



TouchGFX

Technical Training

Agenda

1 Basic Concepts

2 STM32 Graphics

3 TouchGFX Generator

4 Partial Frame Buffer

5 TouchGFX Designer

6 TouchGFX Engine

7 Hands-On

Basic Concepts

Embedded Graphics

- MCU

- 이미지 렌더링
- 이미지를 읽고 RAM에 작성
- RAM의 이미지를 디스플레이로 전송

- RAM

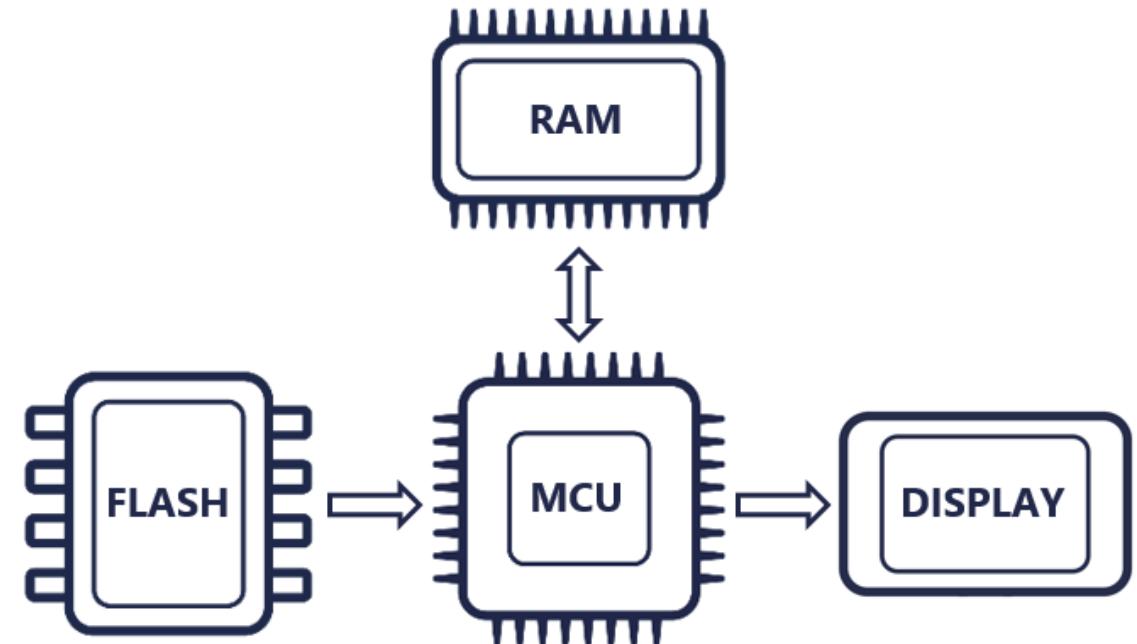
- 렌더링된 이미지가 저장되는 공간(Framebuffer)

- Flash

- 모든 정적 데이터가 저장되는 공간
(Assets - image, font, text)

- Display

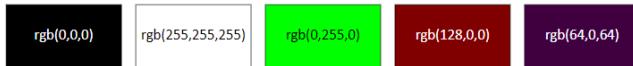
- 이미지를 실제로 보여주는 공간
- 계산된 이미지(Framebuffer)가 RAM에 저장된 후
MCU를 통해 주기적으로 디스플레이에 전송



Color Formats

- Color

- RGB(빨간색, 녹색, 파란색)로 0~255의 값으로 표현

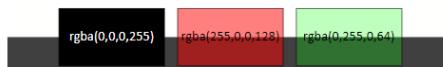


- Grayscale

- 검은색에서 흰색까지 모든 색상을 회색으로 표현

- Opacity(불투명도)

- 불투명도가 지정된 색상 : **RGBA**
- A(Alpha, 불투명도)는 RGB와 동일하게 0~255의 값으로 표현



- Color depth

- Framebuffer에 저장된 색상들을 하나씩 나타낼 때 사용되는 비트 수(bit per pixel, bpp)
- $24\text{bpp} = 2^{24}$, 즉 16777216개의 색 표현 가능

Color Formats

- **Formats**

- Pixel color format
 - RGB888 : 24bit color depth, R/G/B 각 요소에 8bit 사용
 - RGB565 : 16bit color depth, R/B에 5bit 그리고 G에 6bit 사용
- Image format
 - ARGB8888 : 32bit, 각 요소에 8bit
 - L8_ARGB8888 : 8bit index 형식, ARGB8888 Palette
 - GRAY4 : 4bit grayscale
 - BW : 1bit grayscale
- Framebuffer format
 - 일부 image format은 framebuffer format으로 사용 불가.
(L8 format의 경우 두 이미지 간 blending을 할 수 없기 때문에 사용 불가)
- Text format
 - Glyph는 특정 색상 형식으로 메모리에 저장됨
 - A8 : 8bit, 불투명도 전용
 - A4 : 4bit, 불투명도 전용

Color formats

- Dithering

- Image를 여러가지 색상으로 표현할 때 시각적 품질을 높여주는 기법
- Image의 색상에 약간의 노이즈를 추가하여 Image의 색상을 실제 보다 더 많이 보이게 하는 방법



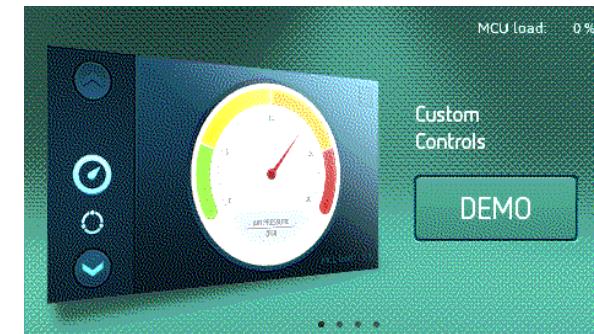
원본 RGB888 image



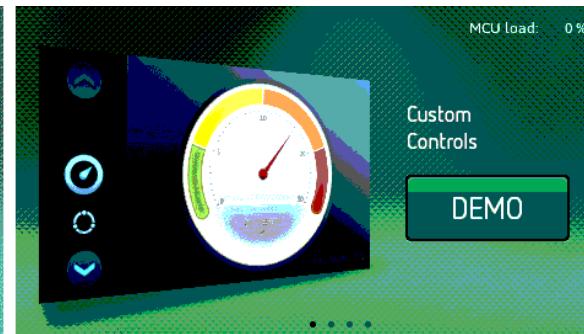
Dithering 적용 된 RGB565 image



Dithering 적용 하지 않은 RGB565 image



Dithering 적용 된 xRGB2222 image



Dithering 적용 하지 않은 xRGB2222 image

Framebuffer

- 그래픽 엔진에서 디스플레이에 표시하고 다음 이미지를 준비할 목적으로 업데이트되는 메모리 공간

- Framebuffer 위치

- 내부 RAM
- 외부 RAM
- 디스플레이 내장 RAM

- 메모리 사용

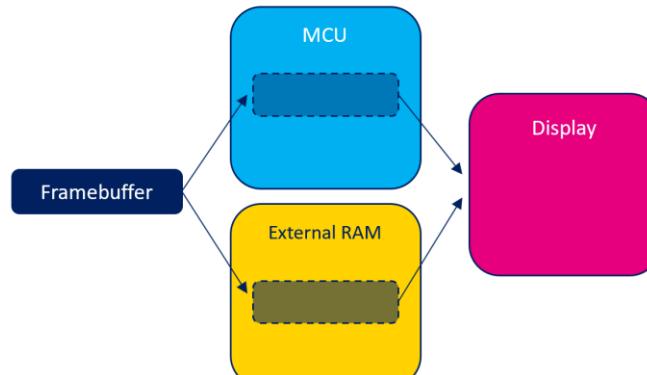
- 가로 X 세로 X Color depth

해상도(픽셀)	색상(bpp)	계산	메모리 사용량(바이트)
800 x 480	16bpp	800 480 16 / 8	768.000B
480 x 272	24bpp	480 272 24 / 8	391.680B
100 x 100	8bpp	100 100 8 / 8	10.000B



2D framebuffer 대/모리

24bpp framebuffer 내용과 그에 따른 디스플레이 결과



가능한 framebuffer 위치

Graphics Engine

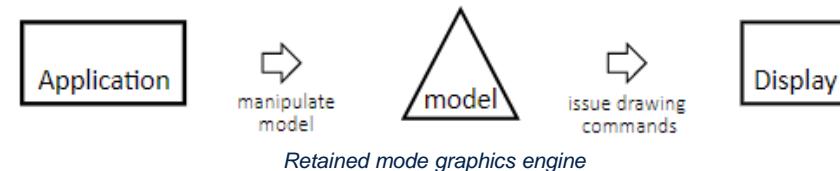
- Scene model

- Immediate mode graphics engine

: 애플리케이션이 디스플레이에 그라픽을 바로 그릴 수 있는 API를 제공. 따라서 그리기 연산을 적시에 호출하는 것은 애플리케이션에 달려 있음

- Retained mode graphics engine

: 사용자가 표시되는 그래픽 구성요소의 추상적 모델을 조작. Engine은 이 그래픽 구성요소 모델을 정확한 그래픽 그리기 연산으로 적시에 변환 (e.g. TouchGFX)



- Main loop

- 이벤트 수집

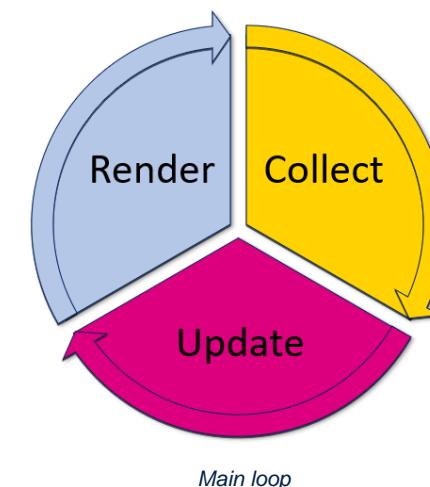
- 터치 스크린, 물리적 버튼 누름, Back-end 시스템의 메시지 등으로 부터 이벤트를 수집

- Scene model 업데이트

- 수집된 이벤트에 대해 model의 위치와 애니메이션, 색상, 이미지 등을 업데이트 하여 반응

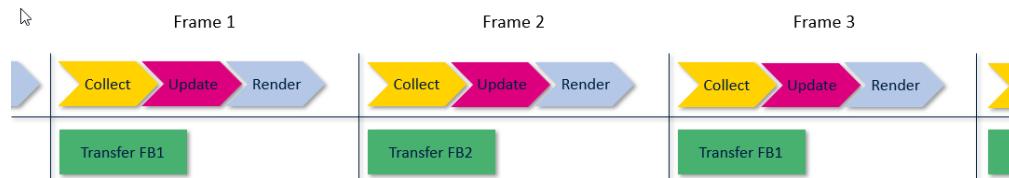
- Scene model 렌더링

- Model에서 업데이트된 부분을 다시 그려 디스플레이에 표시



Handling Framebuffer

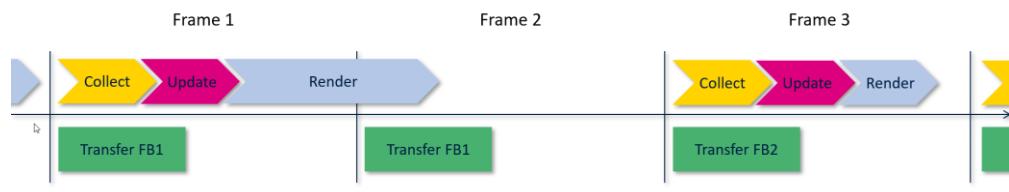
- Double framebuffer



0/중 framebuffer

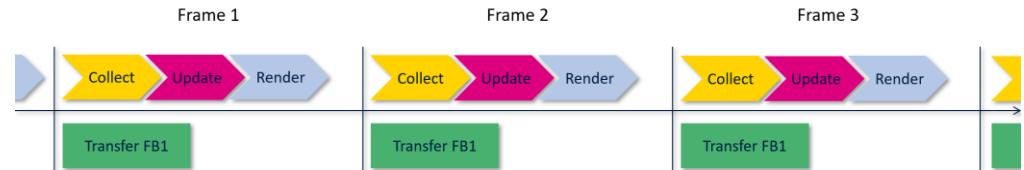


Frame2에서 업데이트가 없는 경우

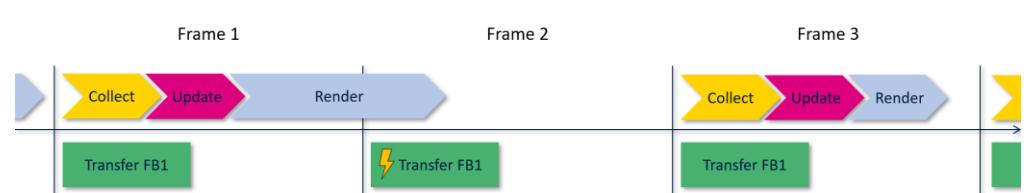


렌더링 시간이 길어진 경우

- Single framebuffer



단일 framebuffer는 모든 프레임에서 재전송



렌더링 시간이 길어진 경우

STM32 Graphics



STM32 hardware embedded graphics HW acceleration

Chrom-ART™ Accelerator

그래픽 작업 중 CPU 로드 감소 효과
낮은 메모리 사용량

뛰어난 GUI 퍼포먼스 – 부드럽고 풍부한 그래픽 효과

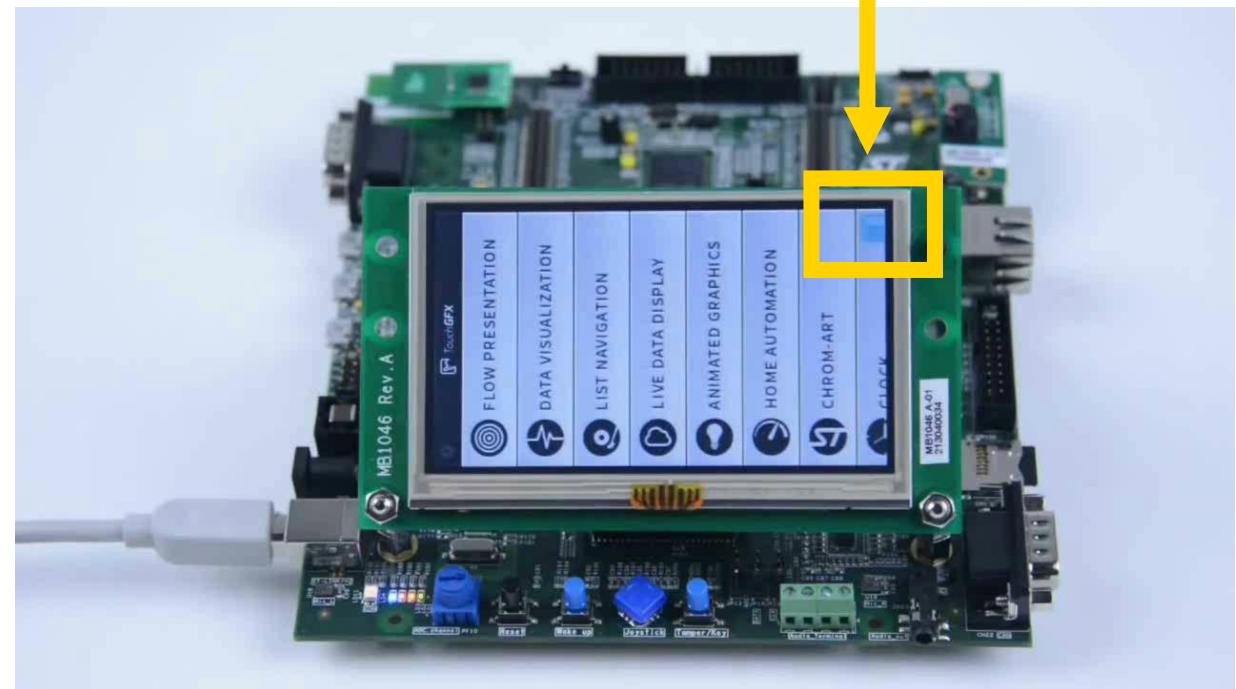
- 80% 적은 CPU 자원으로 부드러운 움직임 및 투명도 효과를 구현

→ The technology behind

- Simple Drawing
- 2D Copy
- Alpha blending
 - For transparency effects
 - Anti aliased bitmap fonts
- Color format conversion

퍼포먼스 결과에
주목!!

4% CPU load with Chrom-ART™
Up to 84% without Chrom-ART™





STM32 hardware embedded graphics HW acceleration

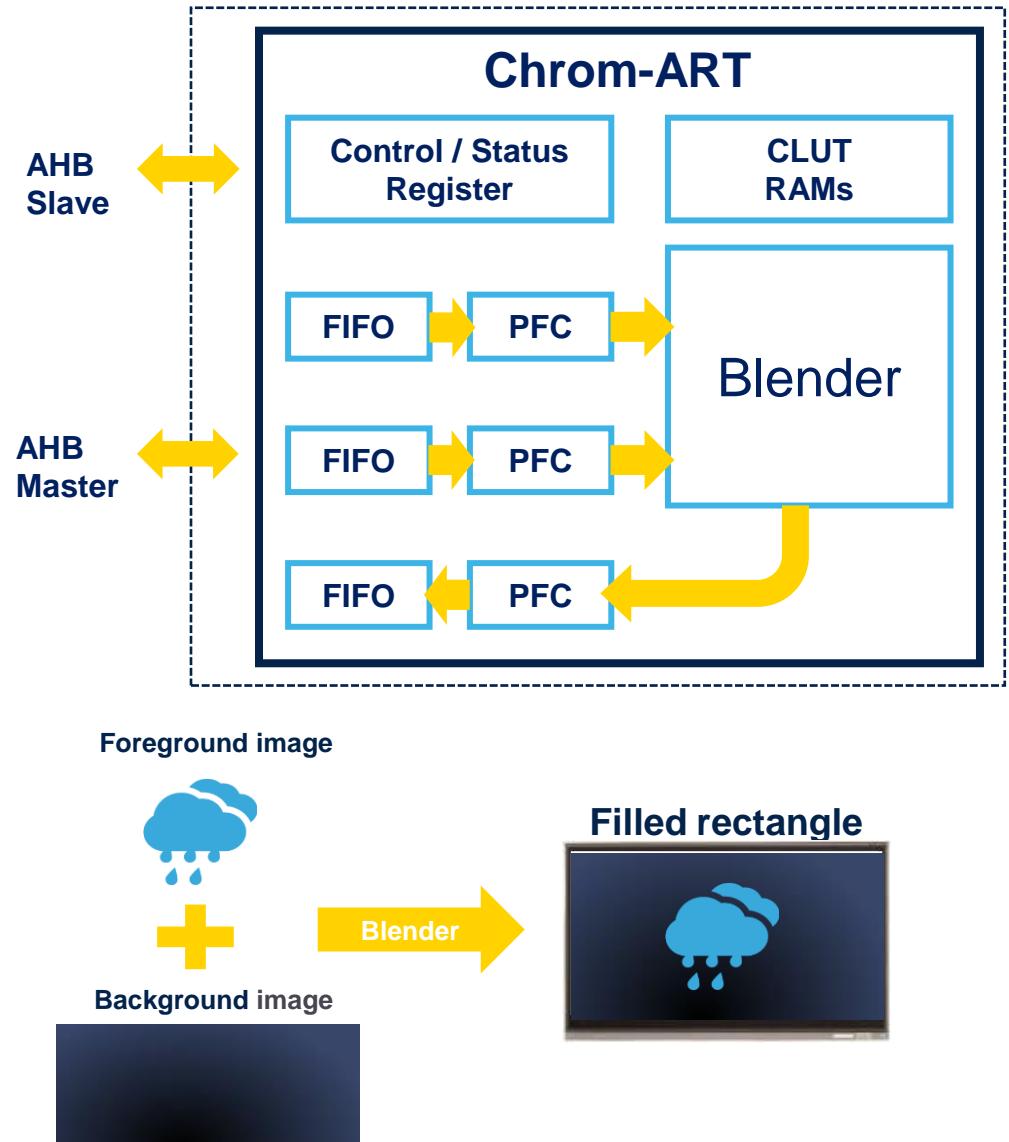
Chrom-ART™ Accelerator

그래픽 작업 중 CPU 로드 감소 효과
낮은 메모리 사용량

뛰어난 GUI 퍼포먼스 – 부드럽고 풍부한 그래픽 효과
• 80% 적은 CPU 자원으로 부드러운 움직임 및 투명도
효과를 구현

→ The technology behind

- Simple Drawing
- 2D Copy
- Alpha blending
 - For transparency effects
 - Anti aliased bitmap fonts
- Color format conversion





STM32 hardware embedded graphics HW acceleration

NeoChrom / NeoChromVG GPU

그래픽 작업 중 CPU 로드 감소 및 낮은 메모리 사용량
뛰어난 GUI 퍼포먼스 – 부드럽고 풍부한 그래픽 효과

→ The technology behind

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

NeoChromVG NEW

- 하드웨어 그래픽 가속기
- SVG 그리기 성능 대폭 향상
- SVG, Fonts, MAP navigation에 활용



NeoChrom video!
NeoChromVG video!





STM32 hardware embedded graphics HW acceleration

JPEG hardware accelerator

브랜드 및 사용자 경험을 향상시키는 영상(MJPEG)을 구현하는데 효과적

- 설명 및 가이드 영상
- 브랜딩 영상

CPU 로드 최소화

고품질 (30-60 FPS)의 motion JPEG(MJPEG) 영상 재생 가능.

→ The technology behind

- JPEG compression and decompression
- Full and easy management of JPEG headers
- ChromART supports the pixel reordering and YCbCr to RGB conversion



Example of instruction video

JPEG HW acc.	FPS	CPU load
OFF	12	>90%
ON	90	<10%

STM32H7, 240 MHZ, 640x480 resolution
JPEG Codec HW accelerator enables:
High quality video
Nine times less CPU resources required



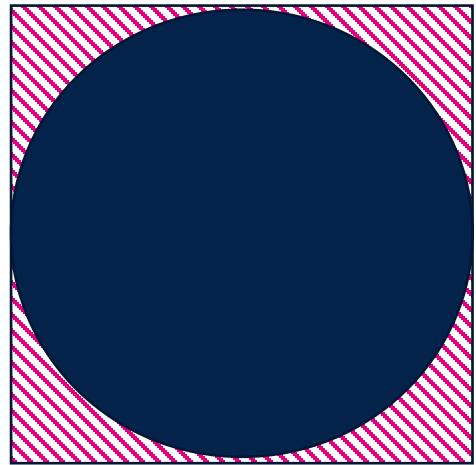
STM32 hardware embedded graphics memory optimization

Chrom-GRC™

원형 디스플레이에 대한 메모리 최적화
프레임 버퍼 공간을 최대 20% 절약

→ The technology behind

- Graphic Resource Cutter for non-square displays
- No modification nor special management at SW level
- RAM needs:
 - For 360x360 round display
 - @16bpp ~205kBytes (vs.253kBytes)
 - @24bpp ~307kBytes (vs.380kBytes)
 - For 400x400 round display
 - @16bpp: 250kBytes (vs.312kBytes)
 - @24bpp: 372kBytes (vs.469kBytes)



절약된 메모리 영역
(최대 20%)



STM32 hardware Supported displays

디스플레이 지원

다양한 임베디드 디스플레이 지원

최대 해상도 1024*768, 약 10 Inch, 디스플레이 크기 지원

디스플레이에 내장된 GRAM의 유무에 상관없이 지원

→ The technology behind

Supporting multiple display interfaces

- LCD-TFT display controller (LTDC)
- MIPI-DSI Host controller
- Parallel (FMC) Interface
- Serial interface

1" - 3"

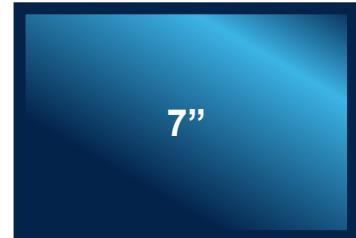
Typical: 400x400

2" – 3"

Typical: 320x240

4" – 5"

Typical: 480x272



7"

Typical: 800x480

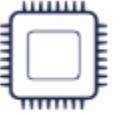
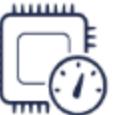


+10"

Typical: 1024x600



STM32 Portfolio For Graphics

				
STM32 SERIES	FREQUENCY	HARDWARE ACCELERATION	DISPLAY INTERFACES	SUPPORTED RESOLUTIONS
STM32G0 (CM0+)	64 MHz	-	SPI	Up to 320*240
STM32L4 (CM4)	80 MHz	Chrom-ART™	Parallel SPI	Up to 480*272
STM32L4+ (CM4)	120 MHz	Chrom-ART™ Chrom-GRC™	Parallel LCD TFT MIPI-DSI	Up to 450*450
STM32F4 (CM4)	180 MHz	Chrom-ART™	Parallel LCD TFT MIPI-DSI	Up to 800*480
STM32F7 (CM7)	216 MHz	Chrom-ART™ HW JPEG Codec	Parallel LCD TFT MIPI-DSI	Up to 1024*768
STM32H7 (CM7)	480 MHz	Chrom-ART™ HW JPEG Codec	Parallel LCD TFT MIPI-DSI	Up to 1024*768
STM32U5 (CM33)	160 MHz	Chrom-ART™ NeoChrom Chrom-GRC	Parallel LCD TFT MIPI-DSI	Up to 800*480

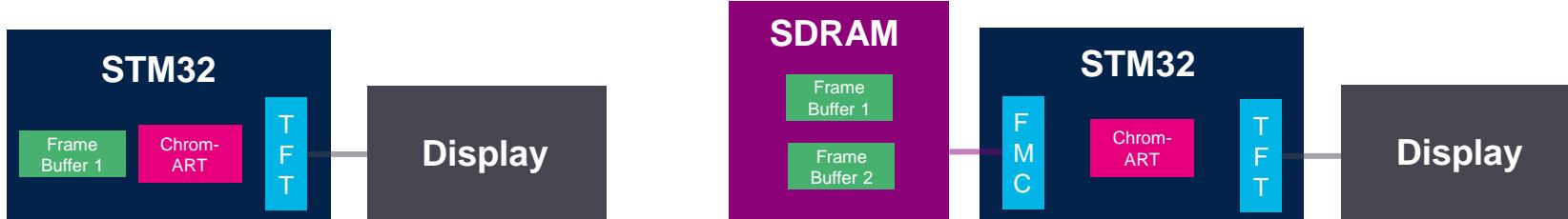


Hardware Configuration For Hardware

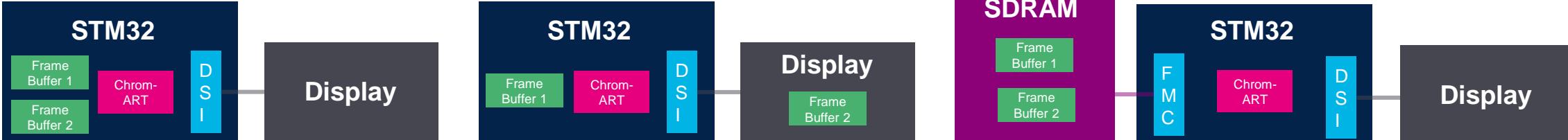
- SPI and Intel 8080 / Motorola 6800 LCD interfaces for small resolutions



- TFT controller for medium resolution (up to XGA)



- MIPI-DSI interface for medium resolution, high pixel density GUI, mainly consumer today





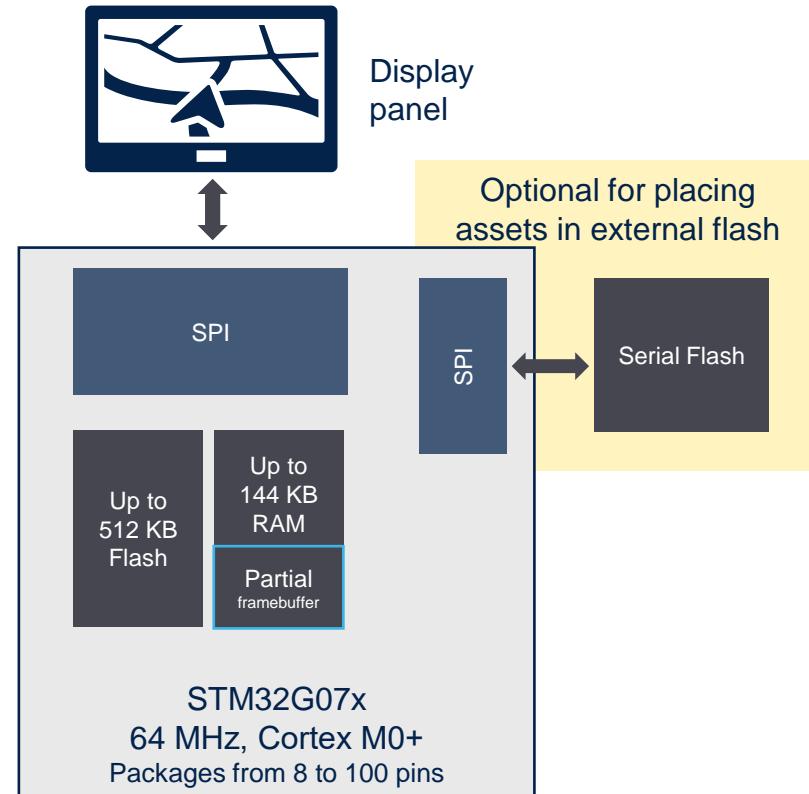
Entry-level graphics on ultra-low-cost STM32G07x

최대 해상도 480x272, 약 4.3 Inch 디스플레이



Resources

- **NUCLEO-G071RB + X-NUCLEO-GFX01M**
 - 2.2" SPI QVGA 320x240 display
- **TouchGFX Designer**
 - Board specific demos in full source
 - TouchGFX Board setup
 - Note:
TouchGFX GUI footprint starting from 30kB flash and 16kB RAM (including partial frame buffer)





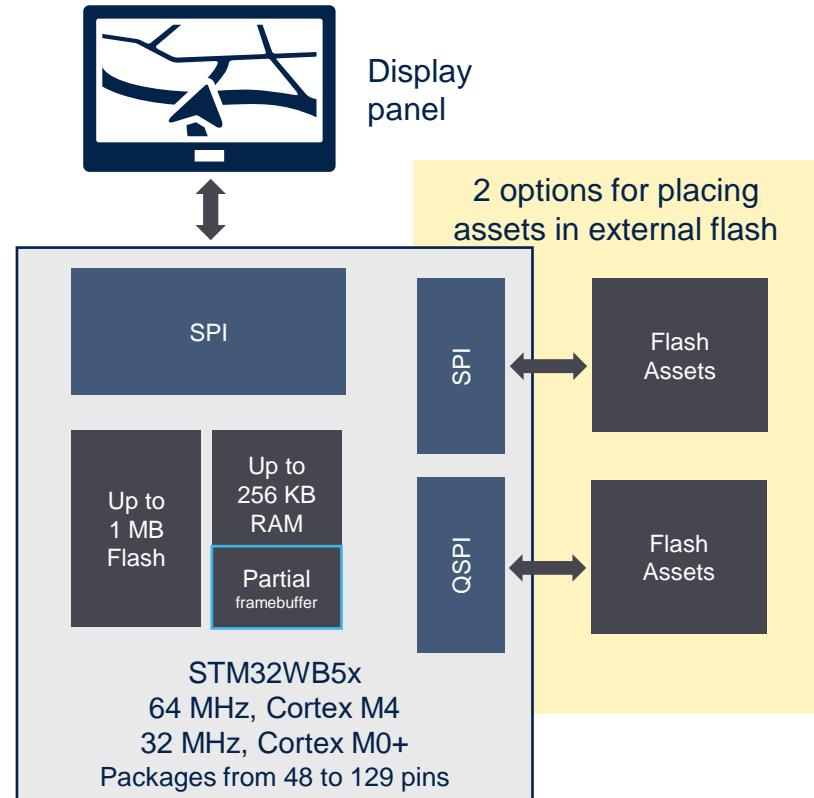
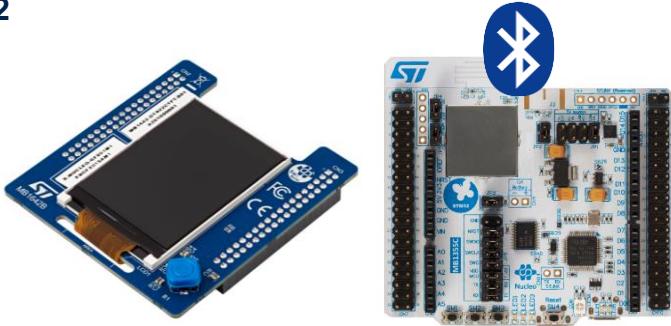
Entry-level graphics coupled with BLE STM32WB55RG

최대 해상도 480x272, 약 4.3 Inch 디스플레이



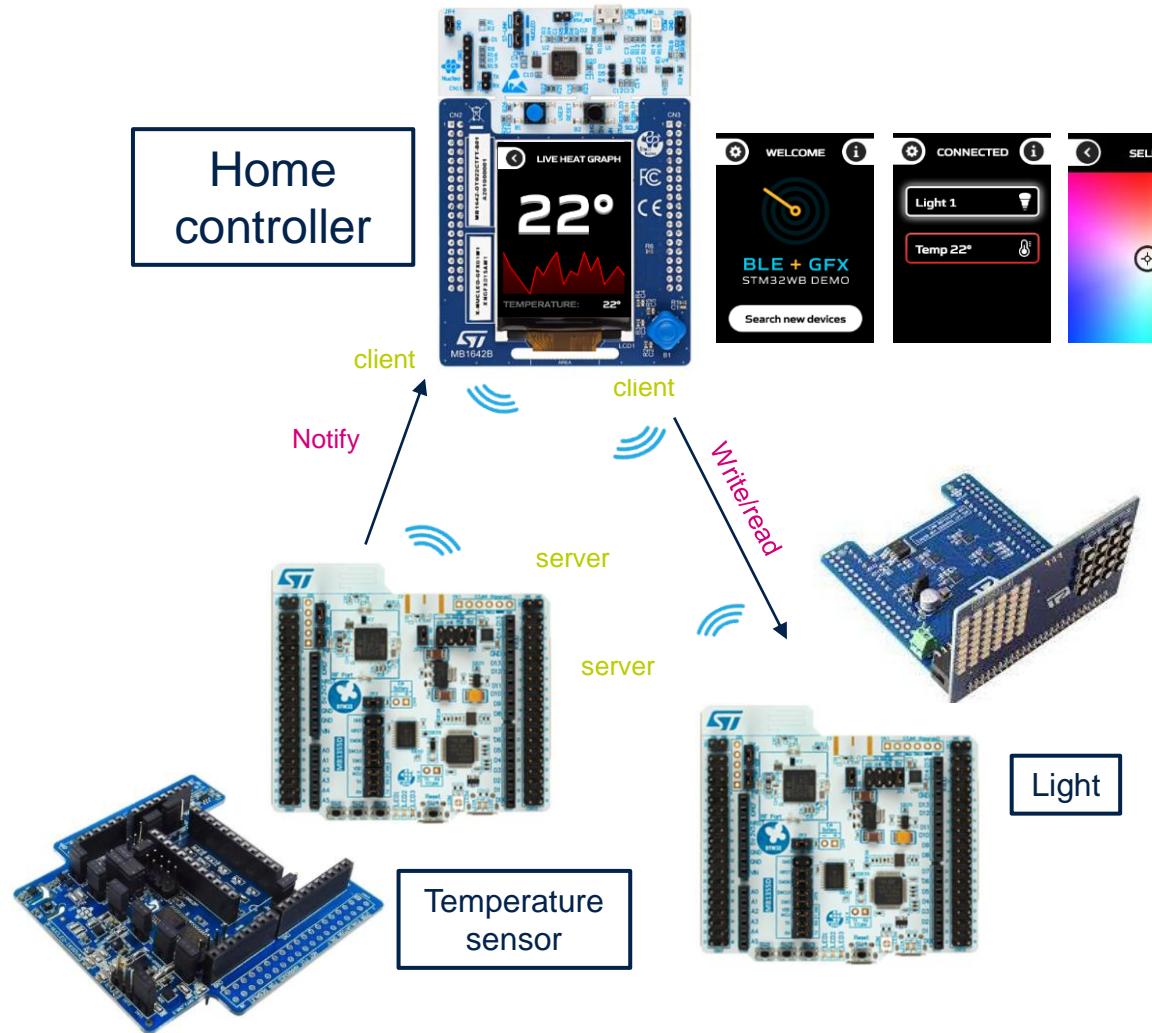
Resources

- NUCLEO-WB55RG + X-NUCLEO-GFX01M2
 - 2.2" SPI QVGA 320x240 display
- TouchGFX Designer
 - UI demo available
 - TouchGFX Board setup
 - Note:
TouchGFX GUI footprint starting from 30kB flash and 16kB RAM (including partial frame buffer)

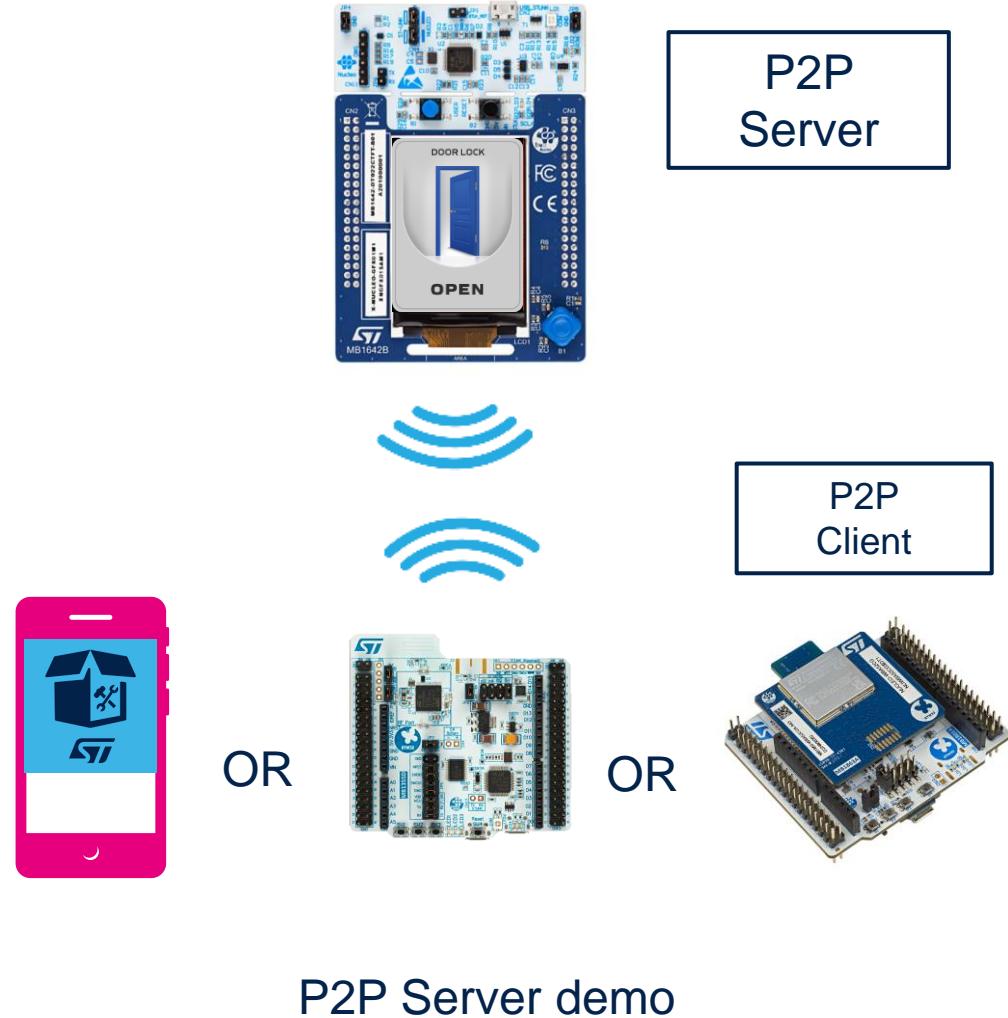




Entry-level graphics coupled with BLE STM32WB55RG | Demos available



Home controller demo



P2P Server demo



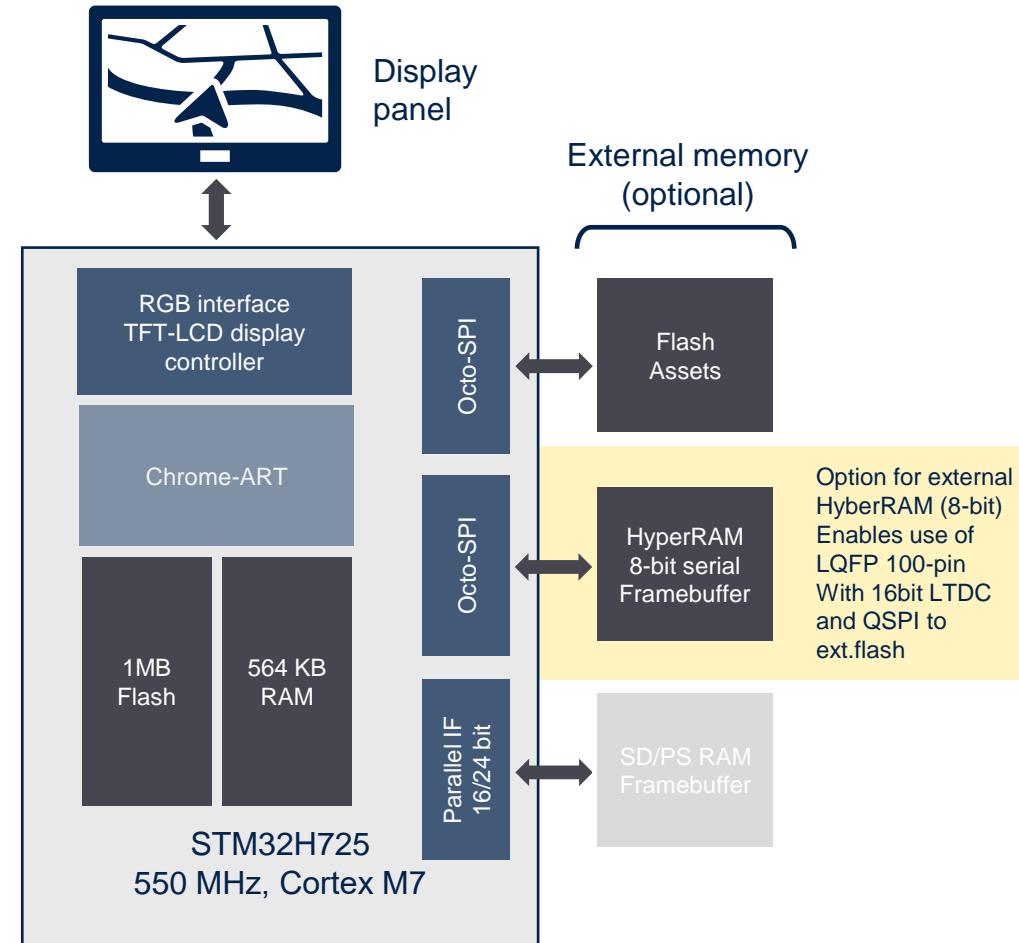
Advanced graphics with external RAM STM32H723/H725/H730

최대 해상도 약 800x480 / 1024x600, 7 Inch 디스플레이



Resources

- **STM32H735G-DK**
 - 4,3" WQVGA, 480x272 display
 - Preloaded TouchGFX demo
- **TouchGFX Designer**
 - Demo in full source
 - TouchGFX Board setup





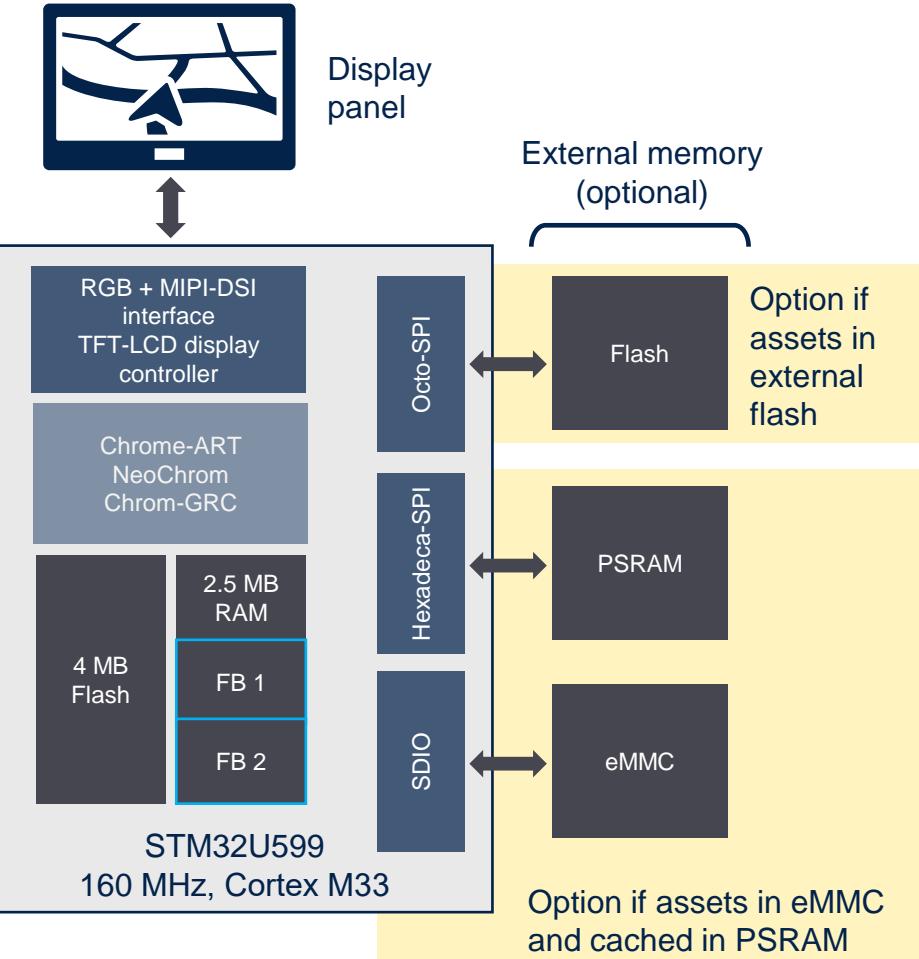
Advanced Graphics on Ultra-Low-Power MCUs STM32U5A9

최대 해상도 약 800x480, 7 Inch 디스플레이



Resources

- **STM32U599 Discovery Kit**
 - 2.5" 480x480, MIPI Video Mode Display
IF, OSPI flash
- **TouchGFX Designer**
 - UI demo in full source
 - TouchGFX Board setup



The TouchGFX solution





High-end UI with limited hardware requirements

TouchGFX GUI library

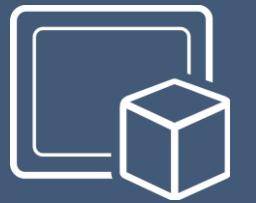


Delivered as an X-Cube-TouchGFX Package



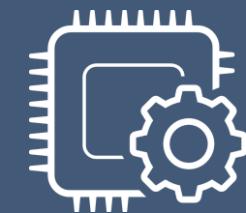
TouchGFX DESIGNER

PC GUI-builder
and -simulator



TouchGFX GENERATOR

Configure and
generate a
TouchGFX
project



TouchGFX ENGINE

Optimized and
hardware
accelerated
graphics library



STM32 software TouchGFX GUI software

TouchGFX Designer

쉬운 개발 환경 제공

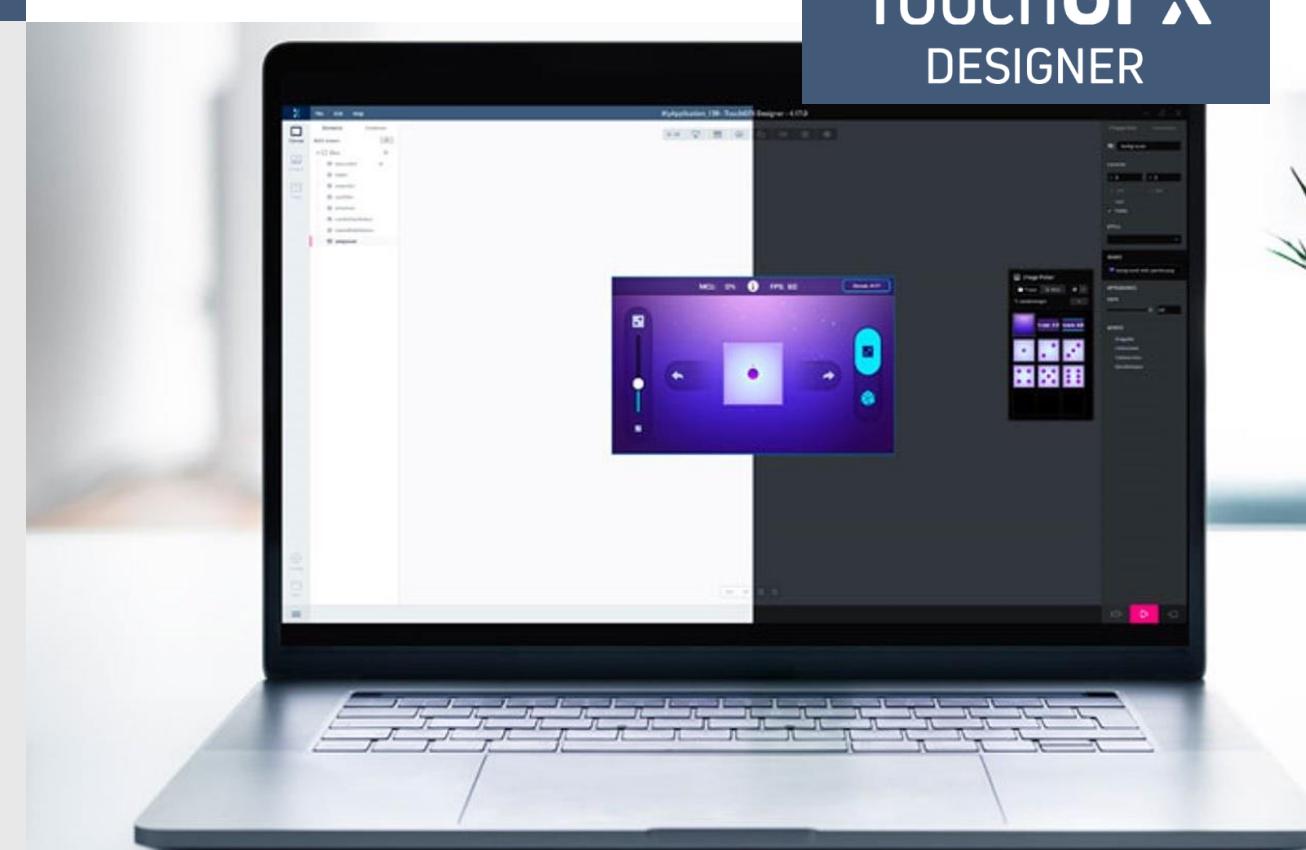
WYSIWYG GUI 개발 환경을 제공하여 훌륭한 GUI를 손쉽게 개발

원하는 모든 GUI를 생성

구조적이고 유연한 TouchGFX를 통해 개발자 고유한 UI를 쉽게 설계하고 만들 수 있도록 제공

→ The technology behind

- PC 시뮬레이터 또는 HW 타깃에서 실행
- 사용자 코드를 TouchGFX Designer에 의해 생성된 코드와 결합
- 내장된 위젯을 사용하여 고유의 소프트웨어 요소를 생성
- 고유의 위젯 설계 Design your own widgets.
- C++ 기반의 GUI
- Model-View-Presenter 패턴은 다른 C/C++ 애플리케이션 구성 요소와 쉽게 인터페이스 할 수 있는 방법을 제공





STM32 software TouchGFX GUI software

TouchGFX Generator

빠른 UI 프로젝트 생성 및 low-lever 개발에 대한 쉬운 환경 구성

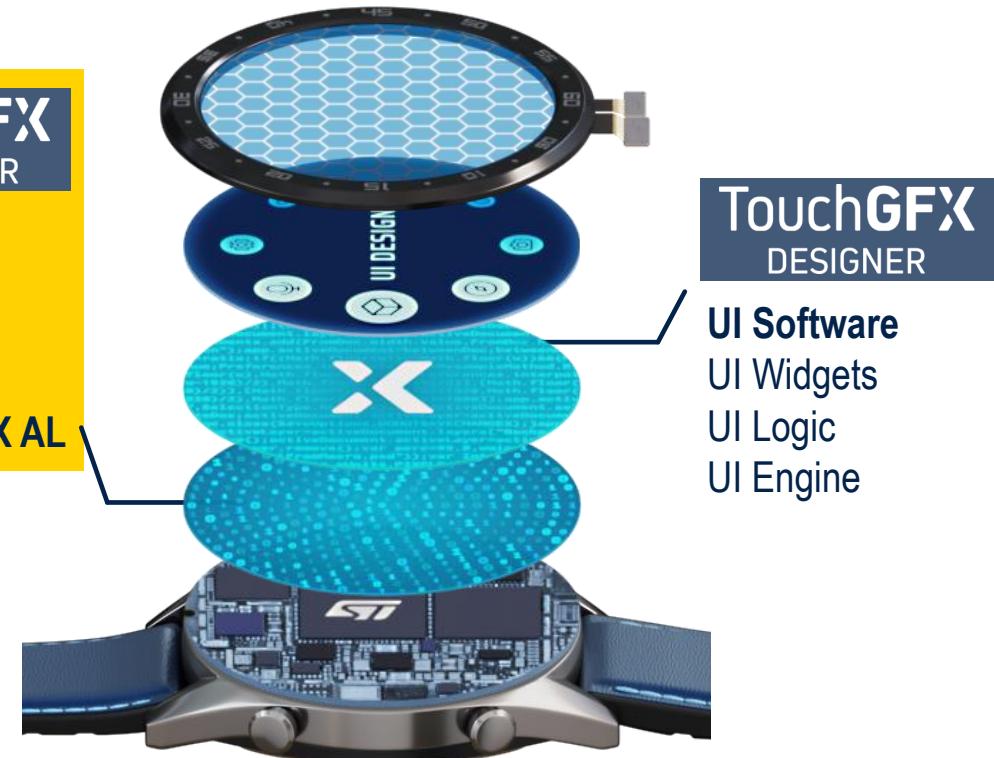
- 메모리
- 프레임 버퍼
- 디스플레이 해상도
- 색심도 (Color depth)

사용자가 선호하는 IDE를 선택

RTOS 선택 및 RTOS 사용 여부 지정

→ The technology behind

- CubeMX 플러그인을 통해 STM32 기반의 하드웨어를 위해 TouchGFX 추상화 계층(AL) 생성 및 환경 구성
- TouchGFX AL은 사용 가능한 그래픽 HW 가속기를 활성화 및 최적화
- CubeIDE, IAR Workbench 그리고 ARM Keil 중 개발자가 선호하는 IDE를 선택하여 프로젝트를 생성할 수 있도록 제공





STM32 software TouchGFX GUI software

TouchGFX Engine

최대 성능 구현

TouchGFX 엔진 기술을 통해 STM32 제품 기반에서 스마트폰과 같은 최고 수준의 GUI 성능을 달성할 수 있도록 한다

→ The technology behind

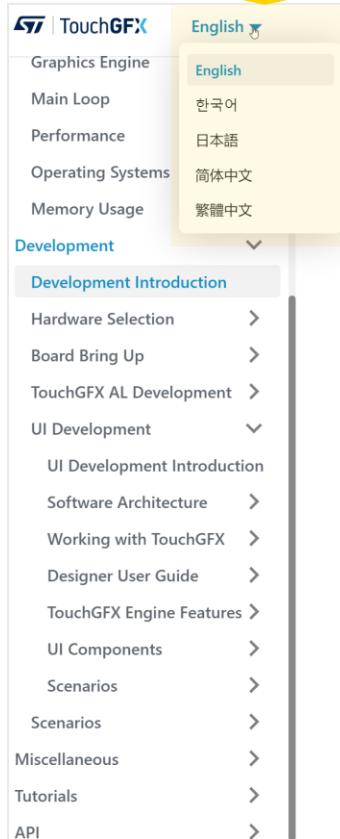
- **최소한의 MCU 로드 및 메모리 사용량에 최적화**
컴파일 및 런타임 분석
STM32 하드웨어 가속기를 적극적으로 활용
- **고급 렌더링 알고리즘**
최적화된 visible surface determination 알고리즘과 맞춤형 무효화
기법으로 그려지는 픽셀 수의 최소화
- **고급 그래픽 객체**
고도로 최적화된 위젯들로 런타임에서 선, 원, 사용자 지정 모양 및
그래픽을 그리거나 이미지에 스케일링 및 3D 회전을 적용

TouchGFX ENGINE



Get access to hours of training and detailed documentation

언어 선택
(5가지 언어 선택 가능)

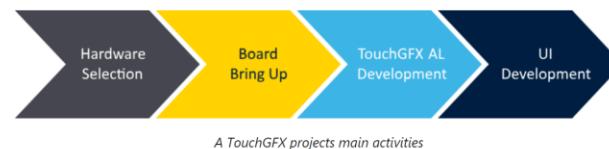


TouchGFX 버전 선택

Development Introduction

Main Activities

A TouchGFX project involves a set of activities that you will be addressing during the development phase. The effort in each of them are dependent on what the goal of your project is. If you are doing a UI prototype you can use premade code for major parts of the application and thus speed up your project development by skipping most of these activities. If you are doing a full project based on a custom made board, you will be addressing each of these activities in your project.



Main Components

Your TouchGFX project is made up of five main software and hardware components. Each of the activities will generate one of the main component for your TouchGFX project. The TouchGFX Engine is not an output of any main activity, this is the starting point for your TouchGFX project and is available when you have downloaded and installed.



4.17 ▾ Community TouchGFX.com

- Main Activities
- Main Components
- Hardware Selection
- Prototyping
- Custom Hardware
- Board Bring Up
- STM32CubeMX
- TouchGFX Board Setups (TBSs)
- TouchGFX AL Development
- TouchGFX Generator
- UI Development
- TouchGFX Designer
- TouchGFX Board Setups
- Demos and Examples
- Custom Hardware
- Workflow
- Generated Code and User Code
- Change of Compiler/IDE

TouchGFX Generator

TouchGFX Generator

- TouchGFX Generator는 X-CUBE-TOUCHGFX의 일부로 STM32CubeMX Additional-Software의 항목 중 하나로 구성.
- TouchGFX Generator 활성화
 - CubeMX에 “Select Component” 를 클릭하여 TouchGFX Generator를 활성화



Pack / Bundle / Component	Status	Version	Selection
> RowBot.I-CUBE-UNISONRTOS	5.5.0-4	Install	
> STMicroelectronics.X-CUBE-AI	6.0.0	Install	
> STMicroelectronics.X-CUBE-ALGOBUILD	1.2.0	Install	
> STMicroelectronics.X-CUBE-AZRTOS-H7	1.0.0		Install
> STMicroelectronics.X-CUBE-BLE1	4.4.0		Install
> STMicroelectronics.X-CUBE-BLE2	3.2.0		Install
> STMicroelectronics.X-CUBE-DISPLAY	1.0.0		Install
> STMicroelectronics.X-CUBE-GNSS1	5.2.0		Install
> STMicroelectronics.X-CUBE-MEMS1	8.3.0		Install
> STMicroelectronics.X-CUBE-NFC4	2.0.2		Install
> STMicroelectronics.X-CUBE-SFXS2LP1	3.0.0		Install
> STMicroelectronics.X-CUBE-SUBG2	4.0.0		Install
> STMicroelectronics.X-CUBE-TOF1	1.0.0		Install
> STMicroelectronics.X-CUBE-TOUCHGFX	4.17.0		Install
< Graphics Application	4.17.0		
Application	4.17.0		TouchGFX...
> VendorAlpha.Pack1	3.0.0		Not selected
> VendorAlpha.Pack2	3.0.0		TouchGFX...

TouchGFX를 활성화

Pack / Bundle / Component	Status	Version	Selection
> RowBot.I-CUBE-UNISONRTOS	5.5.0-4	Install	
> STMicroelectronics.X-CUBE-AI	6.0.0	Install	
> STMicroelectronics.X-CUBE-ALGOBUILD	1.2.0	Install	
> STMicroelectronics.X-CUBE-AZRTOS-H7	1.0.0	Install	
> STMicroelectronics.X-CUBE-BLE1	4.4.0	Install	
> STMicroelectronics.X-CUBE-BLE2	3.2.0	Install	
> STMicroelectronics.X-CUBE-DISPLAY	1.0.0	Install	
> STMicroelectronics.X-CUBE-GNSS1	5.2.0	Install	
> STMicroelectronics.X-CUBE-MEMS1	8.3.0	Install	
> STMicroelectronics.X-CUBE-NFC4	2.0.2	Install	
> STMicroelectronics.X-CUBE-SFXS2LP1	3.0.0	Install	
> STMicroelectronics.X-CUBE-SUBG2	4.0.0	Install	
> STMicroelectronics.X-CUBE-TOF1	1.0.0	Install	
> STMicroelectronics.X-CUBE-TOUCHGFX	4.17.0	Install	
< Graphics Application	4.17.0		
Application	4.17.0		TouchGFX...
> VendorAlpha.Pack1	3.0.0		Not selected
> VendorAlpha.Pack2	3.0.0		TouchGFX...

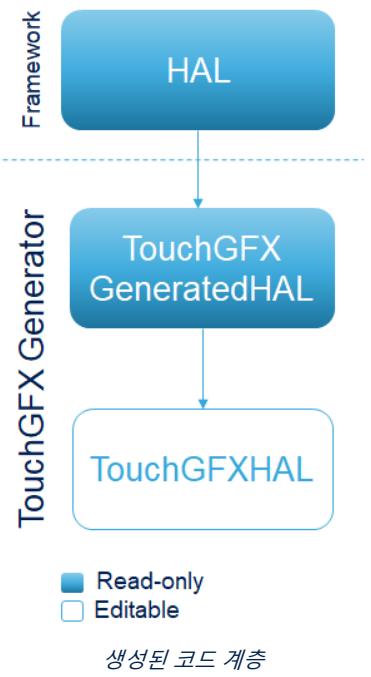
* MCU가 듀얼-코어인 경우 Single context만 활성화 가능

Generated Code Architecture

- **myproject.ioc**
: CubeMX 프로젝트 폴더
- **Core**
: main.c 및 startup 코드
- **Driver**
: CMSIS 및 MCU 제품군 드라이버
- **EWARM**
: IDE 프로젝트 폴더(EWARM, MDK-ARM 또는 STM32CubeIDE)
- **Middlewares**
: TouchGFX 라이브러리/헤더파일과 FreeRTOS와 같은 타사 소프트웨어를 포함
- **ApplicationTemplate.touchgfx.part**
: CubeMX에 의해 생성된 파일로 화면크기, 색심도와 같은 TouchGFX Designer와 관련된 정보를 포함
- **App**
: app_touchgfx.c는 TouchGFX를 초기화하고 메인 루프를 시작하는데 사용되는 함수에 대한 정의를 포함
- **target**
: HAL의 동작을 확장하거나 CubeMX에 의해 생성된 설정을 덮어 쓸 수 있도록 사용자에 의해 수정 가능한 파일을 포함
- **target/generated**
: CubeMX에서 설정을 변경한 경우 CubeMX에 의해 덮어 쓰여지는 read-only 파일들을 포함.
OSWrappers.cpp (OSAL)에는 TouchGFX 엔진과 동기화에 필요한 함수들을 포함

TouchGFX 폴더

```
.mxproject  
myproject.ioc  
Core  
Drivers  
EWARM  
Middlewares  
TouchGFX  
| ApplicationTemplate.touchgfx.part  
| App  
| | app_touchgfx.c  
| | app_touchgfx.h  
| target  
| | STM32TouchController.cpp  
| | STM32TouchController.hpp  
| | TouchGFXGPIO.cpp  
| | TouchGFXHAL.cpp  
| | TouchGFXHAL.hpp  
| generated  
| | OSWrappers.cpp  
| | TouchGFXConfiguration.cpp  
| | TouchGFXGeneratedHAL.cpp  
| | TouchGFXGeneratedHAL.hpp
```



Configuration Overview

• Display

인터페이스, 크기 그리고 버퍼 전략과 같은 디스플레이와 관련된 설정

- **Interface**

- STM32에서 사용 가능한 다양한 디스플레이 인터페이스
 - Parallel RGB(LTDC), MIPI DSI, FMC, SPI

- **Framebuffer Pixel Format**

- 프레임 버퍼의 픽셀 형식 설정.
 - XRGB8888(32bpp) ~ BW(1bpp)
 - 디스플레이 인터페이스 설정에 따라 설정 가능한 픽셀 형식 제약.

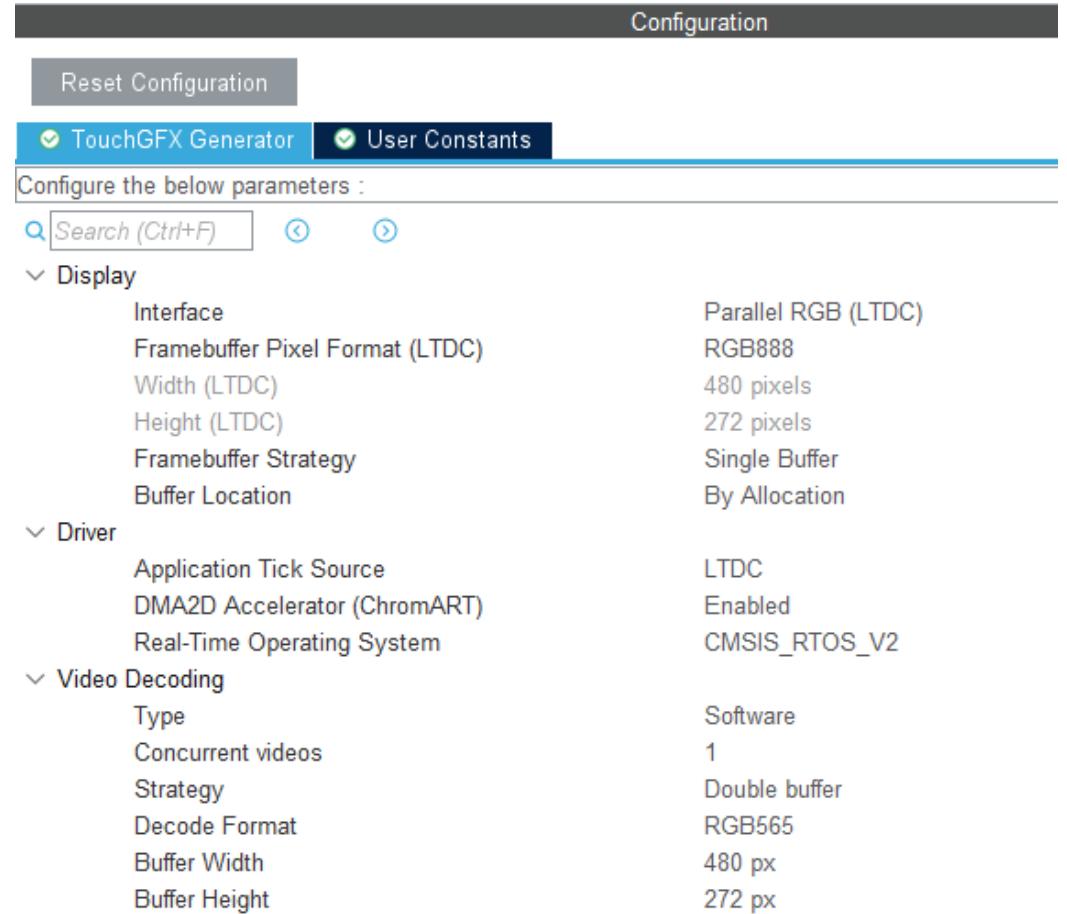
- **Framebuffer Strategy**

- 애플리케이션에 따른 프레임 버퍼의 전략 설정
 - Single Buffer, Double Buffer, Partial Buffer

- **Buffer Location**

- 프레임 버퍼의 위치를 지정하는 방식을 설정

- “Framebuffer Strategy”가 single 혹은 double인 경우:
 - By Allocation – 링커 스크립트에 따라 프레임 버퍼 메모리를 배치
 - By Address – 사용자가 프레임 버퍼 주소를 정의
- “Framebuffer Strategy”가 partial인 경우:
 - Number of blocks (내부 RAM에 항상 배치)
 - Block size (Byte)



Configuration Overview

• Driver

TouchGFX 추상화 계층(AL)의 다양한 책임에 맞는 드라이버를 선택

• Application Tick Source

: TouchGFX 애플리케이션 수행에 토대가 되는 Tick source 선택

- LTDC : TouchGFX Generator 가 OSWrappers::signalVSync()를 호출하는 함수를 LTDC 인터럽트 handler에 생성.
- Custom and FMC : 개발자가 OSWrappers::signalVSync() 를 반복적으로 호출하는 handler를 구현하는 것을 요구.

• DMA2D Accelerator (ChromART)

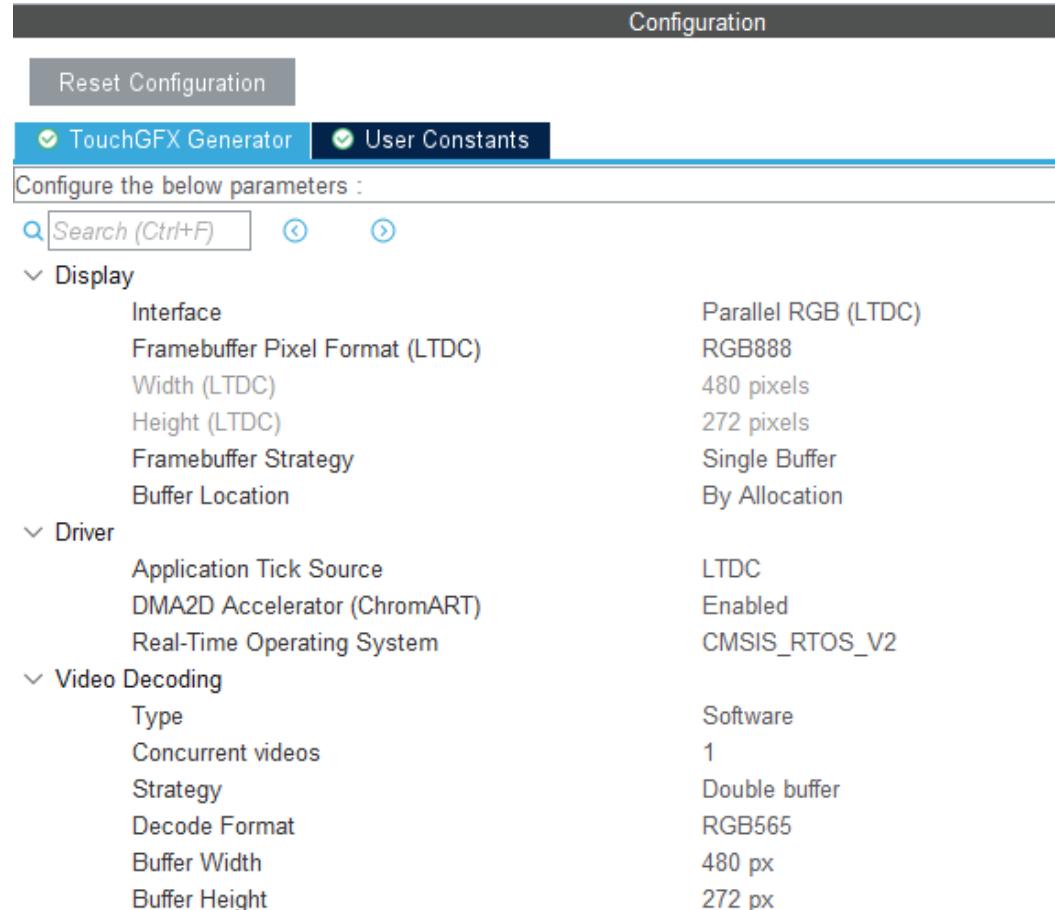
: 그래픽 가속에 관련한 옵션 선택

- None
- DMA2D Accelerator (ChromART)
- GPU2D (NeoChrom) /* 새로운 기능 */

• Real-Time Operating System

: TouchGFX Engine은 OSWrappers 인터페이스를 사용해 메인 이벤트 루프를 비롯한 프레임 버퍼 액세스를 사용자가 선택한 RTOS와 동기화

- No OS : OS를 사용하지 않는 Bare-metal 기반
- CMSIS_RTOS_V1/CMSIS_RTOS_V2 : Middleware 에서 FreeRTOS를 활성화한 경우 사용 가능 하며 CMSIS OS V1 및 V2 를 모두 호환
- ThreadX : AzureRTOS software package 선택 또는 CubeMX에 Native ThreadX Middleware (new MCU 한정)를 활성화 한 경우 사용.
- Custom : FreeRTOS 및 ThreadX 외의 OS 사용하는 경우, 사용하는 OS에 맞춰 OSWrappers 인터페이스를 사용자가 직접 구현.



Configuration Overview

• Addition Features

• External Data Reader

- RGB565 프레임 버퍼 픽셀 형식에서 Data Reader라는 인터페이스 지원
- Non-memory-mapped 플래시 메모리에서 caching 없이 데이터를 직접 읽음
- 추가 버퍼에 대한 메모리가 충분하지 않은 저가 솔루션(STM32G0)에서 사용
- Disable : 메모리의 버퍼에 asserts를 caching 하는 과정이 필요
- Enable : TouchGFX가 생성한 인터페이스를 통해 직접 asserts에 대한 데이터를 검색

• External Data Reader: Line Buffer Size

- 프레임 버퍼에 이미지 또는 텍스트를 혼합할 수 있도록 2개의 버퍼를 생성
- ARGB8888 픽셀 형식에서 풀 사이즈 이미지를 지원하기 위해 기본 값은 [화면 너비 * 4바이트]로 설정

• External Data Reader: Minimum DMA transfer size

- DMA 전송을 시작하기 위해 필요한 최소한의 바이트를 설정
- 더 많은 바이트가 요청되는 경우 DMA 를 사용하지 않음

The screenshot shows the TouchGFX Generator configuration interface. At the top, there are two tabs: 'TouchGFX Generator' (selected) and 'User Constants'. Below the tabs, a search bar says 'Search (Ctrl+F)' and two circular arrows are shown. The main area is titled 'Configure the below parameters :'. It contains three sections: 'Display', 'Driver', and 'Additional Features'. The 'Display' section includes settings for Interface (Custom), Framebuffer Pixel Format (RGB565), Width (480 pixels), Height (272 pixels), Framebuffer Strategy (Single Buffer), and Buffer Location (By Allocation). The 'Driver' section includes Application Tick Source (Custom), Graphics Accelerator (None), and Real-Time Operating System (No OS). The 'Additional Features' section includes External Data Reader (Enabled), External Data Reader: Line Buffer size (1920 bytes), and External Data Reader: Minimum DMA transfer size (40 bytes).

Category	Setting	Value
Display	Interface	Custom
	Framebuffer Pixel Format	RGB565
	Width	480 pixels
	Height	272 pixels
	Framebuffer Strategy	Single Buffer
	Buffer Location	By Allocation
Driver	Application Tick Source	Custom
	Graphics Accelerator	None
	Real-Time Operating System	No OS
Additional Features	External Data Reader	Enabled
	External Data Reader: Line Buffer size	1920 bytes
	External Data Reader: Minimum DMA transfer size	40 bytes

Configuration Overview

• Video Decoding

하드웨어 또는 소프트웨어에 의한 비디오 디코딩 기능을 사용

- **Type**

: Video decoding 구동 방식(HW 혹은 SW)를 선택

- Disable : 기본 값
- Software : CubeMX의 Middleware 섹션에서 LIBJPEG이 enable 되어 있는 경우 software decoder 생성 가능 (S/W MJPEG decoder).
- Hardware : Multimedia 섹션에서 JPEG이 enable 되고 TouchGFX Generator에서 CMSIS compliant RTOS가 선택되어 있는 경우 “Hardware” 옵션 선택 가능.

- **Concurrent videos**

: 동일한 화면에서 동시에 디코딩되는 최대 비디오 수
(최대 4개의 비디오).

- **Strategy**

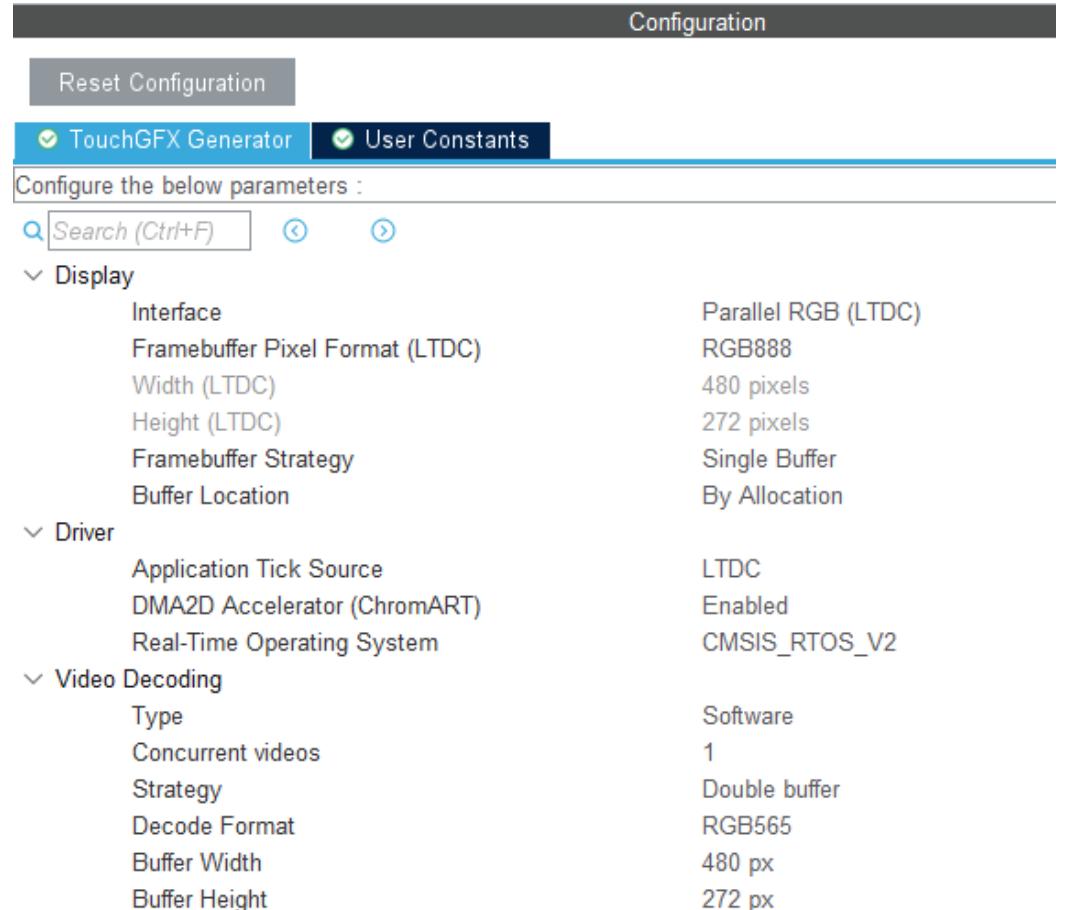
- Direct to Framebuffer
: UI thread에서 비디오 디코딩 수행

- Single Buffer
: 별도의 task에 위치한 전용 버퍼에서 비디오 디코딩 수행

- Double Buffer
: 별도의 task에 위치한 두개의 전용 버퍼체서 비디오 디코딩 수행

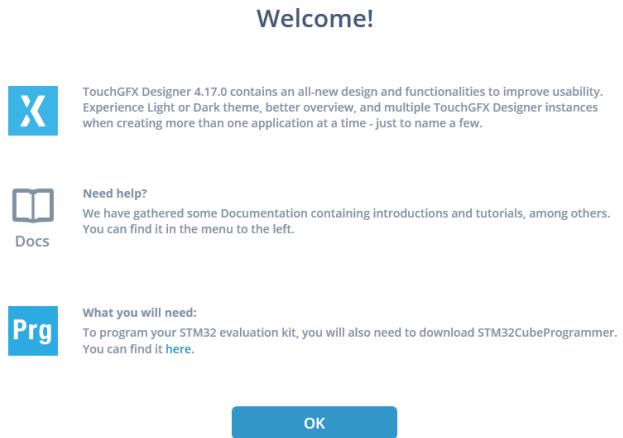
- **Decode Format**

: 프레임 버퍼의 포맷 형식에 상관없이 RGB 버퍼를 선택

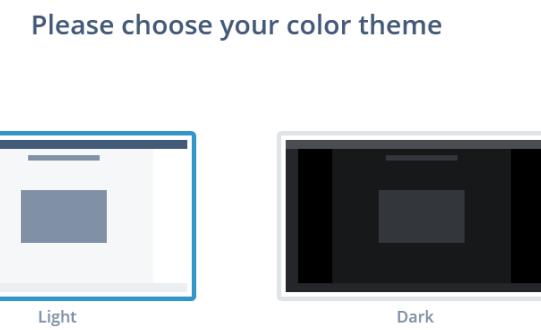


TouchGFX Designer

- TouchGFX Designer를 실행했을 때 나타나는 첫 번째 화면
- Welcome 팝업
- Light 및 Dark 테마 선택



Welcome pop-up



Theme selector pop-up

- Recently used TouchGFX Board Setups(TBS)를 기반한 새로운 프로젝트를 생성 및 실행

The screenshot shows the TouchGFX Home Tab interface. At the top, there is a search bar labeled "Open" with the placeholder text "파일 브라우저를 통해 TouchGFX 프로젝트 파일을 검색하고 실행". Below the search bar are two buttons: "Create New" (highlighted in blue) and "Open". To the left of the main content area, there is a vertical sidebar with icons for "Home", "Create", "Demos", "Docs", and "Q&A".

The main content area has three main sections:

- Create New:** A callout box points to the "Create New" button in the sidebar. The text says "Lobby에 있는 탭이 Home에서 Create로 전환".
- Recently used TouchGFX Board Setups:** A callout box points to a section showing four recently used board setups: STM32H7B3I Evaluation, STM32G071 NUCLEO, STM32F746G Discovery Kit, and STM32F7508 Discovery Kit. Each card displays the board name, operating system, and resolution.
 - My Recent Projects:** A callout box points to a section titled "My Recent Projects" which lists five recent projects: STM32H7B3I-Eval, STM32G071, FlexButton, STM32F746G-Discovery, and STM32F469I-DISCO. Each project entry includes a small icon, the project name, and the dates of creation and modification.
- Open:** A callout box points to the "Open" button at the top right. The text says "Select one of the recently used TouchGFX Board Setups to create an empty project based on that setup. You can also reopen an existing project either by finding it in the recent project list or by browsing your disk by pressing the Open button."

- 새로운 프로젝트를 사용될 TBS를 선택

Producer Filter

- Show All : 모든 TBS 들을 나열
- By STMicroelectronics : STMicroelectronics에 의해 제공되는 TBS들을 나열
- By Partners : 파트너 업체를 통해 제공되는 TBS들을 나열

TBS List
선택된 필터를 토대로
TBS의 항목을 나열.

The screenshot shows the TouchGFX Designer interface with the following elements:

- Searchbar:** "보드의 이름으로 TBS를 필터" (Filter TBS by board name)
- Producer Filter:** Buttons for "Show All", "By STMicroelectronics", and "By Partners".
- TBS List:** A grid of 12 TBS items, each with a thumbnail, name, and brief details. The first item, "Simulator", is highlighted with a blue border.
- Selected TBS:** A detailed view of the "Simulator" TBS on the right, showing its operating system (Windows), description (The simulator runs exclusively on your PC without the need for any embedded target. Use the simulator for prototyping applications, when your hardware is not available, or for evaluating TouchGFX. Any display resolution and any color depth can be chosen. The simulator supports a GCC and a Visual Studio™ based build.), application name ("MyApplication_93"), application directory ("C:\TouchGFXProjects"), color depth ("24 bit"), version ("v 2.0.0"), and resolution ("W 800 H 480"). A large pink "Create" button is at the bottom.
- Create Tab:** A tab labeled "Create" located at the bottom center of the interface.

Example

- 예제와 TBS를 기반으로 새로운 프로젝트를 생성

The screenshot shows the TouchGFX Designer interface version 4.17.0. On the left, there's a sidebar with icons for Home, Create, Examples (which is selected), Demos, Docs, and Q&A. The main area is titled "Example list". A search bar at the top right has the placeholder "Searchbar 이름으로 예제를 필터". A tooltip over the search bar says "Select Board Setup TBS를 선택할 수 있는 팝업을 실행". Another tooltip over the search bar says "Example List 선택된 필터를 토대로 예제의 항목을 나열". Below the search bar, there's a note: "Examples are small applications that show simple usage of either a widget or feature. Click "Select Board Setup" if you want to run the example on a specific board. If not, just leave it and the example will run on the TouchGFX Simulator." To the right of this note, a detailed view of the "Animation Texture Mapper Example" is shown, including its description, application name (MyApplication_90), application directory (C:\TouchGFXProjects), color depth (24 bit), versions (v 2.3.0), resolution (800 x 480 px), and a large "Create" button.

TouchGFX Designer - 4.17.0

Select Board Setup

Searchbar 이름으로 예제를 필터

Example List 선택된 필터를 토대로 예제의 항목을 나열

Select Board Setup TBS를 선택할 수 있는 팝업을 실행

Blank UI Name BlankUI Resolution Any

Animated Image Example Name AnimationExample Resolution 480 x 272, ...

Animation Texture Mapper... Name AnimationTextureMapperExample Resolution 480 x 272, ...

Arabic Text Example Name ArabicTextExample Resolution 480 x 272, ...

Button Example Name ButtonExample Resolution 480 x 272, ...

Clock Example Name ClockExample Resolution 480 x 272, ...

Custom Trigger Action Exa... Name CustomTriggerActionExample Resolution 480 x 272, ...

Custom Widget Example Name CustomWidgetExample Resolution 480 x 272, ...

Dynamic Graph Example

Flex Button Example Name Click Button Resolution UP White

Gauge Example

Keyboard Example

Animation Texture Mapper Example

STMicroelectronics

Description

The Animation Texture Mapper Example shows how to animate a rotation and/or scaling of an image using the extended Texture Mapper class AnimationTextureMapper.

Application Name MyApplication_90

Application Directory C:\TouchGFXProjects

Color Depth 24 bit

Versions v 2.3.0

Resolution 800 x 480 px

Create

Example list

Demos

- 데모 및 TBS 또는 Board Specific Demo(BSD) 기반의 프로젝트 생성

The screenshot shows the TouchGFX Designer software interface. On the left, there's a sidebar with icons for Home, Create, Demos, Docs, and Q&A. The main area has tabs for Demo and Board Specific Demo, with 'Demo' selected. A 'Demos Filter' button is at the top. Below it is a search bar. The 'Demos tab' displays a grid of demo projects:

- Dice Animation (Name: DiceAnimation, Resolution: 800 x 480)
- Knight Hits Zombie Game (Name: KnightHitsZombie, Resolution: 800 x 480)
- Ninja Vs Robot (Name: NinjaVsRobot, Resolution: 800 x 480)
- Pool Demo (Name: PoolDemo, Resolution: 480 x 272)
- Screen Transitions (Name: ScreenTransitions, Resolution: 800 x 480)
- Smart Home (Name: SmartHome, Resolution: 480 x 272)
- TouchGFX Demo 1 (Name: TouchGFXDemo1, Resolution: 480 x 272)
- TouchGFX Demo 2 (Name: TouchGFXDemo2, Resolution: 800 x 480)
- TUTORIAL (Name: TUTORIAL, Resolution: 800 x 480)
- TOUCHGFX (Name: TOUCHGFX, Resolution: 800 x 480)

A callout box labeled 'Selected Board Setup' points to the 'Select Board Setup' button in the sidebar. Another callout box labeled 'Demos List' points to the grid of demos. A third callout box labeled 'Demos Filter' points to the 'Demos Filter' button. A fourth callout box labeled 'Searchbar' points to the search bar. A fifth callout box labeled 'Selected Demos or BSD' points to the right panel, which shows a detailed view of the 'Ninja Vs Robot' demo.

Demos Filter
Demo 또는 Board Specific Demo를 선택

Select Board Setup
TBS를 선택할 수 있는
팝업을 실행

Demos List
선택된 필터를 토대로
데모의 항목을 나열

Demos Filter
Demo 또는 Board Specific Demo를 선택

Searchbar
이름으로 예제 또는 BSD를 필터

Selected Demos or BSD
데모를 선택했을 때 해당 데모의 설명
및 새로운 프로젝트를 생성하기 위한
옵션을 표시

TouchGFX Designer - 4.17.0

Ninja Vs Robot
STMicroelectronics

Description

Small game that showcases the capabilities of TouchGFX. The game also demonstrates, the impact that different level graphic complexity has on performance, by letting the user select the three levels, low, medium, or high. Ninja Vs Robot also has the option to enable/disable Chrom-ART thereby showing the performance enhancement Chrom-ART offers.

Application Name: MyApplication_91

Application Directory: C:\TouchGFXProjects

Color Depth: 24 bit

Versions: v 1.1.0

Resolution: 800 x 480 px

Create

Docs

Q&A

Demos

Home

Create

Demos

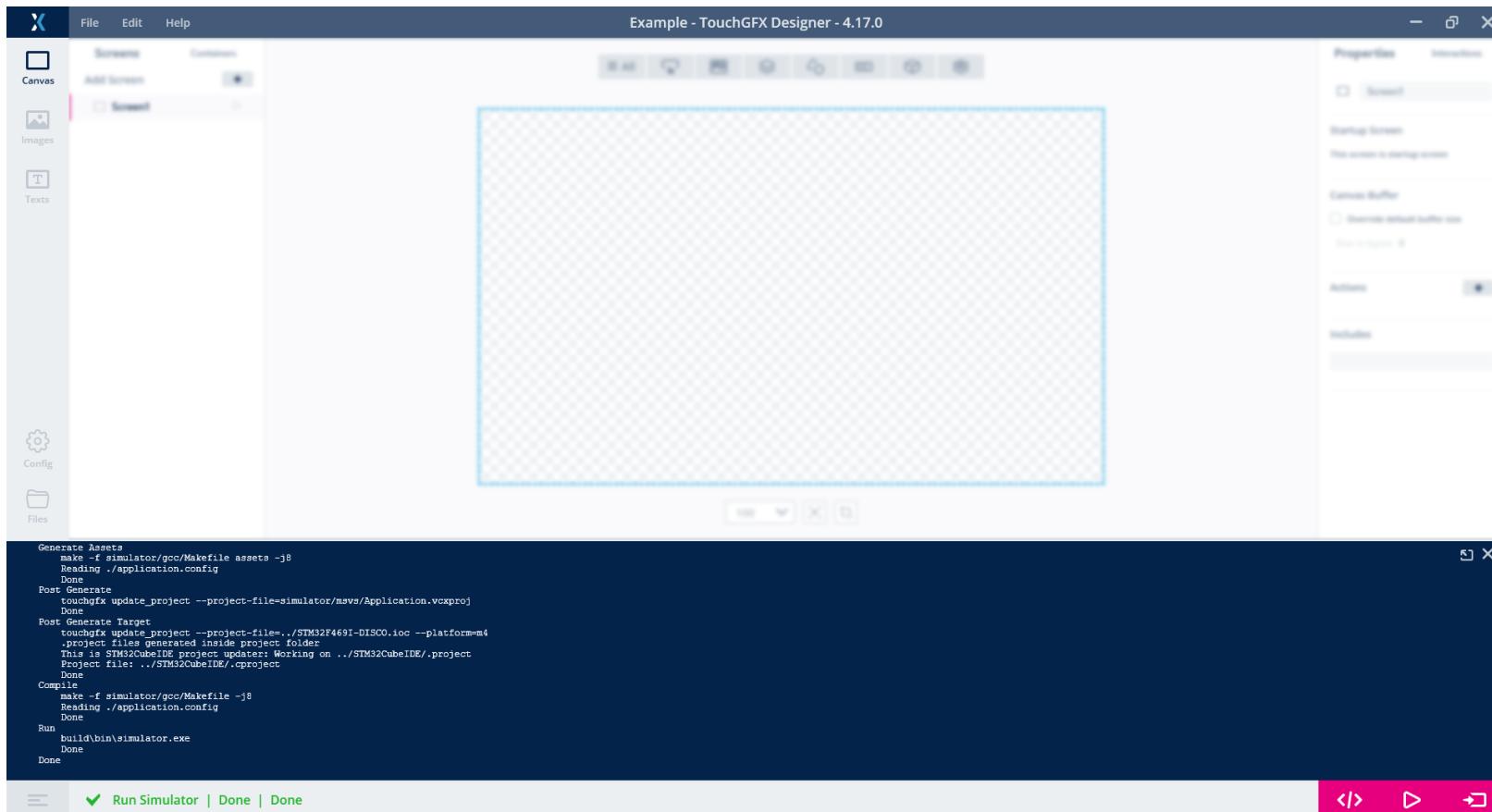
TouchGFX Designer - 4.17.0

Demos tab

ST logo: life.augmented

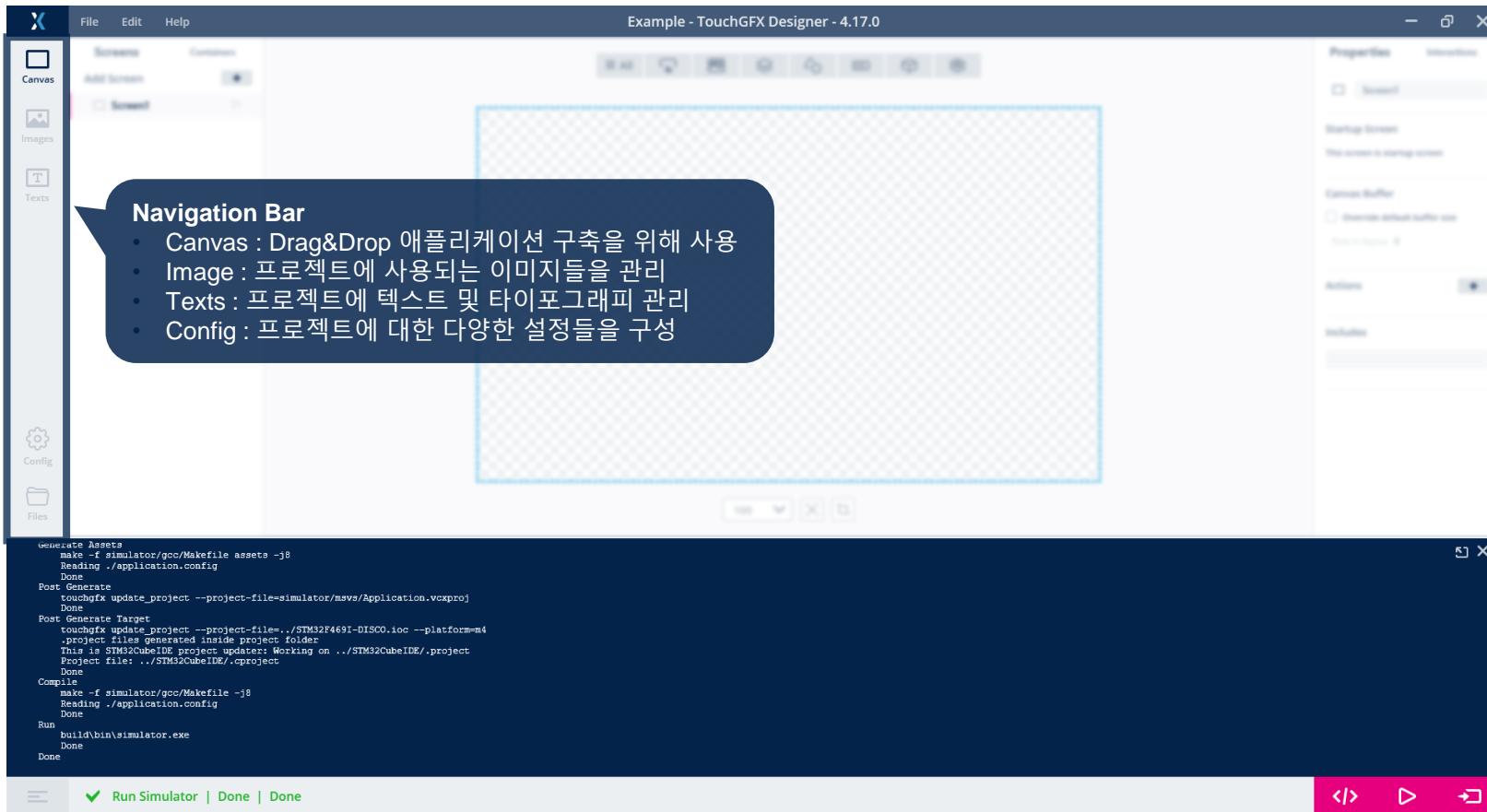
Main Window

- Navigation Bar, Command Buttons, Notification Bar, Notification Bar, Detailed log



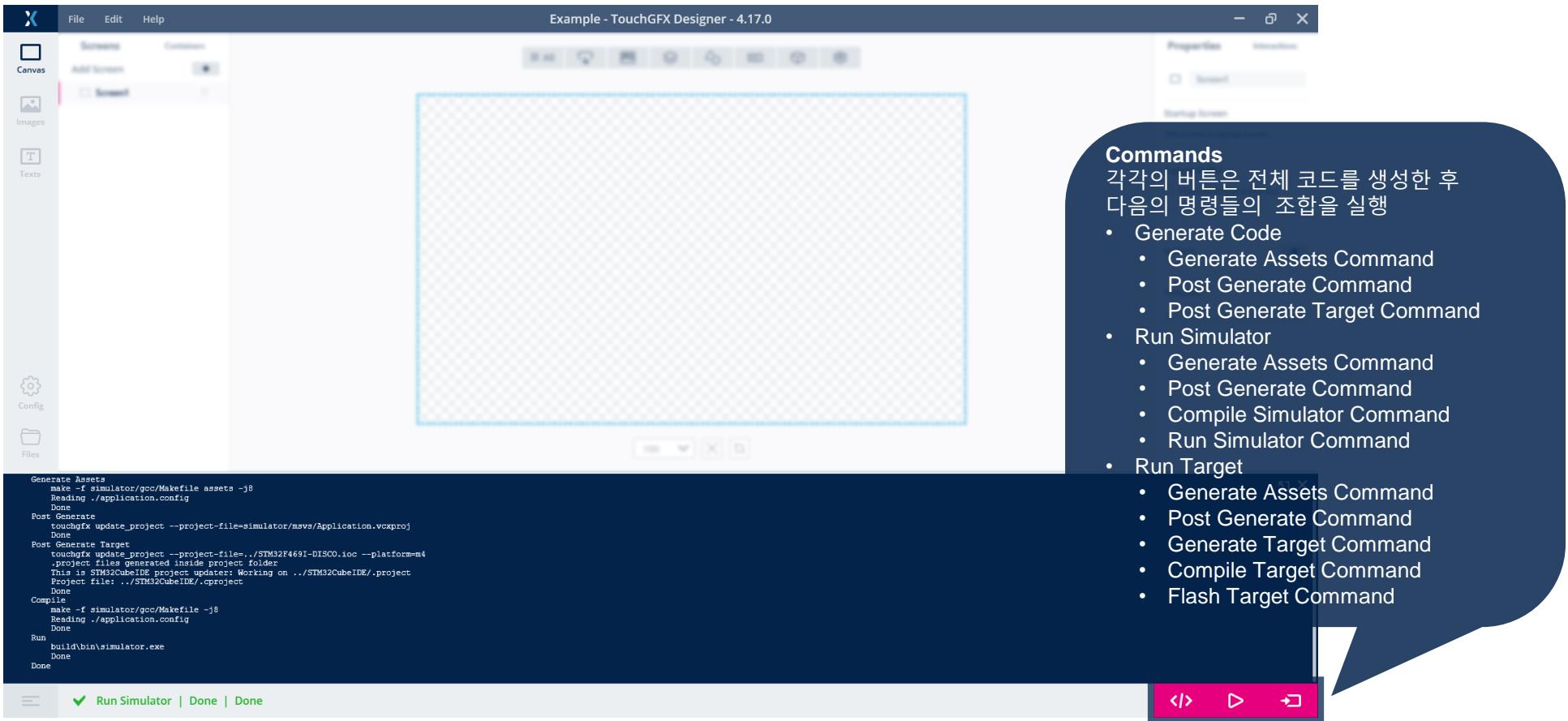
Main Window

- Navigation Bar, Command Buttons, Notification Bar, Detailed log 로 구성



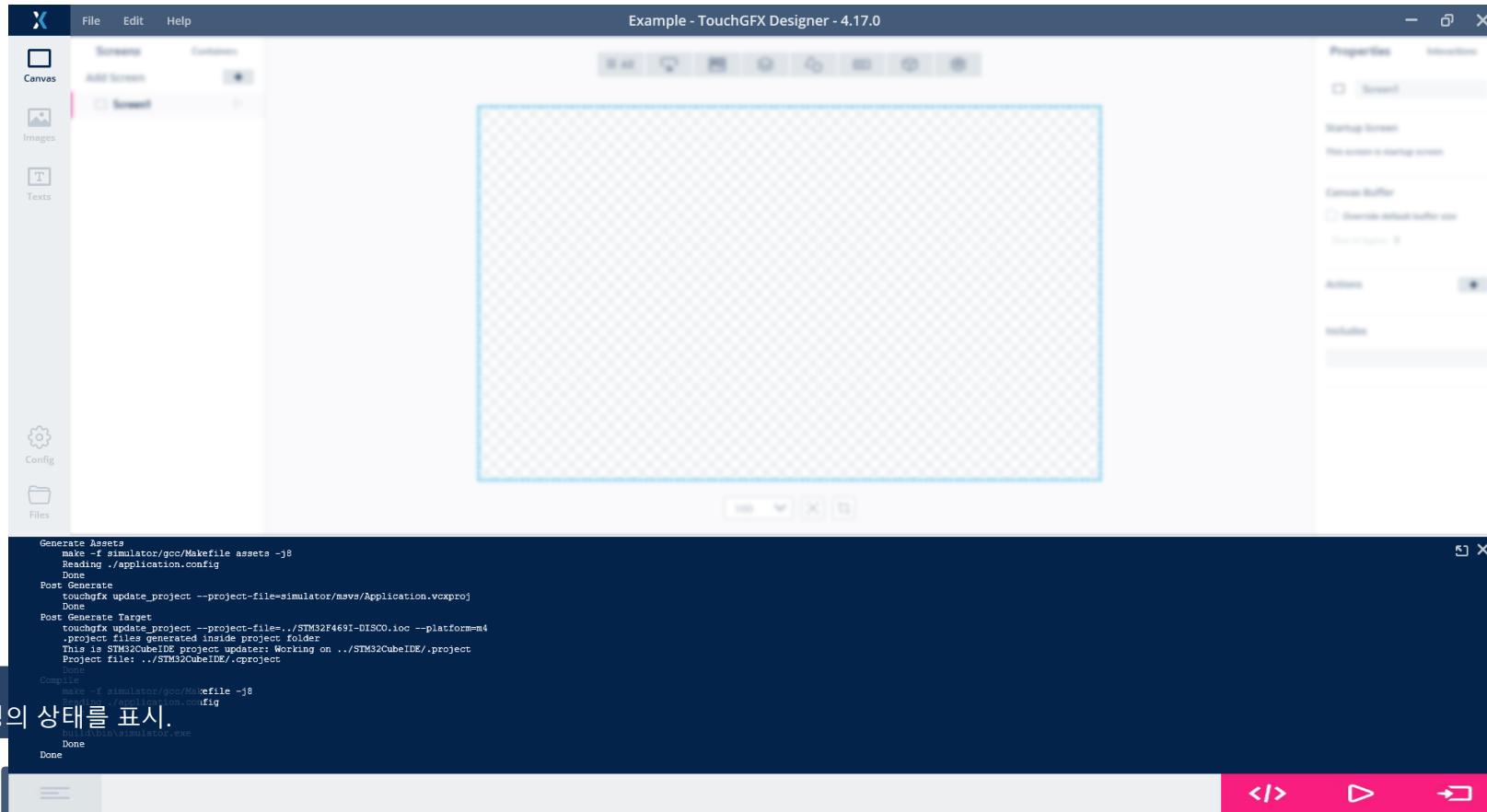
Main Window

- Navigation Bar, Command Buttons, Notification Bar, Detailed log 로 구성



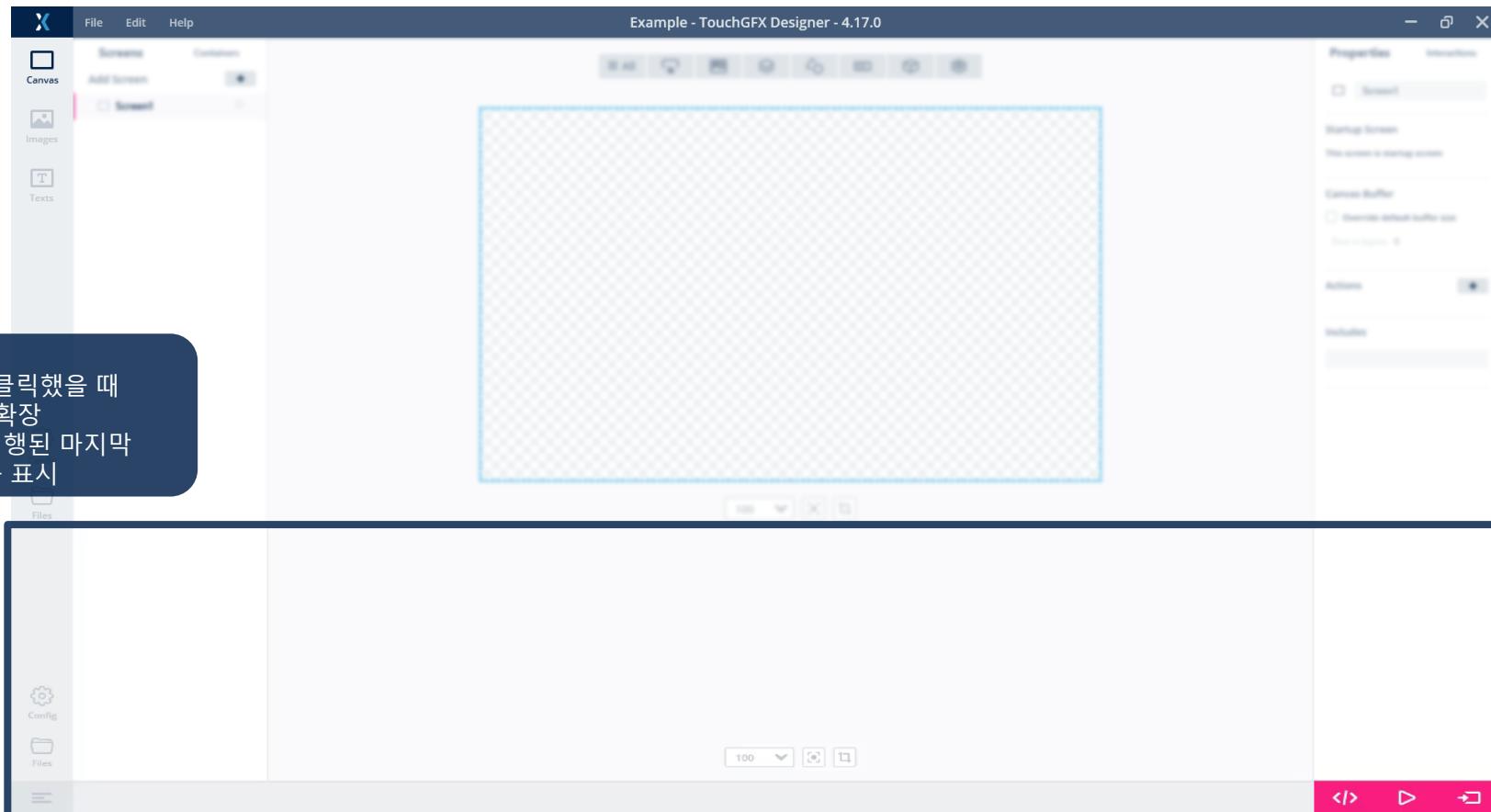
Main Window

- Navigation Bar, Command Buttons, Notification Bar, Detailed log
- 로 구성



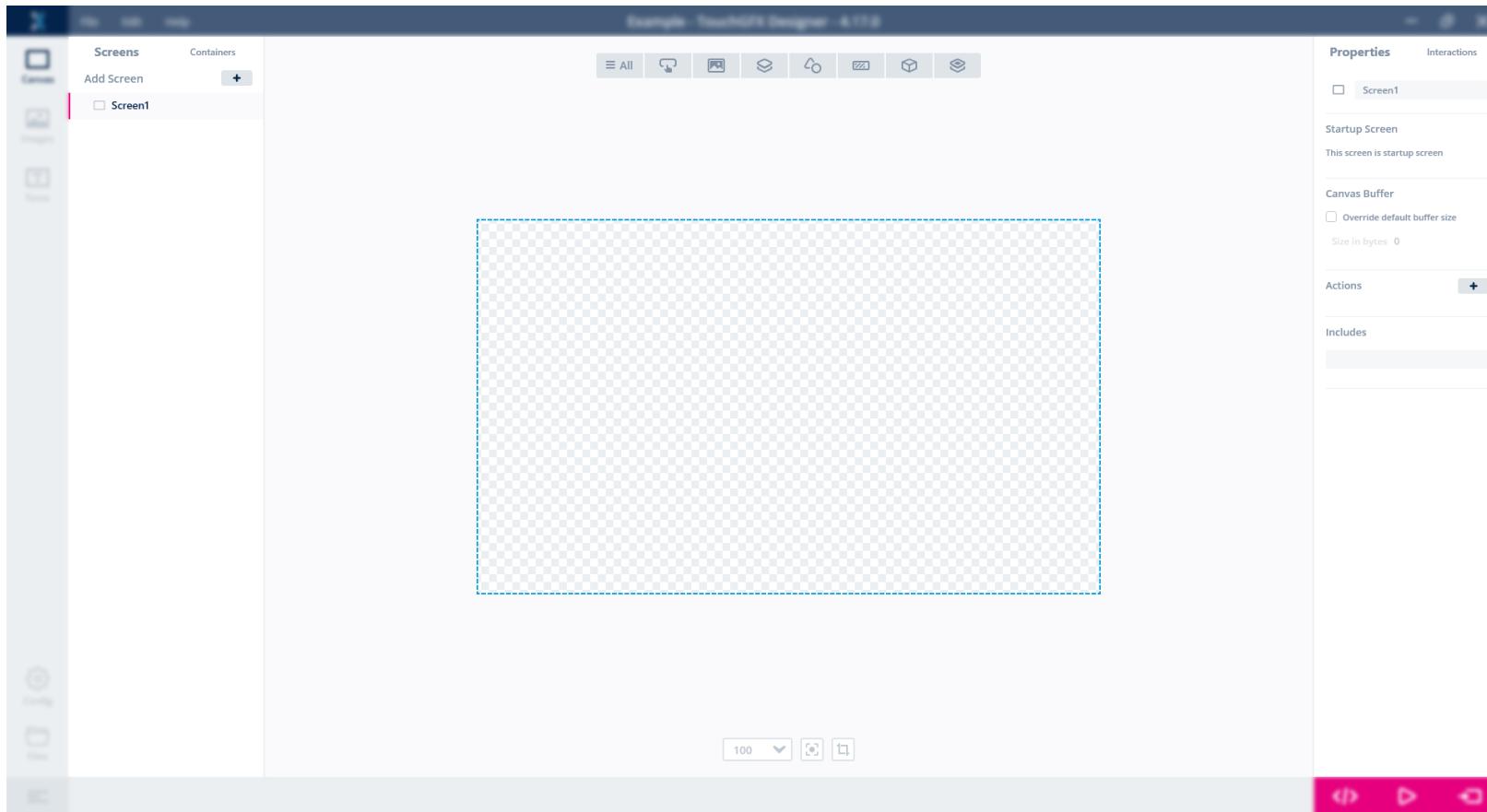
Main Window

- Navigation Bar, Command Buttons, Notification Bar, Detailed log 로 구성



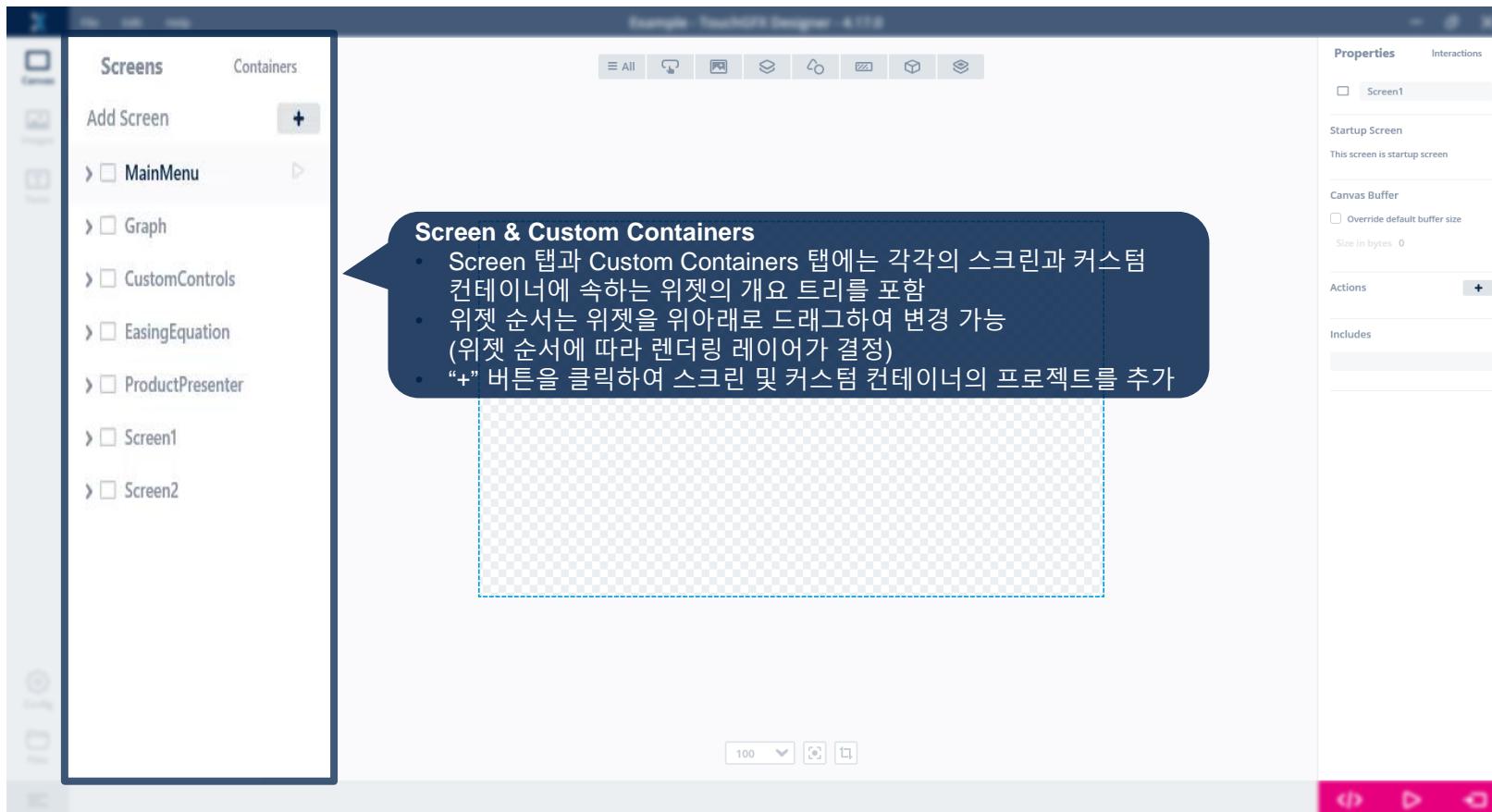
Canvas View

- 애플리케이션의 그래픽 요소를 개발하기 위해 사용하는 View로서 실행했을 때 보여지는 인터페이스를 시각화
- 시스템 요소간 상호작용과 같은 동적요소에 대한 설명 제공



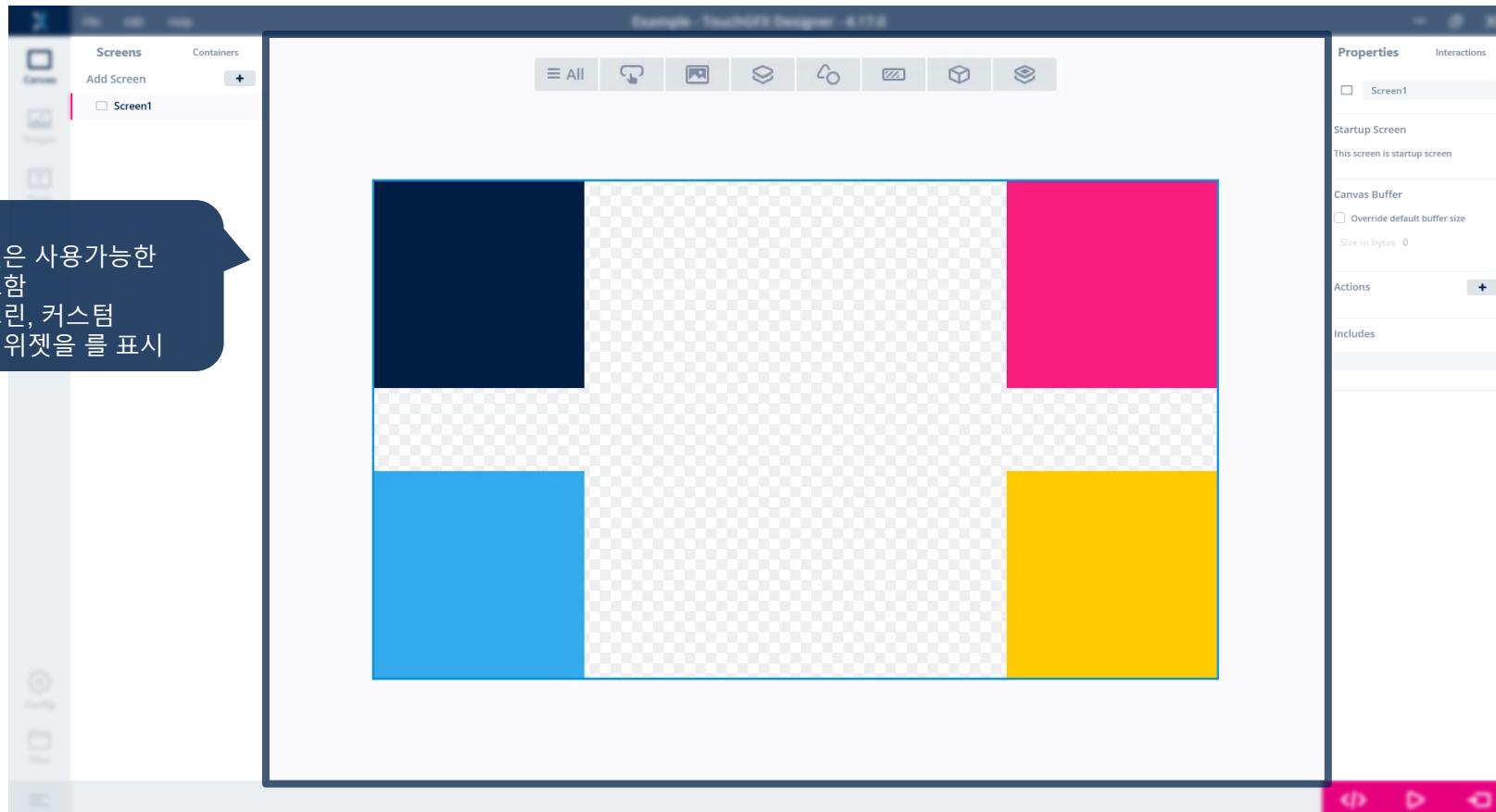
Canvas View

- 애플리케이션의 그래픽 요소를 개발하기 위해 사용하는 View로서 실행했을 때 보여지는 인터페이스를 시각화
- 시스템 요소간 상호작용과 같은 동적요소에 대한 설명 제공



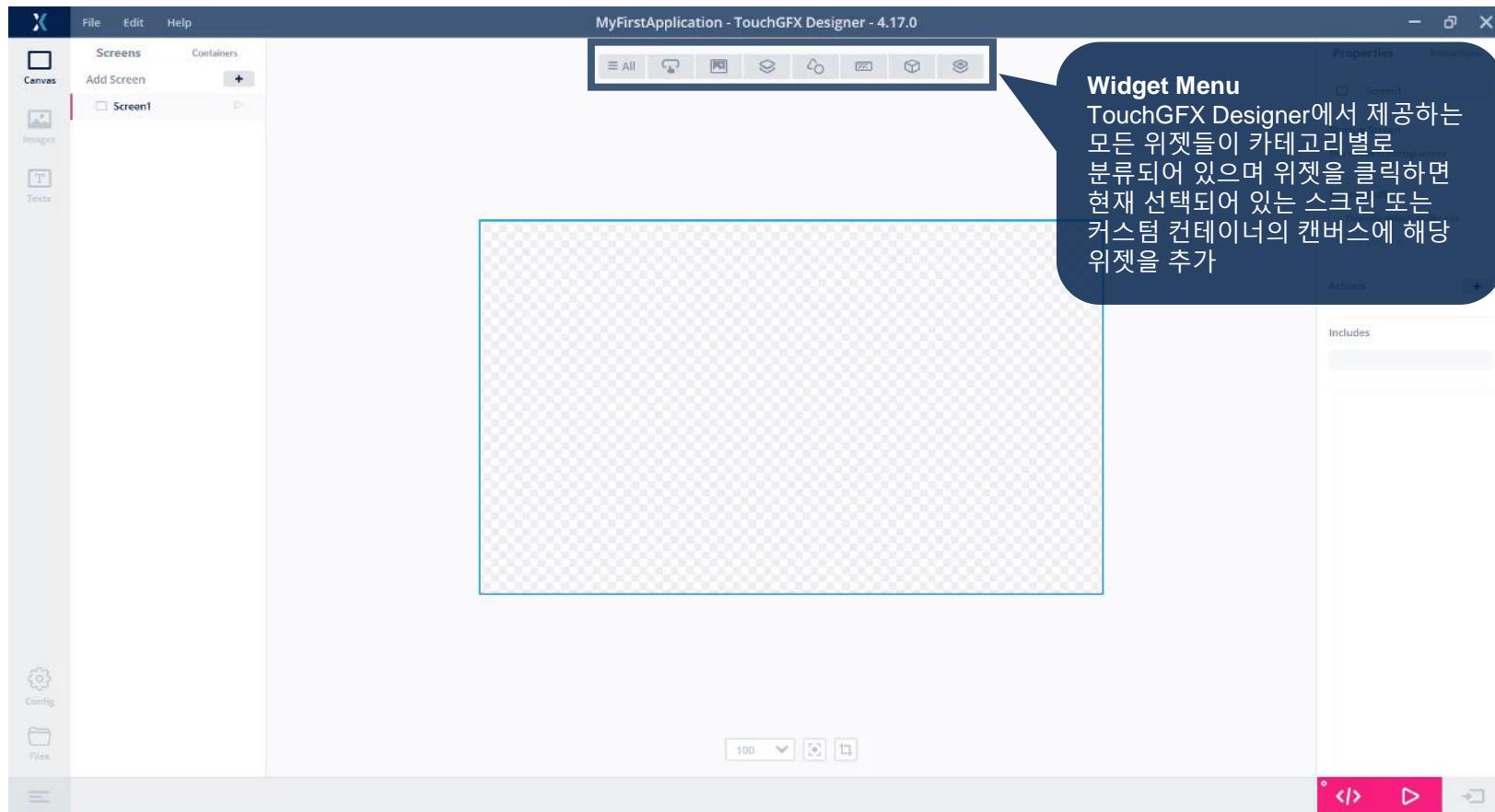
Canvas View

- 애플리케이션의 그래픽 요소를 개발하기 위해 사용하는 View로서 실행했을 때 보여지는 인터페이스를 시각화
- 시스템 요소간 상호작용과 같은 동적요소에 대한 설명 제공



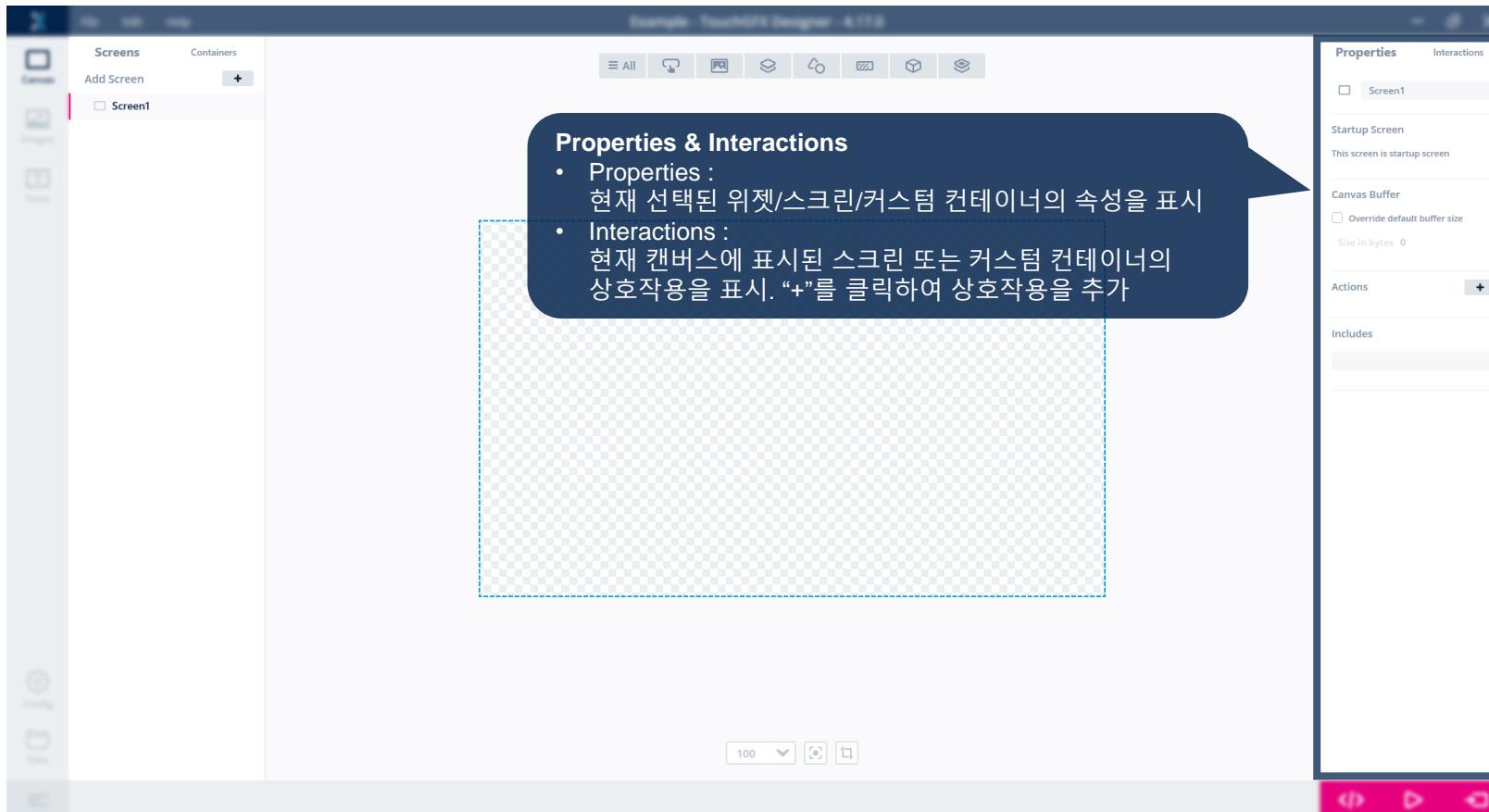
Canvas View

- 애플리케이션의 그래픽 요소를 개발하기 위해 사용하는 View로서 실행했을 때 보여지는 인터페이스를 시각화
- 시스템 요소간 상호작용과 같은 동적요소에 대한 설명 제공



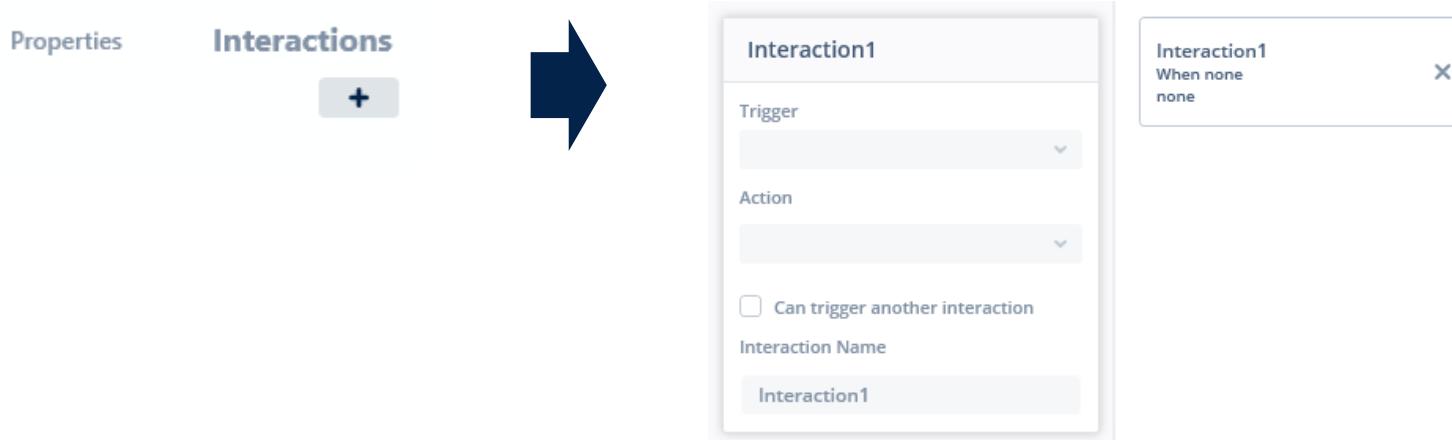
Canvas View

- 애플리케이션의 그래픽 요소를 개발하기 위해 사용하는 View로서 실행했을 때 보여지는 인터페이스를 시각화
- 시스템 요소간 상호작용과 같은 동적요소에 대한 설명 제공



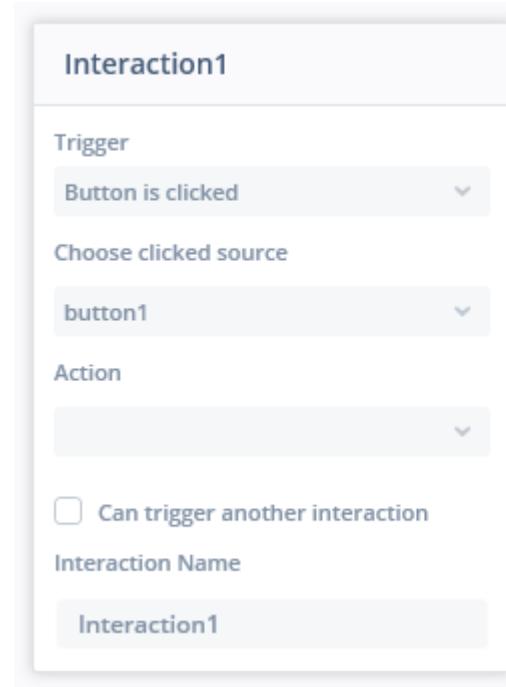
Interaction

- 상호작용에서는 트리거가 발생했을 때 어떠한 액션이 실행되도록 구성
 - 트리거(trigger)는 상호작용을 시작하도록 하는 이벤트로 애플리케이션에서 액션이 실행되기 위해서는 트리거가 먼저 발생해야 함
 - 액션(action)은 트리거가 발생한 후 일어나는 동작으로 트리거 조건이 충족되었을 때 애플리케이션에서 일어나는 상황을 결정
 - 스크린 또는 커스텀 컨테이너에서 Interactions 탭으로 이동 후 “+” 버튼을 클릭하여 상호작용 추가
 - 상호작용 세부 정보에서 상호작용 이름과 사용 가능한 구성 옵션 지정



Interaction

- 트리거
 - 트리거의 메뉴는 현재 스크린 또는 커스텀 컨테이너에 따라 다름.
 - Hardware button is clicked
 - Screen transition begins
 - Screen transition ends
 - Button is clicked

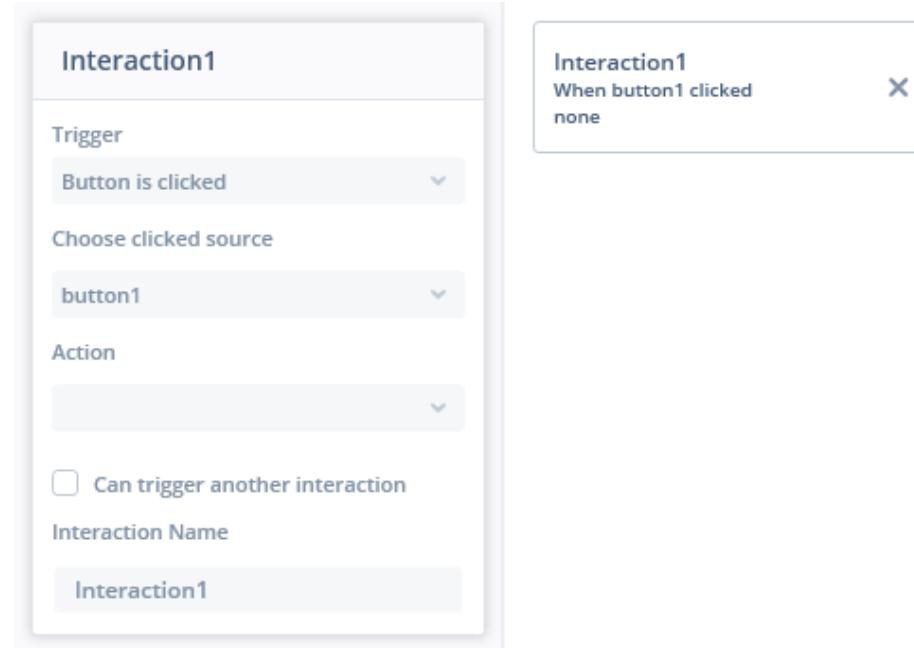


Interaction

- 액션

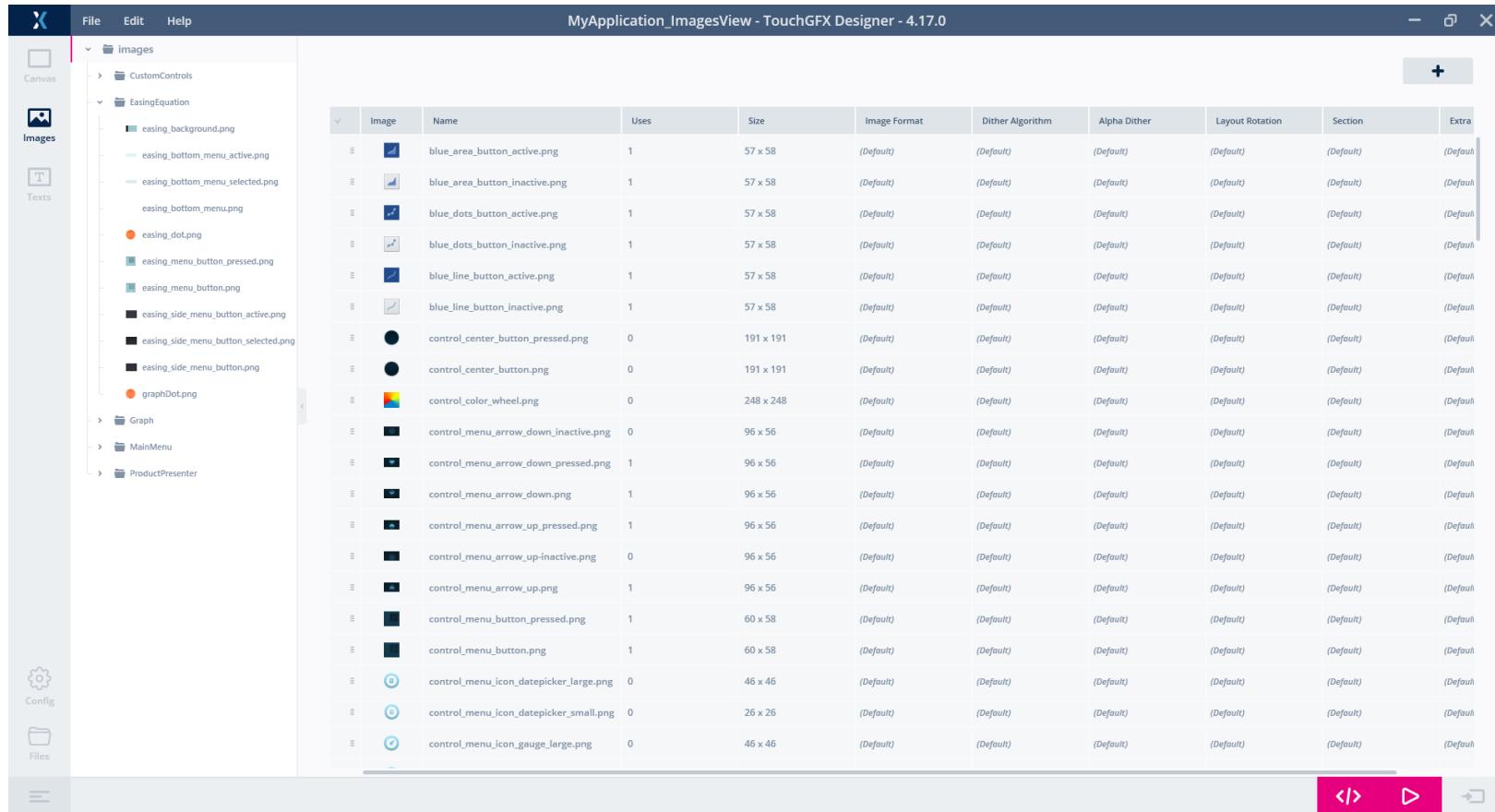
- 액션 메뉴는 현재 스크린 또는 커스텀 컨테이너에 따라 다름
 - Call new virtual function
 - Change screen
 - Execute C++ code
 - Set Language
 - Wait for
 - Move widget
 - Fade widget
 - Hide widget
 - Show widget

- “Can trigger another interaction”을 선택하는 경우 상호작용이 완료되었을 때 또 다른 상호작용의 트리거로 동작.



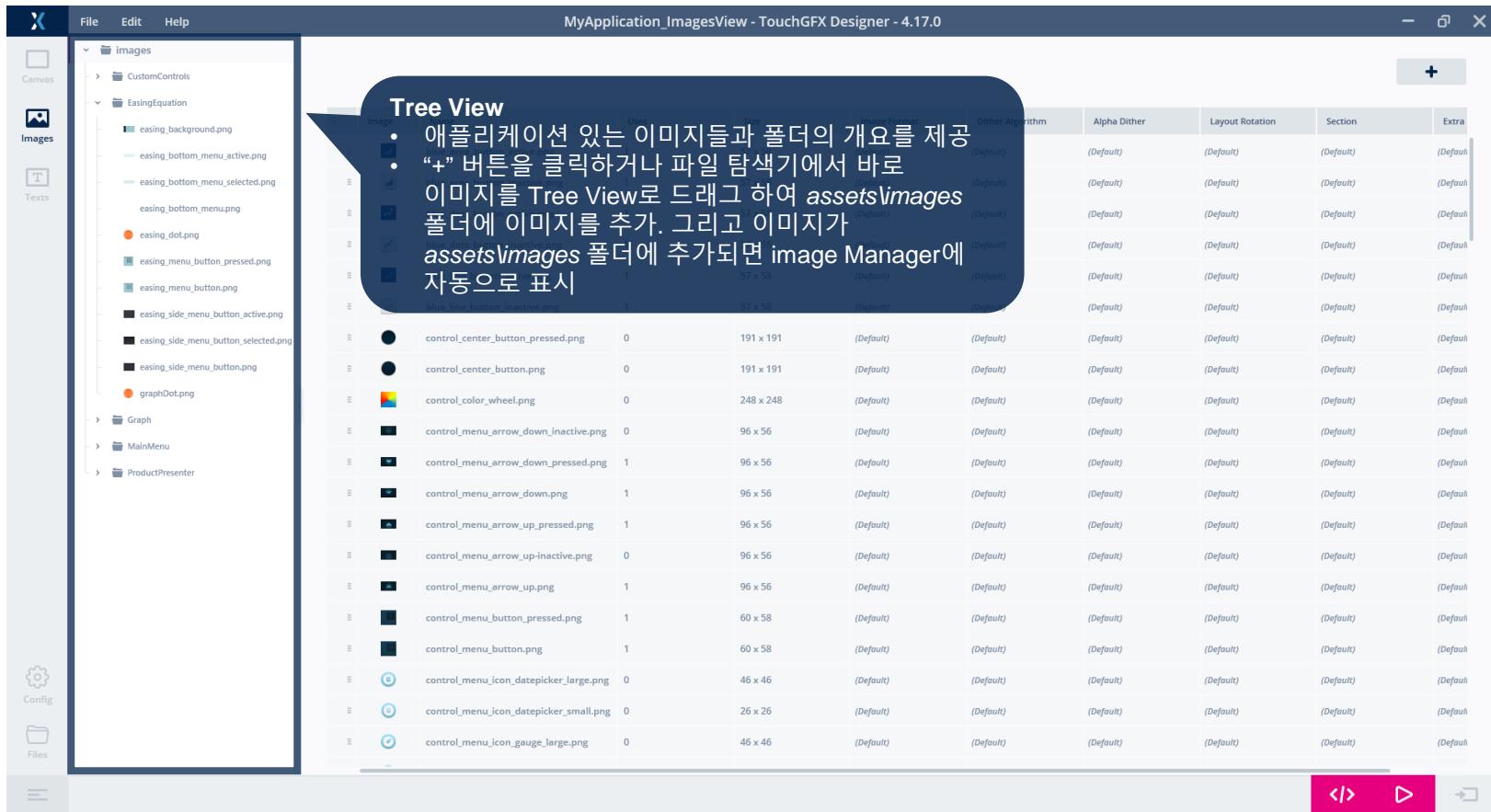
Images View

- TouchGFX 애플리케이션에 사용되는 이미지들("assets/images" 폴더에 위치)을 관리



Images View

- TouchGFX 애플리케이션에 사용되는 이미지들("assets\images" 폴더에 위치)을 관리



Images View

- TouchGFX 애플리케이션에 사용되는 이미지들("assets/images" 폴더에 위치)을 관리

Table View

저장된 이미지의 목록을 나열하고 이미지의 여러 속성들을 표시

- Image : 이미지 미리 보기
- Name : 폴더에 저장된 이미지의 이름
- Uses : 어플리케이션에서 이미지가 사용된 횟수
- Size : 이미지의 가로 X 세로
- Image Format : 이미지 형식
- Dither Algorithm : 이미지에 사용할 디더링 알고리즘 설정
 - 0 – No dither : 이미지에 디더링을 적용하지 않음
 - 1 – Floyd-Steinberg : 오차를 인접 픽셀까지 확산시켜 디더링이 세밀해지지만 선명도가 떨어짐
 - 2 – Javis, Judice & Ninke : Floyd-Steinberg에 비해 오차를 한 단계 더 먼 거리의 픽셀까지 확산시키므로 디더링은 거칠지만 더욱 선명한 이미지를 얻을 수 있음. 디더링 속도가 가장 느림
 - 3 – Stucki : 평균 오차를 최소화한 디더링에 기반을 두고 있지만 더욱 빠르고 선명함
- Alpha Dither : 알파 채널을 통한 디더링 알고리즘 사용 여부
- Layout Rotation : 디더링 이미지에 사용하는 알고리즘.
- Section : 대상 하드웨어에서 이미지가 저장되는 메모리 위치
- Extra Section : L8 이미지 형식을 사용할 경우 생상표를 다른 메모리 위치에 저장하도록 선택 가능

The screenshot shows the 'Images View' window in the TouchGFX Designer. The window title is 'MyApplication/ImagesView - TouchGFX Designer - 4.17.0'. The main area is a table view with the following columns: Image, Name, Uses, Size, Image Format, Dither Algorithm, Alpha Dither, Layout Rotation, Section, and Extra. The table lists numerous images, mostly related to control buttons, with their respective details like size (e.g., 57x58, 191x191, 248x248, 60x58, 46x46, 26x26 pixels), image format (Default), and dithering algorithms (Default). The 'Section' column indicates they are stored in the 'control_center' section. The 'Extra' column also shows 'Default' values. The interface includes standard window controls (minimize, maximize, close) and a toolbar at the bottom with icons for file operations.

Image	Name	Uses	Size	Image Format	Dither Algorithm	Alpha Dither	Layout Rotation	Section	Extra
blue_area_button_active.png	blue_area_button_active.png	1	57 x 58	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
blue_area_button_inactive.png	blue_area_button_inactive.png	1	57 x 58	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
blue_dots_button_active.png	blue_dots_button_active.png	1	57 x 58	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
blue_dots_button_inactive.png	blue_dots_button_inactive.png	1	57 x 58	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
blue_line_button_active.png	blue_line_button_active.png	1	57 x 58	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
blue_line_button_inactive.png	blue_line_button_inactive.png	1	57 x 58	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_center_button_pressed.png	control_center_button_pressed.png	0	191 x 191	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_center_button.png	control_center_button.png	0	191 x 191	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_color_wheel.png	control_color_wheel.png	0	248 x 248	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_arrow_down_inactive.png	control_menu_arrow_down_inactive.png	0	96 x 56	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_arrow_down_pressed.png	control_menu_arrow_down_pressed.png	1	96 x 56	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_arrow_down.png	control_menu_arrow_down.png	1	96 x 56	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_arrow_up_pressed.png	control_menu_arrow_up_pressed.png	1	96 x 56	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_arrow_up_inactive.png	control_menu_arrow_up_inactive.png	0	96 x 56	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_arrow_up.png	control_menu_arrow_up.png	1	96 x 56	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_button_pressed.png	control_menu_button_pressed.png	1	60 x 58	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_button.png	control_menu_button.png	1	60 x 58	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_icon_datepicker_large.png	control_menu_icon_datepicker_large.png	0	46 x 46	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_icon_datepicker_small.png	control_menu_icon_datepicker_small.png	0	26 x 26	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)
control_menu_icon_gauge_large.png	control_menu_icon_gauge_large.png	0	46 x 46	(Default)	(Default)	(Default)	(Default)	(Default)	(Default)

Image View

Texts View

- 프로젝트에서 텍스트, 번역 그리고 타이포그래피(typoography)를 정의하는데 사용
- **Texts Tab**과 **Typography Tab** 으로 구성

The screenshot shows the TouchGFX Designer interface with the 'Texts' tab selected. The main area displays a table of text entries grouped under 'Group1'. The columns are: Id, Typography, Alignment, GBR, DNK, and JPN. The table contains numerous entries, many of which are truncated. The left sidebar includes tabs for Canvas, Images, Texts (which is selected), Config, and Files.

ID	TYPOGRAPHY	ALIGNMENT	GBR	DNK	JPN
Demo_View_Mcu_Load_text	DemoView_McuLoad_Text	MCU load:	<value> %	MCU load:	<value> %
Demo_View_Mcu_Load_value	DemoView_McuLoad_Text	MCU load:	<value> %	MCU load:	<value> %
Demo_View_Mcu_Load_text_small	DemoView_McuLoad_Text...	MCU load:	MCU load:	MCU load:	MCU load:
Demo_View_Mcu_Load_value_small	DemoView_McuLoad_Tex...	MCU load:	<value> %	<value> %	<value> %
Poster_Headline_00	PosterHeadline	About TouchGFX	Om TouchGFX	TouchGFXについて	Cortex-Mのような低成本のハード
Poster_Text_00	PosterText	A unique software framework that	Et unikt softwareframework som muliggør	Alpha Blending	アルファブレンド
Poster_Headline_01	PosterHeadline	Alpha Blending	Alpha Blending	Alpha blending er den proces, hvor et bil...	アルファブレンドとは、画像と背景
Poster_Text_01	PosterText	Alpha blending is the process of	Alpha blending er den proces, hvor et bil...	Fonts and Languages	フォントおよび言語
Poster_Headline_02	PosterHeadline	Fonts and Languages	Fonte og Sprog	TouchGFX supports multiple languages	TouchGFXは多言語仕様となっており、
Poster_Text_02	PosterText	TouchGFX supports multiple languages	TouchGFX understøtter adskillige sprog	Press to Scan	<value> %
Three_Way_Progress_Bar_Percentag	ThreeWayProgressBarPer...	Press to Scan	N/A	N/A	<value> %
Three_Way_Progress_Bar_scan	ThreeWayProgressBarBut...	N/A	N/A	Scanning...	<value> %
Three_Way_Progress_Bar_scanning	ThreeWayProgressBarBut...	Scanning...	N/A	Scan completed	<value> %
Three_Way_Progress_bar_status	CustomControlsStatus	Scan completed	N/A	Percentage Bar	<value>%
Percentage_Bar_readout	PercentageBarReadout	Percentage Bar	N/A	Datepicker_Months	<value>%
Percentage_Bar_headline	PercentageBarHeadline	DatePicker	N/A	Datepicker_Day_28	N/A
Datepicker_Months	DatePicker	January	N/A	Datepicker_Day_29	N/A

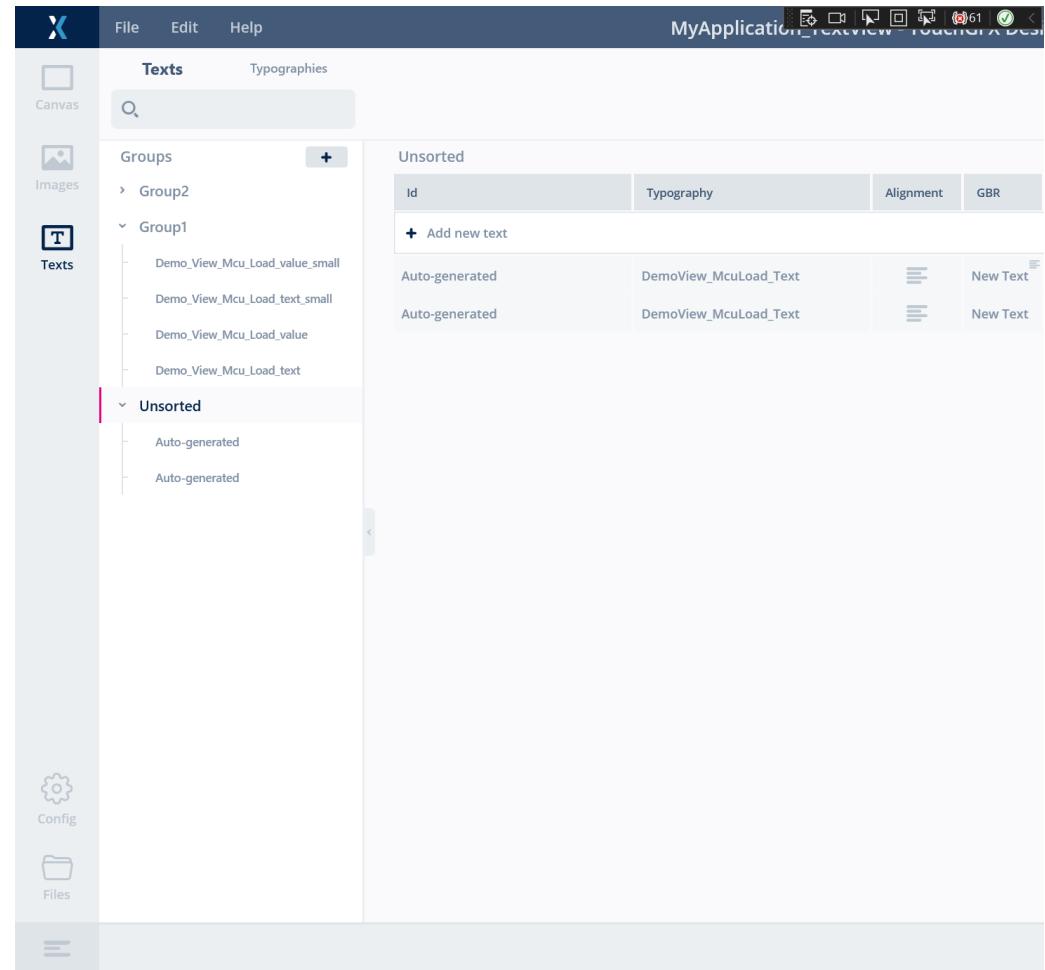
Texts View

- Auto-generated ID

- 자동 생성된 ID를 사용하는 텍스트들은 단 한 번만 사용되며 한 번에 둘 이상의 위젯이나 작업에 사용 불가
- 쉽게 ID 이름을 바꿀 수 있어 여러 위젯 및 작업에서 텍스트를 사용 가능

- Defined ID

- 정의된 ID를 사용하는 텍스트들은 TouchGFX Designer에서 여러 위젯 및 작업에서 재사용 가능
- 정의된 ID로 새 텍스트를 추가하기 위해서는 텍스트 탭에서 “ADD NEW TEXT” 버튼을 클릭하거나 자동 생성된 ID를 사용하는 텍스트의 ID 이름을 변경



Auto-generated texts

Texts View

- **Texts Tab**

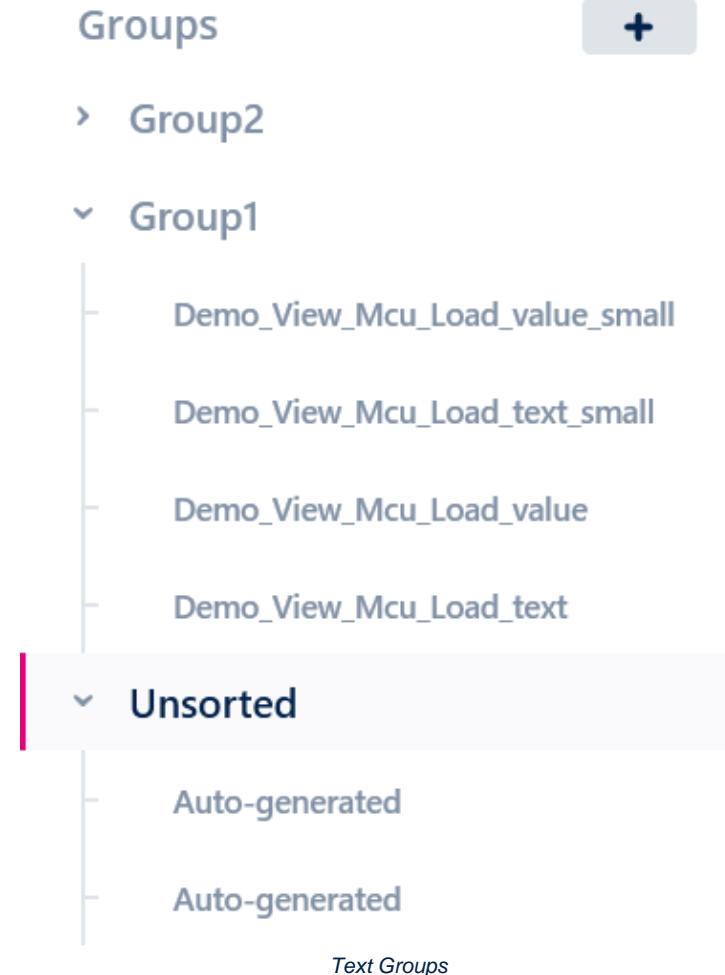
Texts 탭은 모든 텍스트 요소들을 개략적으로 설명

The screenshot shows the TouchGFX Designer application window titled "MyApplication_Textview_TouchGFX_Designer - 4.19.0". The "Texts" tab is selected in the sidebar. The main area displays a table of text entries grouped under "Group1".

ID	Typography	Alignment	GBR	DNK	JPN
Demo_View_Mcu_Load_text	DemoView_McuLoad_Text	MCU load:	<value> %	MCU load:	<value> %
Demo_View_Mcu_Load_value	DemoView_McuLoad_Text	MCU load:	<value> %	MCU load:	<value> %
Demo_View_Mcu_Load_text_small	DemoView_McuLoad_Text...	MCU load:	MCU load:	MCU load:	MCU load:
Demo_View_Mcu_Load_value_small	DemoView_McuLoad_Text...	MCU load:	<value> %	MCU load:	<value> %
Poster_Headline_00	PosterHeadline	About TouchGFX	Om TouchGFX	TouchGFXについて	Cortex-Mのような低成本のハード
Poster_Text_00	PosterText	A unique software framework that	Et unik softwareframework som muliggør	アルファブレンド	アルファブレンド
Poster_Headline_01	PosterHeadline	Alpha Blending	Alpha Blending	Alpha blending er den proces, hvor et bil...	アルファブレンドとは、画像と背景
Poster_Text_01	PosterText	Alpha blending is the process of	Alpha blending er den proces, hvor et bil...	Fonte og Sprog	フォントおよび言語
Poster_Headline_02	PosterHeadline	Fonts and Languages	Fonte og Sprog	TouchGFX supports multiple languages	TouchGFXは多言語仕様となっており、
Poster_Text_02	PosterText	TouchGFX supports multiple languages	TouchGFX understøtter adskillige sprog	N/A	N/A
Three_Way_Progress_Bar_Percentage	ThreeWayProgressBarPer...	<value> %	<value> %	Press to Scan	N/A
Three_Way_Progress_Bar_scan	ThreeWayProgressBarBut...	Scanning...	N/A	Scan completed	N/A
Three_Way_Progress_Bar_scanning	ThreeWayProgressBarBut...	<value> %	<value> %	Scan completed	N/A
Three_Way_Progress_bar_status	CustomControlsStatus	Scan completed	N/A	Scanning...	N/A
Percentage_Bar_readout	PercentageBarReadout	<value>%	<value>%	Press to Scan	N/A
Percentage_Bar_headline	PercentageBarHeadline	Percentage Bar	N/A	Scan completed	N/A
Datepicker_Months	DatePicker	January	N/A	Scanning...	N/A
Datepicker_Years					
Datepicker_Day_28					
Datepicker_Day_29					

Texts View

- Text Tab
 - Search
 - 특정 텍스트 요소 또는 번역을 검색
 - Tree view
 - 그룹 및 텍스트 요소에 대한 개요를 제공
 - Groups
 - 프로젝트에 포함된 모든 텍스트들을 하나 이상의 그룹으로 묶어서 관리 가능
 - Tree view에서 텍스트를 드래그하거나 grid에서 텍스트를 마우스 우(right) 클릭하여 특정 그룹으로 추가

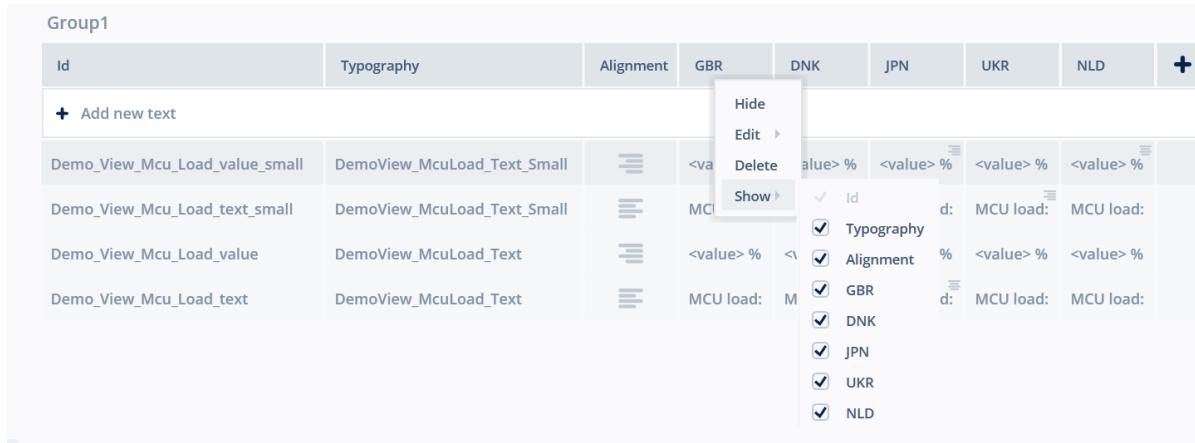


Texts View

- Text Tab

- Header context menu

- Grid에 header를 마우스 우 클릭하면 header context menu가 표시
 - Column을 숨김, 편집, 삭제 그리고 표시

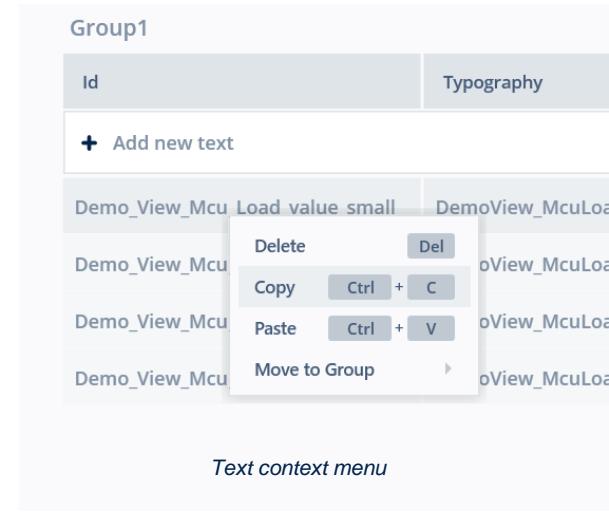


Header context menu

- Text Tab

- Grid context menu

- Grid에 텍스트를 마우스 우 클릭하면 context menu가 표시
 - 텍스트를 삭제, 복사 그리고 다른 그룹으로 이동



Text context menu

Texts View

- Text Tab

- 번역(Translations)

- Typography : 텍스트와 모든 번역에서 사용해야 하는 타이포그래피를 지정
 - Alignment : 텍스트와 모든 번역에서 사용해야 하는 가로 정렬을 지정

- 번역 세부 설정

- 각 번역은 기본 정렬을 덮어 쓰기 가능
 - 더블 클릭을 통해 정렬 선택기를 표시

- 언어 추가

- Header menu에서 “+” 버튼을 클릭
 - 언어 추가 팝업이 나타나고 언어 이름과 초기값을 지정

Group1				
Id	Typography	Alignment	GBR	+
+ Add new text				
Demo_View_Mcu_Load_text	DemoView_McuLoad_Text		MCU load:	
Demo_View_Mcu_Load_value	DemoView_McuLoad_Text		<value> %	

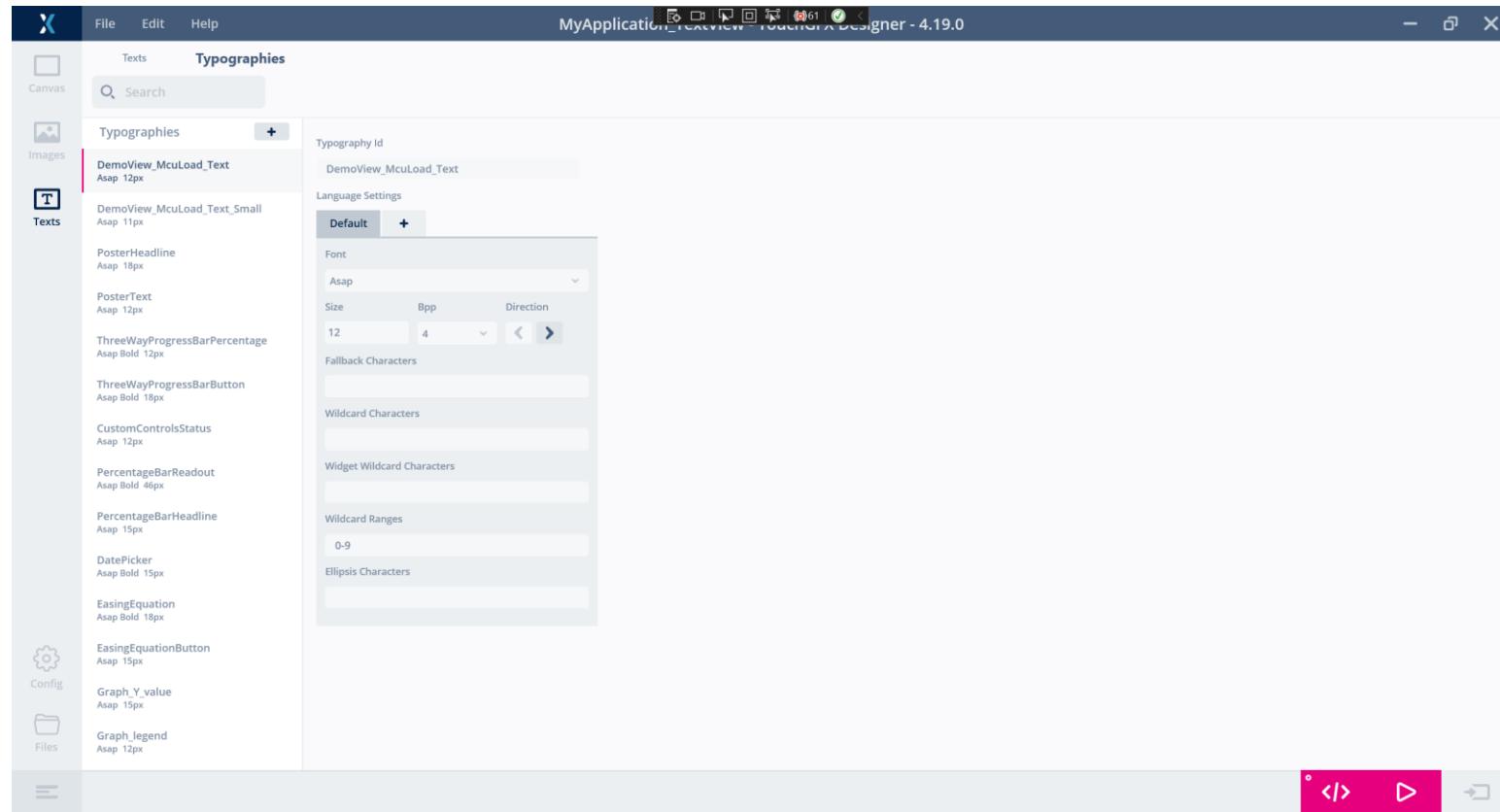
How to set Translation Specific Alignment

Group1								Add Language	
Id	Typography	Alignment	GBR	DNK	JPN	UKR	NLD	+	
+ Add new text									
Demo_View_Mcu_Load_value_small	DemoView_McuLoad_Text_Small		<value> %						
Demo_View_Mcu_Load_text_small	DemoView_McuLoad_Text_Small		MCU load:						
Demo_View_Mcu_Load_value	DemoView_McuLoad_Text		<value> %						
Demo_View_Mcu_Load_text	DemoView_McuLoad_Text		MCU load:						

Add New Language popup

Texts View

- **Typographies Tab**
 - 타이포그래피 탭은 프로젝트의 모든 타이포그래피를 나열하고 설정



Texts View

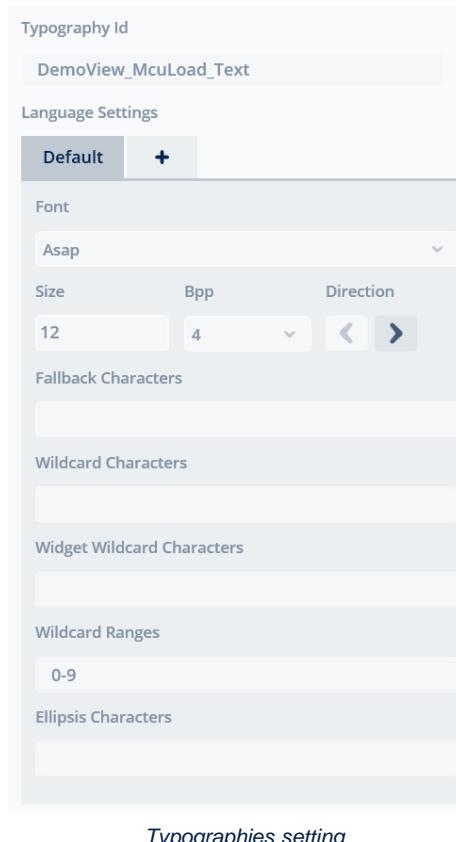
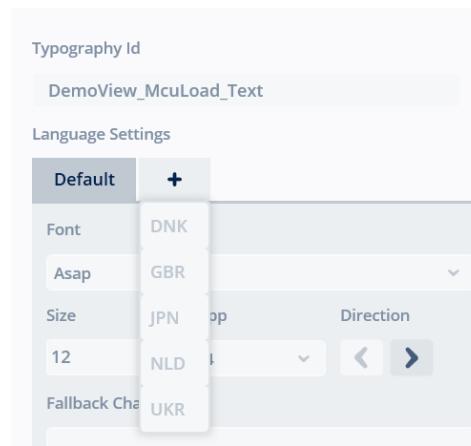
- **Typographies Tab**

- **Typography ID**

- 타이포그래피의 이름

- **Language Settings**

- 타이포그래피의 언어 설정을 포함.
 - “+” 버튼을 클릭하여 원하는 언어를 선택.



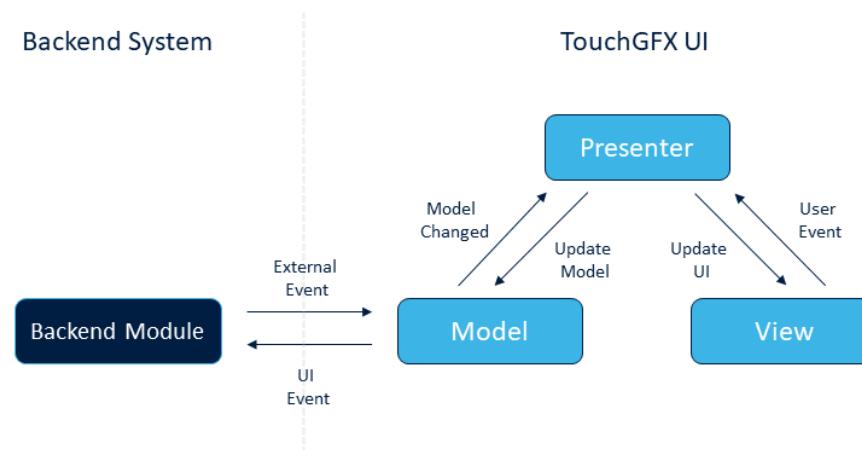
- **Font**
타이포그래피에 사용하기 위한 폰트
- **Size**
타이포그래피의 폰트 크기
- **Bpp**
폰트를 나타내기 위한 픽셀당 사용되는 비트 수. 1, 2, 4, 8 bpp
- **Direction**
텍스트의 방향 지정. LTR(default) 및 RTL
- **Fallback Character**
글리프(glyph)를 사용할 수 없는 경우 이 열에 지정된 문자를 사용.
- **Wildcard Characters**
TouchGFX 애플리케이션에서 표시할 수 있는 문자 (예: "0123456789-")
- **Widget Wildcard Characters**
와일드카드가 필요한 위젯에서 추가하는 문자
(예: digital clock 위젯은 "0123456789:AM")
- **Wildcard Ranges**
Wildcard Characters와 유사하지만 범위 지정이 가능 (예: "0-9, A-F")
- **Ellipsis Character**
텍스트 영역에서 긴 문자를 자르는데 사용

TouchGFX Engine

Software Architecture

- Model-View-Presenter

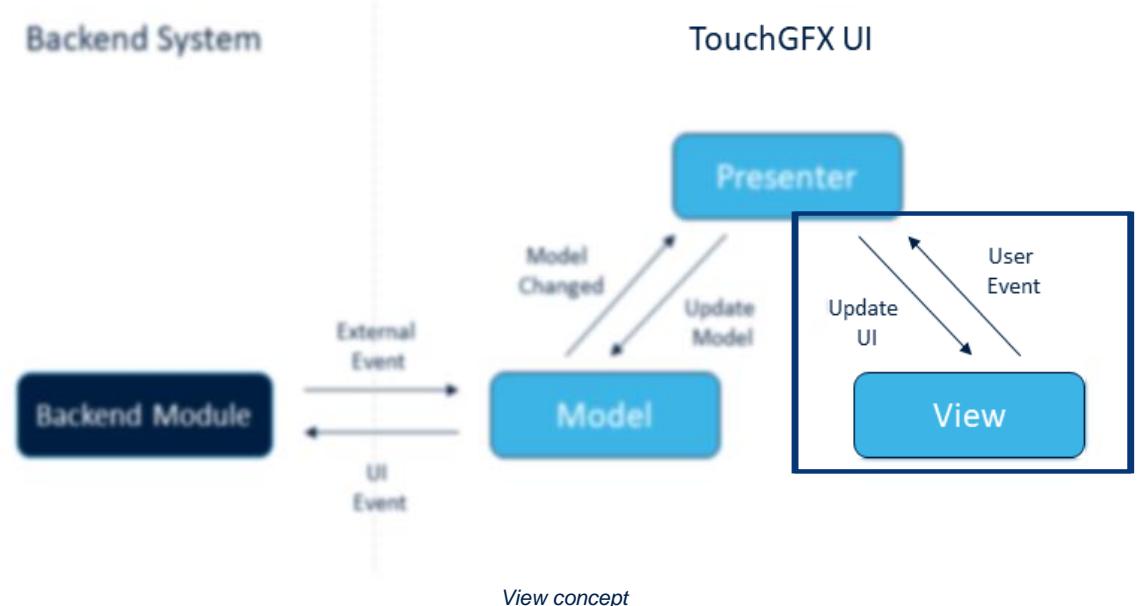
- Model-View-Presenter(MVP) 패턴은 Model-View-Controller(MVC) 패턴에서 파생된 아키텍처 패턴으로 두 패턴 모두 사용자 인터페이스 애플리케이션을 구축할 때 널리 사용되는 패턴으로 TouchGFX는 MVP 패턴을 준수.
 - 관심 영역 분리 :
코드를 각자의 책임을 가진 별개의 영역으로 분할. 코드가 더욱 단순해지므로 재사용 및 유지 관리 용이
 - 유닛 테스트 :
UI 로직(Presenter)이 시각적 계측(View)과 분리되어 있어 쉽게 단독으로 테스트하는 것이 가능



Software Architecture

- View

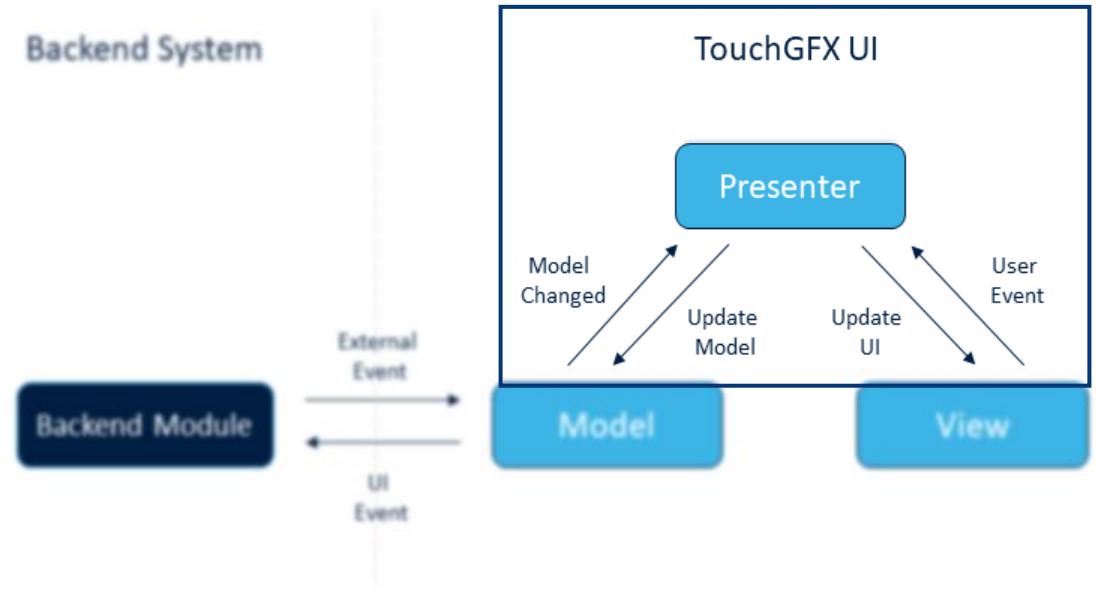
- 전달받은 데이터를 표시하고 해당 그 데이터에 따라 실행할 Presenter에게 사용자 명령(이벤트)을 전송하는 수동적 인터페이스
 - UI 구성 요소들의 생성 및 처리
 - 사용자 입력 수신
 - 사용자 입력을 Presenter에게 전송
 - Presenter로부터 받은 데이터를 기반으로 UI 업데이트



Software Architecture

- **Presenter**

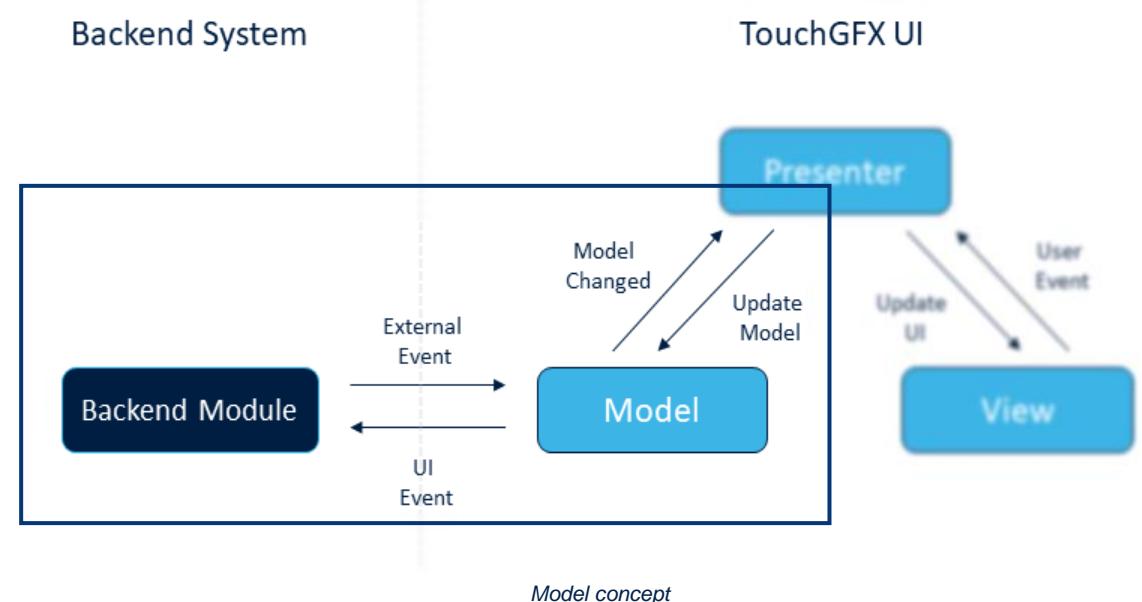
- Model과 View에 따라 실행
- 저장소(Model)에서 데이터를 가져온 후 View에 표시할 수 있도록 형식을 지정
- Model과 View 사이에서 중재자 역할
 - Model로부터 데이터를 받아 View에 전송할 준비
 - View로부터 사용자 입력을 받아 Model로 전송
- UI를 위한 비즈니스 로직
- UI 구현에 관한 지식을 필요로 하지 않음.



Software Architecture

- Model

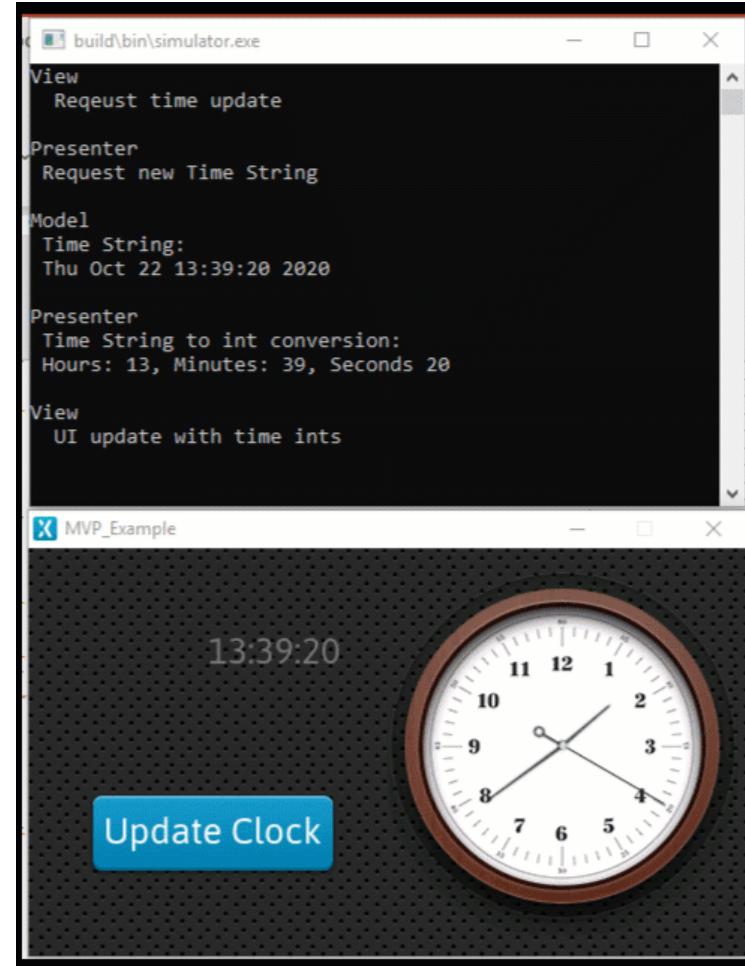
- 사용자 인터페이스에 표시하거나 그 밖에 실행할 데이터를 정의하는 인터페이스
 - Backend 와 통신
 - 데이터 저장
 - Presenter에 데이터를 전달



Software Architecture

- MVP – 예제

1. View에 “Update Clock” 버튼을 클릭 했을 때
현재 시간 요청(이벤트)이 Presenter를 통해
Model에 전송
2. Model은 현재 시간 데이터를 Presenter로 전송
3. Presenter는 View에 전달되어 Clock 위젯이
사용할 수 있도록 데이터를 변환
4. View는 새로운 현재 시간 데이터를 Clock
위젯에 업데이트

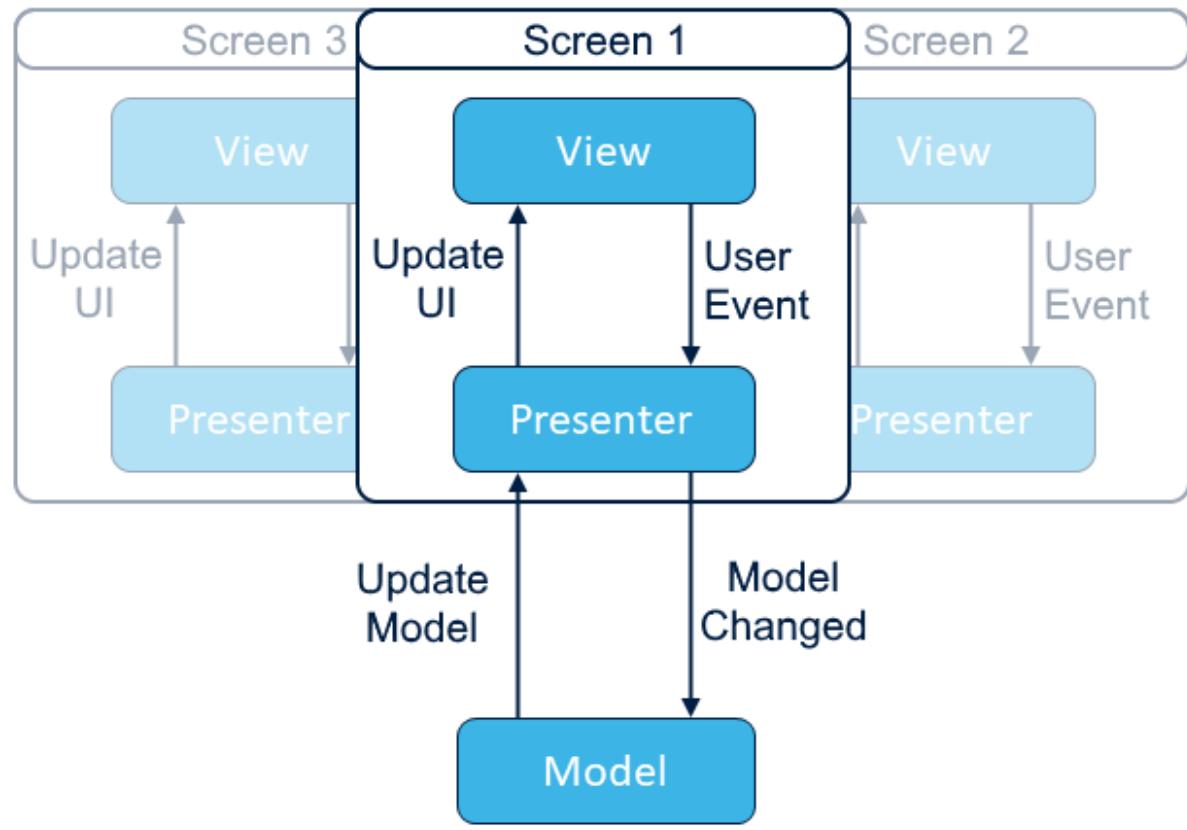


Example of Clock widget on MVP

Software Architecture

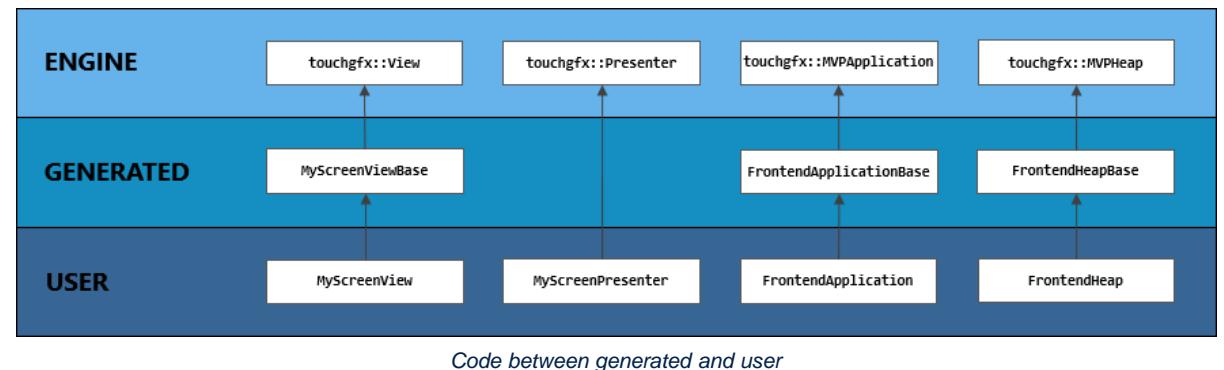
- Screen 개념

- 애플리케이션은 하나 이상의 “Screen”들로 구성
- UI 및 UI 로직의 논리적 그룹화
- 하나의 View class와 하나의 Presenter class
- 한 번에 하나의 Screen만 활성화
(RAM 메모리 사용량 최소화)
- 하나의 프로젝트에는 하나의 Model만 존재
- Model은 항상 활성화 상태를 유지



Software Architecture

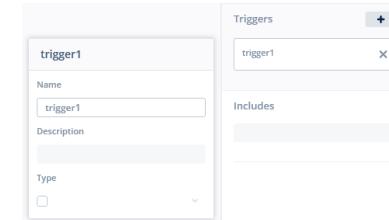
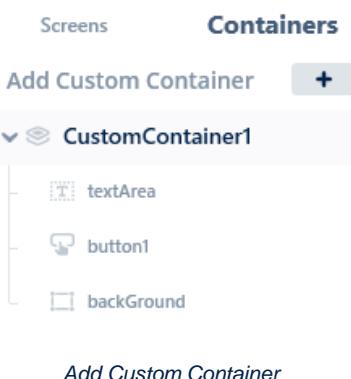
- Generated Code vs. User Code
 - Engine
 - TouchGFX에 의해 제공
 - Generated
 - Designer에 의해 여러 번 생성
 - Designer에서 추가된 위젯을 위한 코드
 - 빈번한 재생성
 - User
 - Designer에 의해 한번만 생성
 - 사용자 작업용 코드로 생성



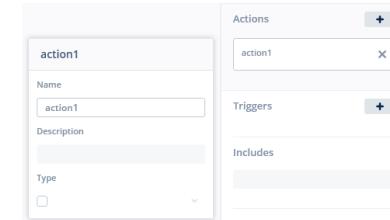
TouchGFX Engine Features

• Custom Container

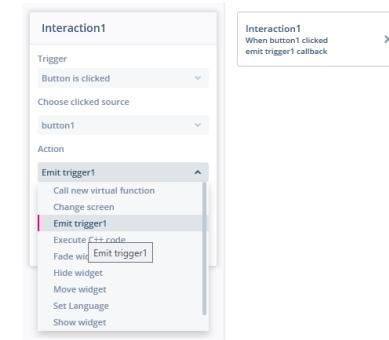
- TouchGFX의 표준 위젯에 없는 위젯 생성이 필요할 때 사용
- 여러 기존 위젯을 묶어 하나의 위젯으로 생성
- 그리기 성능 향상
- 복합 커스텀 컨테이너 생성 가능
- 커스텀 트리거(Callback) 및 커스텀 액션(Method)를 정의



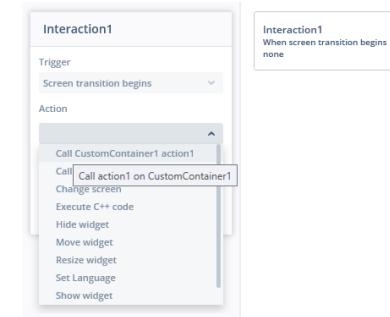
Custom Container의 Properties 탭에서 Custom Trigger를 추가



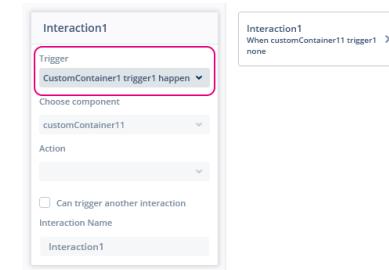
Custom Container의 Properties 탭에서 Custom Action을 추가



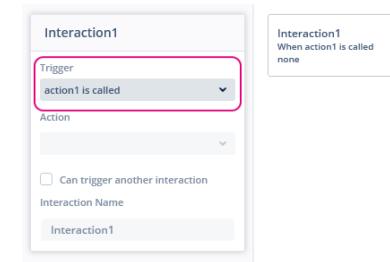
상호작용에 보내는 Custom Trigger의 파라미터를 지정



상호작용에서 실행되는 Custom Action의 파라미터를 지정



Screen의 상호작용에서 트리거로 Custom Trigger를 선택



Custom Container의 상호작용에서 트리거로 Custom Action을 선택 혹은 Custom Container를 포함하는 Screen의 상호작용에서 액션으로 Custom Action 선택 가능

TouchGFX Engine Features

• Caching Bitmaps

- 애플리케이션에서 비트맵을 저장(또는 캐시)할 때 사용하는 전용 버퍼
- 내장 RAM에 비트맵 데이터를 캐시 한 후 렌더링하므로 전반적인 UI 성능 향상
- 속도가 느린 외장 장치에 비트맵을 저장해야 하는 경우 유용
 - 포인터를 통한 직접 액세스가 불가능한 Non-memory mapped flash(예:SD-card, USB disk...)
- BlockCopy
 - 플래시 메모리에서 캐시로 데이터를 복사
 - Non-memory mapped 플래시 메모리와 같은 주소 지정이 불가능한 플래시에 저장하는 경우 인터페이스 구현 필수
 - [Using Non-Memory Mapped flash for storing images](#) 참조

• 비트맵 캐시 구성

- 외장 메모리의 3MB를 비트맵 캐시로 지정

FrontendApplication.cpp (extract)

```
#include <gui/common/FrontendApplication.hpp>
#include <touchgfx/Bitmap.hpp>

FrontendApplication::FrontendApplication(Model& m, FrontendHeap& heap)
    : FrontendApplicationBase(m, heap)
{
    // Place cache start address in SDRAM at address 0xC0008000;
    uint16_t* const cacheStartAddr = (uint16_t*)0xC0008000;
    const uint32_t cacheSize = 0x300000; //3 MB, as example
    Bitmap::setCache(cacheStartAddr, cacheSize);
}
```

- 배열을 선언 하여 배열의 주소와 크기를 캐시로 지정

FrontendApplication.cpp (extract)

```
#include <gui/common/FrontendApplication.hpp>
#include <touchgfx/Bitmap.hpp>

// Define an array for the bitmap cache
uint16_t cache[128*1024]; //256 KB cache

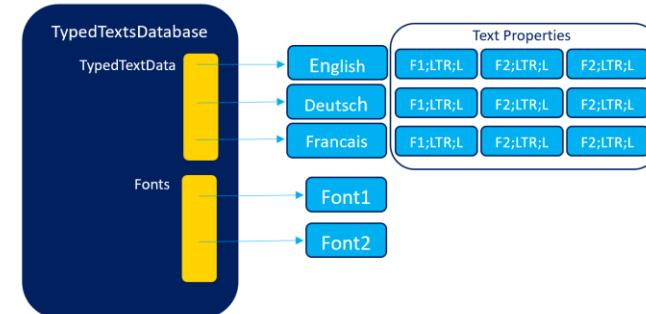
FrontendApplication::FrontendApplication(Model& m, FrontendHeap& heap)
    : FrontendApplicationBase(m, heap)
{
    Bitmap::setCache(cache, sizeof(cache));
}
```

TouchGFX Engine Features

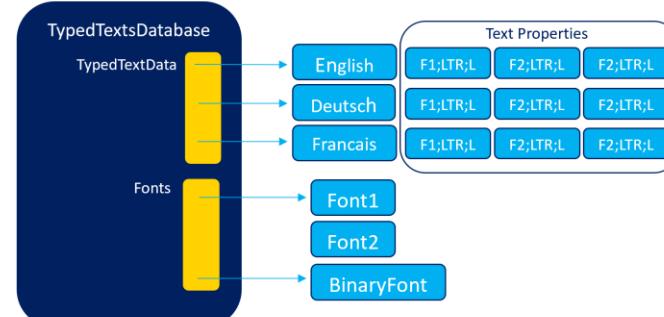
- **Binary Fonts**

- 폰트 정보를 컴파일하여 애플리케이션에 연결하는 기존 방식을 대체하는 것을 목표
- 작은 사이즈의 애플리케이션 바이너리에 다양한 폰트 집합을 제공
- 전체 바이너리 폰트가 RAM 또는 memory-mapped flash에 로딩되어야 하기 때문에 메모리 용량에 주의

- **TypedTextDatabase class**



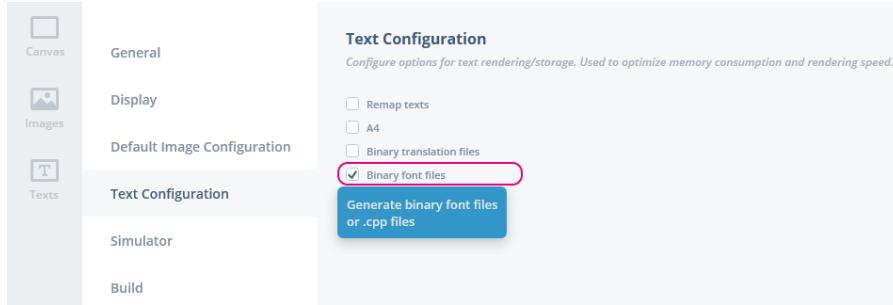
- **바이너리 폰트 사용**



TouchGFX Engine Features

- ## Binary Fonts

- Designer에서 바이너리 폰트 생성 설정
 - [Config] → [Text Configuration]



- 코드를 다시 생성했을 때, generated/fonts/bin/ 폴더에 바이너리 폰트가 생성

- 바이너리 폰트 설치
 - 파일 시스템에서 내부 RAM으로 바이너리 폰트를 로딩하는 예

```
FrontendApplication.cpp

//read the file into this array in internal RAM
uint8_t fontdata[10000];

//binary font object using the data
BinaryFont bf;

FrontendApplication::FrontendApplication(Model& m, FrontendHeap& heap)
    : FrontendApplicationBase(m, heap)
{
    //read the binary font from a file
    FILE* font = fopen("generated/fonts/bin/Font_verdana_20_4bpp.bin", "rb");
    if (font)
    {
        //read data from the file
        fread(fontdata, 1, 10000, font);
        fclose(font);

        //initialize BinaryFont object in bf using placement new
        new (&bf) BinaryFont((const struct touchgfx::BinaryFontData*)fontdata);

        //replace application font 'DEFAULT' with the binary font
        TypedTextDatabase::setFont(DEFAULT, &bf); //verdana_20_4bpp
    }
}
```

TouchGFX Engine Features

- Backend Communication

- UI와 하드웨어 주변 장치 간 통신
- Model class
 - UI 상태 정보 저장
 - 주변 시스템과 연결되는 인터페이스
 - 매 프레임마다 호출되는 tick() 함수

- 시스템 연결

- GUI Task 샘플링 :
Model::tick 함수에서 직접 주변 시스템을 샘플링
- Secondary Task 샘플링 :
샘플링을 담당하는 OS task 생성. Mailbox 또는 Message queue를 통해 GUI Task에 데이터 전송

- GUI 데이터 전송 예

Model.cpp

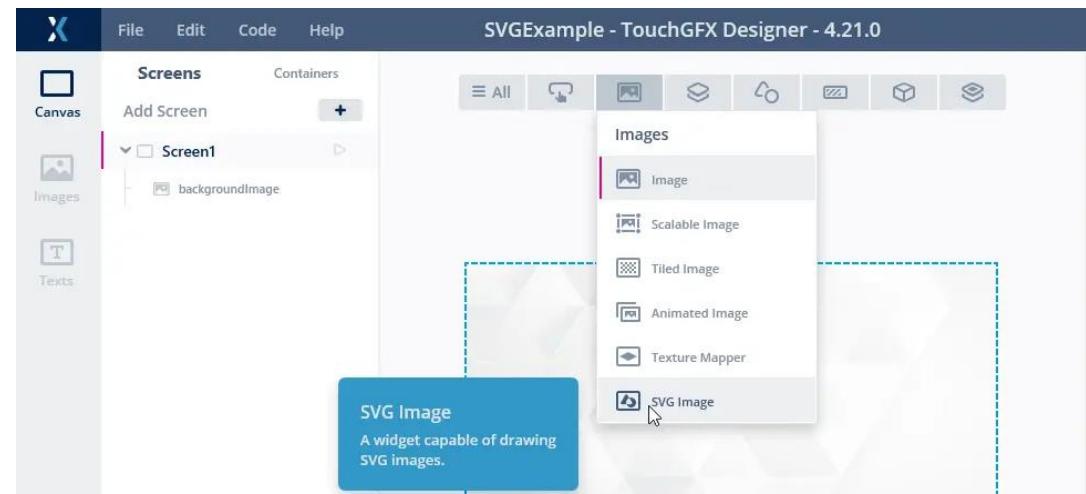
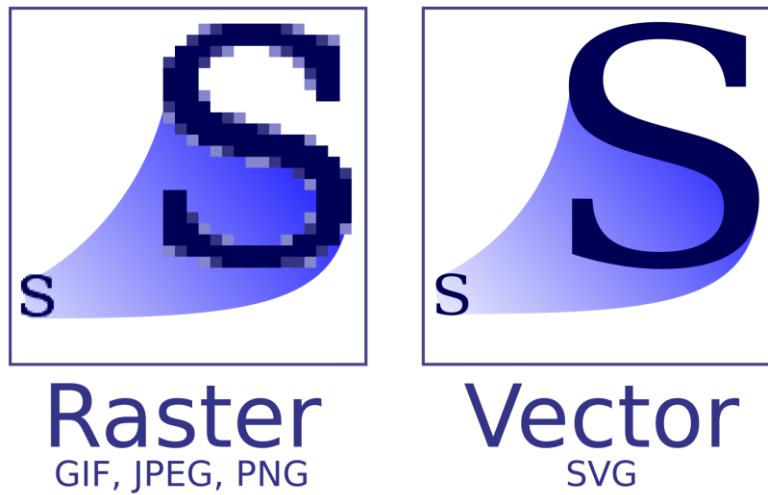
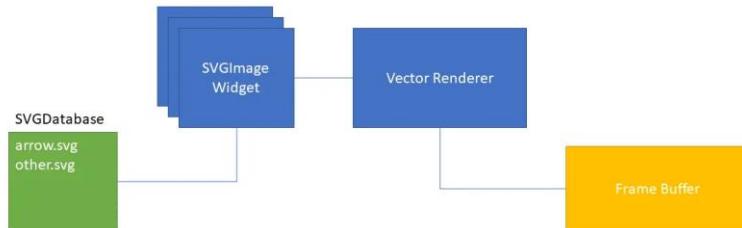
```
void Model::tick()
{
    // Pseudo-code for sampling data
    if (OS_Poll(GuiTaskMBox))
    {
        // Here we assume that you have defined a "Message" struct containing type and data,
        // along with some event definitions.
        struct Message msg = OS_Read(GuiTaskMBox);
        if (msg.eventType == EVT_TEMP_CHANGED)
        {
            // We received information that temperature has changed.
            // First, update Model state variable
            currentTemperature = msg.data;

            // Second, notify the currently active Presenter that temperature has changed.
            // The modelListener pointer points to the currently active Presenter.
            if (modelListener != 0)
            {
                modelListener->notifyTemperatureChanged(currentTemperature);
            }
        }
    }
}
```

Send data to UI via Message queue

TouchGFX Engine Features

- **SVG (Scalable Vector Graphics)**
 - 2D 벡터 그래픽을 표현하기 위한 XML 기반의 파일 형식
 - 품질의 저하 없이 이미지 확대 및 축소 가능
 - Vector 기반의 이미지 포맷
 - 메모리 사용량 감소
- **SVG 위젯**
 - `SVGImage` 위젯은 `VectorRenderer`를 사용해 렌더링을 수행



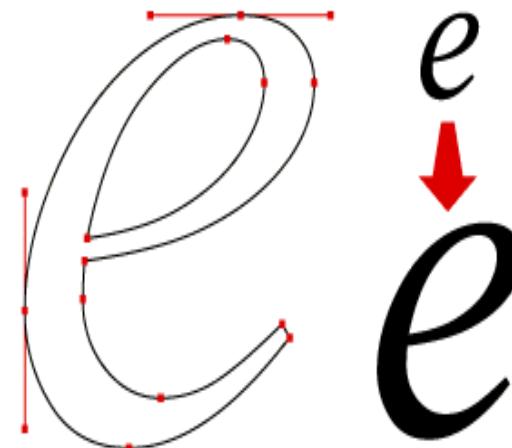
TouchGFX Engine Features

- Vector 기반 그래픽의 이점
 - 메모리 사용량의 감소

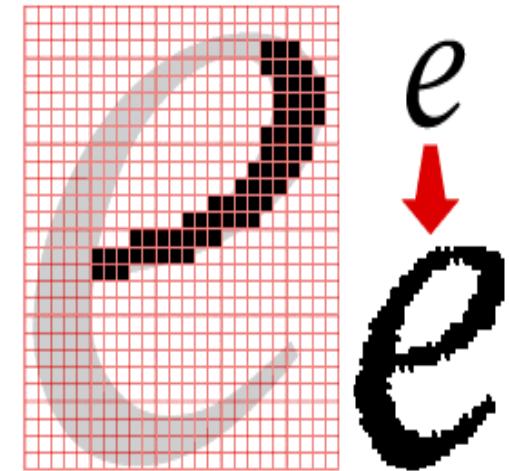
Image	Bitmap 200x200 RGB888	SVG	Memory saving
	120KBytes	4.2Kbytes	99.97%
	120Kbytes	350 bytes	99.99%

- 높은 품질의 zoom 및 rotation 지원

VECTOR GRAPHICS



BITMAPPED (RASTER) GRAPHICS



TouchGFX Engine Features

• Image Compression

- L8 이미지에 가장 적합한 압축 형식을 찾아 렌더링 성능 저하를 제한하면서 메모리 사용량을 감소
- 사용 가능한 압축 형식:
 - L4: 4bit로 인코딩, 최대 16가지 색상 지원
 - RLE (Run-length encoding):
픽셀의 실행 길이를 인코딩, 최대 64 가지 생상 지원
 - LZW9 (Lempel-Ziv-Welch Compression Encoding):
Dictionary 기반으로 인코딩, 최대 256 가지 생상 지원
- L8에 추가로 선택된 압축은 이미지 메모리 크기를 20%에서 99%까지 줄일 수 있다

Image	Bitmap size	Number of colors	Memory saving
	10 Kbytes (50x50)	32	97.41% (259 bytes)
	25.6 Kbytes (80x80)	193	93.26% (1.7 Kbytes)
	60 Kbytes (150x200)	128	74.46% (15.3 Kbytes)

Partial Frame Buffer

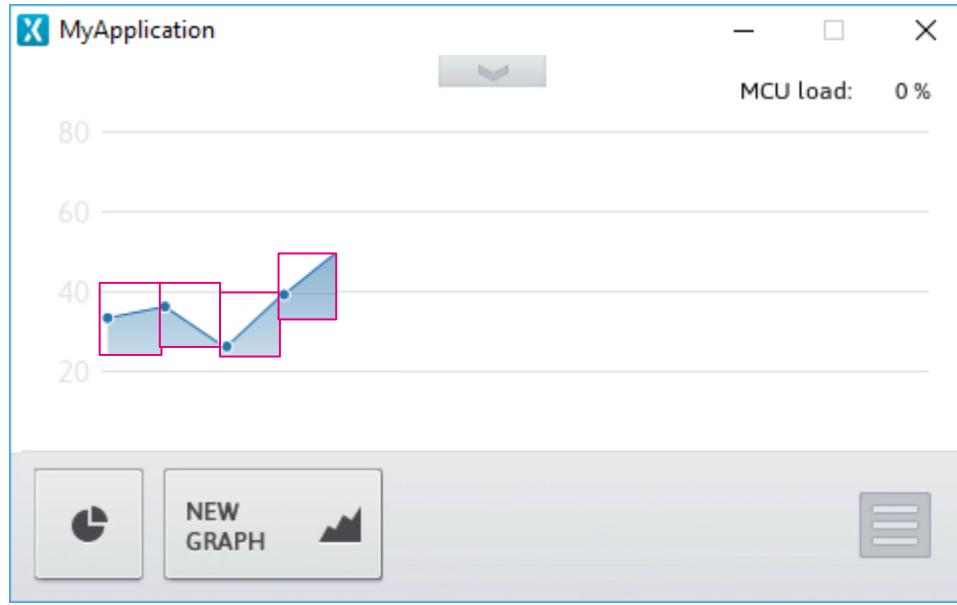
STM32 low cost GUI solution

- **Partial Frame Buffer**

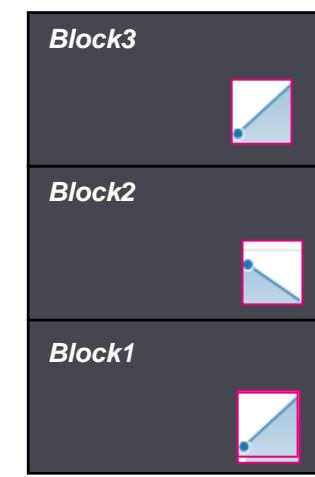
- Improve your UIs with less MCU and Memory resources
- Create an exciting and impactful entry-level GUI
- Intuitive and simple animations
- Configurable framebuffer size starting from 6KBytes
- Useable on any STM32 including Cortex-M0+ cores
- Display interface: DSI, SPI, Parallel/8080/FMC
- Limitations:
 - Partial display update limits UI performance like Texturemapper.
 - Require Display holding display controller and GRAM

STM32 Low-cost GUI solution

- Partial FrameBuffer



*Framebuffer in internal RAM
E.g. Each 6 KBytes*



TouchGFX renders the parts of View that needs to be updated into many small framebuffer blocks.

Whenever a block is rendered, it can be transferred to the display, and the block memory can be reused for rendering one of the next parts.

STM32 low cost GUI solution

Video

Cost-effective GUI solution with STM32 Graphics

STM32 Graphic
Cost-effective GUI Solution



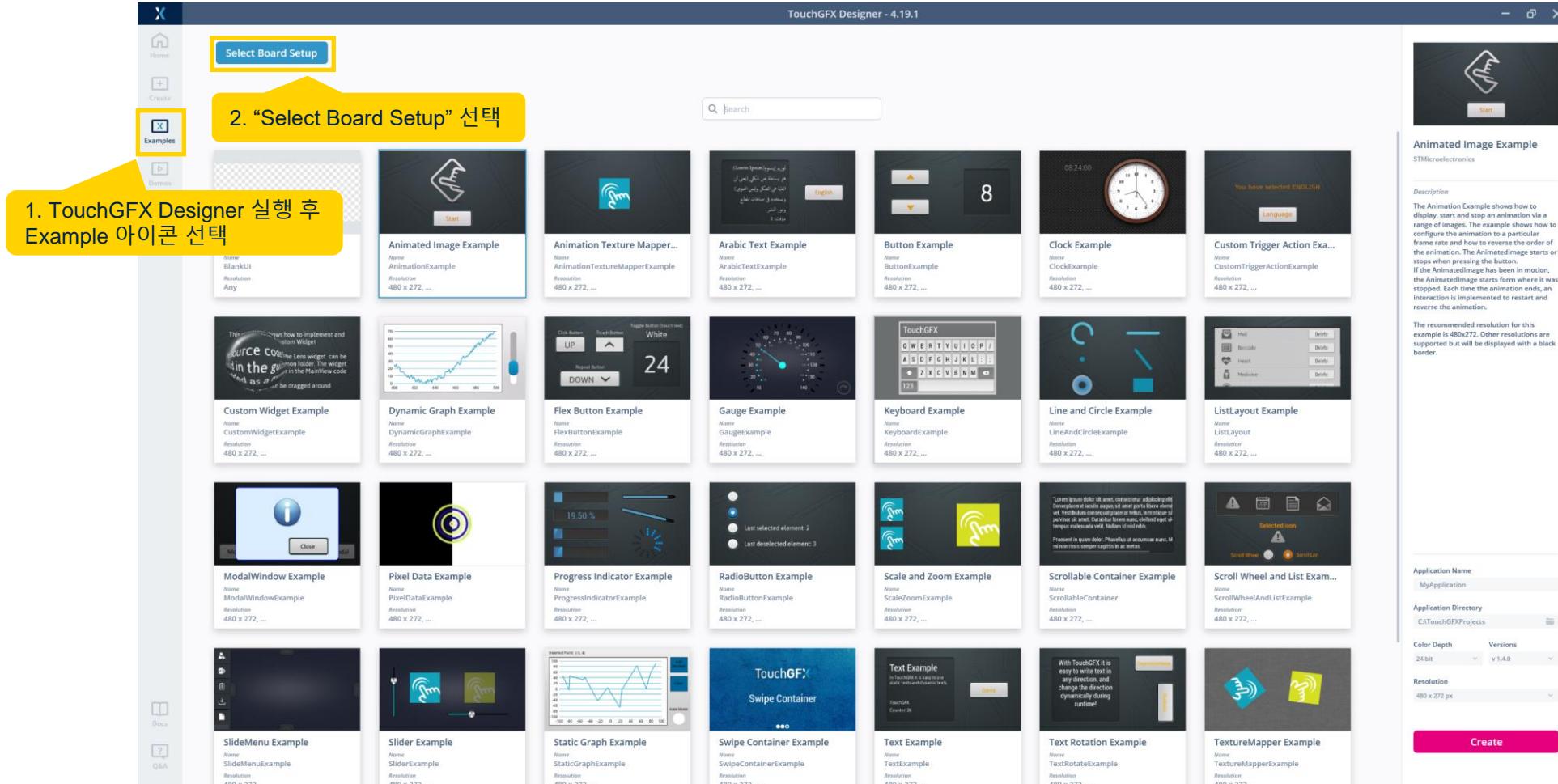
Hands-On with STM32H7B3I-DK

STM32H7B3I-DK

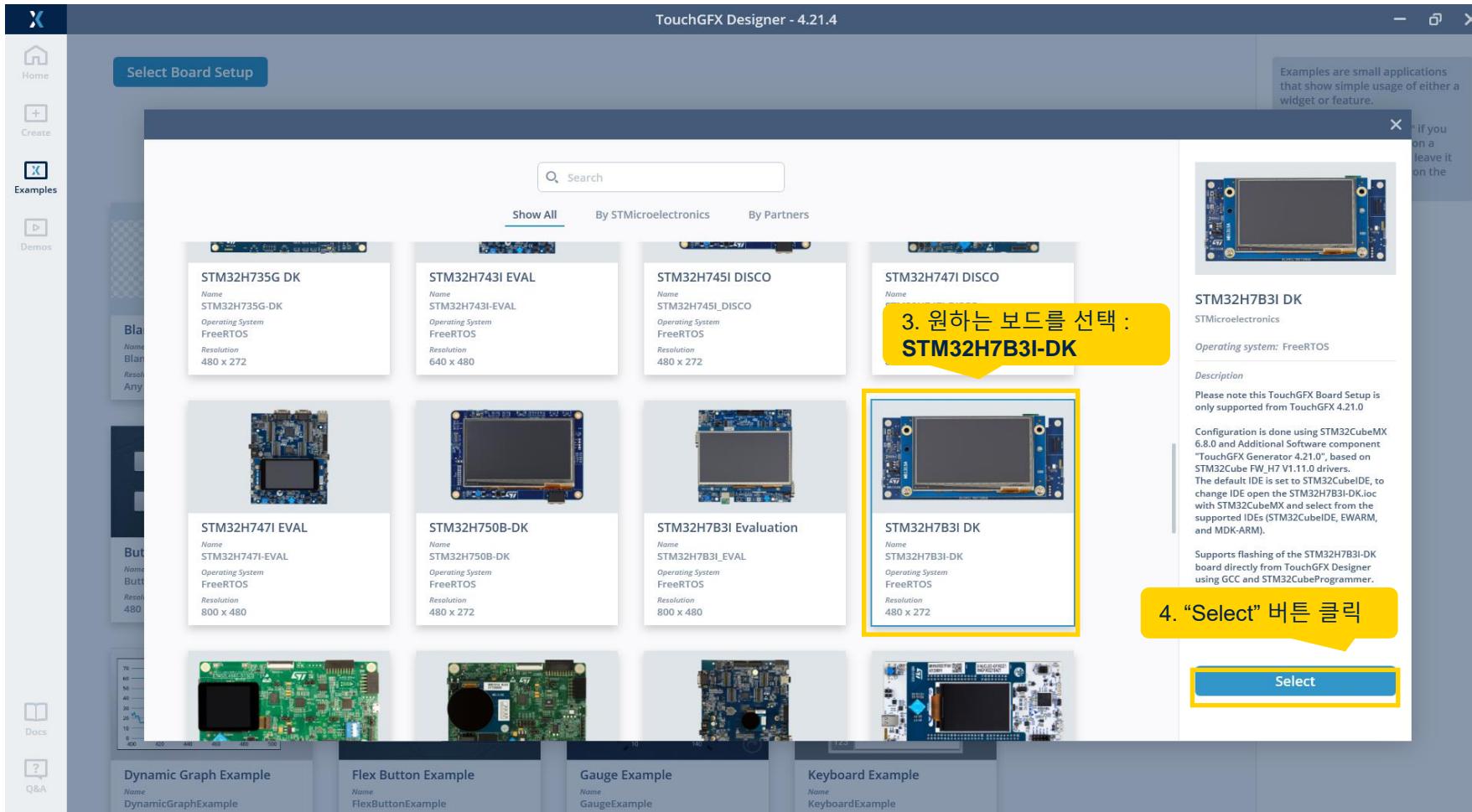
- STM32H7B3LIH6 (BGA255)
- 2MB Flash memory 및 1.4MB SRAM
- 512 Mbit Octo-SPI NOR Flash memory
- 128 Mbit SDRAM
- 4.3" (480x272 pixels) TFT color LCD include :
 - RGB interface
 - Capacitive touch panel
- Micro USB connect
- ST-LINK/V3



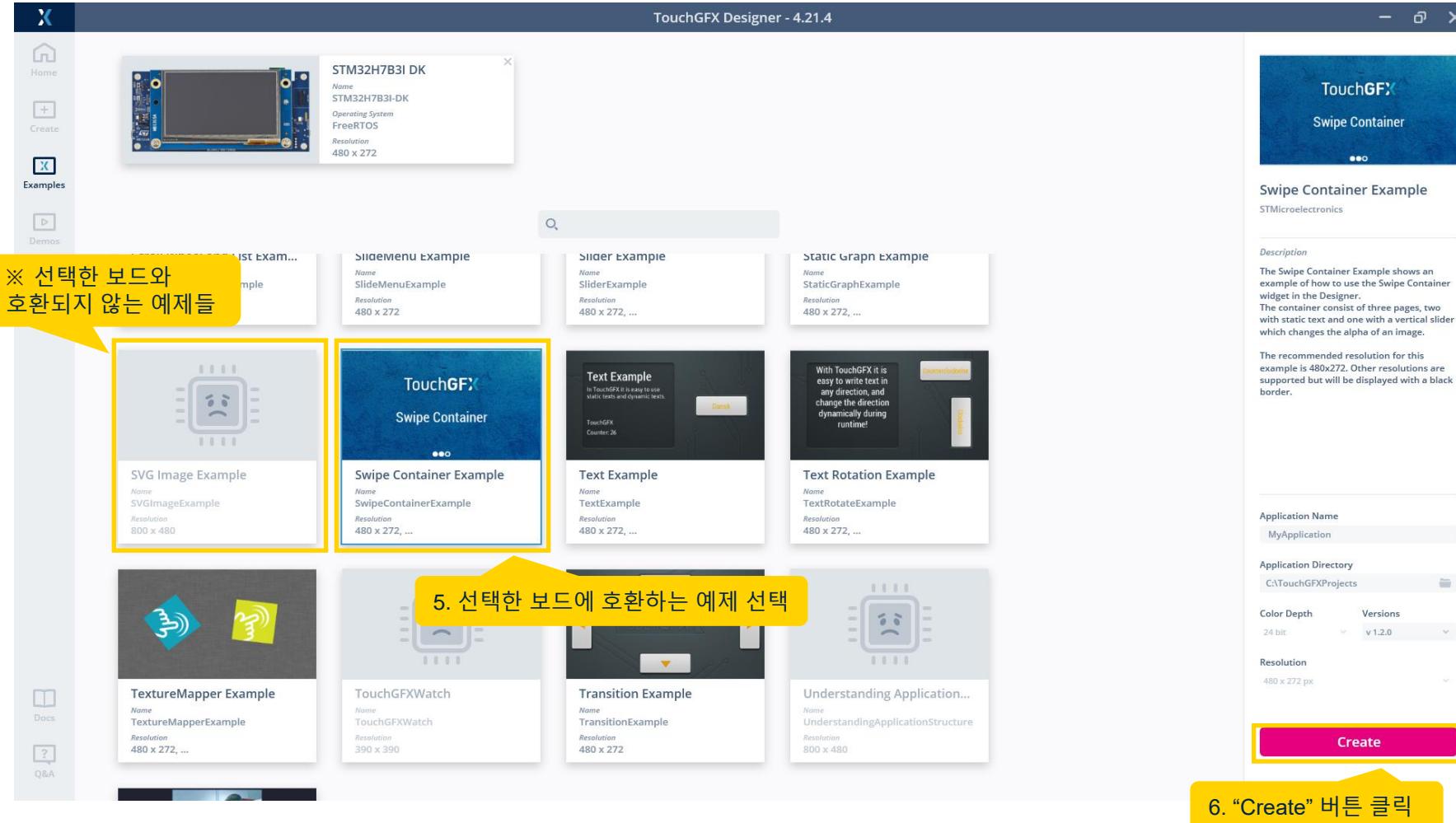
튜토리얼1 : STM32 키트에서 예제 실행하기



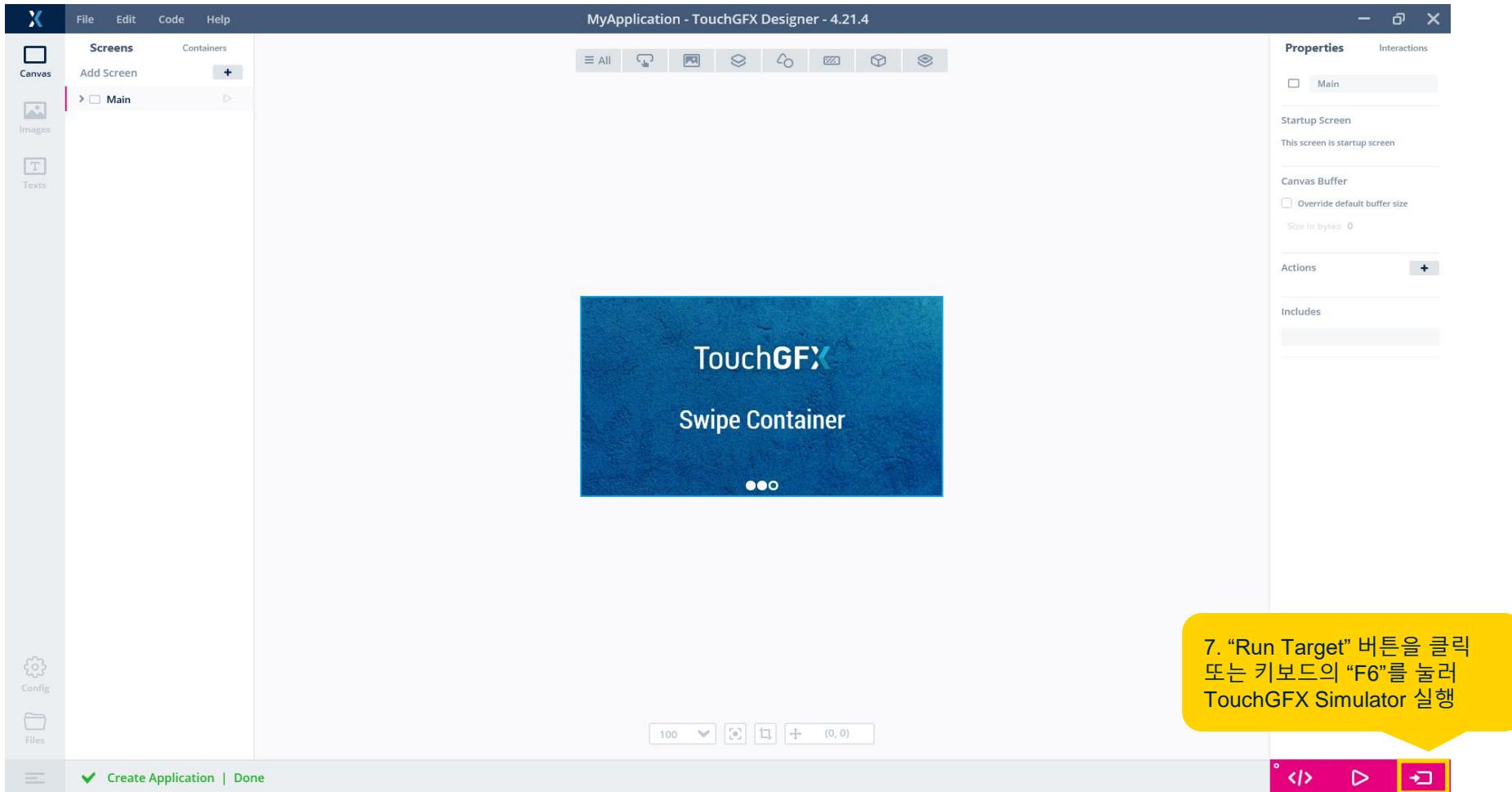
튜토리얼1 : STM32 키트에서 예제 실행하기



튜토리얼1 : STM32 키트에서 예제 실행하기



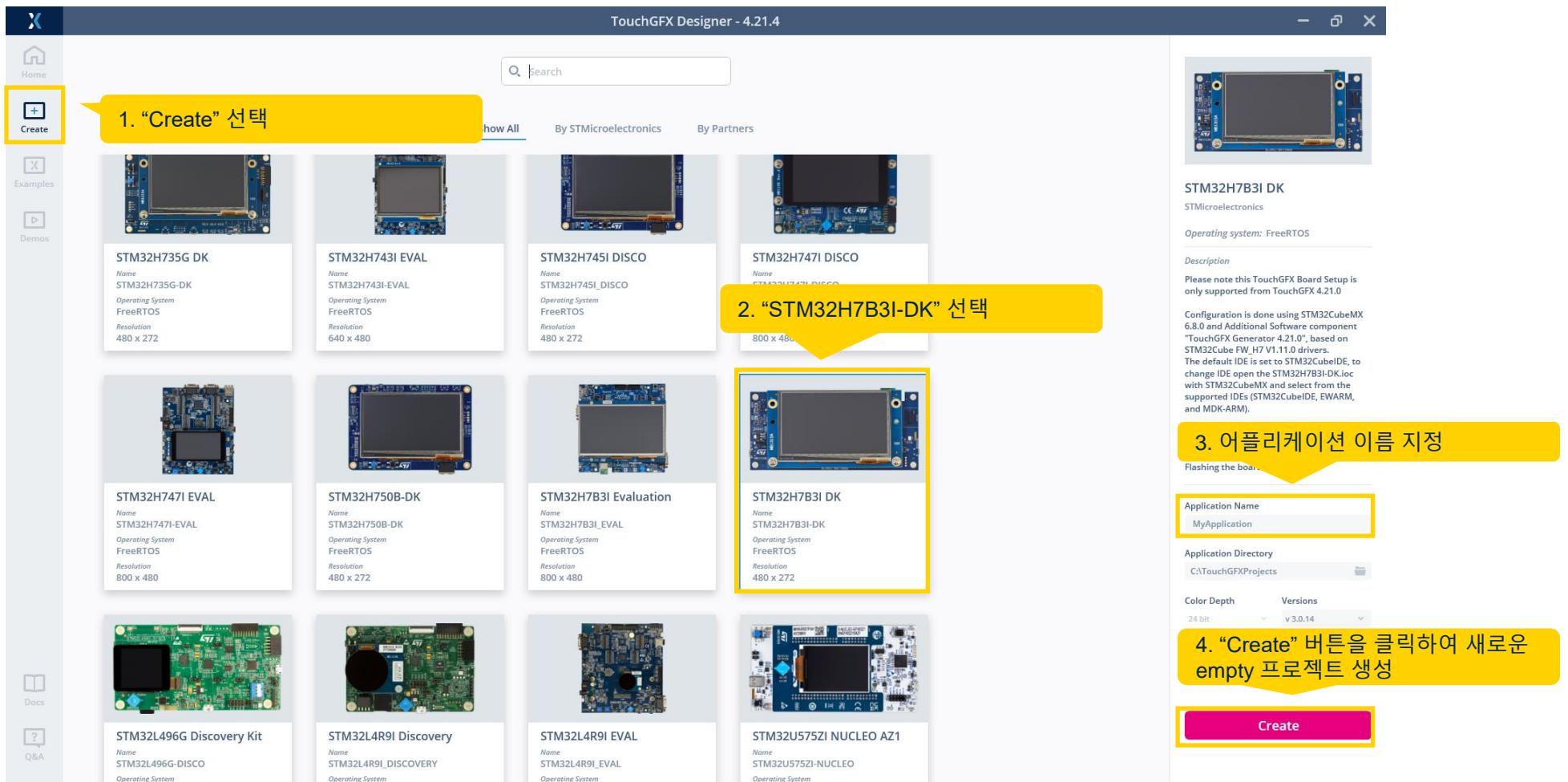
튜토리얼1 : STM32 키트에서 예제 실행하기



※ 타겟으로 플래싱하려면 STM32CubeProgrammer가 설치되어 있어야 합니다.

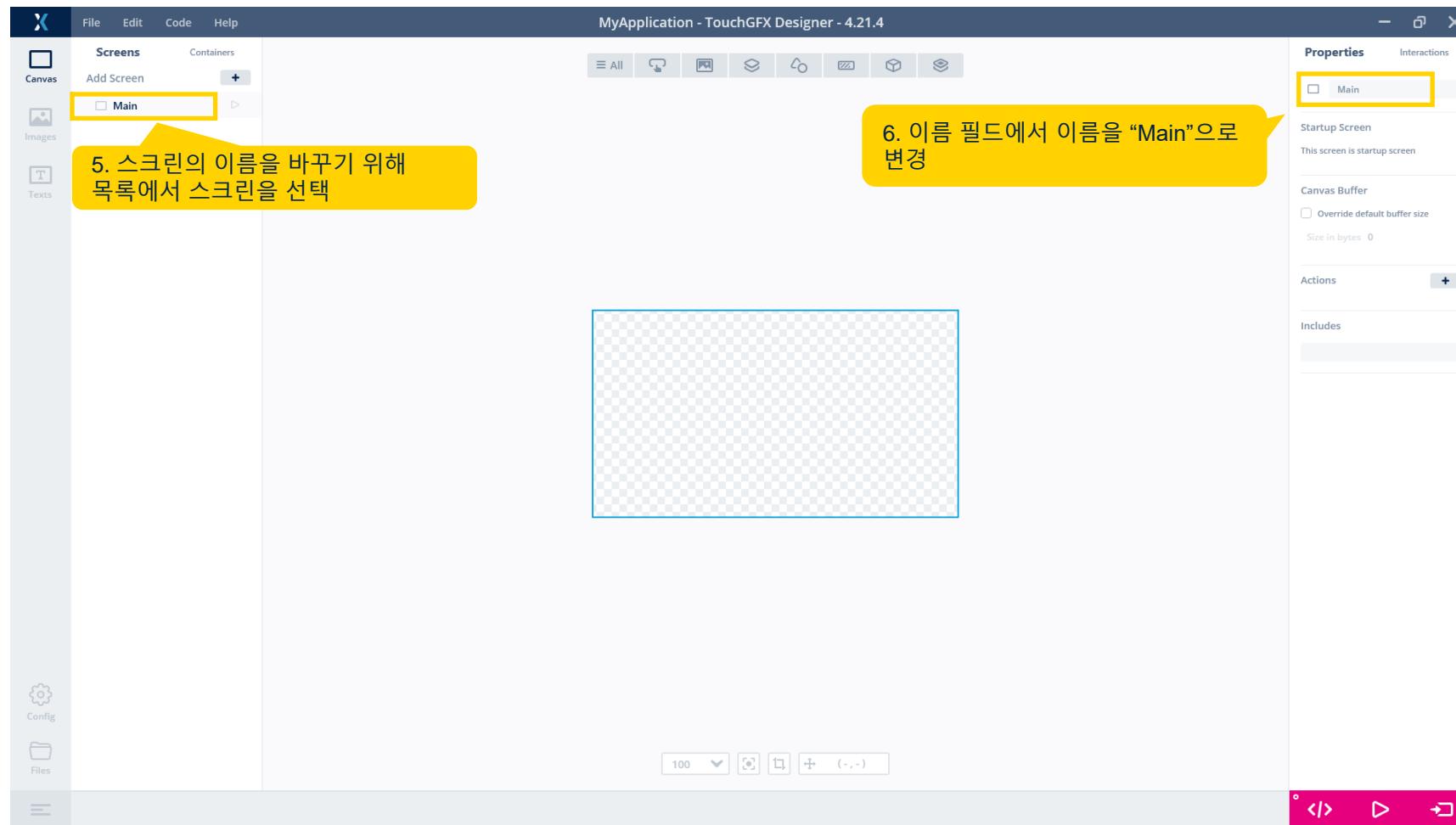
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 1단계: 배경 이미지 설정하기



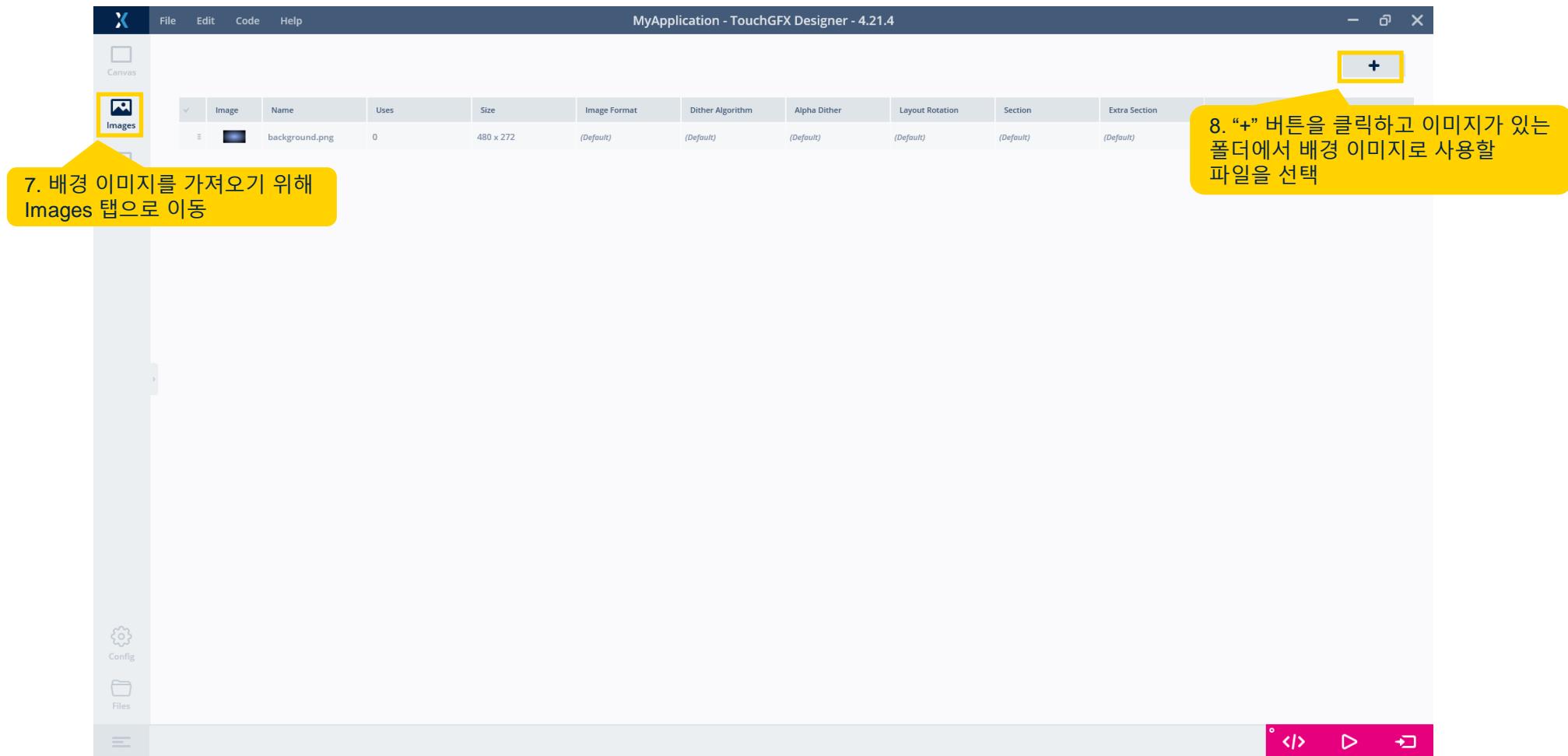
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 1단계: 배경 이미지 설정하기



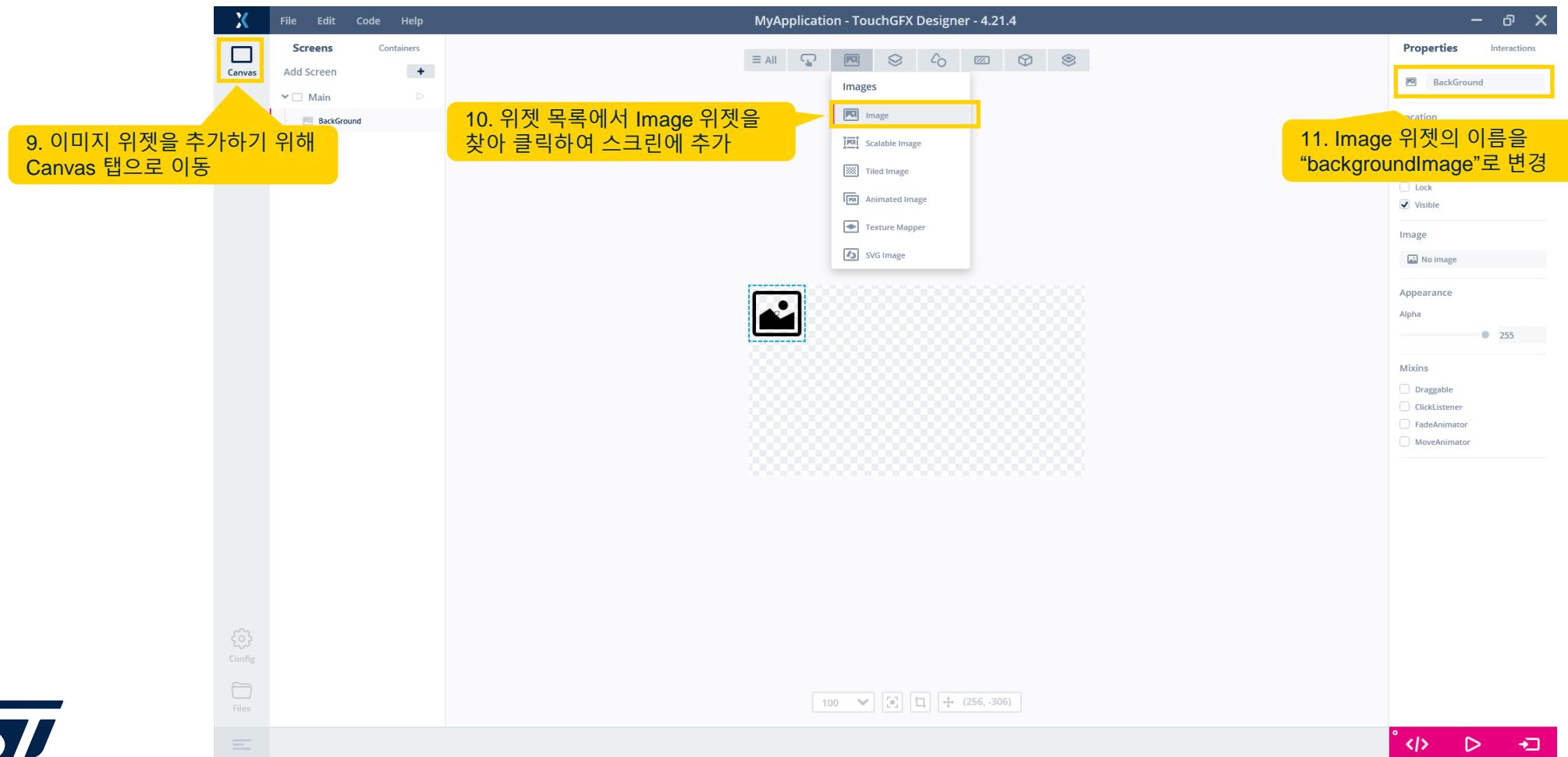
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 1단계: 배경 이미지 설정하기



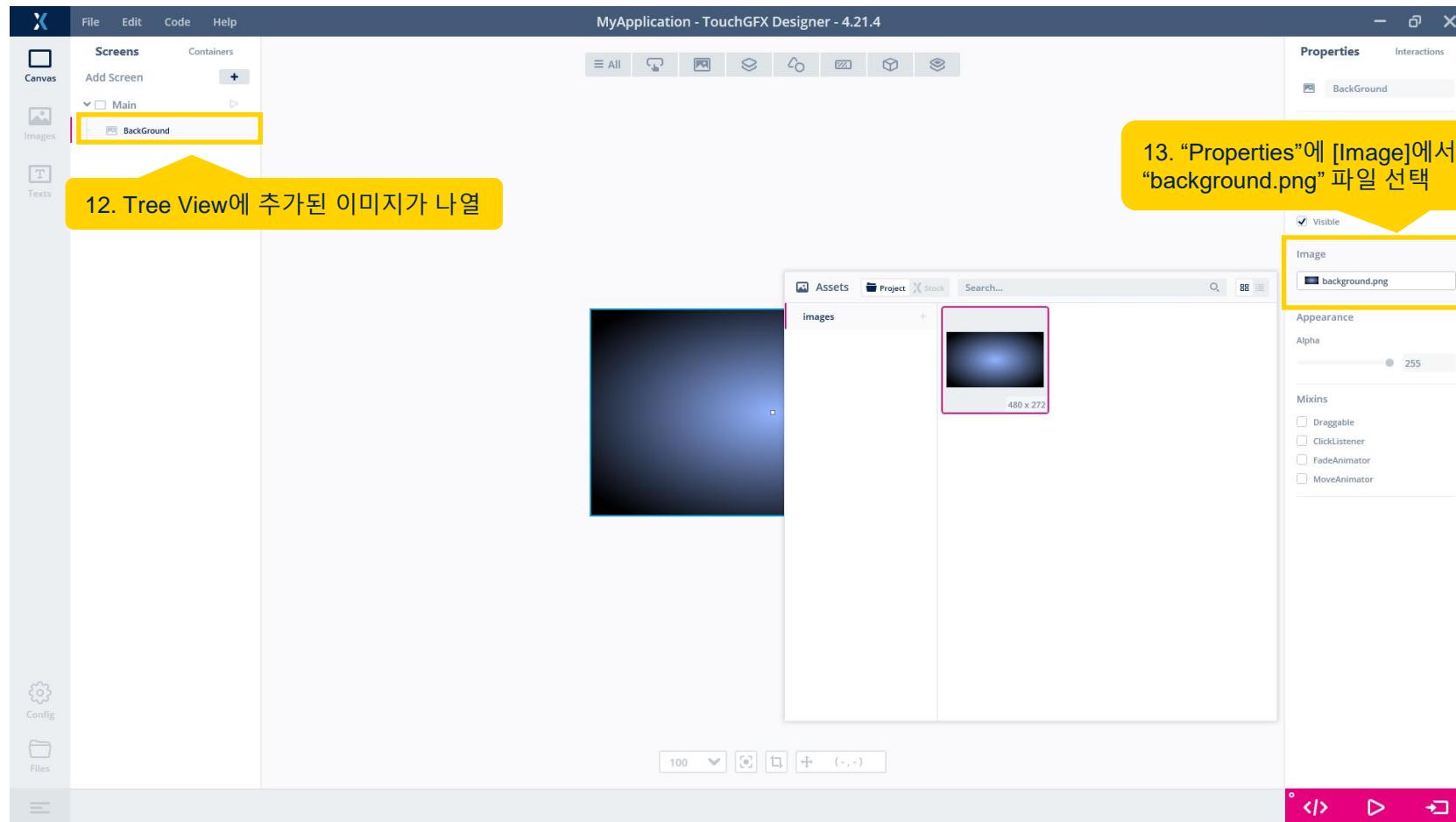
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 1단계: 배경 이미지 설정하기



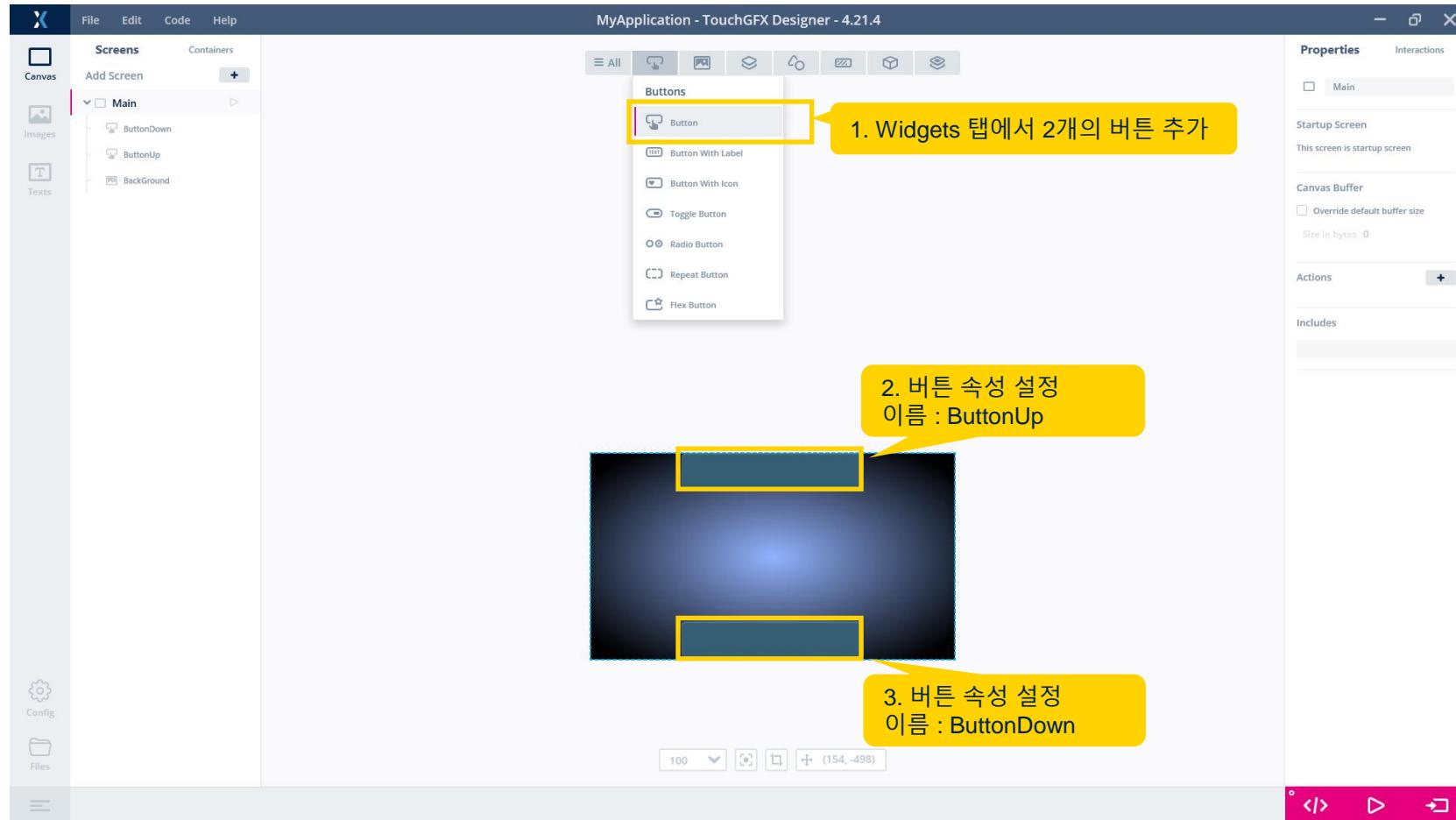
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 1단계: 배경 이미지 설정하기



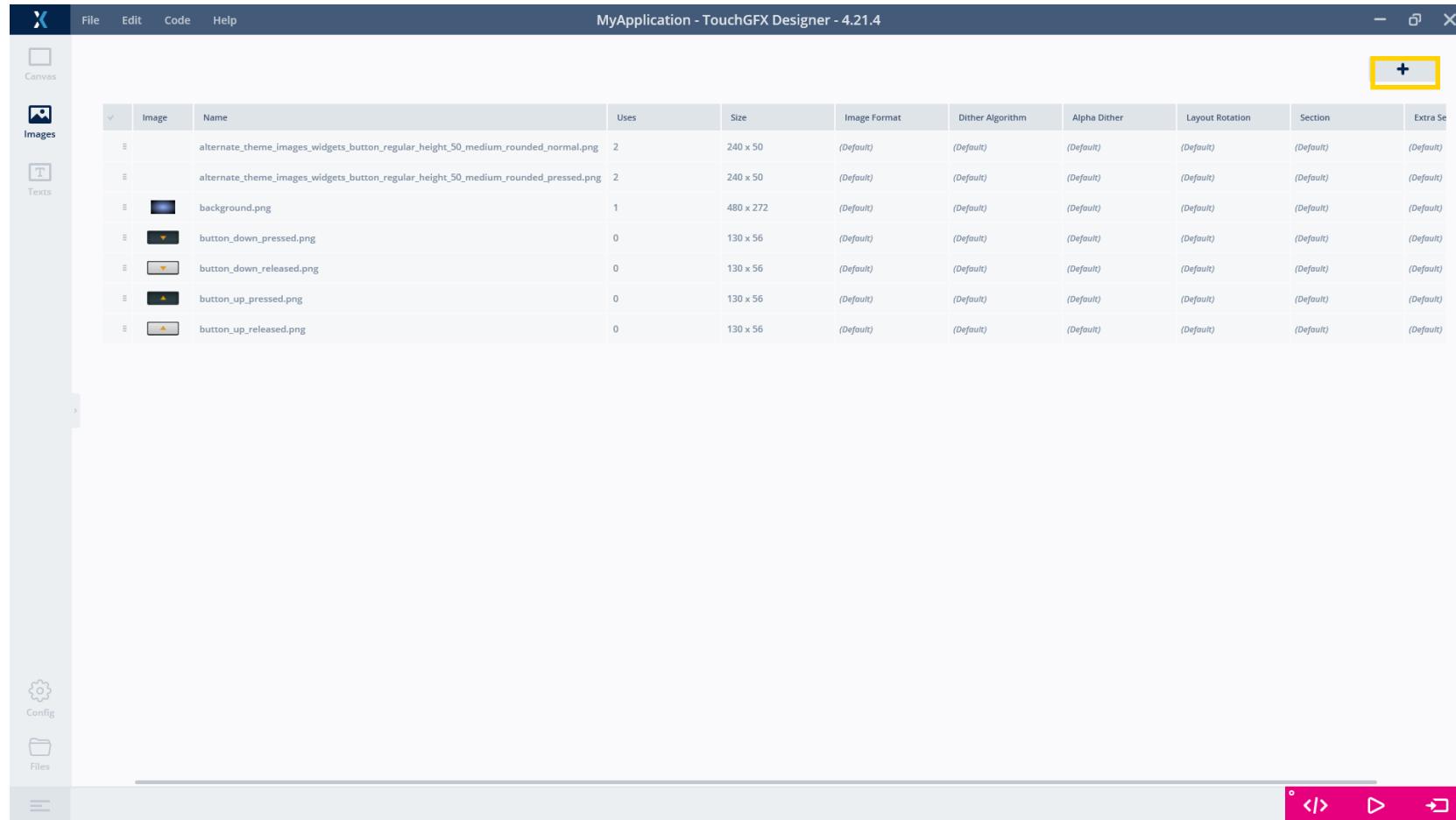
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 2단계: 버튼 추가하기



튜토리얼2 : 자신만의 애플리케이션 생성하기

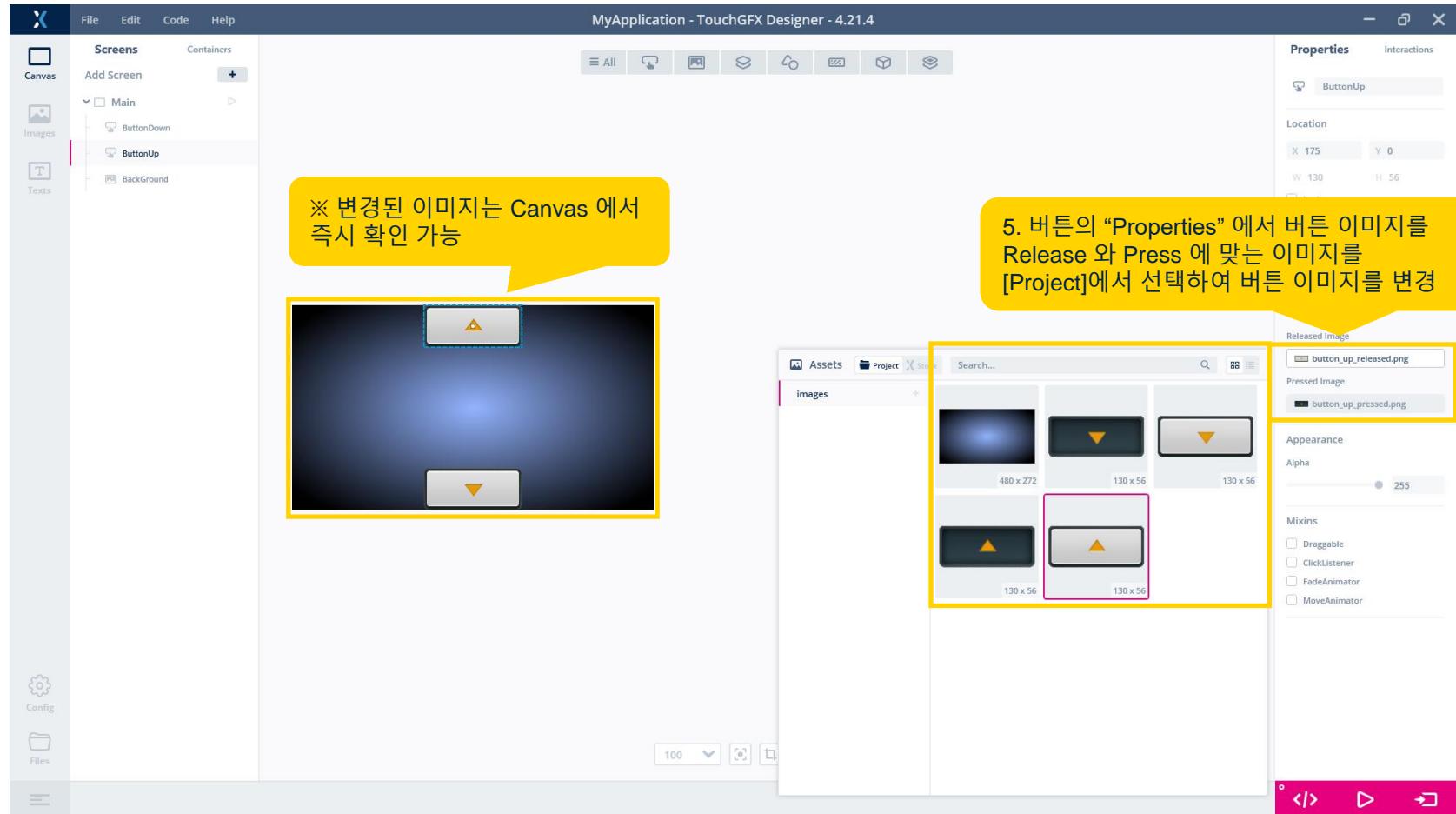
- 2단계: 버튼 추가하기



4. "+" 버튼을 클릭하여
"button_down_pressed.png",
"button_down_released.png",
"button_up_pressed.png",
"button_up_released.png"
이미지를 추가

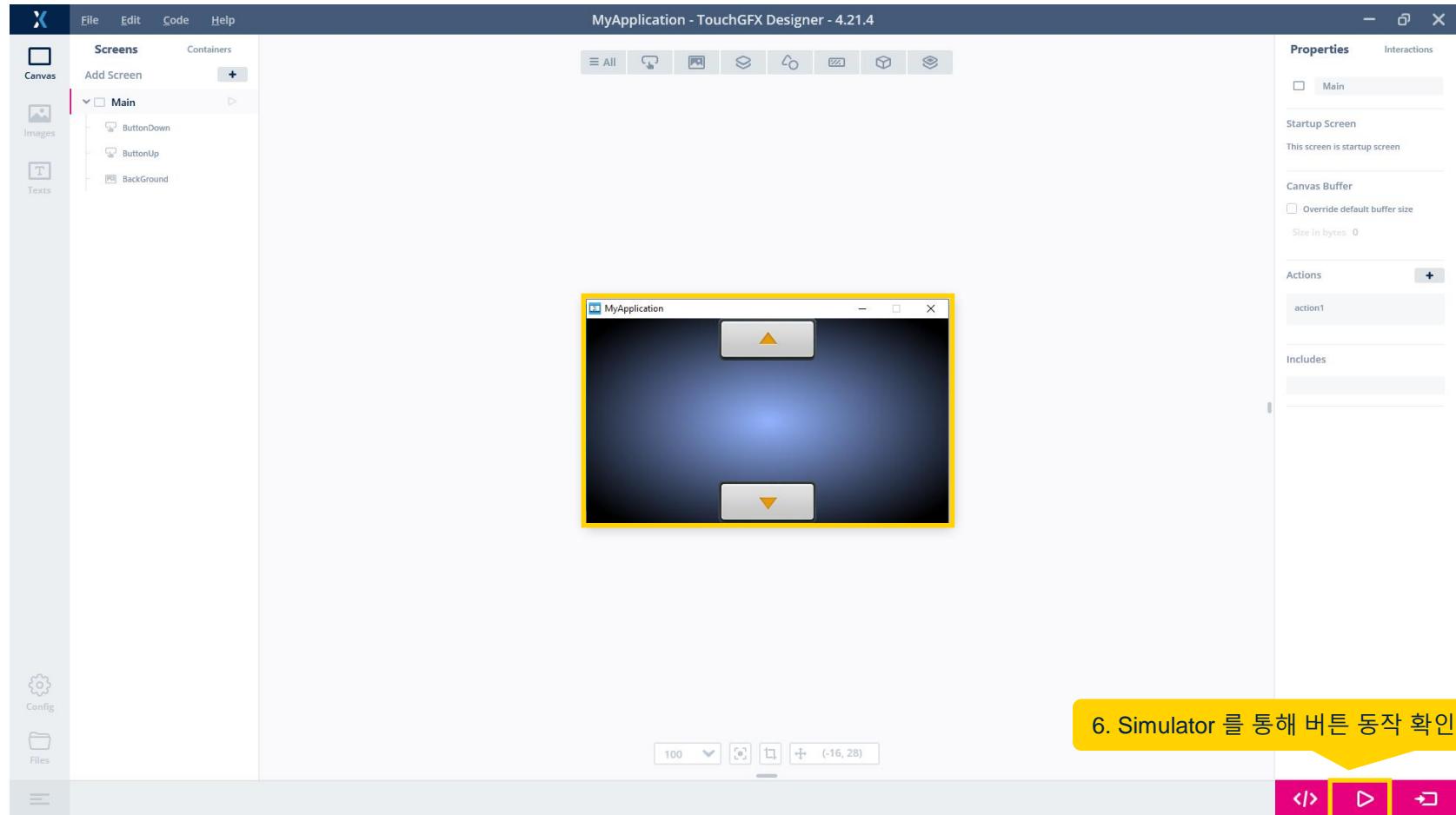
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 2단계: 버튼 추가하기



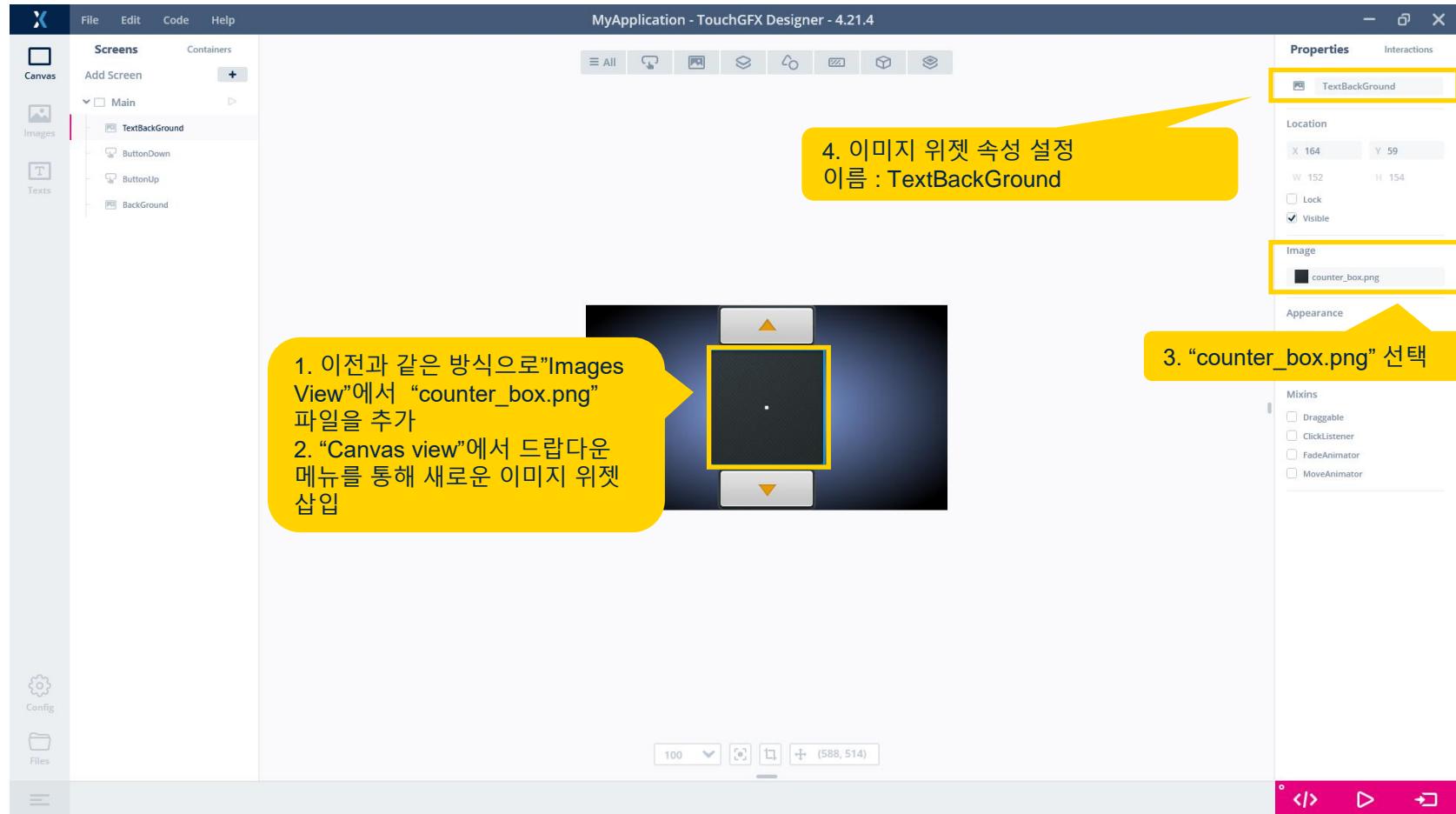
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 2단계: 버튼 추가하기



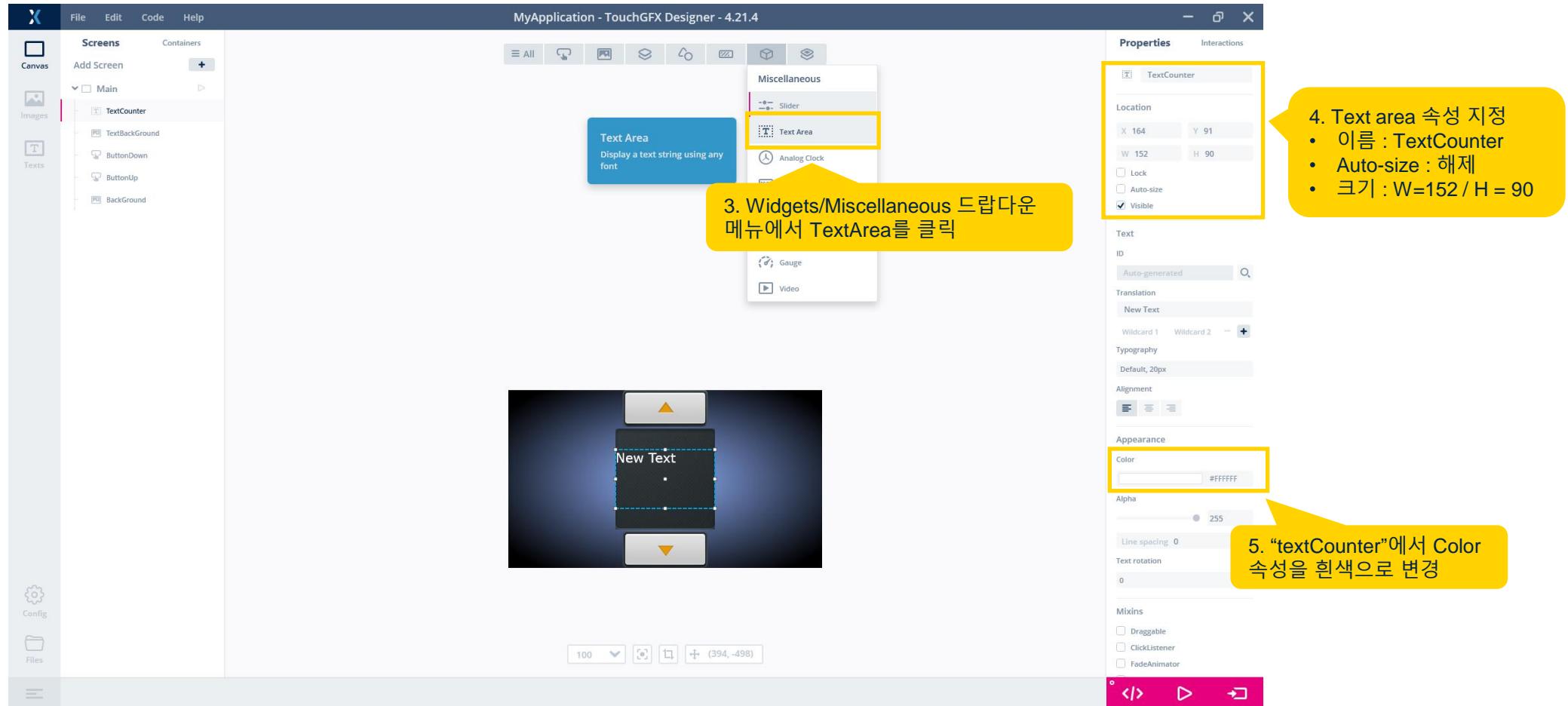
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 3단계: 텍스트 추가하기



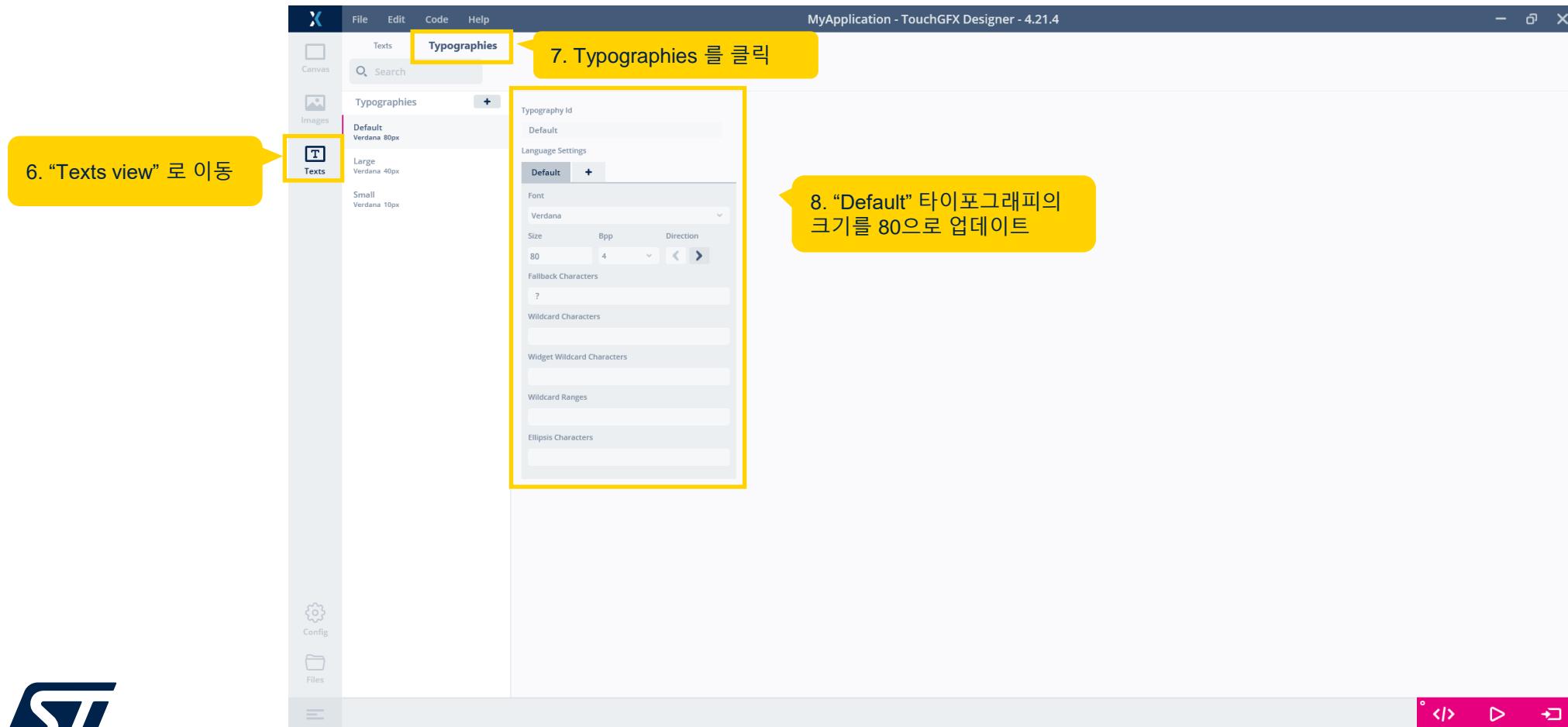
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 3단계: 텍스트 추가하기



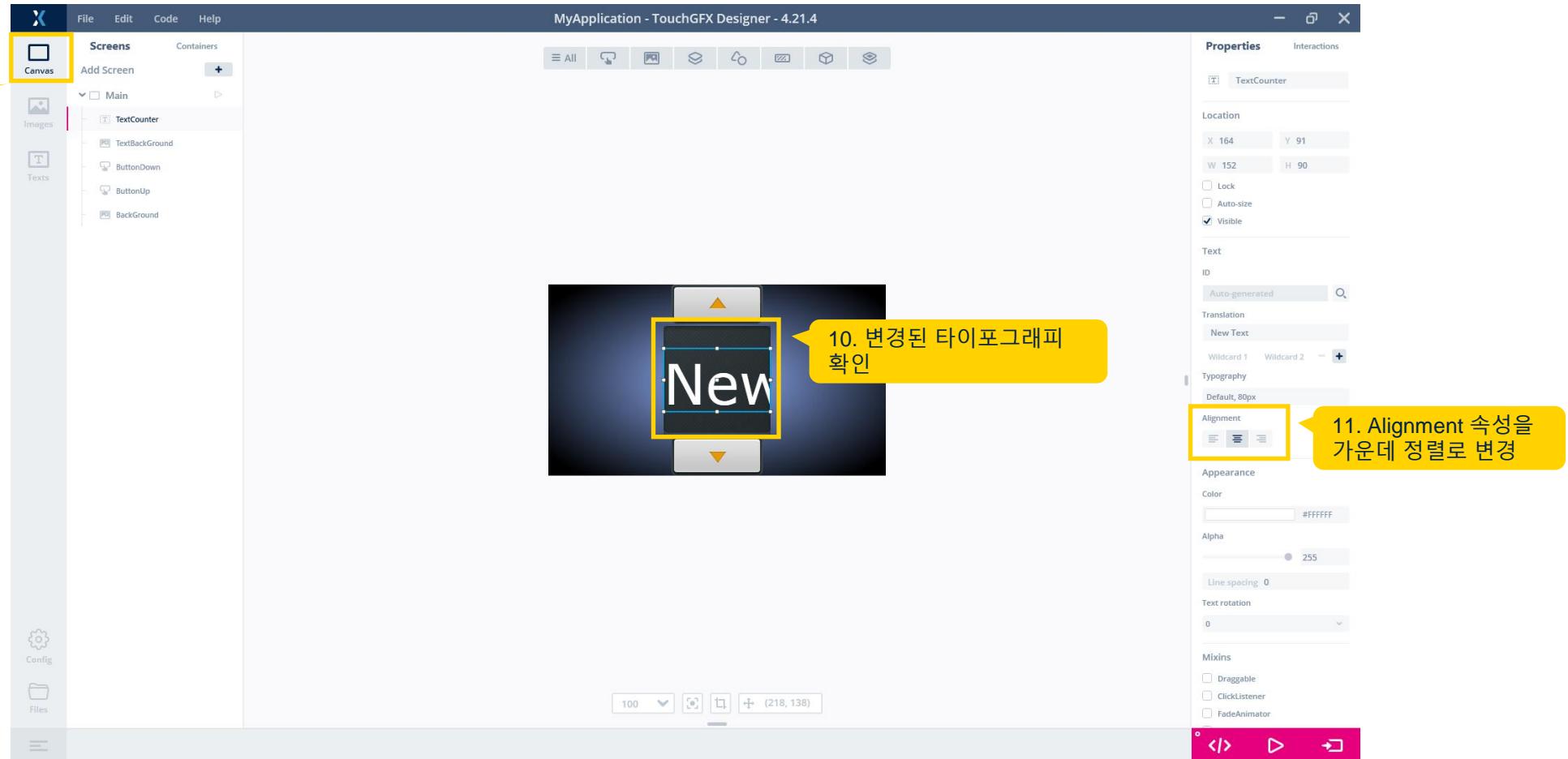
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 3단계: 텍스트 추가하기



튜토리얼2 : 자신만의 애플리케이션 생성하기

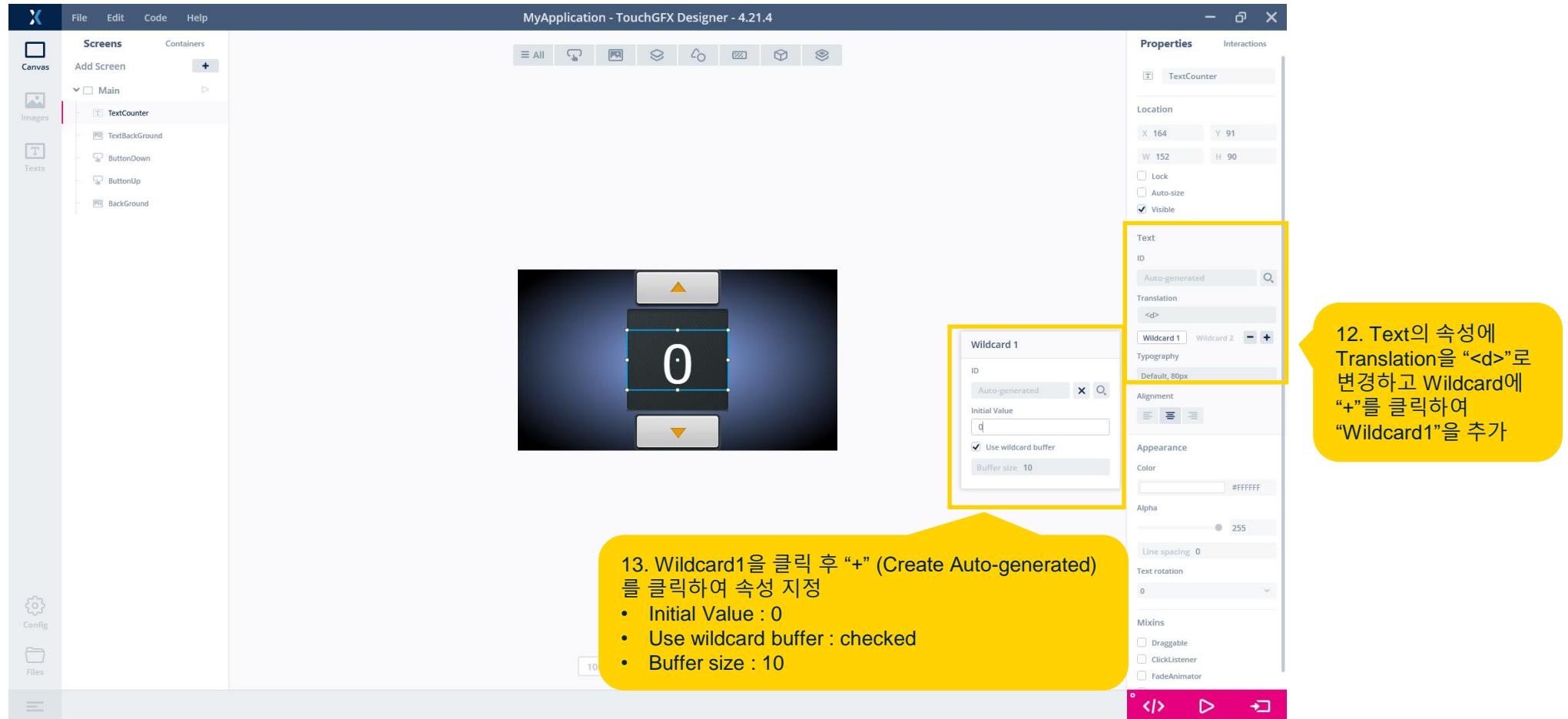
- 3단계: 텍스트 추가하기



튜토리얼2 : 자신만의 애플리케이션 생성하기

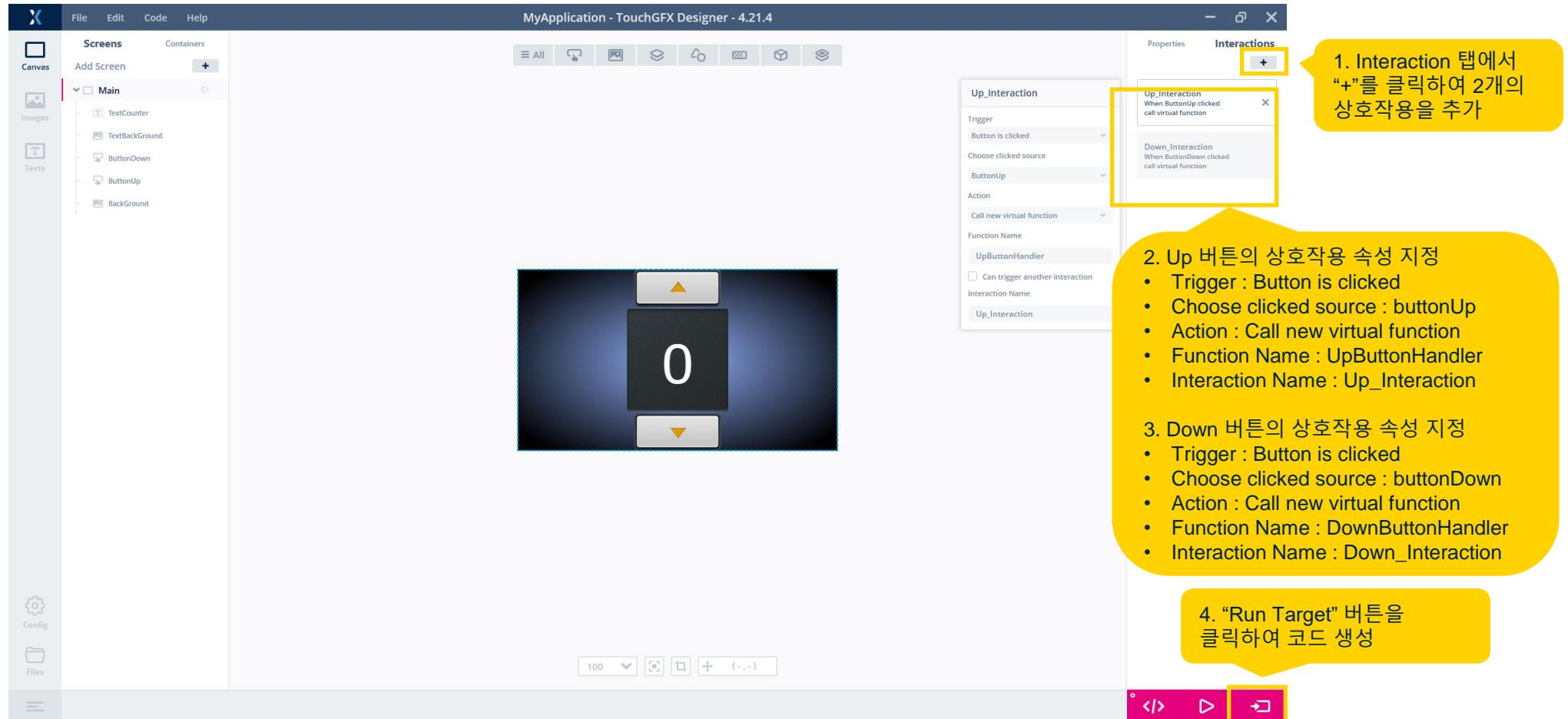
• 3단계: 텍스트 추가하기

※ WildCard : 런타임에서 텍스트 영역을 숫자와 같이 변수로 대체할 수 있는 마커(예, "<d>").
동적인 부분을 고정 텍스트와 함께 결합 가능(예, "Temperature: <temp>" °C)



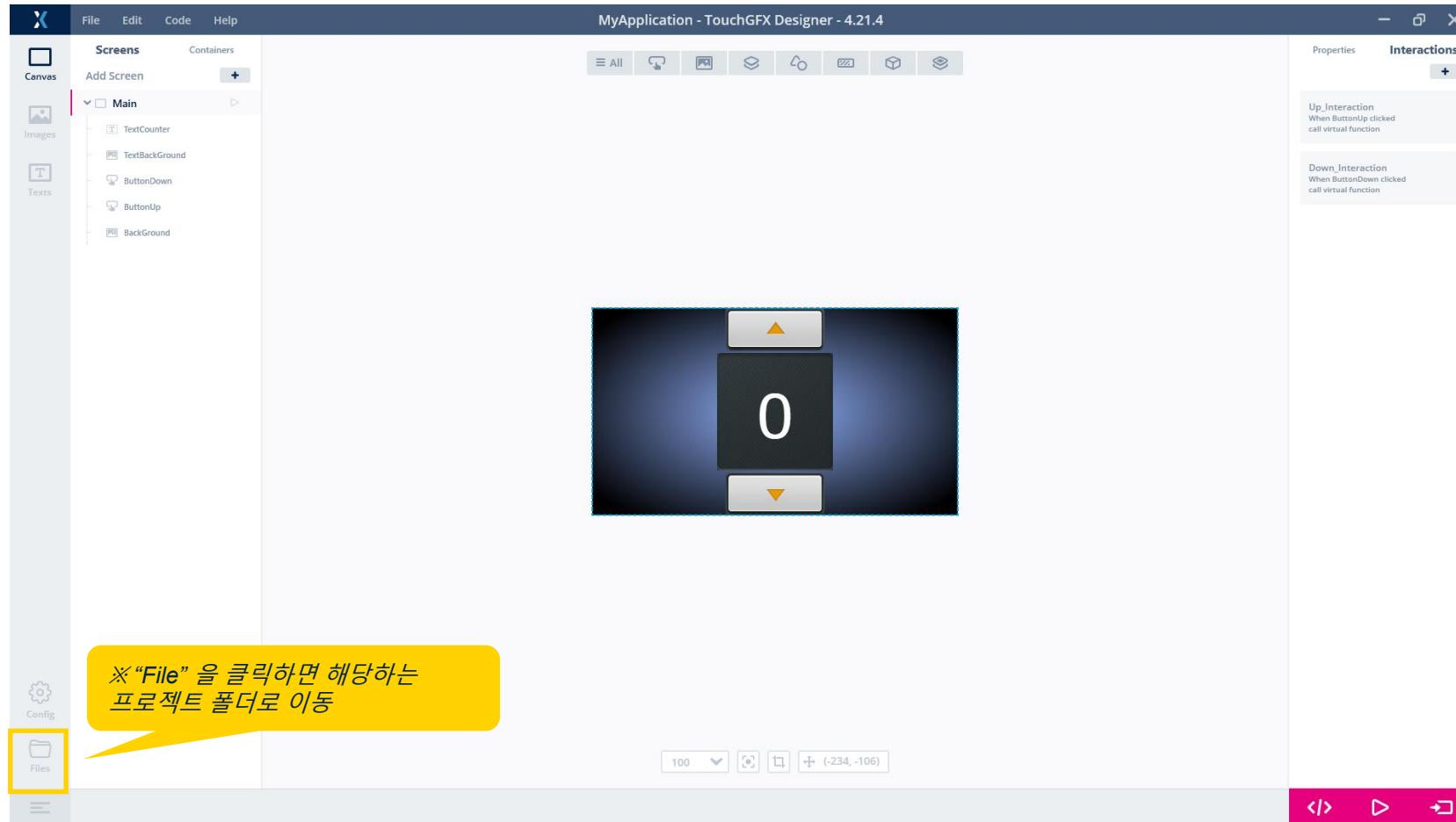
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 4단계: 상호동작, Interaction 추가하기



튜토리얼2 : 자신만의 애플리케이션 생성하기

- 5단계: 코드 추가하기



튜토리얼2 : 자신만의 애플리케이션 생성하기

- 5단계: 코드 추가하기

```
..\TouchGFXTraining\TouchGFX\generated\gui_generated\include\gui_generated\main_screen\MainViewBase.hpp
```

```
class MainViewBase : public touchgfx::View<MainPresenter>
{
public:
    MainViewBase();
    virtual ~MainViewBase();
    virtual void setupScreen();

    /*
     * Custom Actions
     */
    virtual void action1()
    {
        // Override and implement this function in Screen1
    }

    virtual void action2()
    {
        // Override and implement this function in Screen1
    }
}
```

```
/*
 * Virtual Action Handlers
 */
virtual void UpButtonHandler()
{
    // Override and implement this function in Main
}
virtual void DownButtonHandler()
{
    // Override and implement this function in Main
}
```

※ Interaction에서 생성한 비어 있는 두개의 메소드
하위 클래스에서 두 메소드를 직접 구현

튜토리얼2 : 자신만의 애플리케이션 생성하기

- 4단계: 코드 추가하기

```
..\TouchGFXTraining\TouchGFX\gui\include\gui\main_screen>MainView.hpp

#include <gui_generated/main_screen/MainViewBase.hpp>
#include <gui/main_screen/MainPresenter.hpp>

class MainView : public MainViewBase
{
public:
    MainView();
    virtual ~MainView() {}
    virtual void setupScreen();
    virtual void tearDownScreen();

    virtual void UpButtonHandler();
    virtual void DownButtonHandler();
```

1. MainView 클래스에서 두 메소드를 선언

```
protected:
};
```

튜토리얼2 : 자신만의 애플리케이션 생성하기

- 4단계: 코드 추가하기

```
..\TouchGFXTraining\TouchGFX\gui\src\main_screen>MainView.cpp
```

```
#include <gui/main_screen/MainView.hpp>
#include <touchgfx/Utils.hpp>
```

```
MainView::MainView()
```

2. touchgfx_printf() 함수를 호출하기 위해
<touchgfx/utils.hpp> 헤더 선언

```
{
```

```
void MainView::setupScreen()
{
    MainViewBase::setupScreen();
}
```

```
void MainView::tearDownScreen()
{
    MainViewBase::tearDownScreen();
}
```

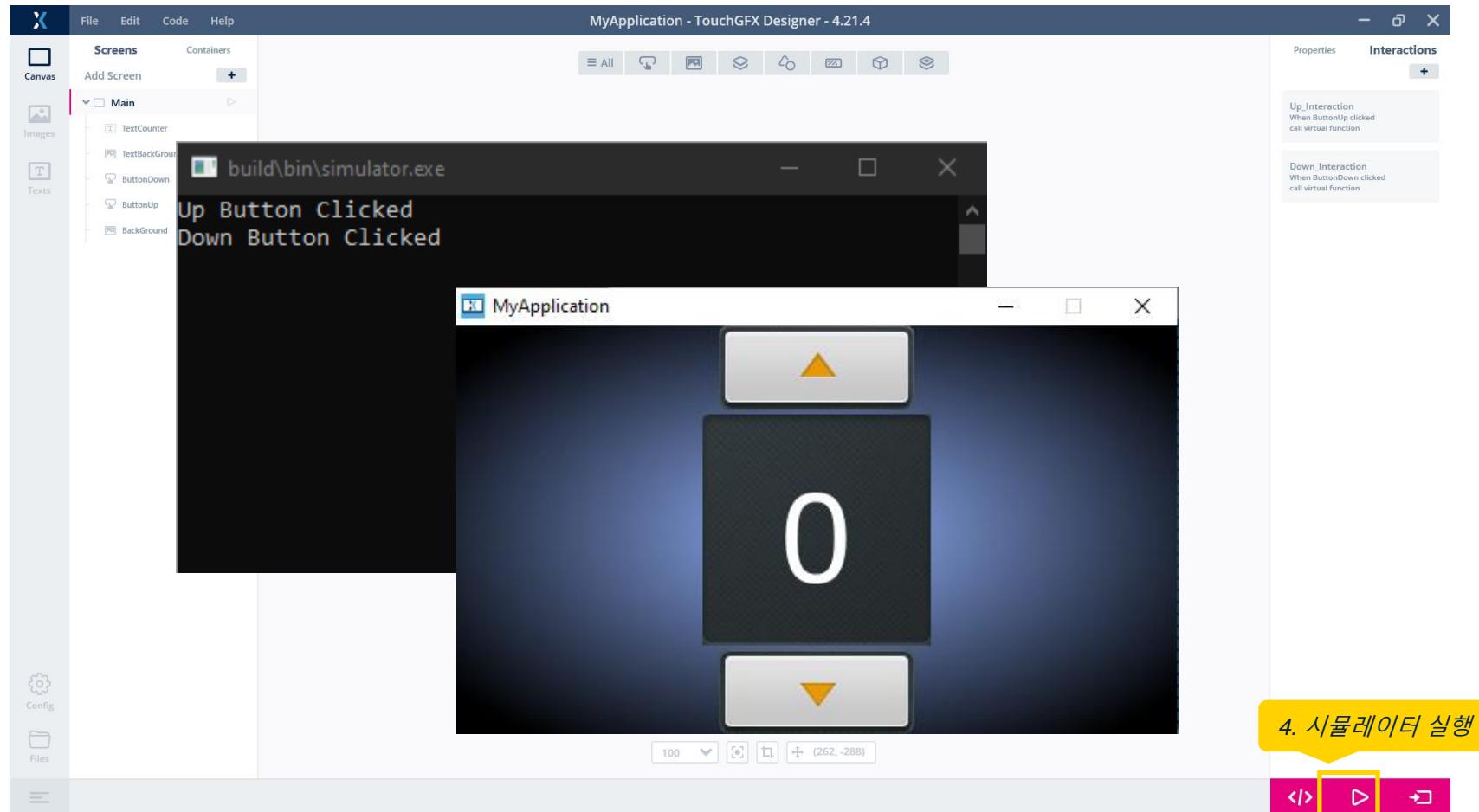
```
void MainView::UpButtonHandler()
{
    touchgfx_printf("Up Button Clicked\n");
}
```

3. 버튼을 클릭했을 때 두 함수가 호출되어
printf를 통한 로그가 출력되도록 구현

```
void MainView::DownButtonHandler()
{
    touchgfx_printf("Down Button Clicked\n");
}
```

튜토리얼2 : 자신만의 애플리케이션 생성하기

- 4단계: 코드 추가하기



튜토리얼2 : 자신만의 애플리케이션 생성하기

- 4단계: 코드 추가하기

```
..\TouchGFXTraining\TouchGFX\gui\include\gui\main_screen>MainView.hpp

#ifndef MAINVIEW_HPP
#define MAINVIEW_HPP

#include <gui_generated/main_screen/MainViewBase.hpp>
#include <gui/main_screen/MainPresenter.hpp>

class MainView : public MainViewBase
{
public:
    MainView();
    virtual ~MainView() {};
    virtual void setupScreen();
    virtual void tearDownScreen();

    virtual void UpButtonHandler();
    virtual void DownButtonHandler();

protected:
    int counter;
};

#endif // MAINVIEW_HPP
```

5. 버튼을 눌렀을 때 카운터 값을 대입할
변수 counter를 MainView 클래스에 추가

튜토리얼2 : 자신만의 애플리케이션 생성하기

- 4단계: 코드 추가하기

```
..\TouchGFXTraining\TouchGFX\gui\include\gui\main_screen>MainView.hpp

#include <gui/main_screen/MainView.hpp>
#include <touchgfx/Utils.hpp>

(생략) ...

void MainView::UpButtonHandler()
{
    touchgfx_printf("Up Button Clicked\n");

    counter++;
    Unicode::snprintf(TextCounterBuffer, TEXTCOUNTER_SIZE, "%d", counter);
    //Invalidate text area, which will result in it being redrawn in text tick
    TextCounter.invalidate();
}

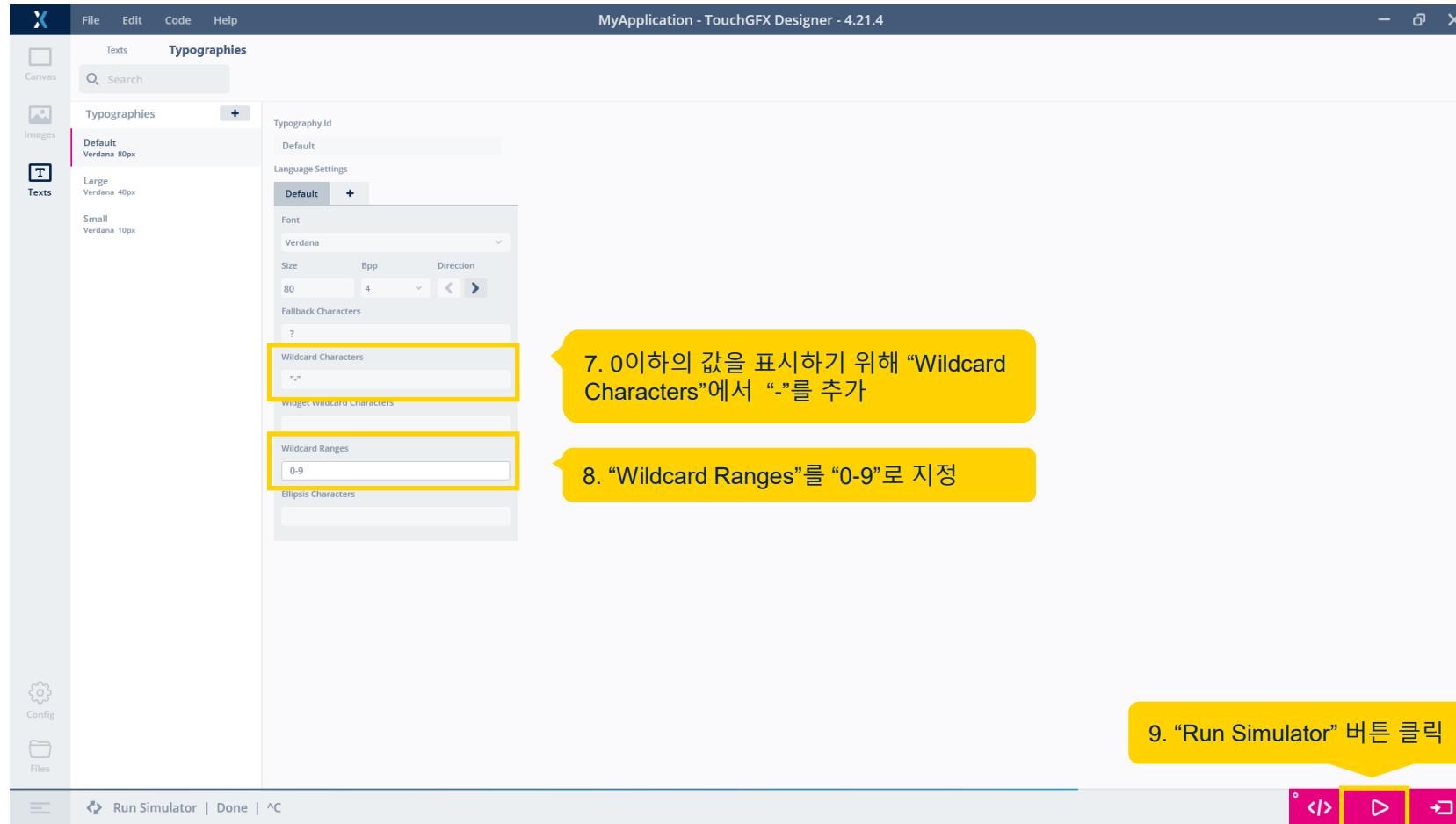
void MainView::DownButtonHandler()
{
    touchgfx_printf("Down Button Clicked\n");

    counter--;
    Unicode::snprintf(TextCounterBuffer, TEXTCOUNTER_SIZE, "%d", counter);
    //Invalidate text area, which will result in it being redrawn in text tick
    TextCounter.invalidate();
}
```

6. TextArea와 Wildcard 버퍼 업데이트 코드 추가
 - UpButtonHandler 메소드에서 카운터 값을 증가.
 - 새로운 값이 문자열로 변환되어 이전 단계에서 TextArea에 구현한 Wildcard 버퍼로 복사.
 - TextArea는 Unicode를 사용하기 때문에 Unicode 버퍼 작성성을 지원하는 snprintf 함수를 사용
 - 버퍼 업데이트 후 textCounter위젯에서 invalidate() 를 호출하여 다시 그리기 실행

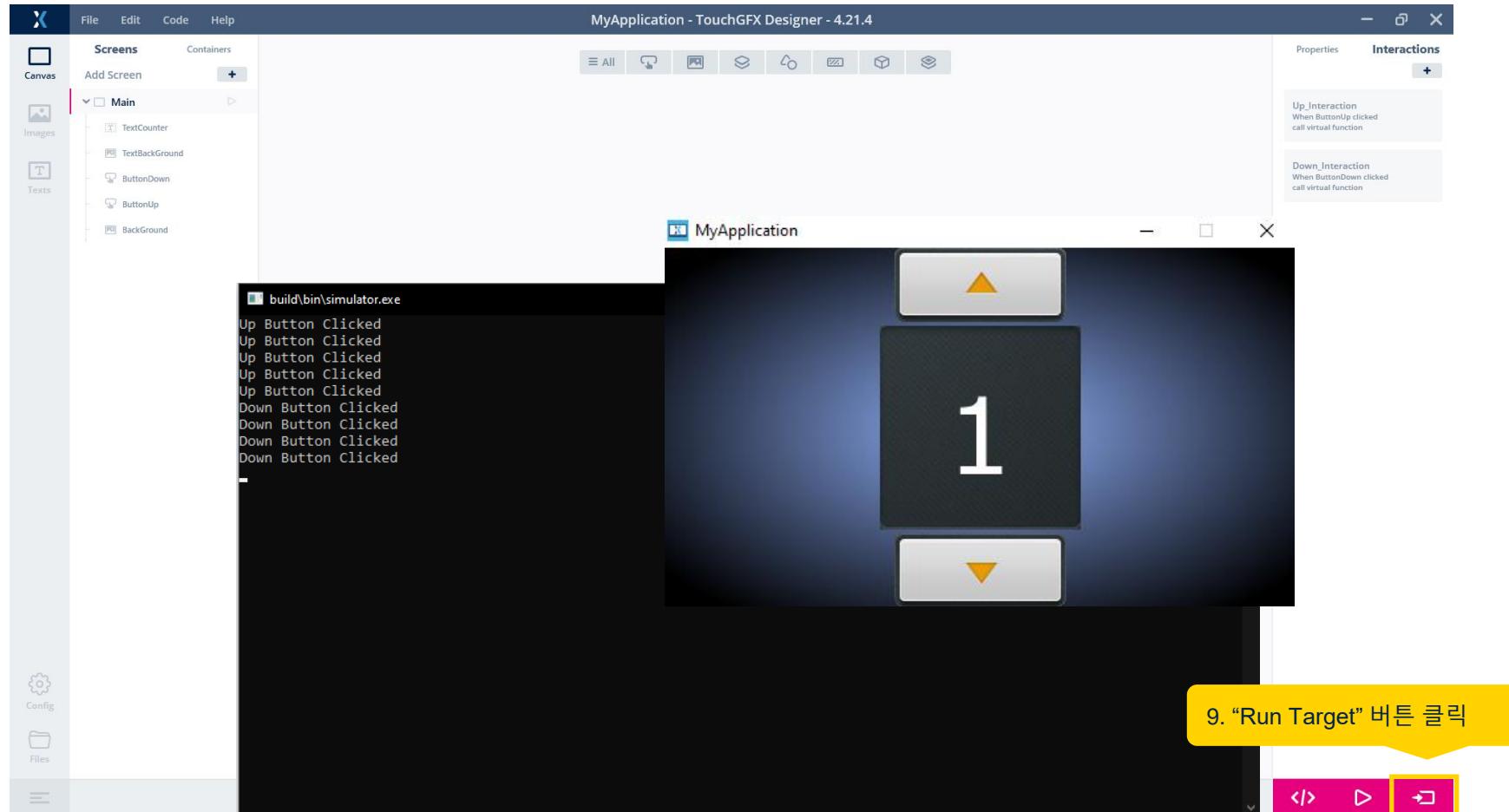
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 4단계: 코드 추가하기



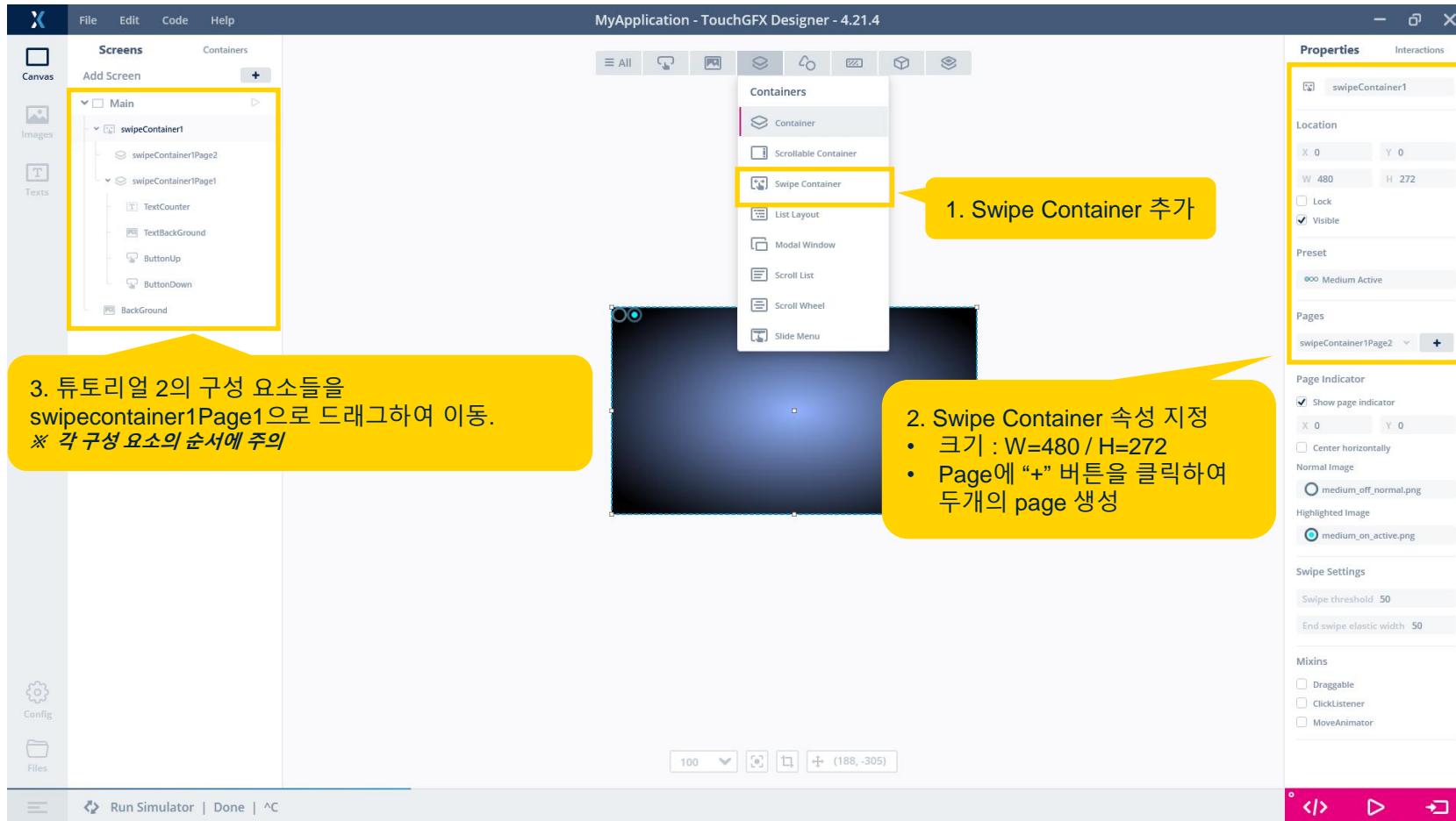
튜토리얼2 : 자신만의 애플리케이션 생성하기

- 4단계: 코드 추가하기



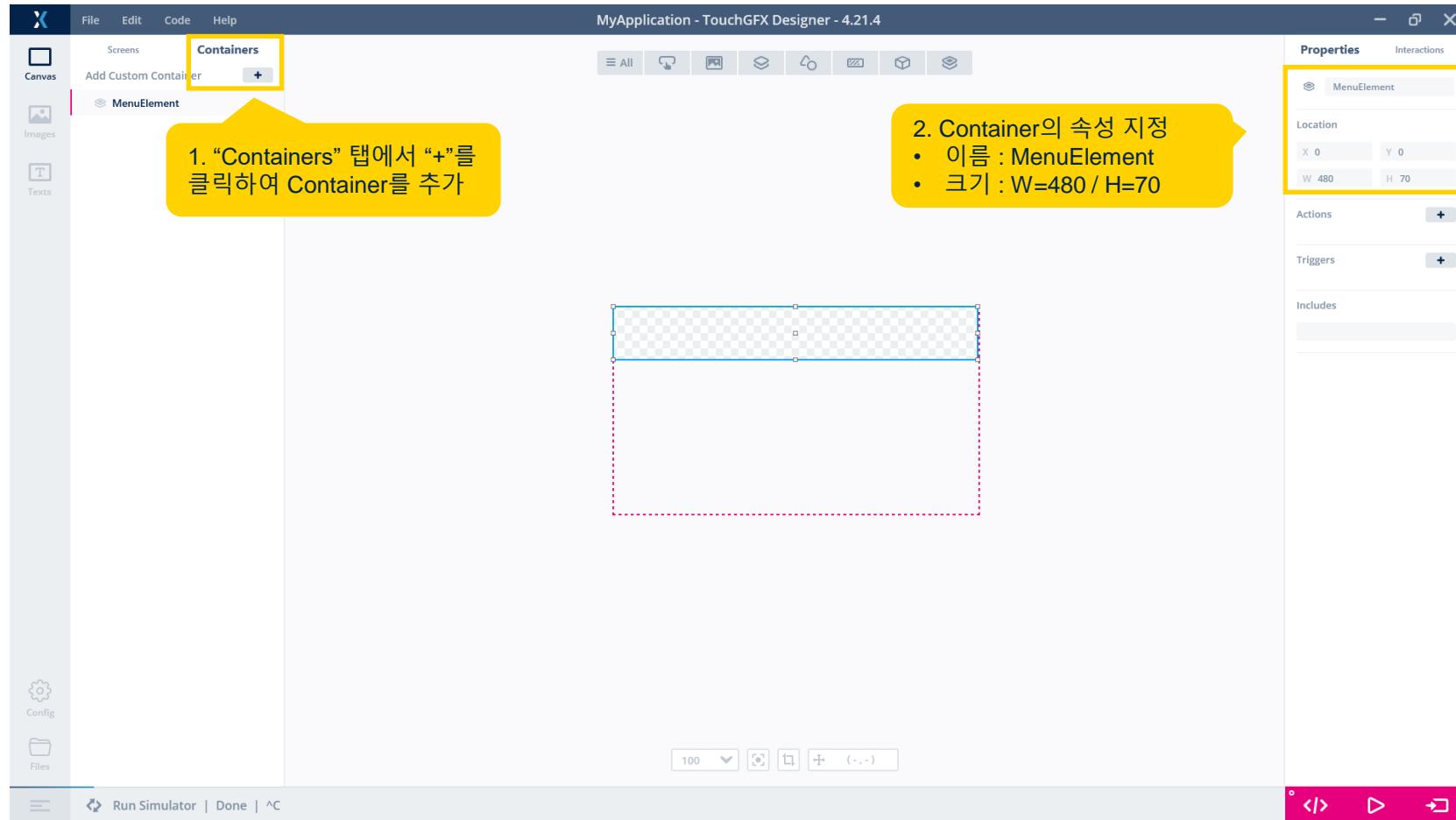
튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 사전준비 : Swipe Container 추가



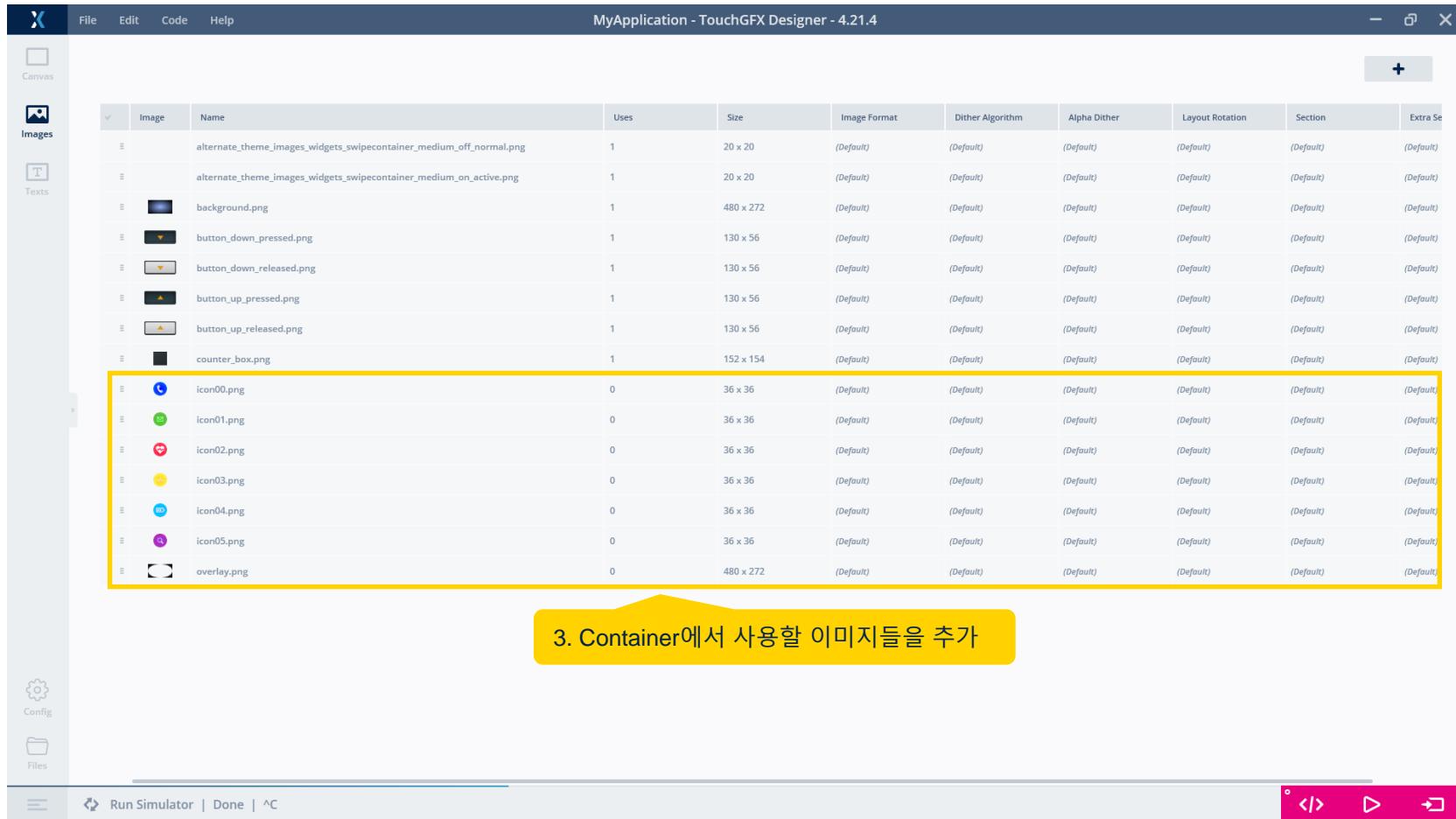
튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 1단계: Custom Container 생성하기



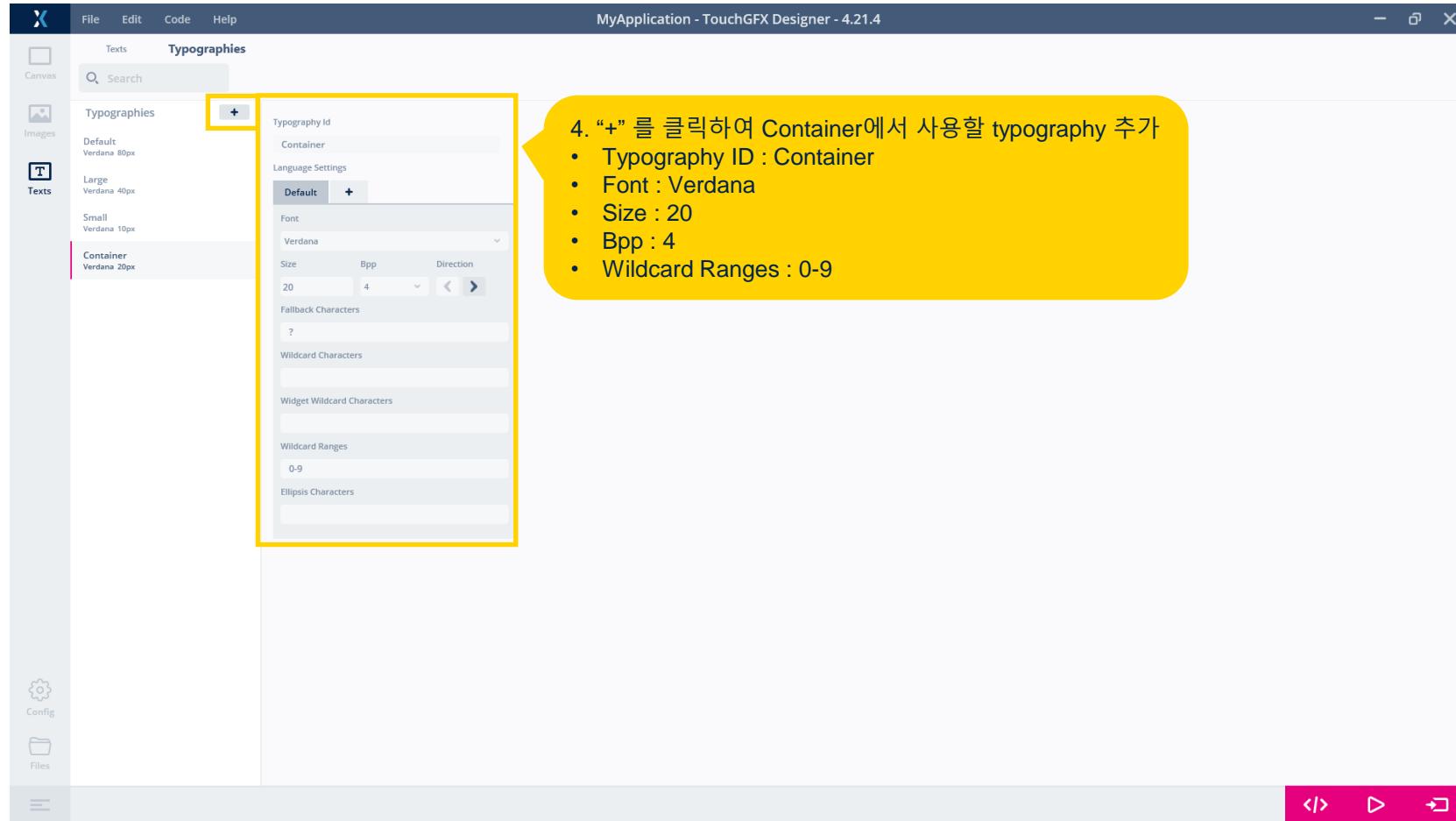
튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 1단계: Custom Container 생성하기



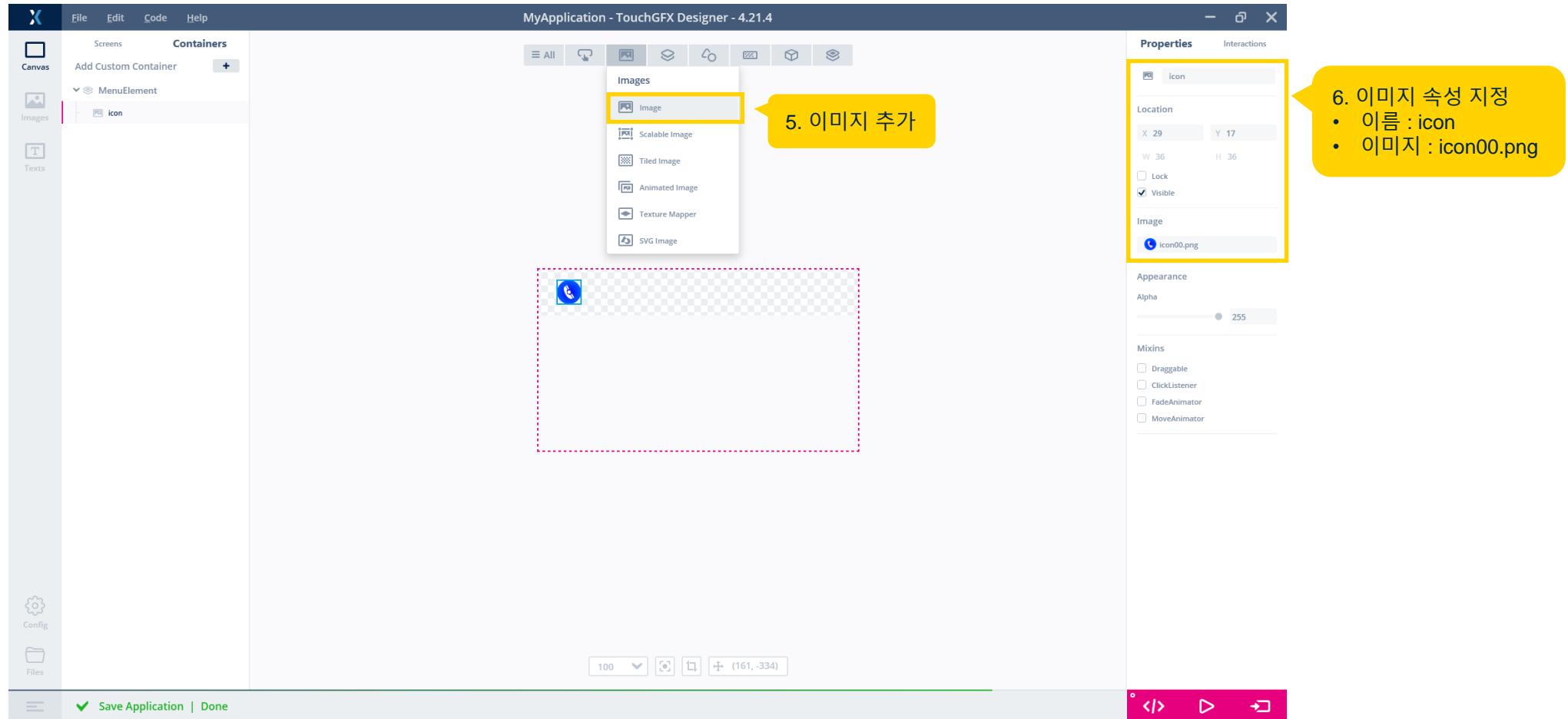
튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 1단계: Custom Container 생성하기



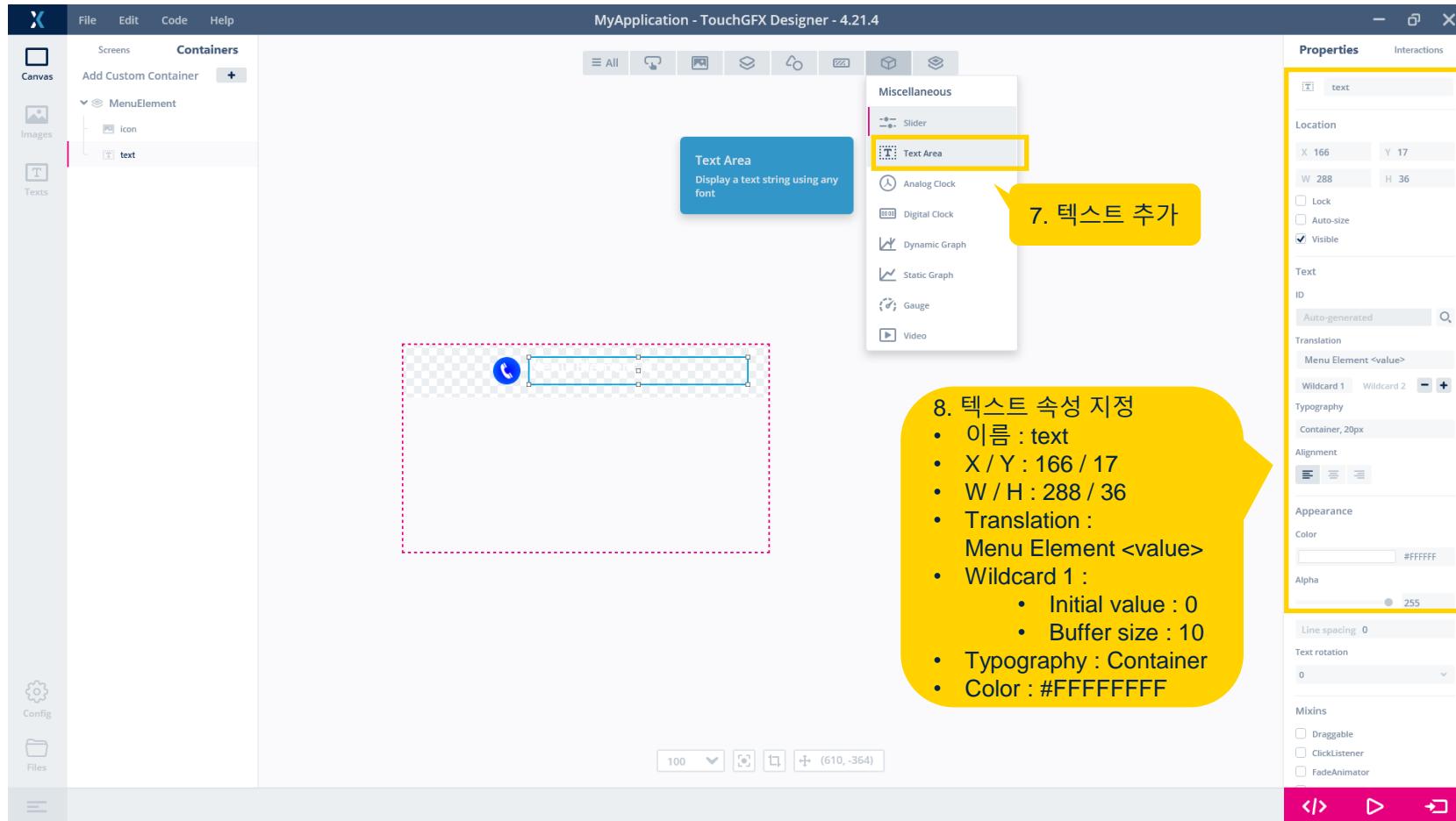
튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 1단계: Custom Container 생성하기



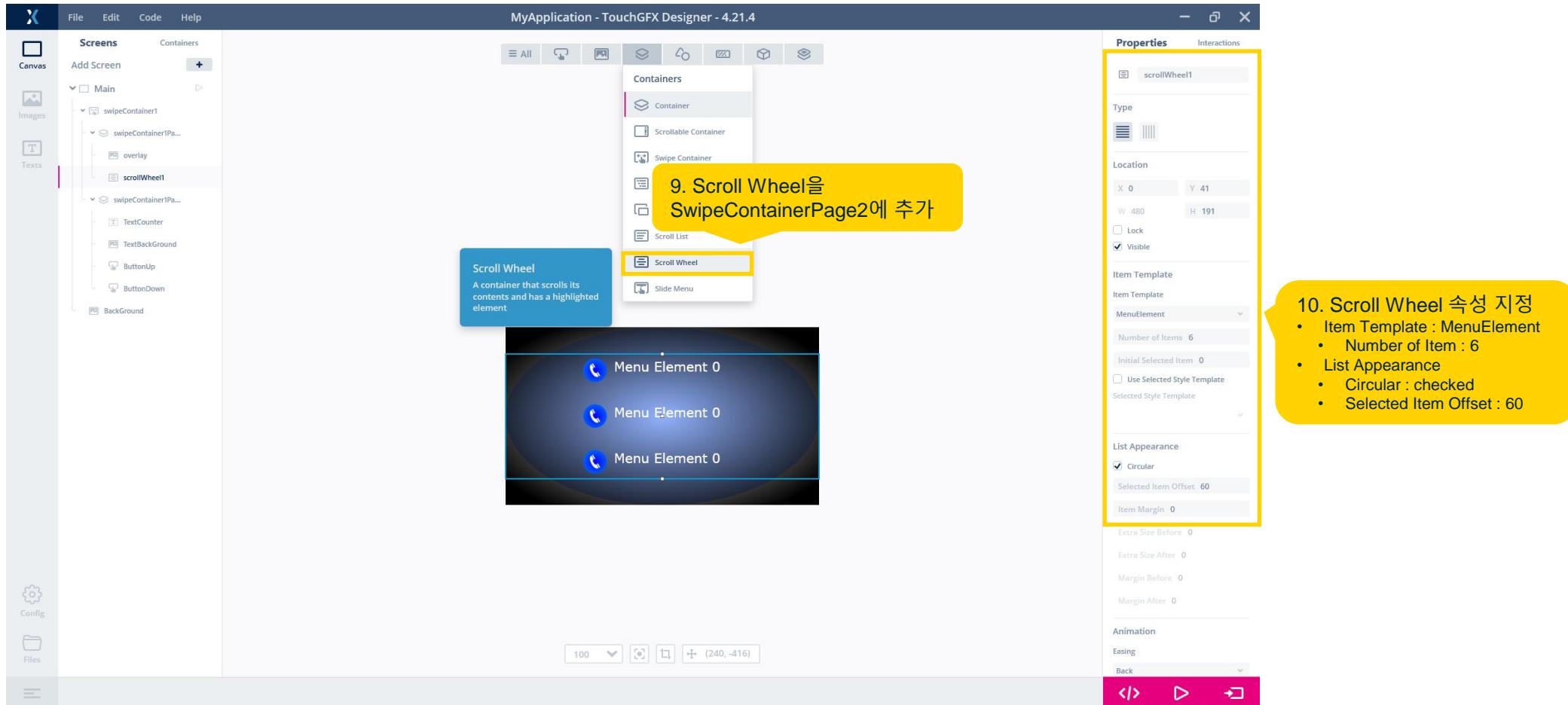
튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 1단계: Custom Container 생성하기



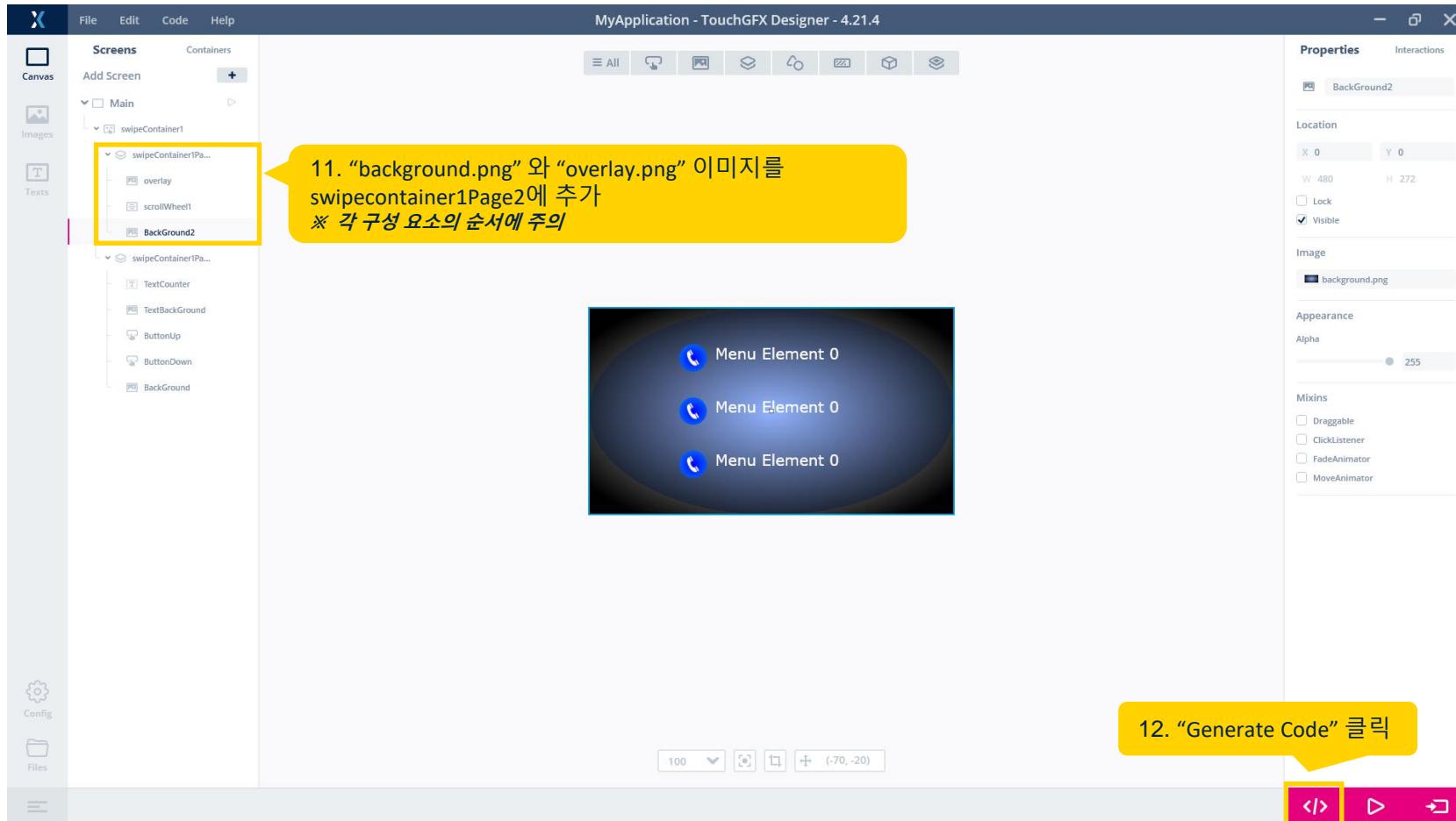
튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 2단계: Scroll Wheel 생성하기



튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 2단계: Scroll Wheel 생성하기



튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 3단계: 사용자 코드를 Scroll Wheel에 추가하기

```
..\TouchGFXTraining\TouchGFX\gui\include\gui\containers\MenuElement.hpp
```

```
#include <BitmapDatabase.hpp>
...(생략)
class MenuElement : public MenuElementBase
{
public:
...(생략)
    void setNumber(int no)
    {
        Unicode::itoa(no, textBuffer, TEXT_SIZE, 10);
        switch (no % 6)
        {
            case 0:
                icon.setBitmap(Bitmap(BITMAP_ICON00_ID));
                break;
            case 1:
                icon.setBitmap(Bitmap(BITMAP_ICON01_ID));
                break;
            case 2:
                icon.setBitmap(Bitmap(BITMAP_ICON02_ID));
                break;
            case 3:
                icon.setBitmap(Bitmap(BITMAP_ICON03_ID));
                break;
            case 4:
                icon.setBitmap(Bitmap(BITMAP_ICON04_ID));
                break;
            case 5:
                icon.setBitmap(Bitmap(BITMAP_ICON05_ID));
                break;
        }
    }
...(생략)
```

11. `setNumber(int no)` 함수를 추가하여
“scrollWheel” 항목에서 텍스트와 아이콘을 변경.
“no” 변수를 통해 이미지 위젯에서 표시할 아이콘을
결정하고 TextArea 위젯의 Wildcard를 no 값으로 변경

튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 3단계: 사용자 코드를 Scroll Wheel에 추가하기

```
..\TouchGFXTraining\TouchGFX\gui\include\gui\containers>MainView.hpp

#ifndef MAINVIEW_HPP
#define MAINVIEW_HPP

#include <gui_generated/main_screen/MainViewBase.hpp>
#include <gui/main_screen/MainPresenter.hpp>

class MainView : public MainViewBase
{
public:
    MainView();
    virtual ~MainView() {}
    virtual void setupScreen();
    virtual void tearDownScreen();

    virtual void buttonUpClicked();
    virtual void buttonDownClicked();

    virtual void scrollWheel1UpdateItem(MenuElement& item, int16_t itemIndex)
    {
        item.setNumber(itemIndex);
    }

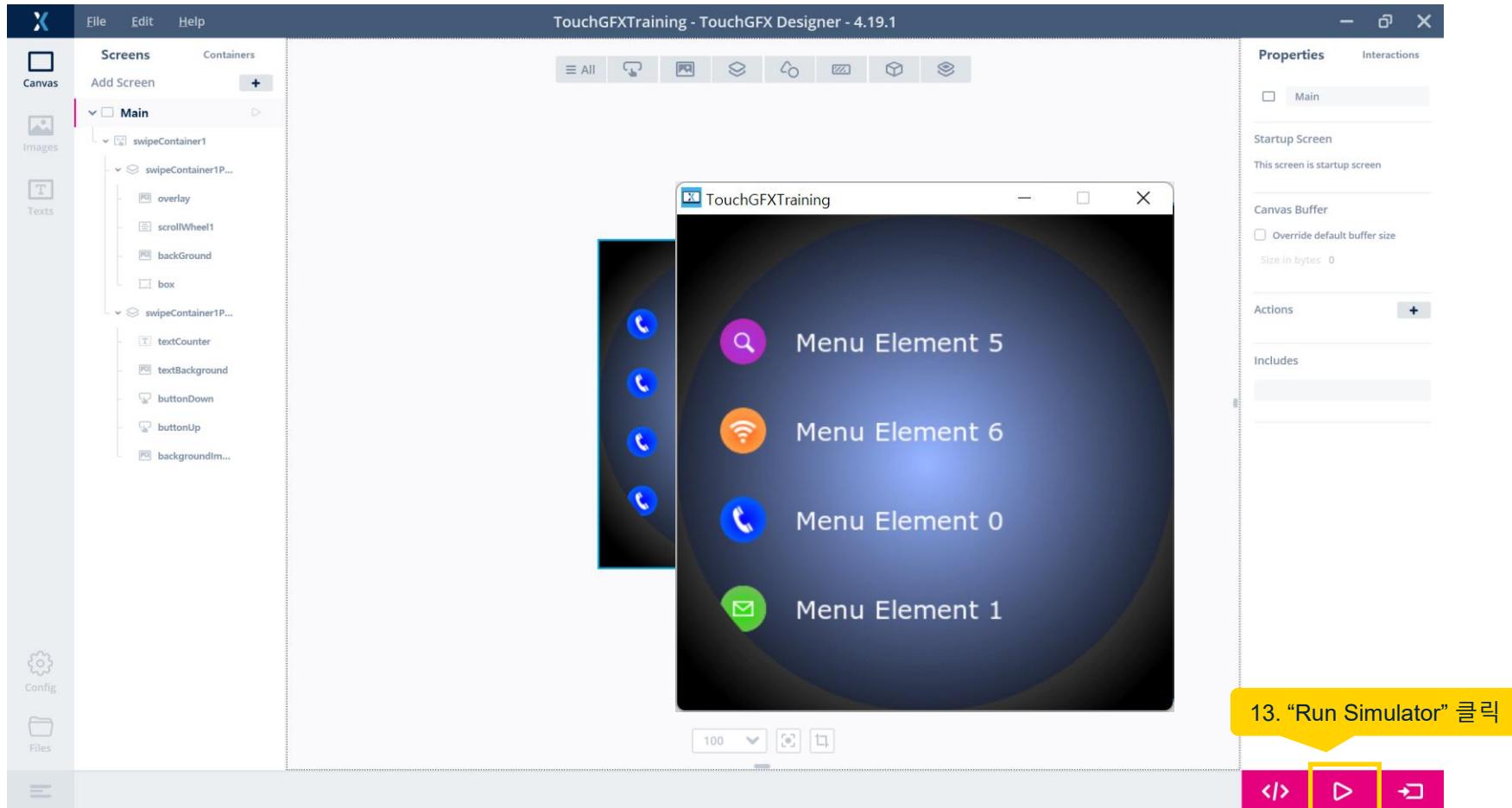
protected:
    int counter;
};

#endif // MAINVIEW_HPP
```

12. Scroll Wheel에 새로운 항목이 표시될 때마다
호출되는 함수를 선언.
“itemIndex”는 현재 업데이트 중인 항목을 알려주는
인덱스이며, “item”은 현재 Scroll Wheel에 표시되는
MenuElement 인스턴스에 대한 참조.

튜토리얼3 : 커스텀 동작을 추가하여 Scroll Wheel 생성하기

- 3단계: 사용자 코드를 Scroll Wheel에 추가하기



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.