



life.augmented

# TouchGFX

## Technical Introduction

This presentation is available on [support.touchgfx.com](https://support.touchgfx.com)

# Agenda

1 Introduction

5 Advanced TouchGFX  
Development

2 What Is TouchGFX ?

3 TouchGFX Development

4 Starting a TouchGFX project

# Introduction



life.augmented

## Goal of this presentation

- Getting to know TouchGFX and the workflow of a GUI project
  - Introduction of required tools
  - Understanding the development steps
  - Discovering the workflow

Target audience: Developers new to TouchGFX

## Further reading

- You will find a lot of help afterwards in the TouchGFX documentation site:

<http://support.touchgfx.com/>

- Slides in this presentation will refer to relevant documentation pages. Links will be in the lower right-hand corner of the slides
- A good place to start reading following this workshop is:

[Getting started – What's next](#)

## TouchGFX Development

### Main Activities:



### Main components:



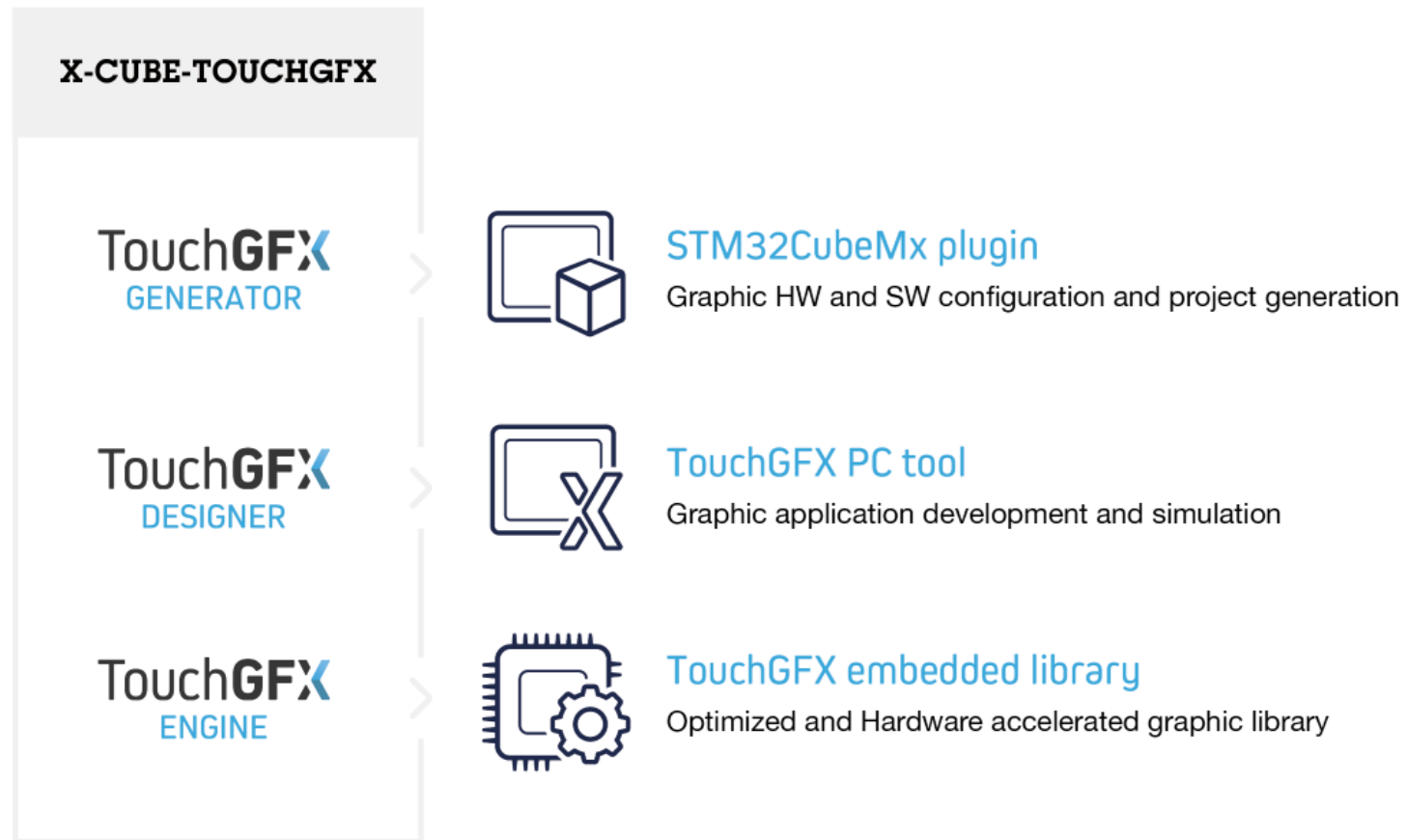
- In this presentation we introduce TouchGFX tools and the TouchGFX project development workflow
- We go deeper into the different main activities in other presentations

# What Is TouchGFX?



# TouchGFX X-Cube Package

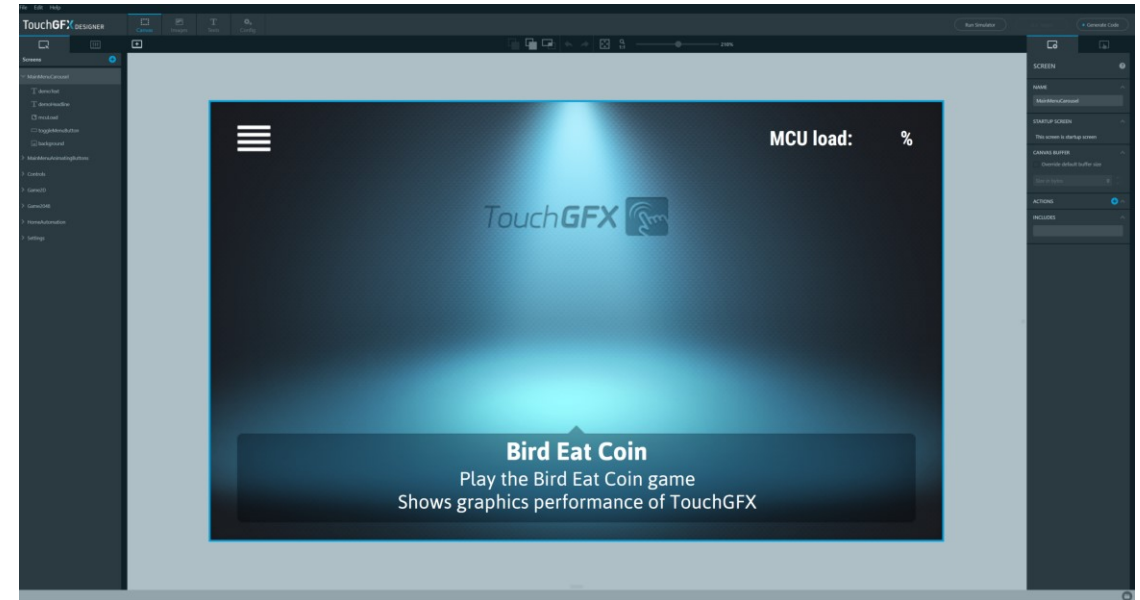
- TouchGFX is a GUI library delivered as a X-Cube package and consists of three components:





# TouchGFX Introduction

- TouchGFX is a fast and easy environment for developing advanced embedded user interfaces with high-end graphics on a wide range of STM32 microcontrollers
- TouchGFX Designer offers drag-and-drop programming with automatic C++ code generation
  - Developers can add C++ user code for additional configuration and UI development



*Canvas view of a TouchGFX project in TouchGFX Designer*

# Tools Required

To be able to create GUI projects on STM32 evaluation kits, you need to download:

- TouchGFX Designer available in the X-Cube-TouchGFX package
  - the TouchGFX PC tool is needed to start developing GUI prototypes
- STM32CubeProgrammer
  - a free software tool for programming STM32 products. It is required in order to flash projects on a board from the TouchGFX Designer



It is recommended to download ST-Link utility in addition to STM32CubeProgrammer to flash some of the oldest projects

See the [Installation Guide](#) for help

# Tools for Custom Product Development

To create GUI projects on custom STM32-based platforms or configure the hardware, it is required to download:

- CubeMX, a graphical tool used to configure STM32 microcontrollers
  - The X-Cube-TouchGFX package is installed through the Software Packs in CubeMX. Follow the *Custom Product Development* section of the documentation



Optional:

- Use your favorite IDE to execute projects on target outside of the Designer
  - All application templates available within TouchGFX Designer include project files and libraries for the toolchains STM32CubeIDE, Keil uVision and IAR Embedded Workbench

# TouchGFX Development

# TouchGFX Development

## Main Activities:



## Main components:



- A TouchGFX project involves a set of activities that need to be addressed during the development phase
- A TouchGFX project is made up of five main software and hardware components
  - Each activity will generate one of the main components

# Hardware Selection

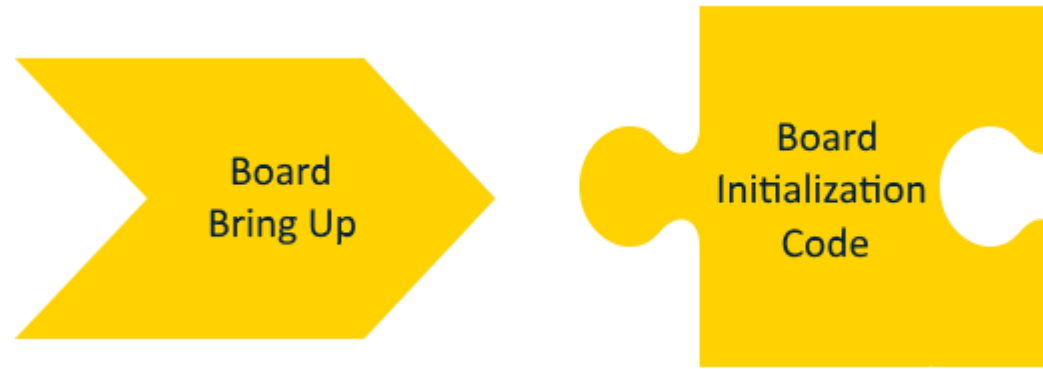
- Activity and Component:



- First step is to evaluate different hardware for GUI applications
  - Different STM32-based display boards can be used to evaluate and prototype
- Select the hardware on which the application will run on
- The selected hardware components and PCB design can influence the GUI application.

# Board Bring Up

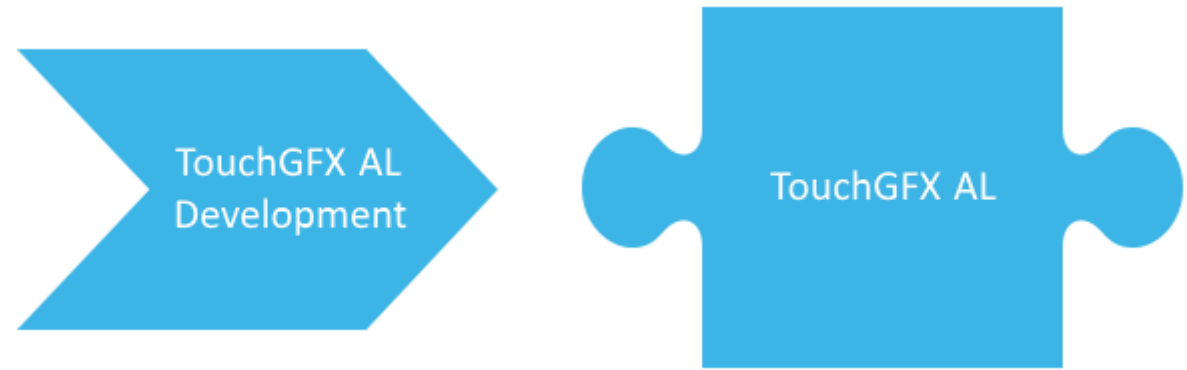
- Activity and Component:



- This activity is a central task to enable execution of TouchGFX on the board
- The output component is a general initialization code that sets up the MCU and all peripherals
  - This step is independent of TouchGFX. It purely handles hardware setup

# TouchGFX AL Development

- Activity and Component:

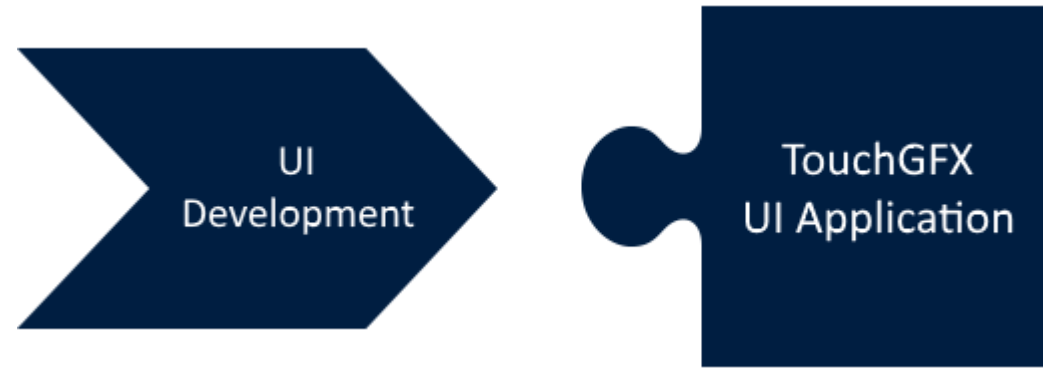


- This activity is key to making the TouchGFX Engine run on top of the initialized display board
- The output component is a software layer called TouchGFX Abstraction Layer (AL).It enables the TouchGFX Engine to run on the board
  - The main tool is the TouchGFX Generator which configures and generates most of the TouchGFX AL code



# TouchGFX UI Development

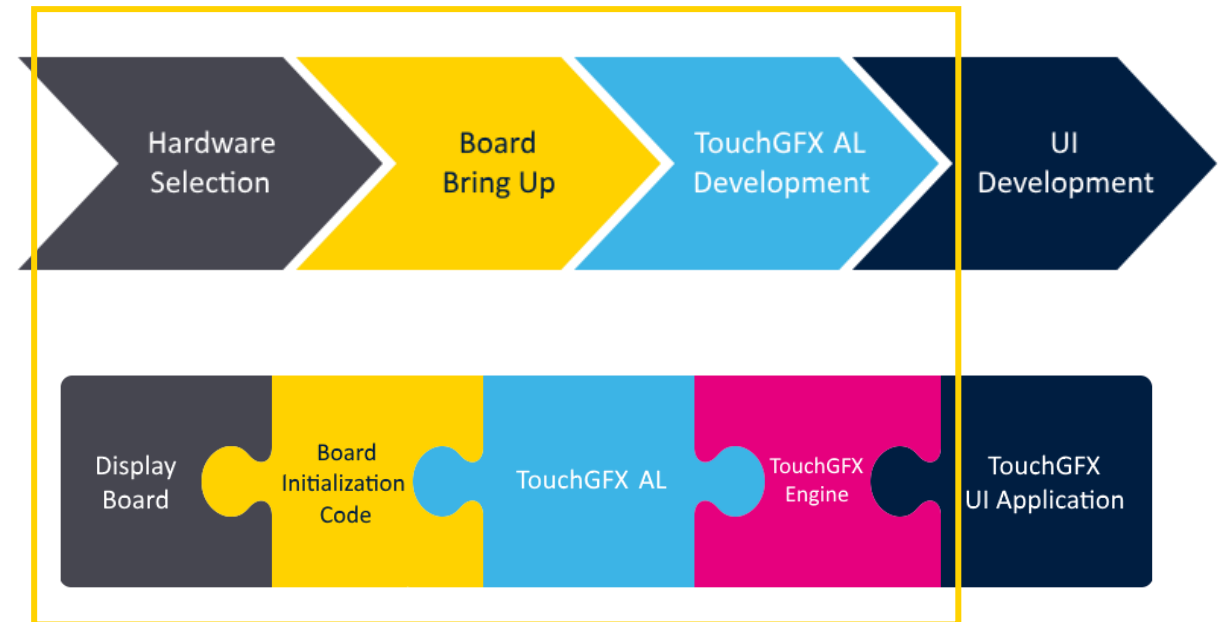
- Activity and Component:



- This activity is where the User Interface code is created. This is what will make up the visual part of the TouchGFX project
- The output component is the TouchGFX UI App. The main tools in this activity are TouchGFX Designer and an IDE or text editor
  - TouchGFX Designer will help start the setup, design and creation of the screens of the application. It will generate the main part of the UI application as C++ code

# Working with an Application Template

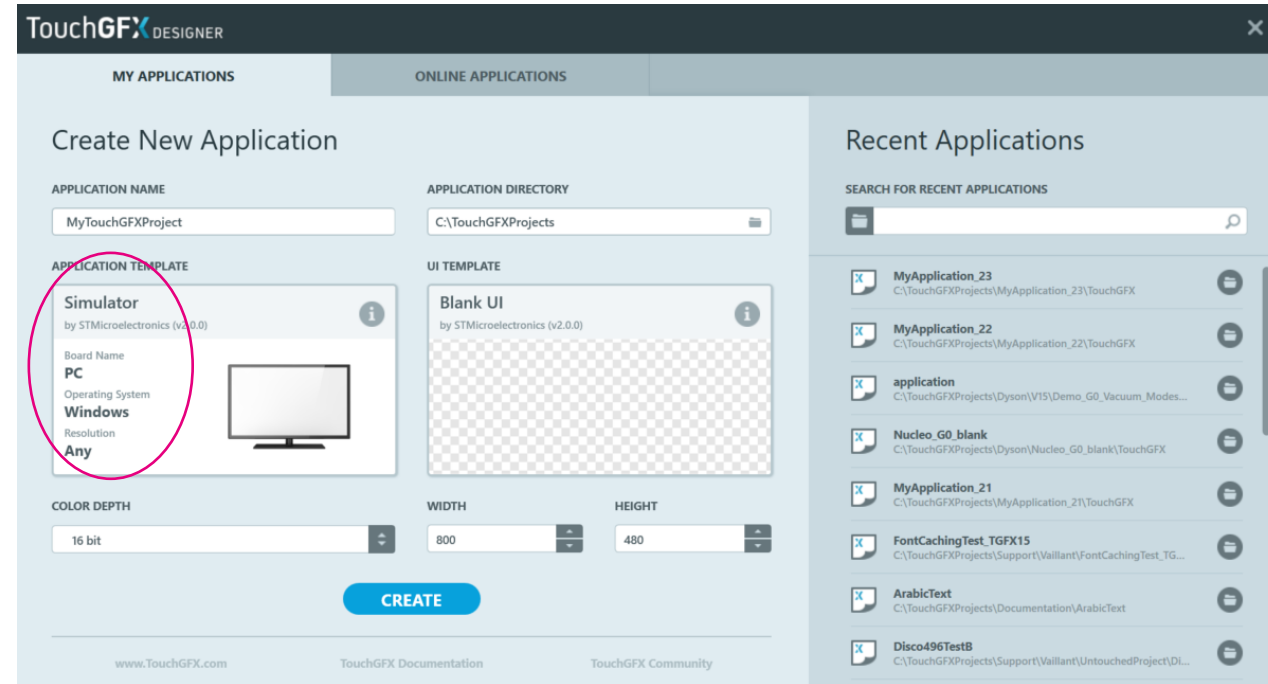
- In TouchGFX Designer users can start from a pre-configured project by selecting an application template of a supported STM32 based display kit
- TouchGFX Application templates are already configured for the first three activities:
  - A display board
  - Board initialization code
  - TouchGFX Abstraction Layer
- Users can start building their UI



# Starting a TouchGFX project

# Getting started with Designer

- It is possible to start a project from scratch or to select from a wide range of pre-made UI demos and ST kit application templates
- If the aim is to test TouchGFX or develop a UI without any hardware platform, the project can be created and run on the PC Simulator
  - The simulator can be used for UI development in parallel to the hardware development

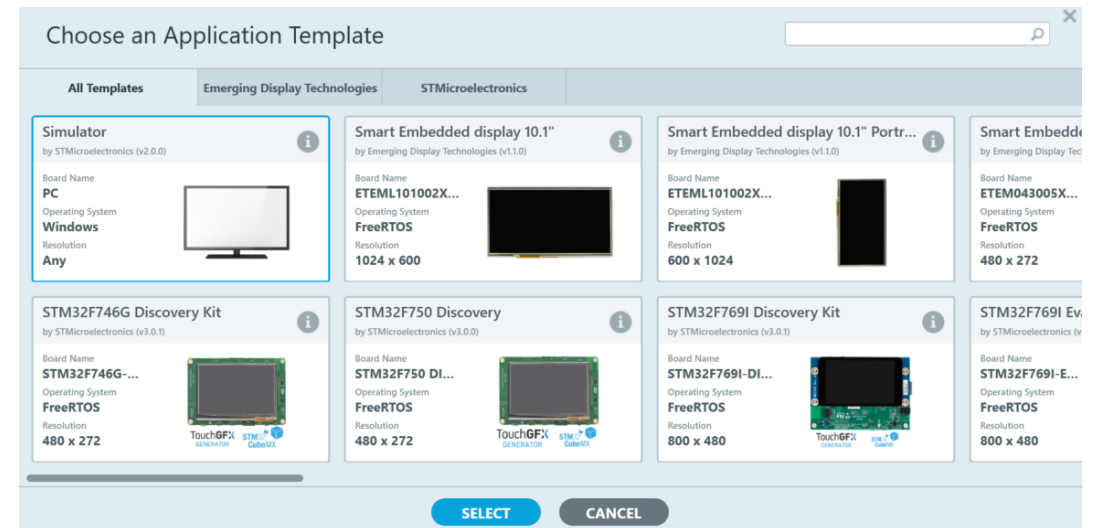
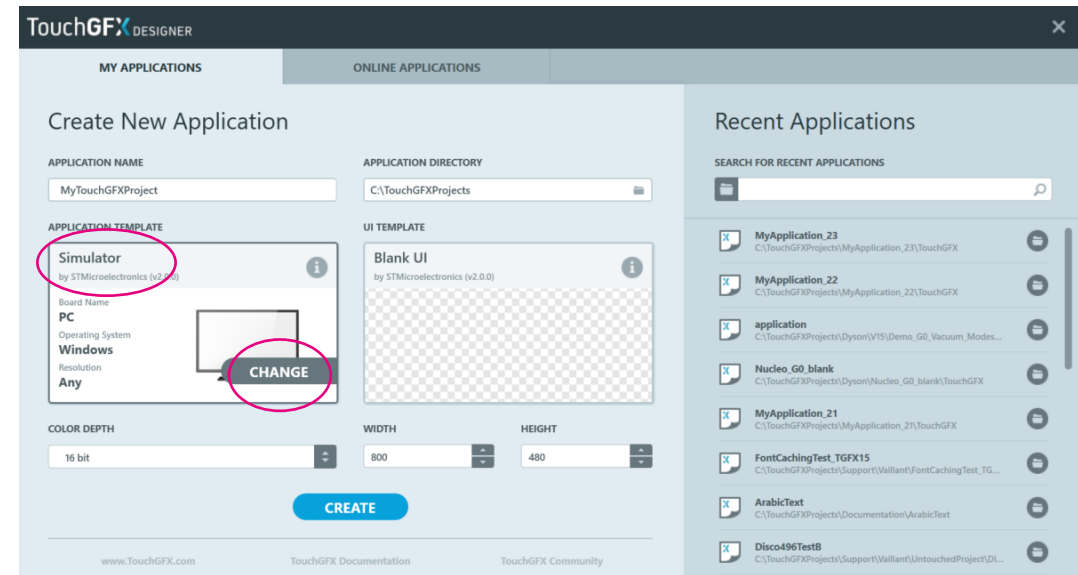


*Startup screen in TouchGFX Designer*

# Selecting an Application Template

It is possible to use preconfigured application templates for some of ST display kits and partner's platforms

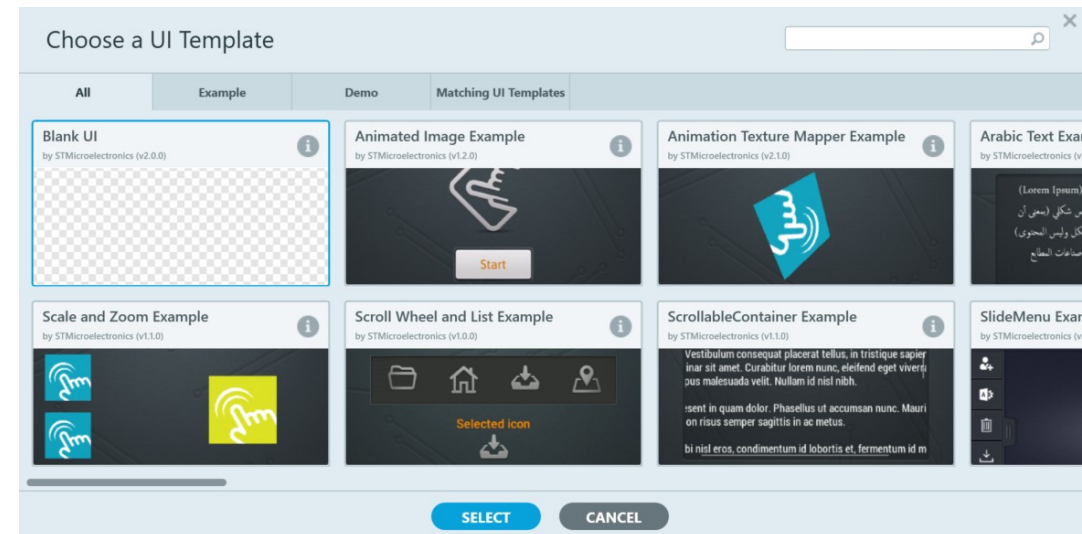
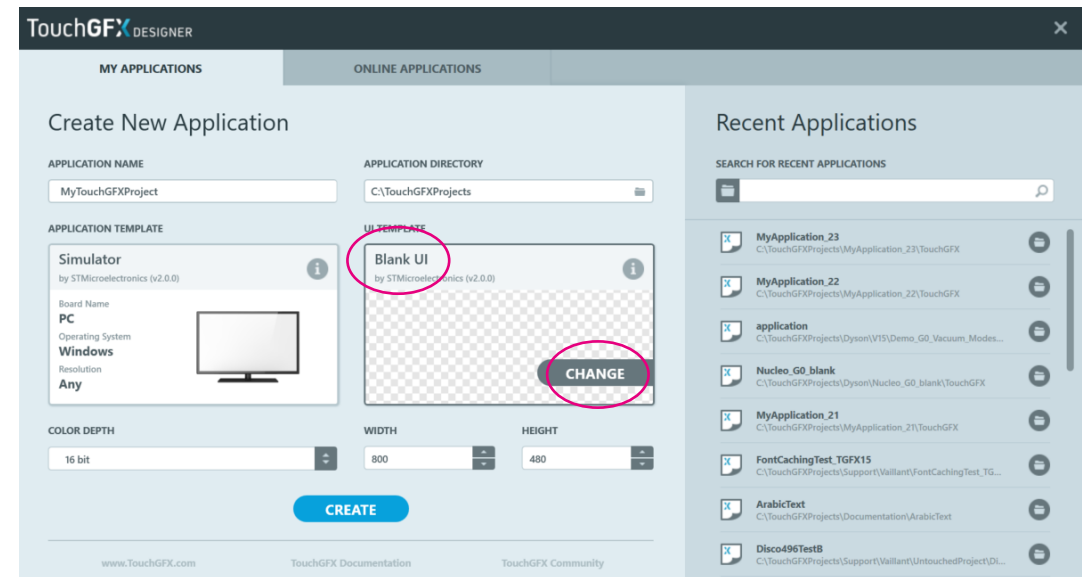
- It is possible to add custom application templates for STM32-based platform



# Selecting a UI Template Setup

The Graphical User Interface can be started from scratch or using pre-configured GUI examples and demos

- The UI examples can be used to understand the concept and functionalities of TouchGFX Designer. They can be used as a start or as inspiration for user code

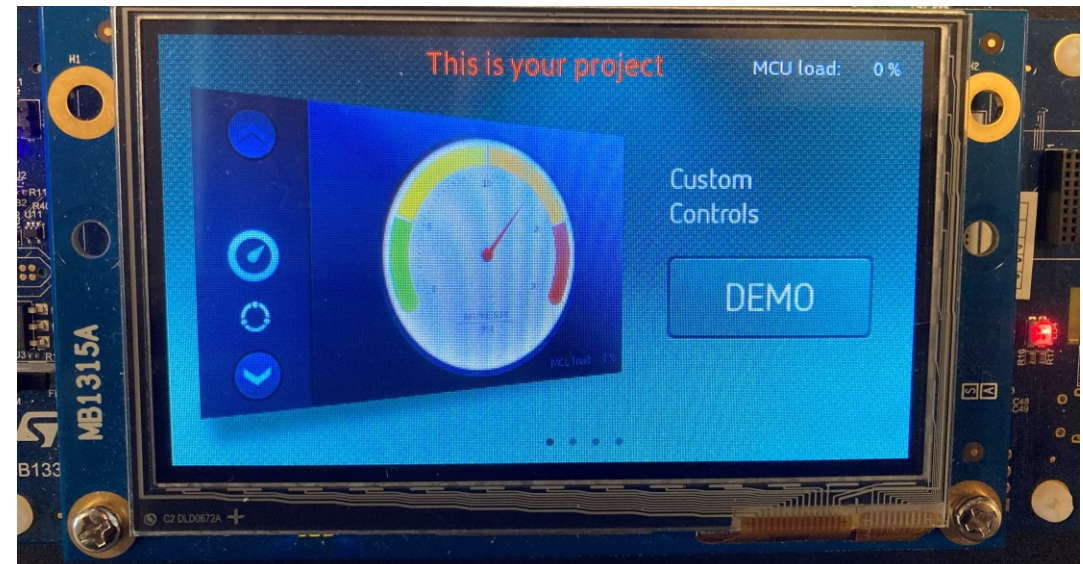
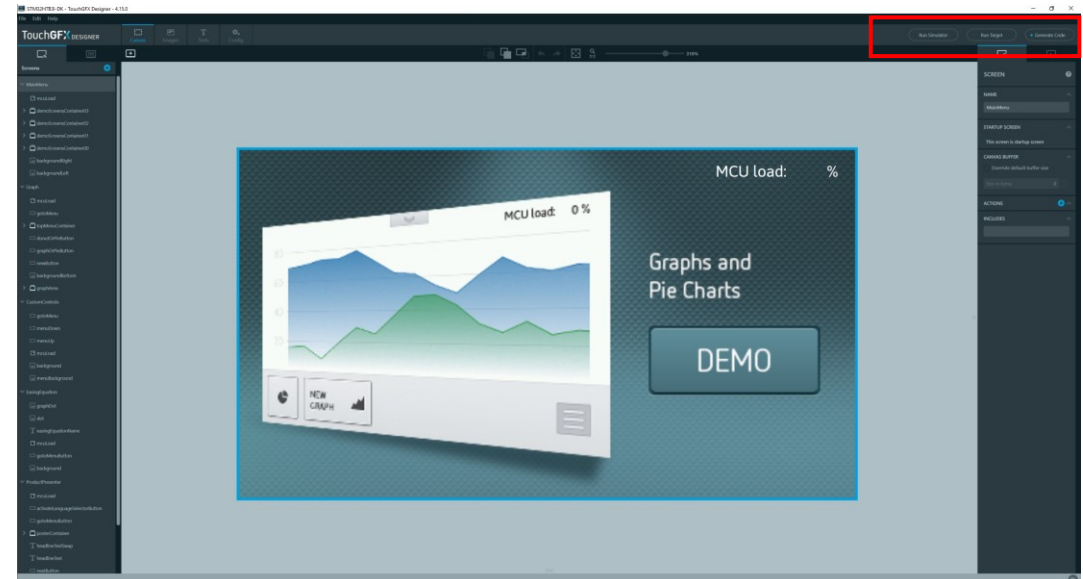


UI Template tab selection

# Creating and Running a Project

After creating a project and entering the main window, the user can directly run the UI with the simulator or on a board within TouchGFX Designer

- The simulator will not reproduce the hardware limitations and configuration errors



Flashing the project on the board from TouchGFX Designer



# Working with STM32CubeIDE

Application Templates for STM32-based kits will use STM32CubeIDE as the preselected toolchain

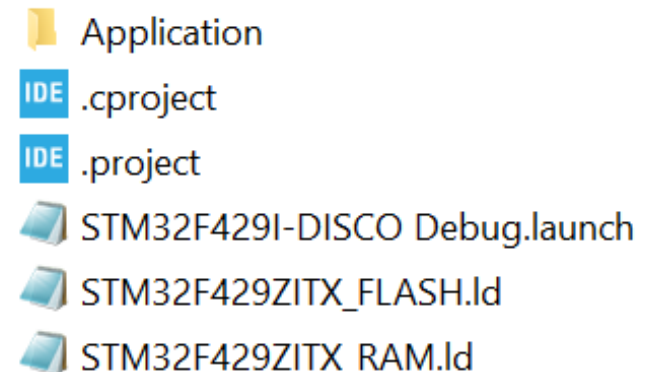
- CubeIDE is a free software tool used for programming STM32 products
- Projects can be flashed on target directly from this tool

The project's folder has a generated STM32CubeIDE project

- Opening .project or .cproject files will open the TouchGFX project in the toolchain



*STM32CubeIDE logo*



*CubeIDE project path: ProjectName/STMCubeIDE*



# Advanced TouchGFX Development

# Modifying the UI through Code

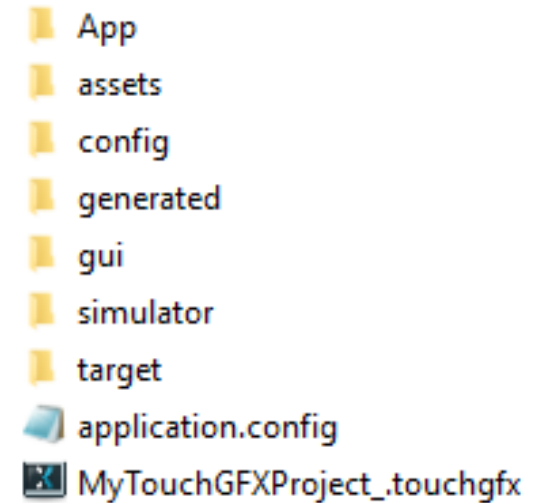
TouchGFX Designer will generate C++ code . Users can write extending code to create the logic behind the GUI project

- User code needs to be implemented in the /gui folder or in the identified USER CODE sections
- As a rule, TouchGFX Designer will only regenerate files within the generated folder: the code generated is by design not allowed to restrict the user in his process doing TouchGFX applications
- UI templates are a good starting point to understand how to work with both generated and user code

# TouchGFX folder

TouchGFX Designer will generate C++ code. Users can write extending code to create the logic behind the GUI project

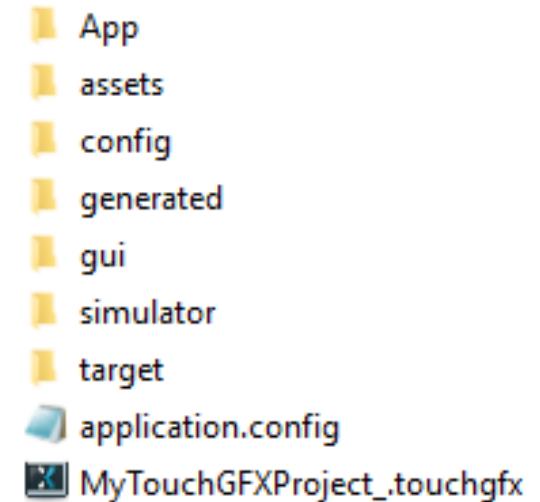
- User code needs to be implemented in the GUI folder
- After creation, TouchGFX Designer will only regenerate files within the generated folder
- A UI template is a good starting point to understand how to work with both generated and user code



*Generated folder of a TouchGFX project with an application template*

# TouchGFX folder

- The assets folder is where images, text and fonts are stored
  - Assets can be manually added through this folder or directly from TouchGFX Designer
- The App, config and simulator folders should not be modified
  - Those folders and files are configured to be able to work with CubeMX, other toolchains and with the simulator available in TouchGFX Designer
- The target folder is where the TouchGFXHAL, GPIOs and the touch controller are configured
  - The folder follows the same logic with generated files and other configurable files



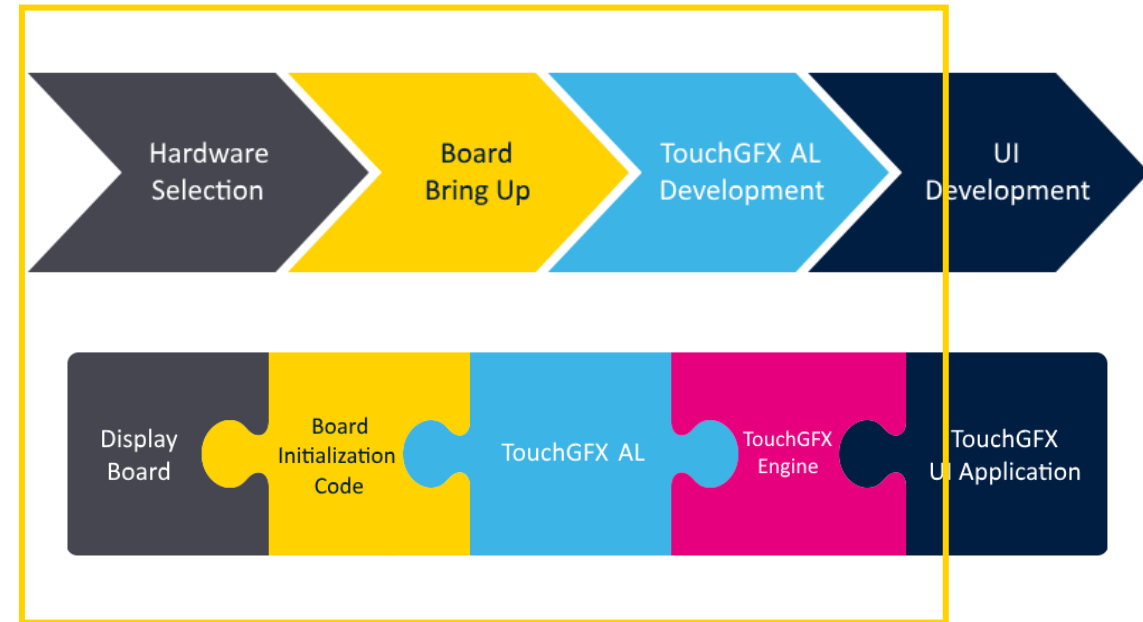
*TouchGFX project path:  
ProjectName/TouchGFX*

# Custom Board Configuration

Hardware and MCU configuration for STM32-based custom boards involve the first three activities of a TouchGFX project

CubeMX is a graphical tool that will allow users to easily configure an MCU and generate general startup code

- Peripherals such as external RAM will require additional initialization code and specific drivers
- Users can set the toolchain to work within the Project Manager's tab in CubeMX
- The ST kits application templates can be used as inspiration to help configure a custom platform



# Starting a Project from CubeMX

It is recommended to start a project from CubeMX only when working on a custom hardware solution. The steps to follow to start a project from CubeMX from scratch is explained in the [Board Bring Up presentation](#) and [Documentation](#)

It is possible to work on the GUI and the board bring up in parallel:

- one team configures the hardware and TouchGFX AL development through CubeMX
- another team develops the GUI on TouchGFX Designer by working on the simulator

# Thank you

© 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](http://www.st.com/trademarks).

All other product or service names are the property of their respective owners.



life.augmented

For further support and content,  
**you can visit the TouchGFX online  
documentation**  
<https://support.touchgfx.com>

