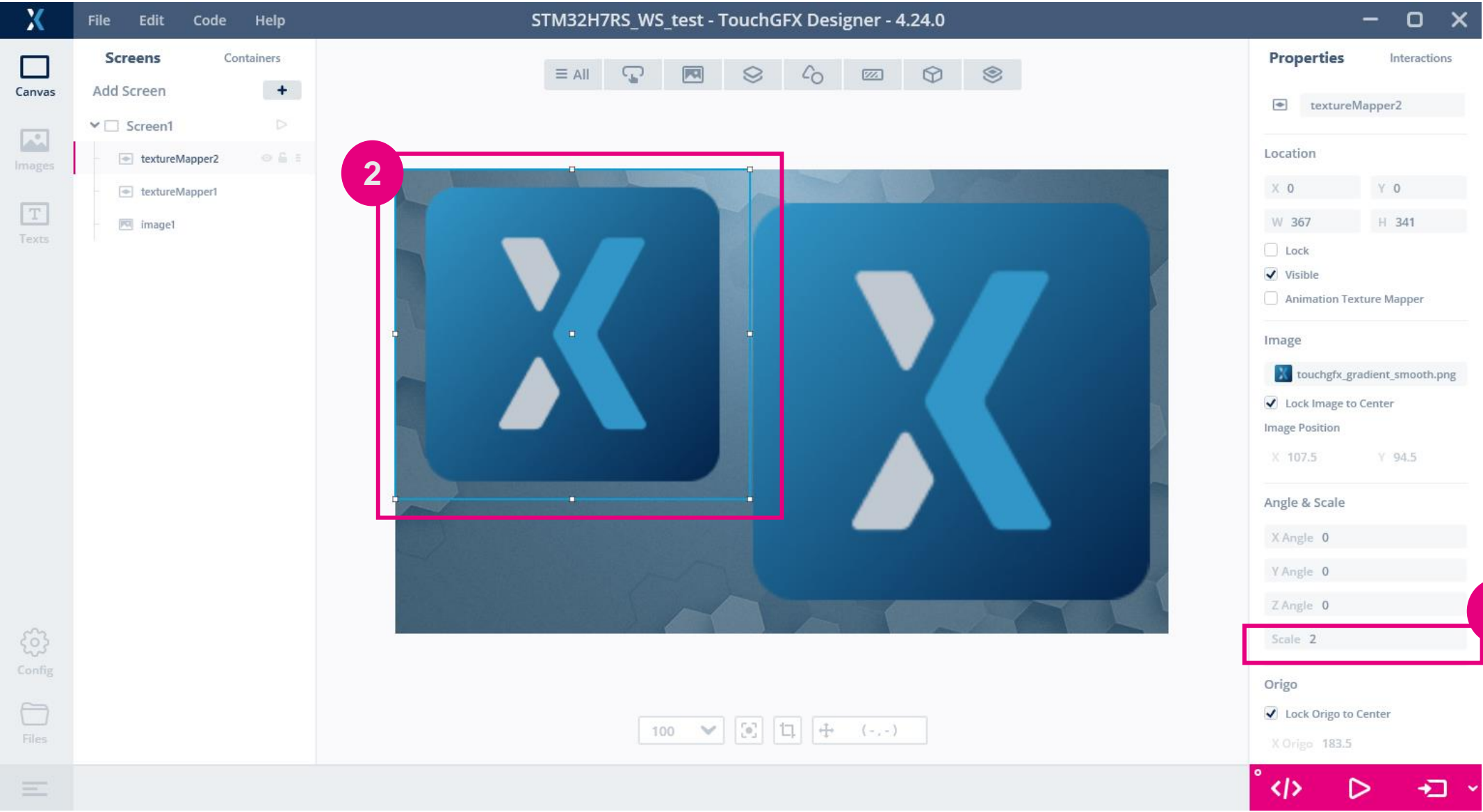


The screenshot displays the TouchGFX Designer interface for a project named "STM32H7RS_WS_test". The main canvas shows a screen with a blue background and a large blue "X" logo. The Properties panel on the right shows the settings for the selected object, "textureMapper1". The Location section shows X: 351, Y: 0, W: 449, and H: 480. The Image section shows the image "touchgfx_gradient_smooth.png" with "Lock Image to Center" checked. The Angle & Scale section shows X Angle: 0, Y Angle: 0, Z Angle: 0, and Scale: 2.7. The Origo section shows "Lock Origo to Center" checked and X Origo: 224.5. A red box highlights the Scale property, and a red circle highlights the Scale value 2.7. Another red circle highlights the top-left corner of the image on the canvas.

Insert a texture mapper

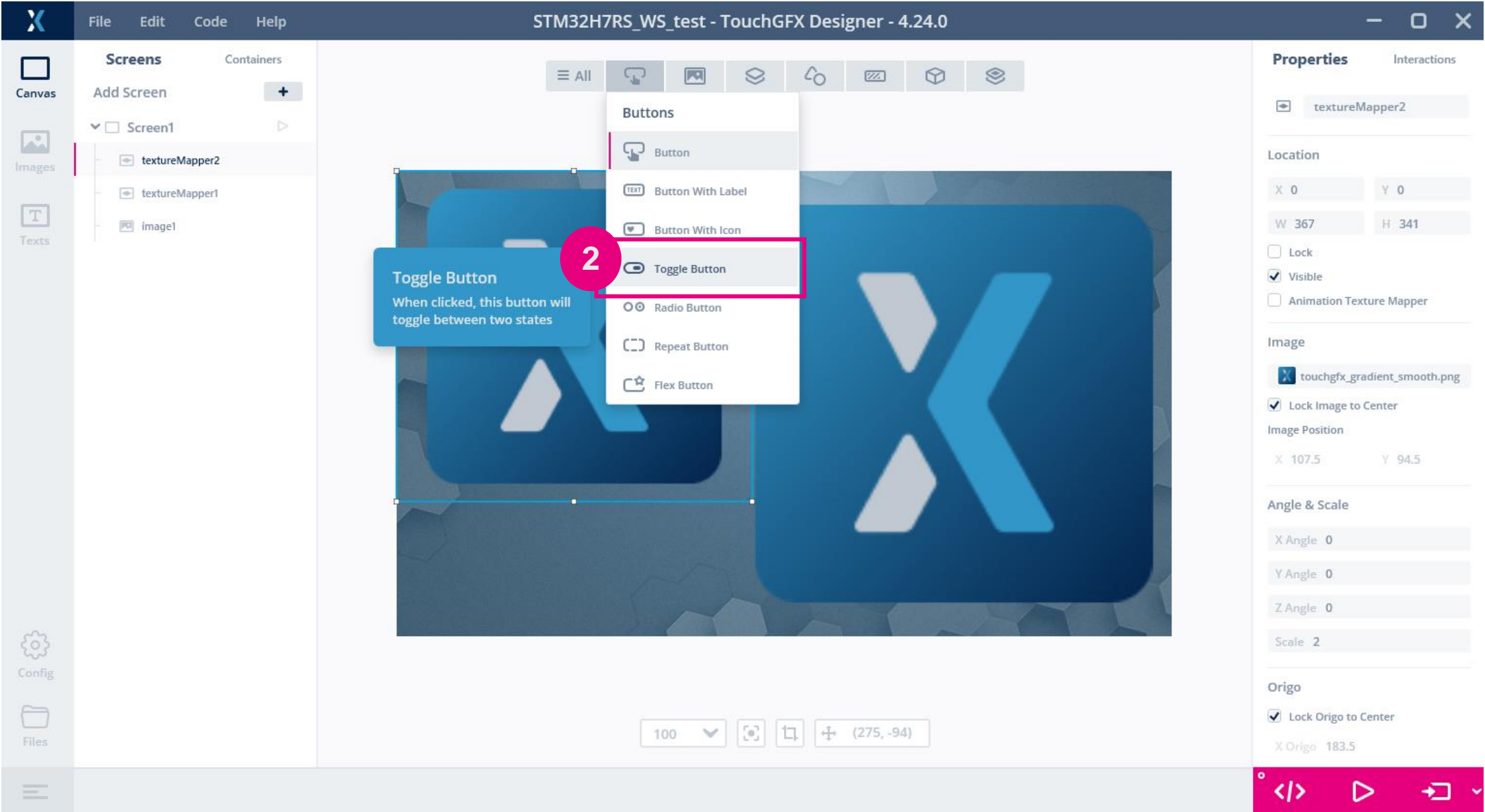


Increase the size and update location

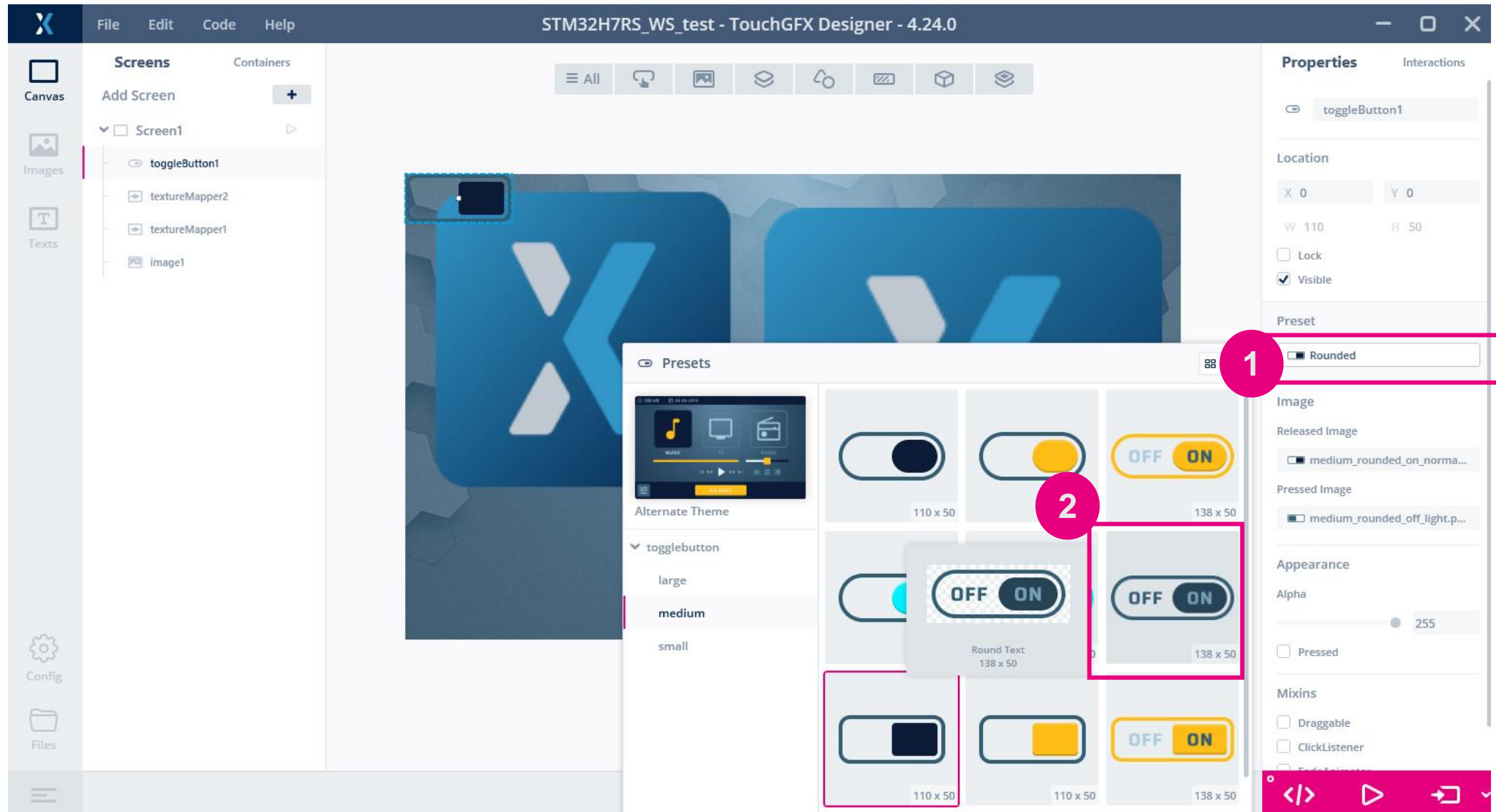


The screenshot shows the TouchGFX Designer 4.24.0 interface. The main canvas displays a screen with a blue background and two large blue 'X' logos. A pink box labeled '2' highlights the left logo, indicating it is the target for size and location adjustments. The Properties panel on the right shows the selected object, 'textureMapper2'. The 'Location' section shows X: 0, Y: 0, W: 367, and H: 341. The 'Image' section shows the image 'touchgfx_gradient_smooth.png' with 'Lock Image to Center' checked. The 'Angle & Scale' section shows X Angle: 0, Y Angle: 0, Z Angle: 0, and Scale: 2. A pink box labeled '1' highlights the 'Scale' property, indicating it is the target for size adjustment. The 'Origo' section shows 'Lock Origo to Center' checked and X Origo: 183.5. The bottom status bar shows a zoom level of 100 and various tool icons.

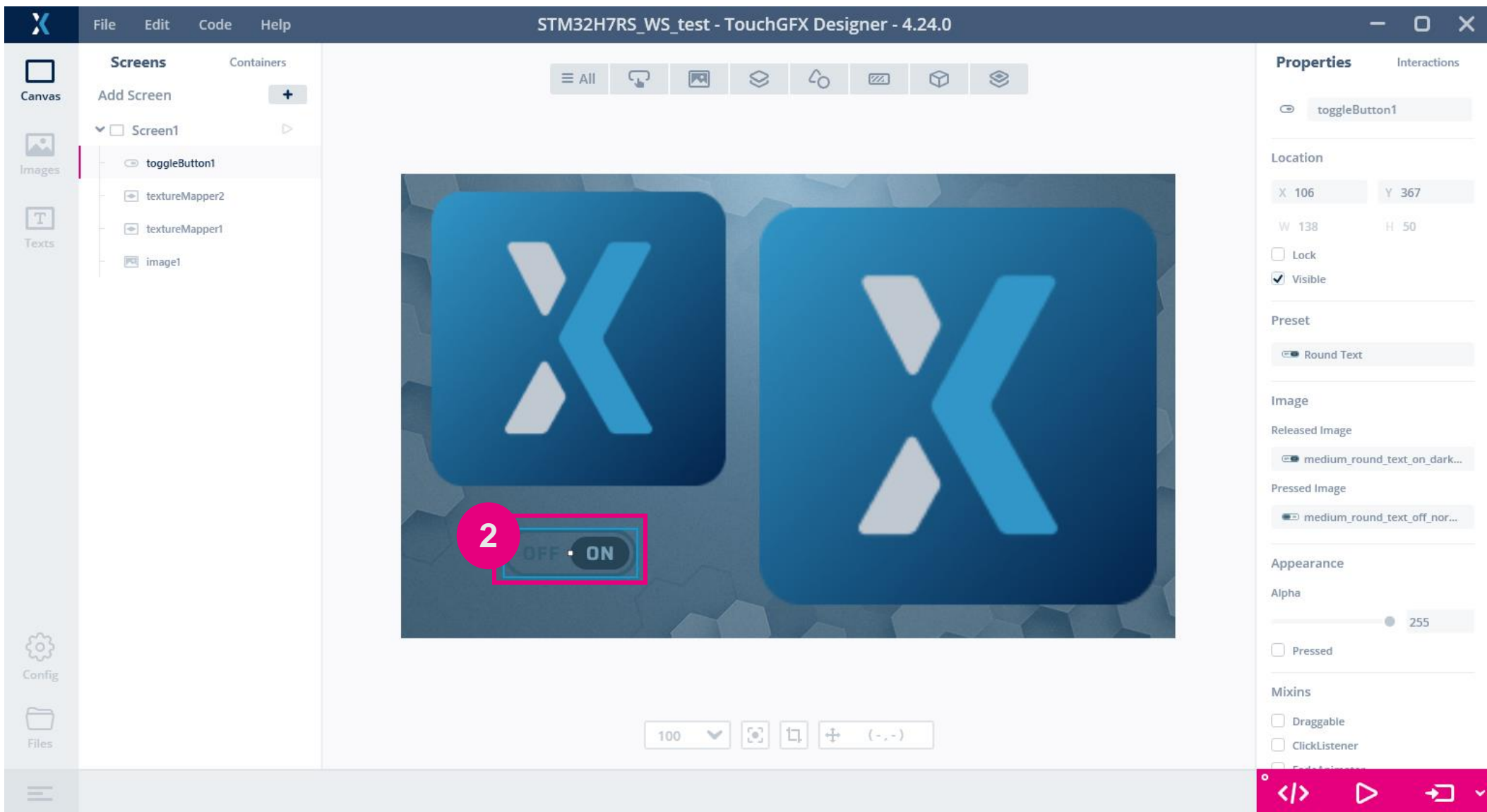
Add a toggle button



Customize the button



Update the button location



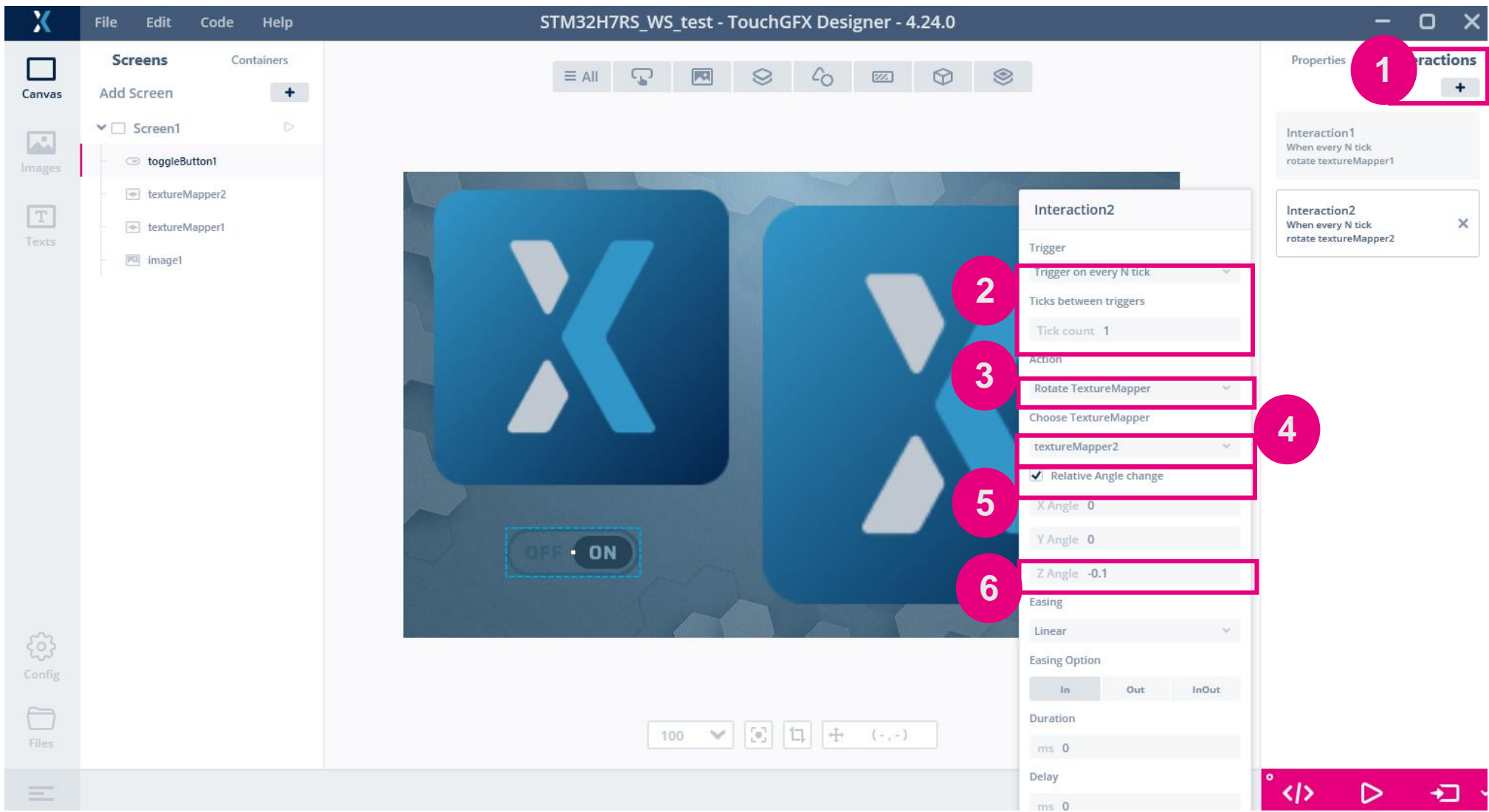
Rotate the texture mapper 1

The screenshot displays the TouchGFX Designer interface for a project named 'STM32H7RS_WS_test'. The central canvas shows a blue 'X' logo on a textured background. A 'toggleButton1' is visible at the bottom left. The left sidebar contains 'Screens' (Screen1) and 'Containers' (toggleButton1, textureMapper2, textureMapper1, image1). The right sidebar shows the 'Properties' and 'Actions' panels. The 'Actions' panel for 'Interaction1' is expanded, showing the following configuration:

- 1. **Actions** panel (indicated by a pink circle with '1')
- 2. **Trigger** dropdown set to 'Trigger on every N tick' (indicated by a pink circle with '2')
- 3. **Action** dropdown set to 'Rotate TextureMapper' (indicated by a pink circle with '3')
- 4. **Choose TextureMapper** dropdown set to 'textureMapper1' (indicated by a pink circle with '4')
- 5. **Relative Angle change** checkbox checked (indicated by a pink circle with '5')
- 6. **Z Angle** set to '0.1' (indicated by a pink circle with '6')

Other visible settings include 'Ticks between triggers' (1), 'Easing' (Linear), 'Duration' (0 ms), and 'Delay' (0 ms). The bottom status bar shows a zoom level of 100% and various tool icons.

Rotate the texture mapper 2



The screenshot displays the TouchGFX Designer interface for a project named "STM32H7RS_WS_test". The left sidebar shows the "Screens" panel with "Screen1" selected, containing elements like "toggleButton1", "textureMapper2", "textureMapper1", and "image1". The central canvas shows a UI design with two large blue squares containing a stylized 'X' logo and a toggle switch labeled "OFF" and "ON". The right sidebar shows the "Interactions" panel with two interactions listed: "Interaction1" and "Interaction2". A pink box highlights the "Interactions" panel header with a pink circle containing the number 1. A context menu for "Interaction2" is open, showing the following configuration: "Trigger" set to "Trigger on every N tick", "Ticks between triggers" set to "Tick count 1", "Action" set to "Rotate TextureMapper", "Choose TextureMapper" set to "textureMapper2", and "Relative Angle change" checked. The "X Angle" is 0, "Y Angle" is 0, and "Z Angle" is -0.1. The "Easing" is set to "Linear" and the "Easing Option" is "In". The "Duration" is 0 ms and the "Delay" is 0 ms. Pink circles with numbers 2 through 6 highlight the "Trigger", "Ticks between triggers", "Action", "Choose TextureMapper", "Relative Angle change", and "Z Angle" fields respectively. A pink box highlights the "Choose TextureMapper" dropdown with a pink circle containing the number 4. The bottom status bar shows a zoom level of 100% and various tool icons.

Copy the code to activate / deactivate NeoChrom

From STM32H7RSWorkShop-Benefit_of_NeoChrom_Demo git

https://github.com/ST-TOMAS-Examples-Gfx/stm32h7rs_touchgfx_neochrom_benefit?tab=readme-ov-file#23-add-the-activation--deactivation-of-neochrom

```
#ifndef SIMULATOR
static uint8_t b_NeoChromEnabled = 1;
    if (b_NeoChromEnabled)
    {
        b_NeoChromEnabled = 0;
        // function natively in TouchGFX for STM32H7RS
        ((TouchGFXHAL*)touchgfx::HAL::getInstance())->activateNeoChrom(false);
    }
    else
    {
        b_NeoChromEnabled = 1;
        // function natively in TouchGFX for STM32H7RS
        ((TouchGFXHAL*)touchgfx::HAL::getInstance())->activateNeoChrom(true);
    }
#endif /*SIMULATOR*/
```


Paste the code to activate / deactivate NeoChrom

The screenshot shows the TouchGFX Designer 4.24.0 interface. The main canvas displays a background image with a red box (6) highlighting the 'Code' editor. The 'Code' editor contains the following C++ code:

```
#ifndef SIMULATOR
static uint8_t b_NeoChromEnabled = 1;
if (b_NeoChromEnabled)
{
    b_NeoChromEnabled = 0;
    // function natively in TouchGFX for STM32H7RS
    ((TouchGFXHAL*) touchgfx::HAL::getInstance())->activateNeoChrom(false);
}
else
{
    b_NeoChromEnabled = 1;
    // function natively in TouchGFX for STM32H7RS
    ((TouchGFXHAL*) touchgfx::HAL::getInstance())->activateNeoChrom(true);
}
#endif /*SIMULATOR*/
```

On the right side, the 'Interactions' panel (1) shows a list of interactions. Interaction3 (2) is selected, and its configuration is shown in the 'Interaction3' dialog box (3). The dialog box has the following settings:

- Trigger: Button is clicked
- Choose clicked source: toggleButton1
- Action: Execute C++ code
- Code (5):

```
#ifndef SIMULATOR
static uint8_t b_NeoChromEnabled = 1;
if (b_NeoChromEnabled)
{
    b_NeoChromEnabled = 0;
    // function natively in TouchGFX for STM32H7RS
    ((TouchGFXHAL*) touchgfx::HAL::getInstance())->activateNeoChrom(false);
}
else
{
    b_NeoChromEnabled = 1;
    // function natively in TouchGFX for STM32H7RS
    ((TouchGFXHAL*) touchgfx::HAL::getInstance())->activateNeoChrom(true);
}
#endif /*SIMULATOR*/
```

The 'Includes' section is empty. The 'Can trigger another interaction' checkbox is unchecked. The 'Interaction Name' is 'Interaction3'. The bottom status bar shows the code editor icon (4) and the play button icon (5).

Copy the include to activate / deactivate NeoChrom

From STM32H7RSWorkShop-Benefit_of_NeoChrom_Demo git

https://github.com/ST-TOMAS-Examples-Gfx/stm32h7rs_touchgfx_neochrom_benefit?tab=readme-ov-file#23-add-the-activation--deactivation-of-neochrom

```
#ifndef SIMULATOR  
#include <TouchGFXHAL.hpp>  
#endif/*SIMULATOR*/
```

Paste the include to activate / deactivate NeoChrom

The screenshot shows the TouchGFX Designer 4.24.0 interface. The main canvas displays a UI design with a blue header and two buttons. The left sidebar shows the 'Screens' panel with 'Screen1' selected, containing 'toggleButton1', 'textureMapper2', 'textureMapper1', and 'image1'. The right sidebar shows the 'Interactions' panel with 'Interaction3' selected. The 'Interaction3' configuration window is open, showing the trigger 'Button is clicked', the source 'toggleButton1', and the action 'Execute C++ code'. The C++ code block is highlighted with a red box and labeled '2'. The 'Includes' section of the code block is highlighted with a red box and labeled '3'. The 'Interaction3' configuration window is also highlighted with a red box and labeled '1'.

3 Includes

```
#ifndef SIMULATOR
#include <TouchGFXHAL.hpp>
#endif /*SIMULATOR*/
```

1 Interaction3

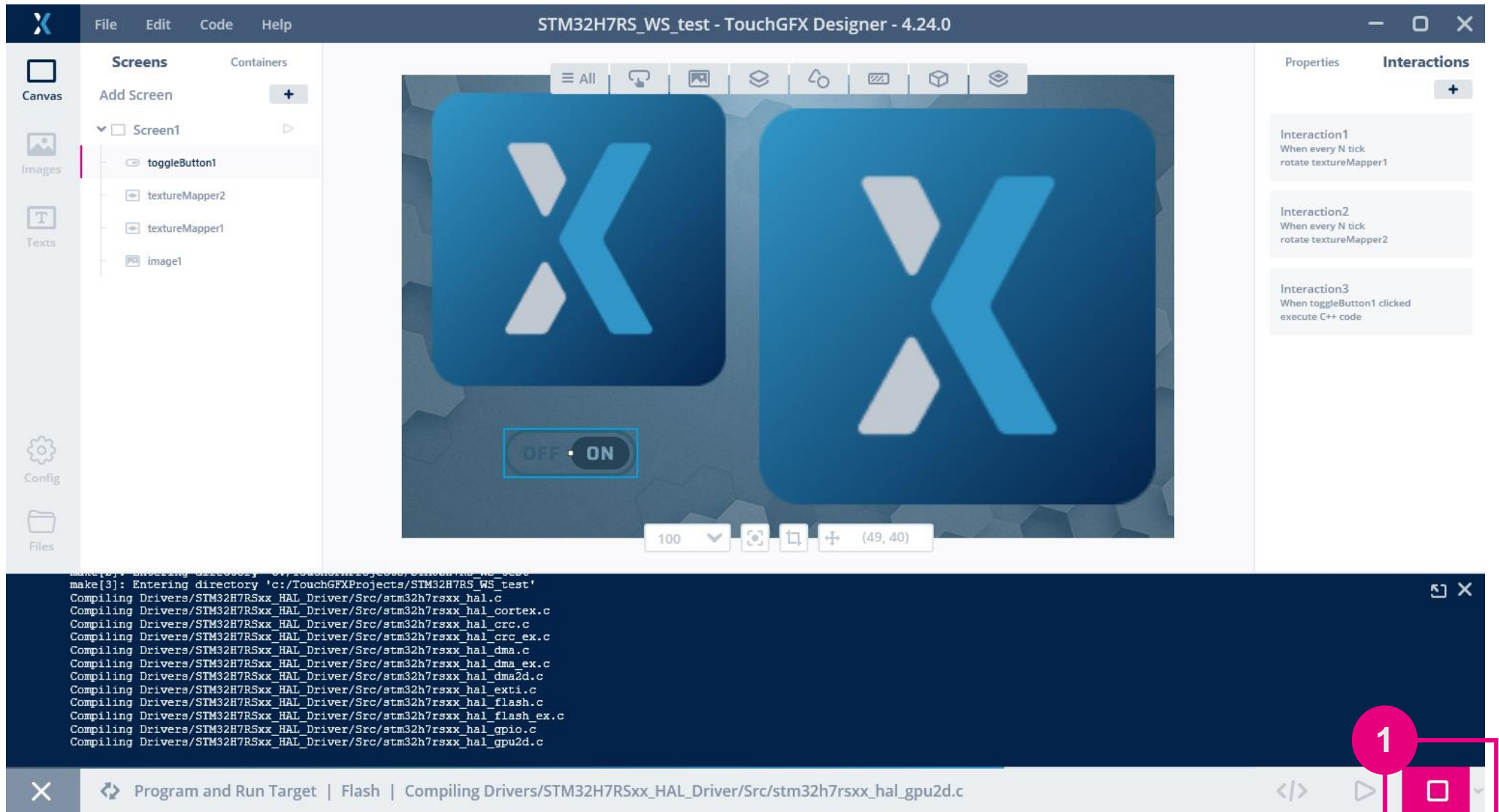
When toggleButton1 clicked
execute C++ code

2

```
#ifndef SIMULATOR
static uint8_t b_NeoChromEnabled;
if (b_NeoChromEnabled)
{
    b_NeoChromEnabled = 0;
    // function native
    ((TouchGFXHAL*) t_...
}
else
{

```

Generate and download to the target

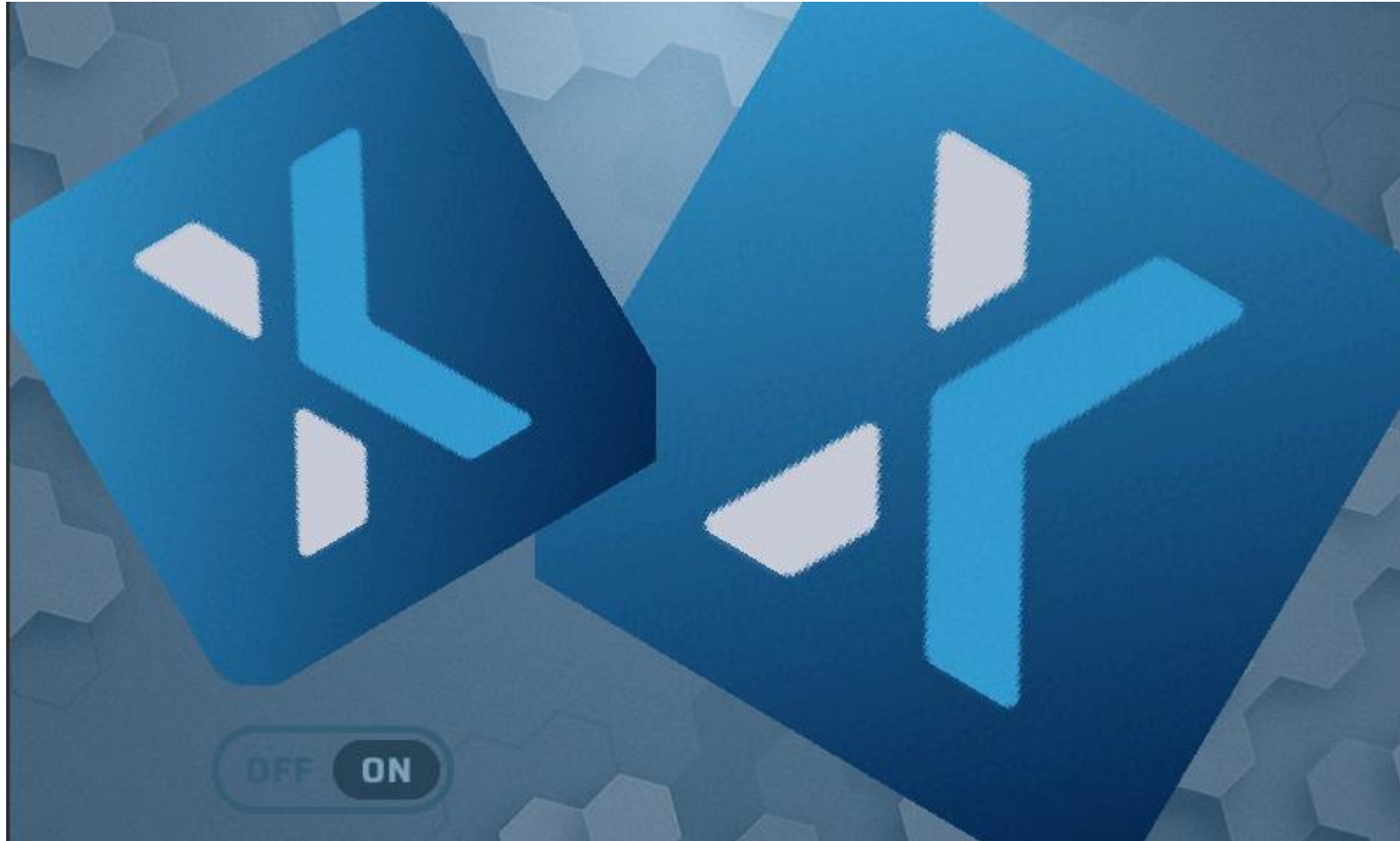


The screenshot displays the TouchGFX Designer 4.24.0 interface for the STM32H7RS_WS_test project. The main workspace shows a canvas with two large blue 'X' logos and a toggle button labeled 'OFF' and 'ON'. The left sidebar contains a 'Screens' panel with a tree view showing 'Screen1' containing 'toggleButton1', 'textureMapper2', 'textureMapper1', and 'image1'. The right sidebar shows the 'Interactions' panel with three defined interactions: 'Interaction1' (When every N tick, rotate textureMapper1), 'Interaction2' (When every N tick, rotate textureMapper2), and 'Interaction3' (When toggleButton1 clicked, execute C++ code). The bottom status bar indicates the current action is 'Program and Run Target'.

make[3]: Entering directory 'c:/TouchGFXProjects/STM32H7RS_WS_test'
make[3]: Entering directory 'c:/TouchGFXProjects/STM32H7RS_WS_test'
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_cortex.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_crc.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_crc_ex.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_dma.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_dma_ex.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_dma2d.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_exti.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_flash.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_flash_ex.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_gpio.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_gpio_ex.c
Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_gpu2d.c

Program and Run Target | Flash | Compiling Drivers/STM32H7RSxx_HAL_Driver/Src/stm32h7rsxx_hal_gpu2d.c

Expected result



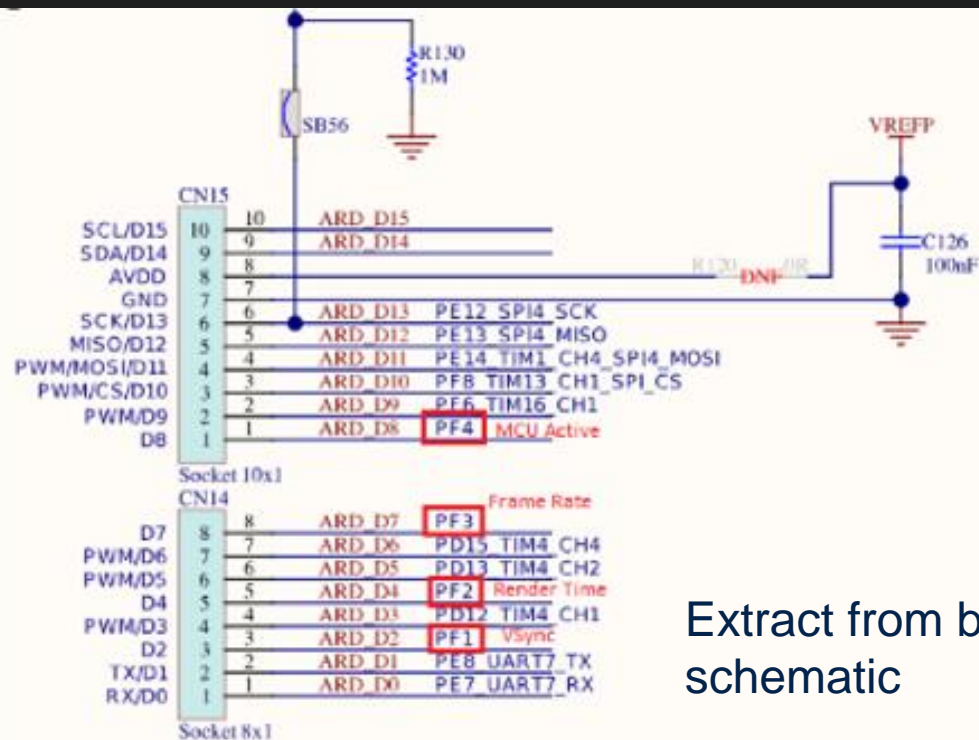
Rotation speed should slow down when NeoChrom is deactivated.

An other way to check MCU activity and platform activity ?

STM32H7S78-DK TBS

Performance testing can be done using the GPIO pins designated with the following signals in CN10 connector on the underside of the board:

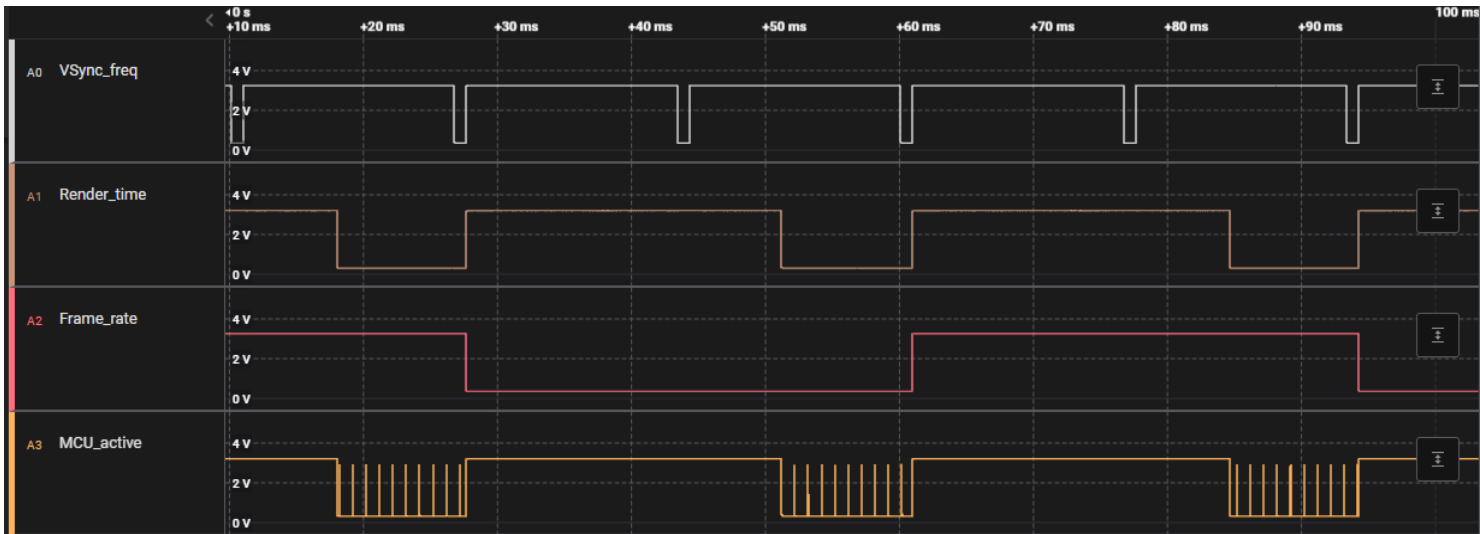
- VSYNC_FREQ - CN14-D2 (PF1)
- RENDER_TIME - CN14-D4 (PF2)
- FRAME_RATE - CN14-D7 (PF3)
- MCU_ACTIVE - CN15-D8 (PF4)



Extract from board schematic

Let measure the benefit of NeoChrom

NeoChrom
OFF



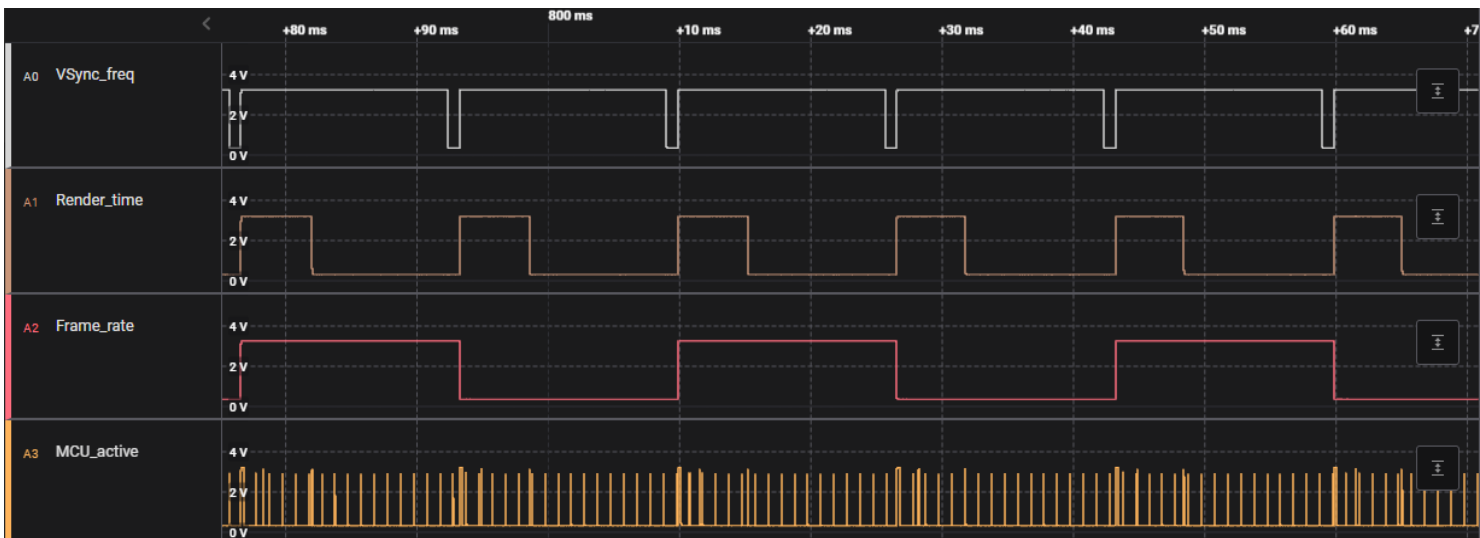
VSync_Freq

Rendering time

Frame rate

MCU active

NeoChrom
ON



VSync_Freq

Rendering time

Frame rate

MCU active

TouchGFX Take-Aways!

- STM32H7RS boot flash allows to select the external RAM / FLASH which fits your graphic UI requirement.
- The integration of NeoChrom 2.5D graphic accelerator allows to have an advanced GUI application without loading the Cortex M7 at 600 MHz
- STM32H7S78-DK is delivered with nice demonstration software which demonstrate the platform capabilities!
- On top of that, TouchGFX designer associated with STM32H7S78-DK allows to create prototype of your UI and evaluate the benefit of the overall STM32H7RS architecture !

Our technology starts with You



Find out more at www.st.com

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

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.



life.augmented