



The presentation is aiming at introducing TouchGFX. It will cover what is TouchGFX and the development workflow of a TouchGFX embedded GUI project .

The main chapters covered are:

(<https://support.touchgfx.com/docs/introduction/welcome>)

(<https://support.touchgfx.com/docs/development/development-introduction>)

This presentation takes approximately ~ 15 min

A presentation for understanding basic key concepts for Embedded Graphics before starting the project development can be the presentation:

(<https://support.touchgfx.com/docs/resources/presentations#embedded-graphics---basic-concepts>)

Presentations for getting started with UI development and TouchGFX can

either be the presentation:

UI Development – Fundamentals

(<https://support.touchgfx.com/docs/resources/presentations#ui-development---fundamentals>)

Or the workshop for a step-by-step guide:

UI Development - Getting Started

(<https://support.touchgfx.com/docs/resources/presentations#ui-development---getting-started>)

Agenda

- 1 Introduction
- 2 What Is TouchGFX ?
- 3 TouchGFX Development
- 4 Starting a TouchGFX project
- 5 Advanced TouchGFX Development



The agenda for the presentation is a short introduction to what is TouchGFX, to the workflow and the main activities to follow during development, and to how to get started with a TouchGFX project.

Introduction



Introduction

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



What to expect of this presentation: it is aimed to developers new to touchGFX and people interested in developing graphical user interfaces. The objective is to get to know TouchGFX and the workflow of a GUI (graphical user interface) project by discovering the tools and understanding the development steps

Introduction

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](#)



5

For further reading, you can find the slides in the TouchGFX online documentation

(<https://support.touchgfx.com/docs/resources/presentations#touchgfx-technical-introduction>). You can go through the other presentations, workshops and tutorials to understand how TouchGFX works and how to get started. (<https://support.touchgfx.com/docs/resources/presentations> and <https://support.touchgfx.com/docs/tutorials/tutorial-01>)

<https://support.touchgfx.com/docs/introduction/getting-started#whats-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

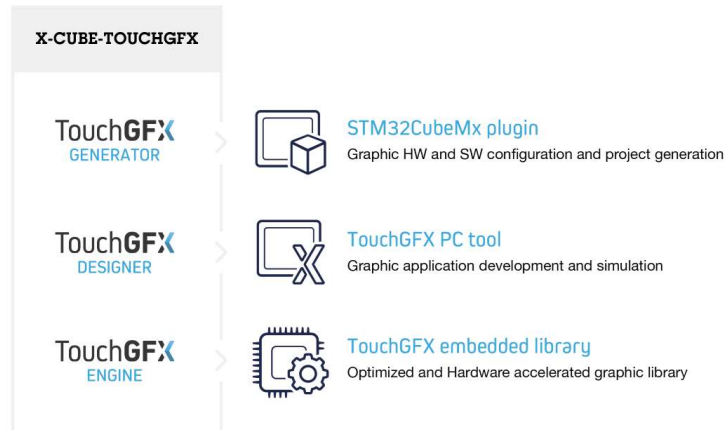
What Is TouchGFX?



life.augmented

TouchGFX X-Cube Package

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



[Documentation link - What is TouchGFX ?](#) 8

Components of TouchGFX X-Cube package:

TouchGFX Generator

STM32CubeMX plugin

TouchGFX Designer

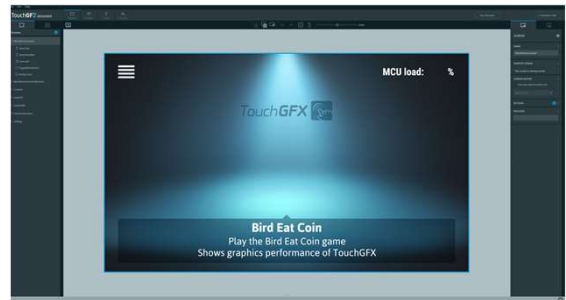
PC tool, for GUI development and simulation testing

TouchGFX Engine

Embedded library for STM32 mcu

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



life.augmented

[Documentation link - What is TouchGFX ?](#) 9

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

TouchGFX
TouchGFX logo

STM32
CubeProgrammer
STM32CubeProgrammer logo

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

[Documentation link - Installation](#) 10

Follow the installation guide to install the necessary tools :
<https://support.touchgfx.com/docs/introduction/installation>

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



[Documentation link - Installation](#) 11

TouchGFX Development



In this section will be explained the structure of a TouchGFX Project, the workflow and the tools involved in the entire project development cycle.

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



[Documentation link - Development Introduction](#) 13

A TouchGFX project involves a set of activities that will be addressed during the development phase. The effort in each of them depends on what the goal of the project is. For a UI prototype it is possible to use premade code for major parts of the application, speeding up the development by skipping most of these activities. For a full project based on a custom-made board, it is mandatory to address each of these activities.

A TouchGFX project is made up of five main software and hardware components. Each of the activities will generate one of the main component. The TouchGFX Engine is not an output of any main activity but is the starting point for a TouchGFX project.

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.



[Documentation link - Hardware Selection](#) 14

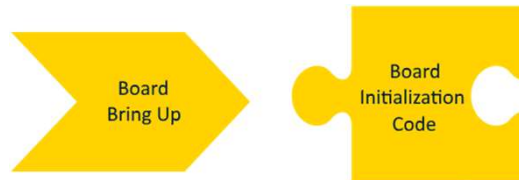
The hardware selection activity is the initial activity in a TouchGFX project.

For a UI prototype the Hardware Selection activity is only a matter of selecting an available STM32 Evaluation kit.

For a custom hardware solution there are a lot of choices to be made and issues to consider, like the MCU performance, memory setup and display size. The selected hardware components and PCB can influence the Graphical user interface application. If a board has not be set yet, it is possible to start the GUI development by just using the TouchGFX Simulator that runs on a PC.

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

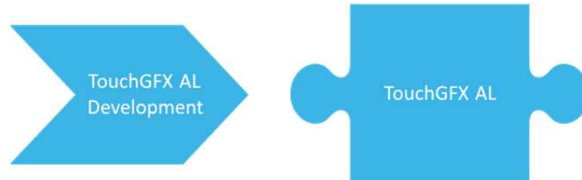


[Documentation link - Board Bring Up](#) 15

This activity is a central task to enable TouchGFX to be executed on your board. The output component is called Board Initialization Code which is a general initialization code that setups the MCU and all peripherals, preparing it for application execution. This code is independent of TouchGFX, as it only handles the pure hardware setup. The main tool for this activity is CubeMX, to configure the MCU and generate general startup code. For peripherals, such as external RAM and Display, additional initialization code and specific peripheral drivers need to be added by users. It is also possible to do the entire Board Initialization Code without use of CubeMX, but is not recommended unless you have expert knowledge on STM32 MCUs and board bring up

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



life.augmented

[Documentation link - AL Development](#) 16

This activity is key in making the TouchGFX Engine run on top of the fully initialized Display Board (Display Board + Board Initialization Code). The output component is called TouchGFX Abstraction Layer (AL). It is a software layer that is an abstraction of your hardware and enables the TouchGFX Engine to run on your board.

The main tool in this activity is TouchGFX Generator which is a CubeMX plugin that allows you to configure and generate most of the TouchGFX AL code.

It is important to notice that for the TouchGFX AL to work it is crucial that the Board Bring Up activity has been successfully configured.

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



[Documentation link - UI Development](#) 17

The UI Development activity is where the User Interface code is developed. This activity might take most of the project development time since this is what will make up the visible part of the TouchGFX project. The component is called the TouchGFX UI Application.

The main tools in this activity are TouchGFX Designer and an IDE or text editor. In TouchGFX Designer users will setup, design and create the screens of the application and generate main parts of the UI Application as C++ code. For the application logic (handling events, communicating with the non-UI part of the system) users will use an IDE or text editor to write C++ code, that will coexist and interact with the generated code from TouchGFX Designer.

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



[Documentation link - Application Templates](#) ¹⁸



life.augmented

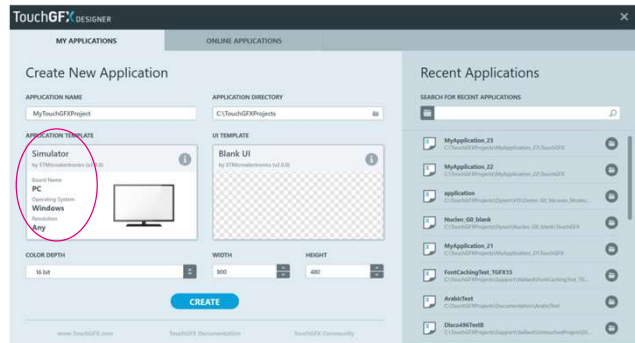
If the aim is to do a UI Prototype or just to try out TouchGFX the project can be created based on an existing Application Template (AT) for one of the standard STM32 Evaluation Kits available in TouchGFX Designer. These include all the Board Initialization and TouchGFX AL Code needed. The ATs are based on a CubeMX configuration, so it is possible for users to modify the configuration for experiments.

If you are doing a UI Prototype or just want to try out TouchGFX you can base your application on an existing Application Template (AT) for one of the standard STM32 Evaluation Kits available in TouchGFX Designer. This includes all the code you need. The ATs are based on a CubeMX and TouchGFX Generator configuration, so it is possible for you to modify the configuration if you want to experiment later on.

Please watch the video of this presentation for a live explanation running through the TouchGFX Designer tool. The following slides are a summary of the live presentation.

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



[Documentation link - Designer Quick Start 20](#)

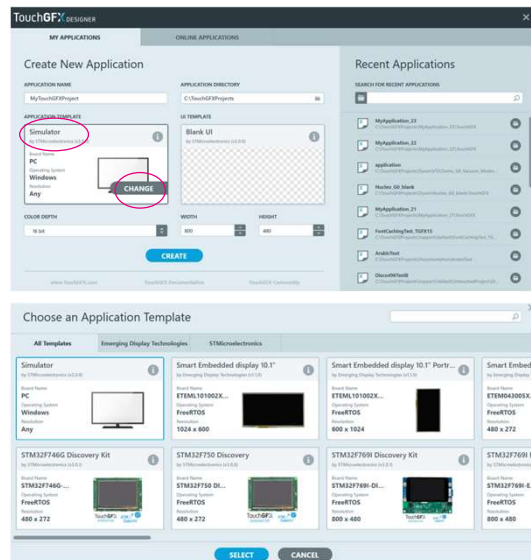
Designer User Guide documentation link:

<https://support.touchgfx.com/docs/development/ui-development/designer-user-guide/startup-window>

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



Application Template tab selection



life. augmented

21

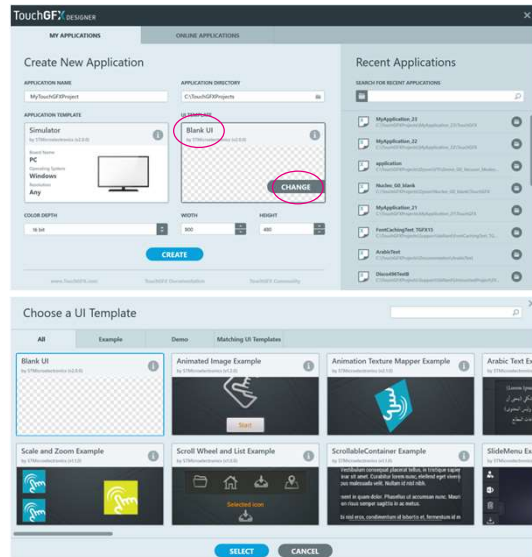
Application Templates documentation link :

<https://support.touchgfx.com/docs/development/ui-development/designer-user-guide/startup-window#application-templates>

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



life.ougmented

UI Templates documentation link :

<https://support.touchgfx.com/docs/development/ui-development/designer-user-guide/startup-window#ui-templates>

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



23

Running the Simulator documentation :

<https://support.touchgfx.com/docs/development/ui-development/working-with-touchgfx/simulator>

Compiling & Flashing on boards documentations :

<https://support.touchgfx.com/docs/development/ui-development/working-with-touchgfx/compiling-and-flashing>

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`

[Documentation link - Compiling & Flashing using IDEs](#) 24

Selecting an application template for STM32 based kits will generate a fully configured CubeMX project. The generated project folder also contains project files for GCC, CubeIDE, IAR and Keil. The active toolchain is set from CubeMX and is set to CubeIDE by default

Compiling & Flashing using IDEs documentation :

<https://support.touchgfx.com/docs/development/ui-development/working-with-touchgfx/compiling-and-flashing>

Advanced TouchGFX Development



life.augmented

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



[Documentation link - Creating Your Own Application](#) 26

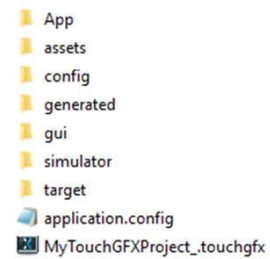
The tutorials available in the documentation are a good starting point to understand how to customize your project through code :

<https://support.touchgfx.com/docs/tutorials/tutorial-01>

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

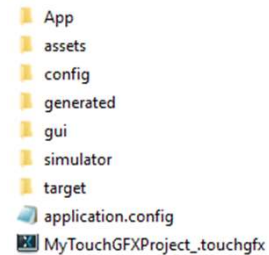


life.augmented

[Documentation link - TouchGFX Designer Project Structure](#) 27

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



[Documentation link - TouchGFX Designer Project Structure](#) 28

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



[Documentation link - Hardware Selection](#) 29

A cubeMX project is created when starting from an application template in Designer. CubeMX is a graphical tool that will allow users to easily configure an MCU and generate general startup code. To work with a different IDE than STM32CubeIDE, it is possible to modify the toolchain in the project's manager tab.

When working with a custom board it is necessary to go through the first three activity of a touchGFX project. The ST kits application templates can be used as inspiration to help configure the custom platform.

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](#)

The configuration of the MCU with the STM32CubeMX tool and the rest of the board bring up phase are not dependent on TouchGFX and should not involve any TouchGFX code. The goal is to make sure that your hardware and lower-level software is in fact working in accordance with your expectations for the final GUI application.

Overall Process guide : <https://support.touchgfx.com/docs/development/board-bring-up/board-introduction#overall-process>

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.

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



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

