



life.augmented

TouchGFX STM32 전용 무료 그래픽 솔루션

STMicroelectronics Korea MCD team

김두형 과장

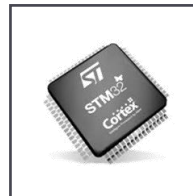
Accelerating the HMI of Things



Enabling high-end user experience in embedded devices
Smarter and richer devices requiring Advanced Graphic User Interfaces

STM32 graphics solutions

Enabling you to create high-end user experience in embedded devices



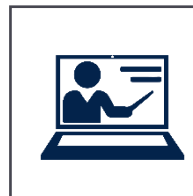
Advanced Graphic MCU Portfolio



State-of-the-art Graphic Software and Tools



Reference Designs and Worldwide Support



How to design TouchGFX with STM32CubeIDE

Advanced graphics MCU portfolio

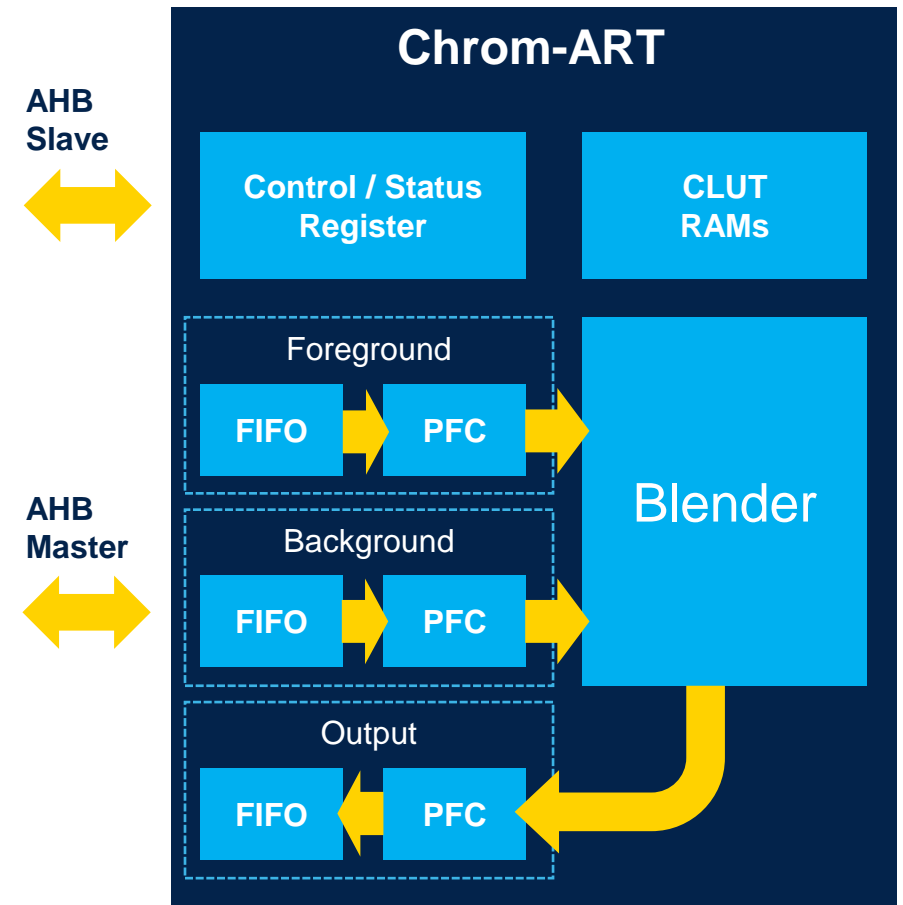




Advanced graphics MCU portfolio

Efficient 2D graphics acceleration for high-end transitions and effects

- **Offloads the CPU from repetitive graphics tasks**
 - Efficient 2D image copy
 - Transparency
 - Pixel format conversion
 - Efficient Fonts management





Advanced graphics MCU portfolio

Chrom-GRC™ for memory optimization

- **Chrom-GRC™**
 - Graphic Resources Cutter for non square displays
 - No modification nor special management at SW level.
 - Saving up to 20% of RAM needs



- For **360x360** round display
 - @16bpp : ~**205KB** (vs.253KB)
 - @24bpp : ~**307KB** (vs.380KB)
- For **400x400** round display
 - @16bpp : ~**250KB** (vs. 312KB)
 - @24bpp : ~**372KB** (vs. 469KB)

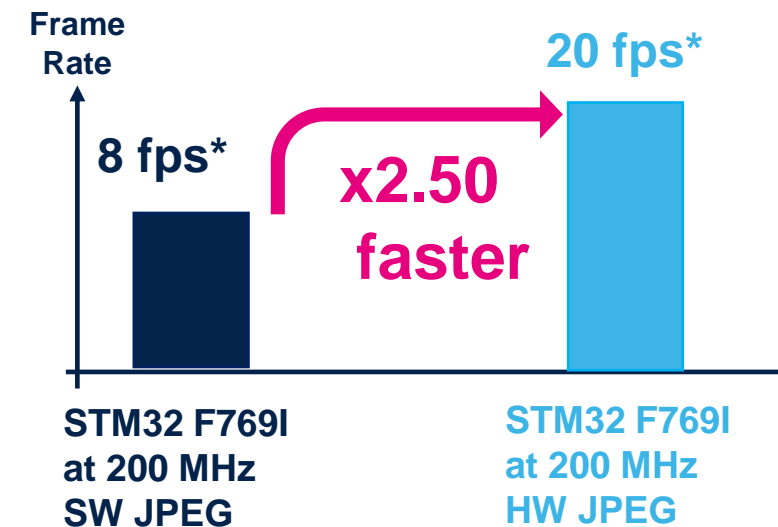
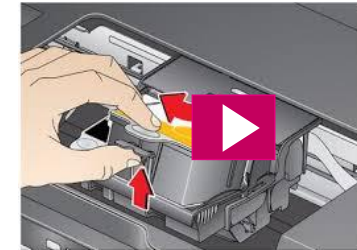


Advanced graphics MCU portfolio

MJPEG video acceleration for branding and tutorial videos

- **HW JPEG accelerator**

- Fast and simple hardware JPEG compression and decompression
- Full management of JPEG headers
- Supporting motion JPEG videos
 - Saving CPU load for MJPEG management
 - Enhancing branding and user experience
- Branding animations at startup
- End-product embedded tutorials



* : frames per second



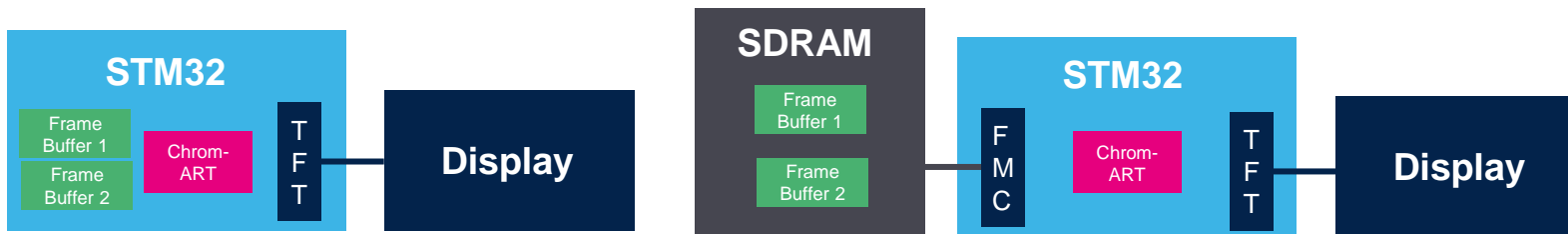
Advanced graphics MCU portfolio

Support for a wide range of display interfaces

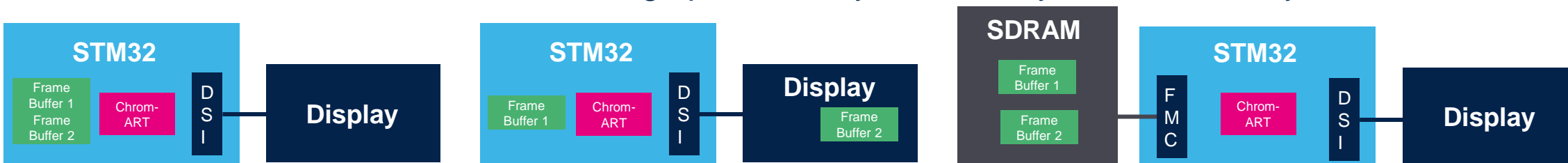
- SPI and Intel 8080 / Motorola 6800 LCD interfaces for small resolutions



- TFT controller for medium resolution (up to XGA)



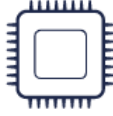

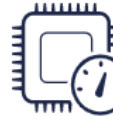


- MIPI-DSI interface for medium resolution, high pixel density GUI, mainly consumer today





Advanced graphics MCU portfolio

Different display interface support

 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™ MJPEG	Parallel LCD TFT MIPI-DSI	Up to 1024*768
STM32H7 (CM7)	480 MHz	Chrom-ART™ MJPEG	Parallel LCD TFT MIPI-DSI	Up to 1024*768

State-of-the art graphic software and tools





State-of-the-art graphic software and tools

TouchGFX – Unbeatable GUI performance on STM32



Maximum Performance

TouchGFX technology enables you to achieve the highest level of smartphone GUI performance on STM32 devices



Create Anything

The structure and flexibility of TouchGFX gives the developer control to easily create unique UI designs



Easy to Use

TouchGFX combines a WYSIWYG designer, auto code generation and a PC-simulator with the efficiency and flexibility of the C++ language



State-of-the-art graphic software and tools

TouchGFX – All you need to quickly start and achieve a high-end GUI

- **Two main development steps**
 - Running TouchGFX on your STM32-based board
 - Use TouchGFX Generator – an STM32CubeMX plugin that lets you to configure and generate the TouchGFX setup code
 - Develop the UI application
- **Advantage of the new TouchGFX Generator solution**
 - Intuitive and seamlessly interaction and workflow between CubeMX/TouchGFX Generator and the TouchGFX Designer.
 - An open solution – no restriction on selected IPs
 - All STM32 devices with Arm Cortex-M0+, M4 and M7 cores are supported
 - Support for special cases where custom code is required



State-of-the-art graphic software and tools



STM32CubeMx plugin

Graphic HW and SW configuration and project generation



STM32CubeMx plugin

Graphic HW and SW configuration and project generation



State-of-the-art graphic software and tools



STM32CubeMx plugin

Graphic HW and SW configuration and project generation

System Core	Packs	Version	Selection
Analog			
Timers	> ARM.CMSIS	5.6.0	
Connectivity	> STMicroelectronics.X-CUBE-AI	5.0.0	
	> STMicroelectronics.X-CUBE-BLE1	5.0.0	
Multimedia	> STMicroelectronics.X-CUBE-BLE2	1.0.0	
Security	> STMicroelectronics.X-CUBE-GNSS1	4.1.0	
	> STMicroelectronics.X-CUBE-MEMS1	7.1.0	
Computing	> STMicroelectronics.X-CUBE-NFC4	1.5.2	
Middleware	> STMicroelectronics.X-CUBE-SUBG2	1.1.0	
Trace and De	✓ STMicroelectronics.X-CUBE-TOUCHGFX	4.13.0	
	✓ Graphics Application	4.13.0	
Power and Tr	✓ Application		TouchGFX Generator

Smooth generation of TouchGFX project

- Helps create and configure your project on your custom hardware based on any STM32 MCU, making you ready to develop your UI application in TouchGFX Designer.

Seamless interoperability with STM32CubeMX

- When developing your project, you can now change project configuration in STM32CubeMX, which automatically updates the graphical settings in your project in TouchGFX Designer

IDE support

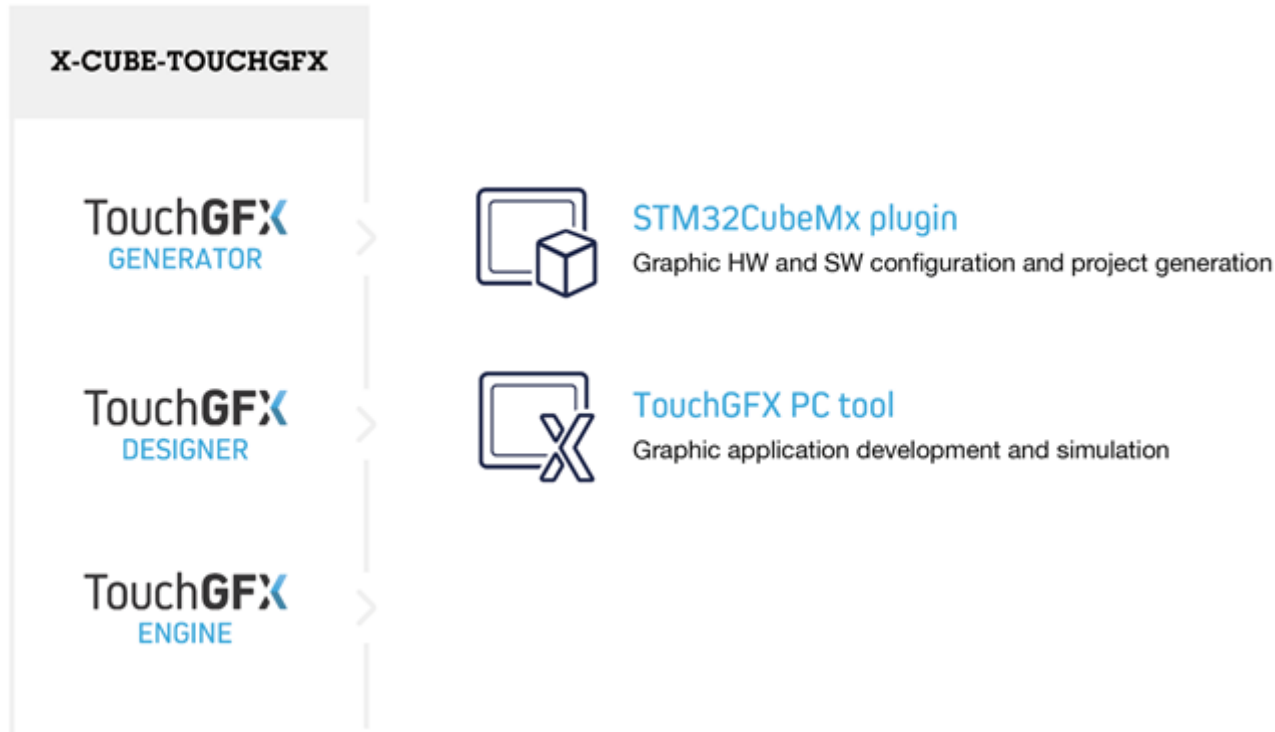
- You can select your preferred IDE (CubeIDE, IAR or Keil) in STM32CubeMX and the TouchGFX project files will be generated for your selected IDE.



TouchGFX PC tool

Graphic application development and simulation

State-of-the-art graphic software and tools



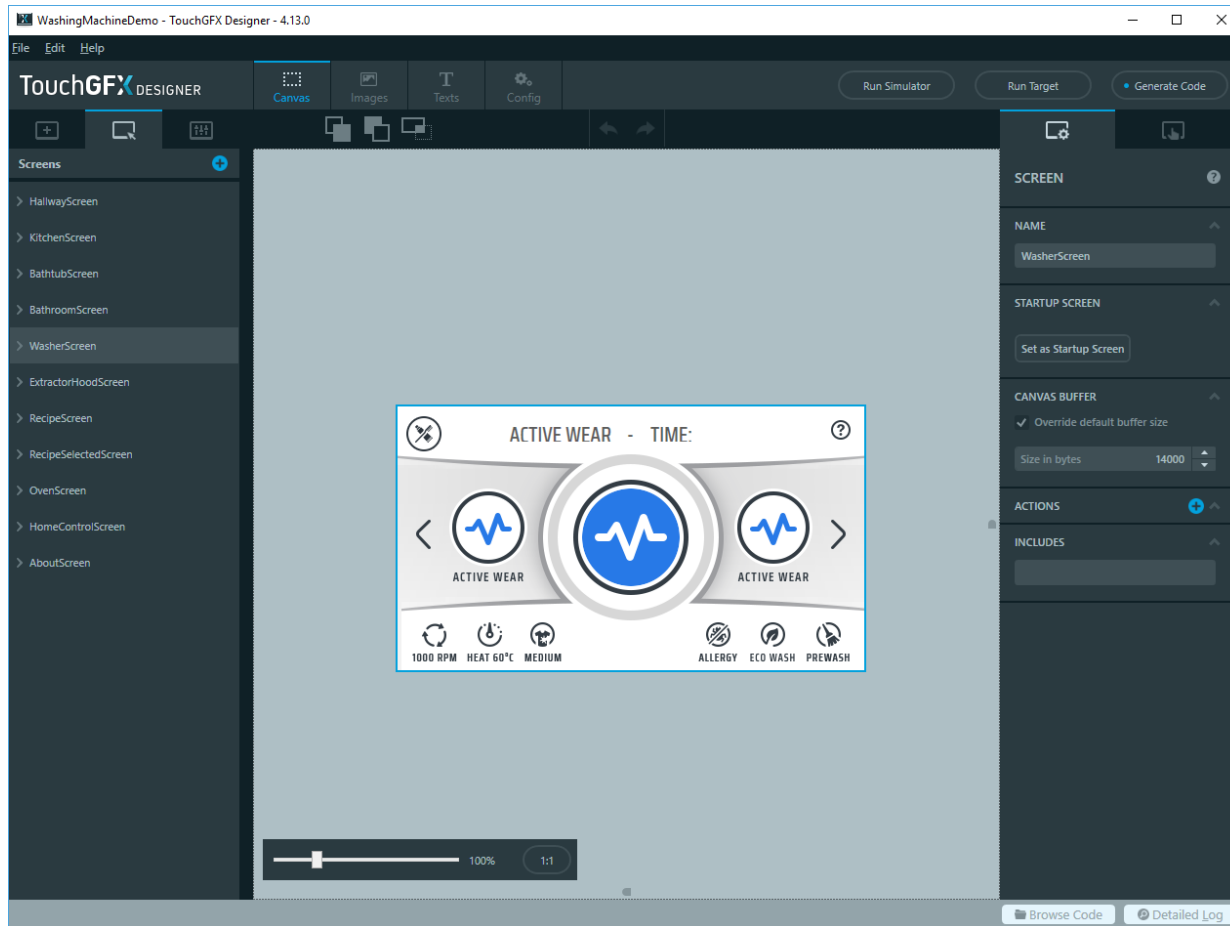


State-of-the-art graphic software and tools



TouchGFX PC tool

Graphic application development and simulation



TouchGFX Designer

From Idea to Prototype

A simple drag n' drop approach combined with ready-to-use high quality sample graphics enable you to create stunning prototypes in minutes with no need for advanced design and programming skills or TouchGFX knowledge.

From Prototype to Product

TouchGFX Designer will support you throughout your entire UI project by simplifying the process of creating the visual design and layout of your screens and custom controls. Your TouchGFX application code is automatically updated with the changes done in the Designer.

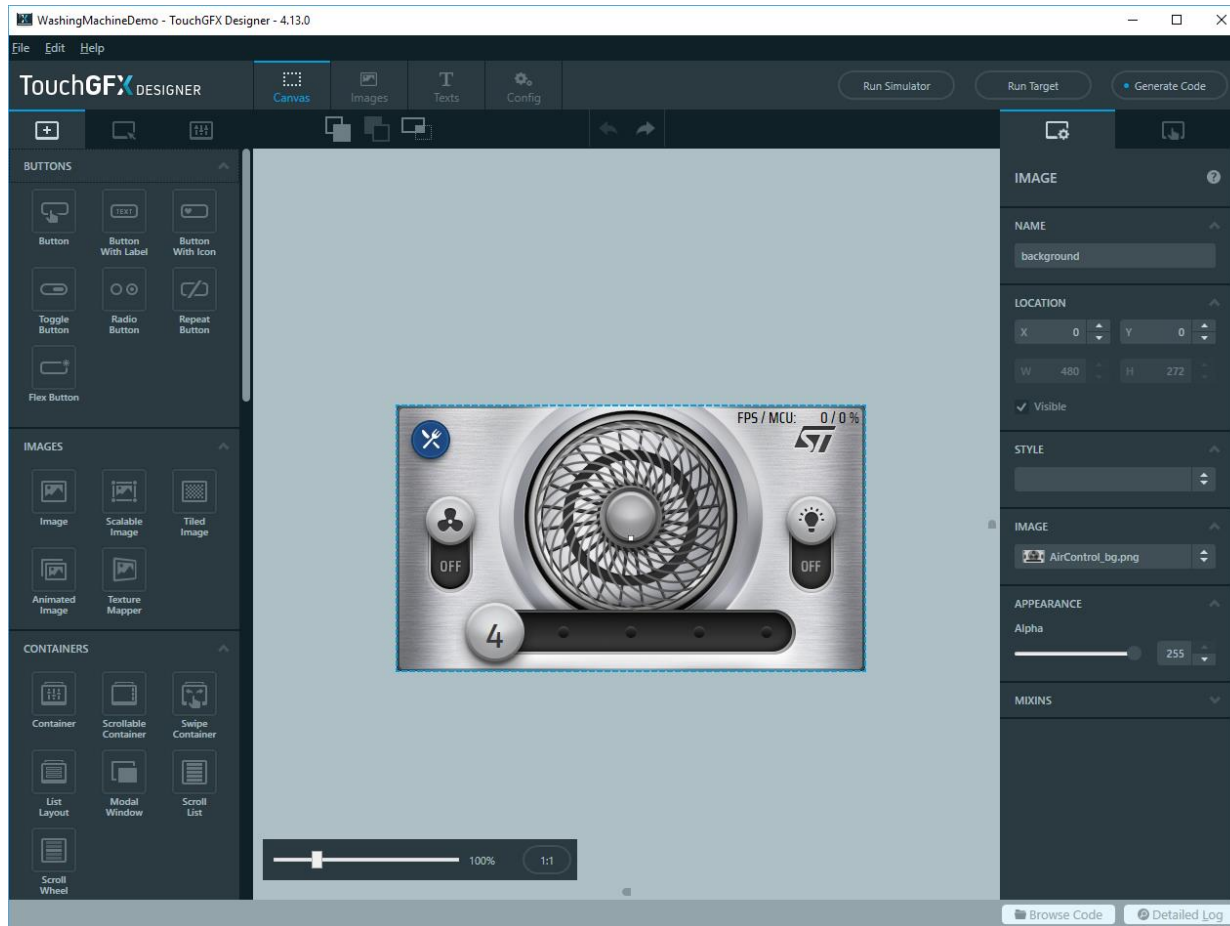


State-of-the-art graphic software and tools



TouchGFX PC tool

Graphic application development and simulation



TouchGFX Designer

- **Structure** : Easy creation of multiple screen contents and associated transitions.
- **Widgets** : Wide selection of customizable widgets like Swipe container, Scrollable list etc.
- **Interactions** : Dynamic interactions for the creation of user-friendly applications.
- **Custom Container** : Create custom reusable controls for your application.
- **Text Handling** Fonts and typographies specified and managed in on single place.
And multiple alphabets and scripts, such as Korean, Latin, Cyrillic, Arabic, Chinese ...

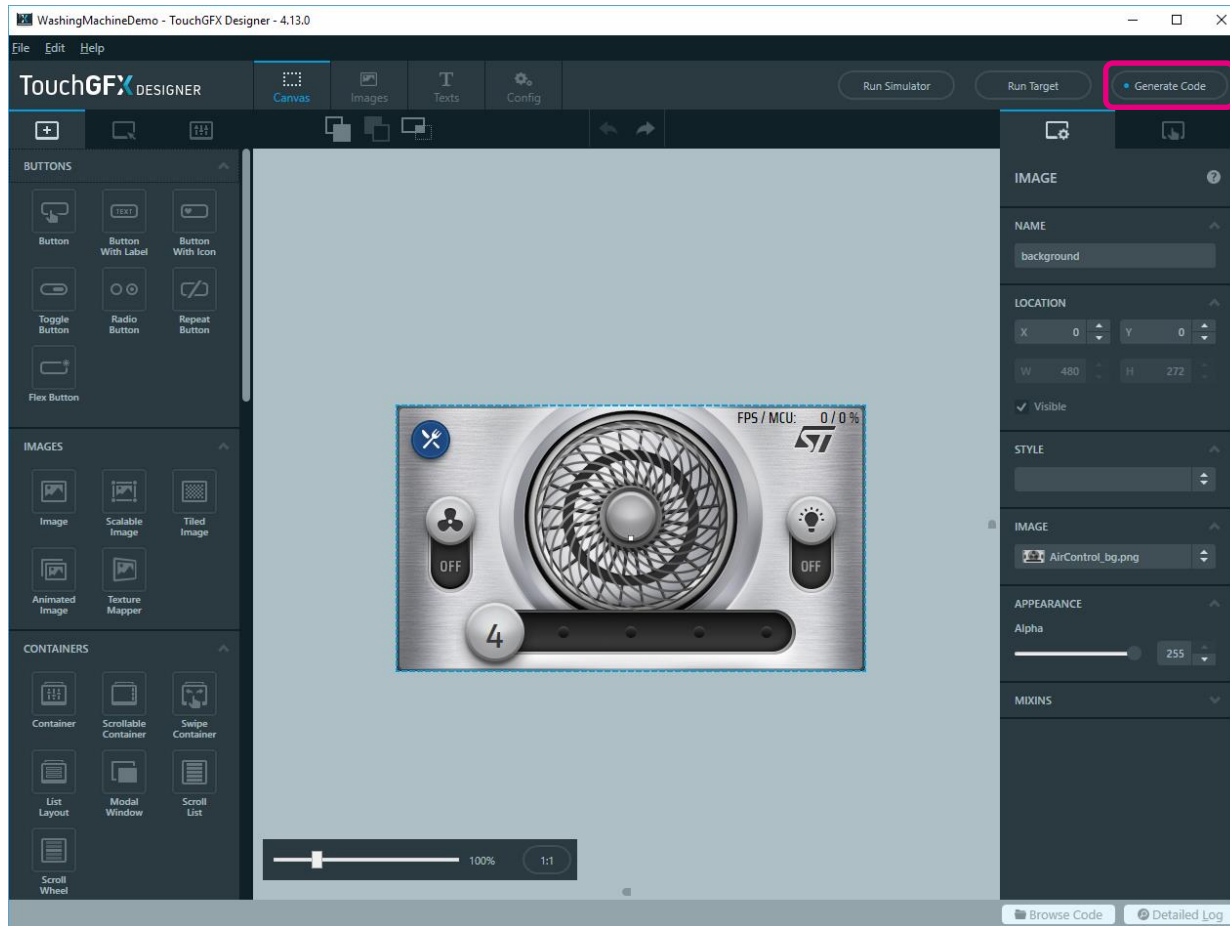


State-of-the-art graphic software and tools



TouchGFX PC tool

Graphic application development and simulation



TouchGFX Designer

• Code Generation :

- Generate and maintains performant C++ code.
- Tool-generated code entirely separated from user code
- All types of code extensions possible for unique animations as well as system interconnections
- Support of several integrated development environment such as STM32CubeIDE, IAR, KEIL and GCC-based IDEs.

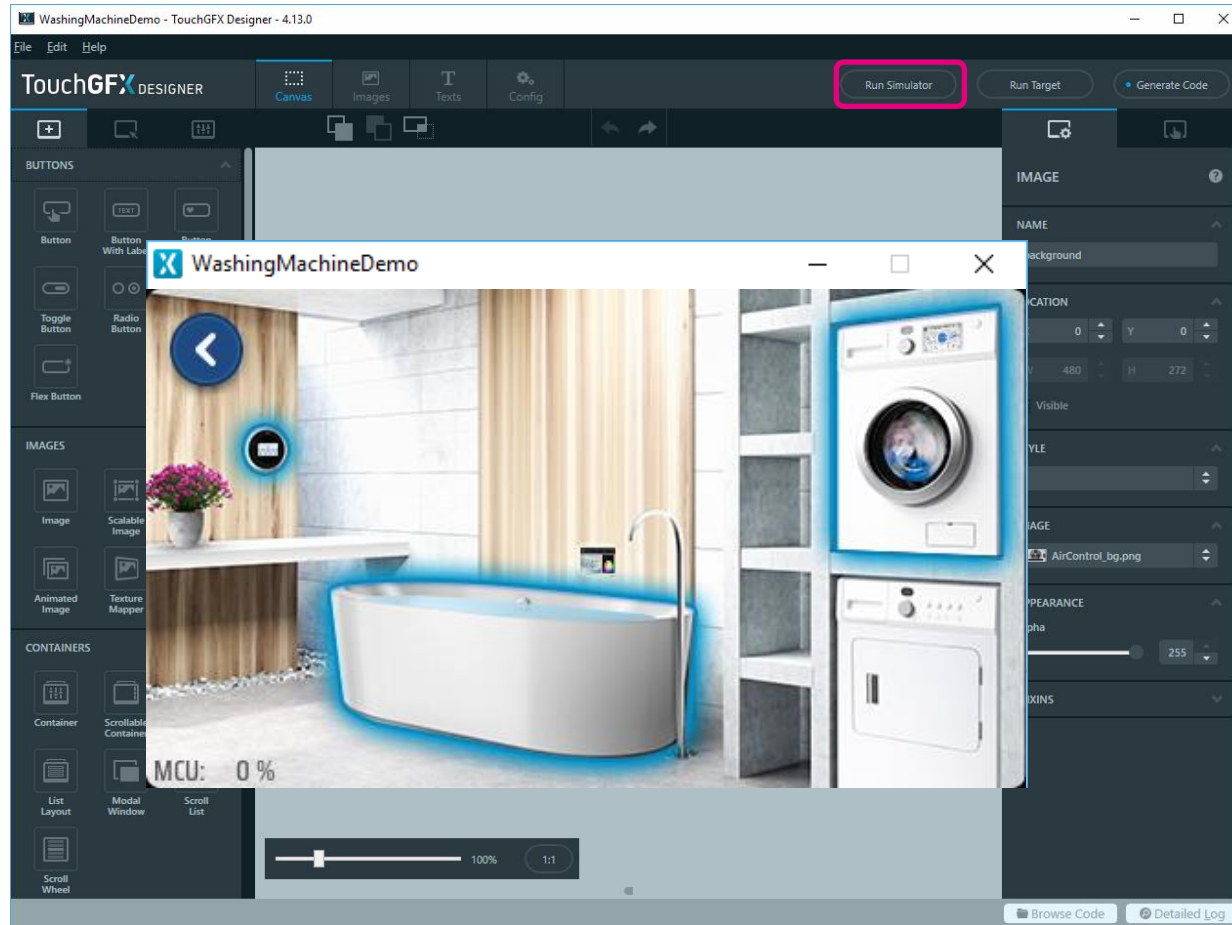


State-of-the-art graphic software and tools



TouchGFX PC tool

Graphic application development and simulation



TouchGFX Designer

PC Simulator for easy UI development

TouchGFX Designer show your UI project through PC simulator.

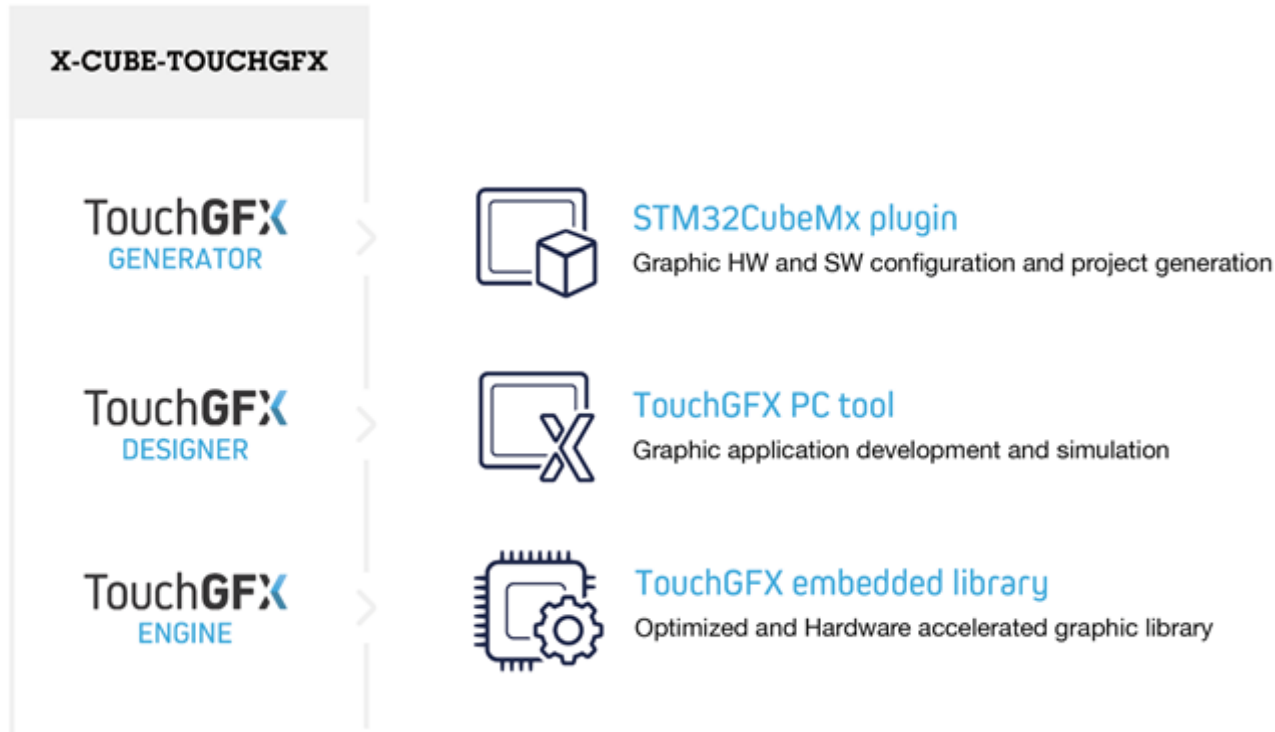
It will provide efficiency and give useful in order to make UI application you desired.



TouchGFX embedded library

Optimized and Hardware accelerated graphic library

State-of-the-art graphic software and tools



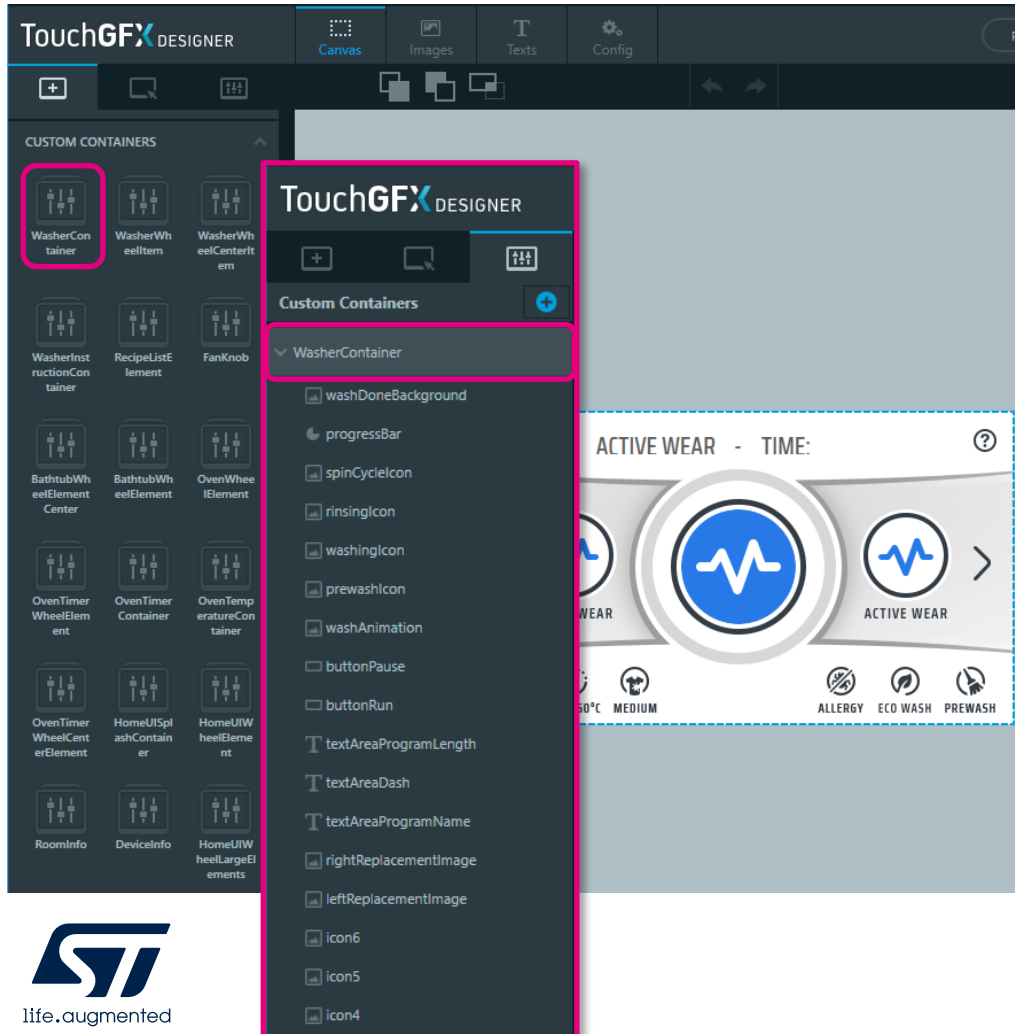


State-of-the-art graphic software and tools



TouchGFX embedded library

Optimized and Hardware accelerated graphic library



• Custom Containers

• Complexed object

Contains other existing widget and combines the visual appearance and behaviors of these widgets.

• High drawing performance

It will utilize the underlying drawing mechanisms of TouchGFX and will determine which parts of a container and the children needs to be redrawn automatically.



State-of-the-art graphic software and tools



TouchGFX embedded library

Optimized and Hardware accelerated graphic library

- **Caching Bitmaps**

- The dedicated RAM buffer where bitmaps can be stored (or cached) by the application.
- Use the RAM cache as pixel source when drawing the bitmap.
- Anticipate to increase the performance of drawing UI.
 - Generally, reading data from RAM is often faster than reading from flash (e.g. when using the Texturemapper because it use non-linear memory access)
- Useful when need to store your bitmaps on slow external storage like an USB-disk or SD card.

- **Bitmap Cache Configuration**

BoardConfiguration.cpp (extract) – Pass the start address and size of the buffer

```
/* Place cache start address in SDRAM at address 0xC0008000; */  
uint16_t cacheStartAddr = (uint16_t)0xC0008000;  
uint32_t cacheSize = 0x300000; //3 MB, as example  
HAL& hal = touchgfx_generic_init<STM32F4HAL>(dma, display, tc, DISPLAY_WIDTH, DISPLAY_HEIGHT, cacheStartAddr, cacheSize);
```

BoardConfiguration.cpp (extract) – Declare an array and just pass the address and size of the array

```
/* Define an array for the bimap cache */  
uint16_t cache[128*1024]; // 128 KB cache  
HAL& hal = touchgfx_generic_init<STM32F4HAL>(dma, display, tc, DISPLAY_WIDTH, DISPLAY_HEIGHT, &cache, sizeof(cache));
```





State-of-the-art graphic software and tools



TouchGFX embedded library

Optimized and Hardware accelerated graphic library

- **Partial FrameBuffer**

- 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
- Total RAM needs for UI starting from 12KBytes (Framebuffer + TouchGFX framework)
- Useable on any STM32 including Cortex-M0+ cores
- Display interface : DSI, SPI, Parallel/8080/FMC
- Limitation :
 - Partial display update limits UI performance like Texturemapper.
 - Require Display holding display controller and GRAM





State-of-the-art graphic software and tools

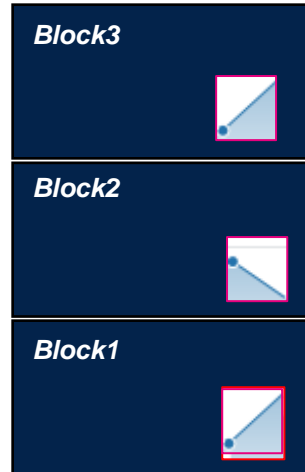
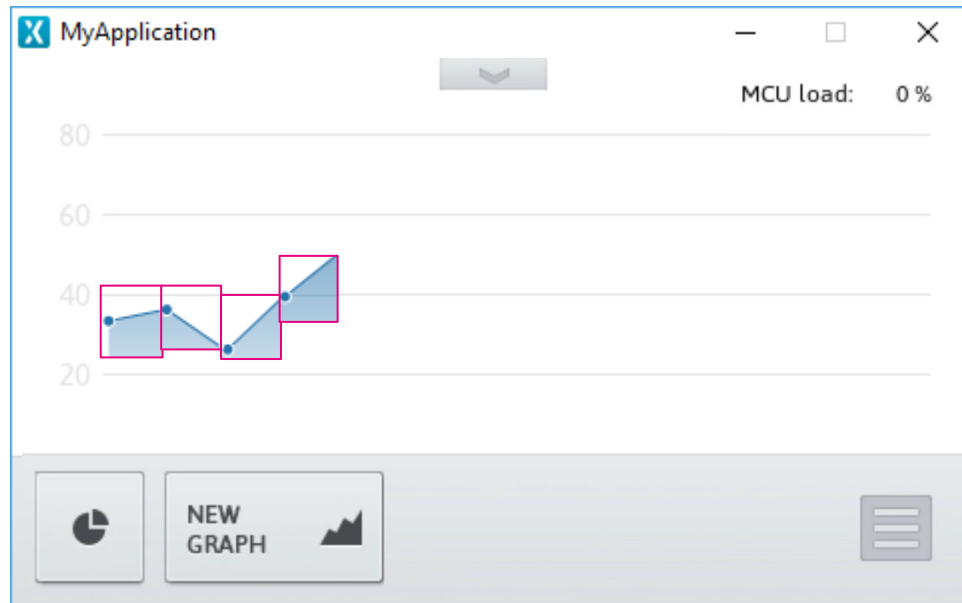


TouchGFX embedded library

Optimized and Hardware accelerated graphic library

- **Partial FrameBuffer**

Framebuffer in internal RAM
E.g. 20KB in total



TouchGFX renders the parts of the 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.





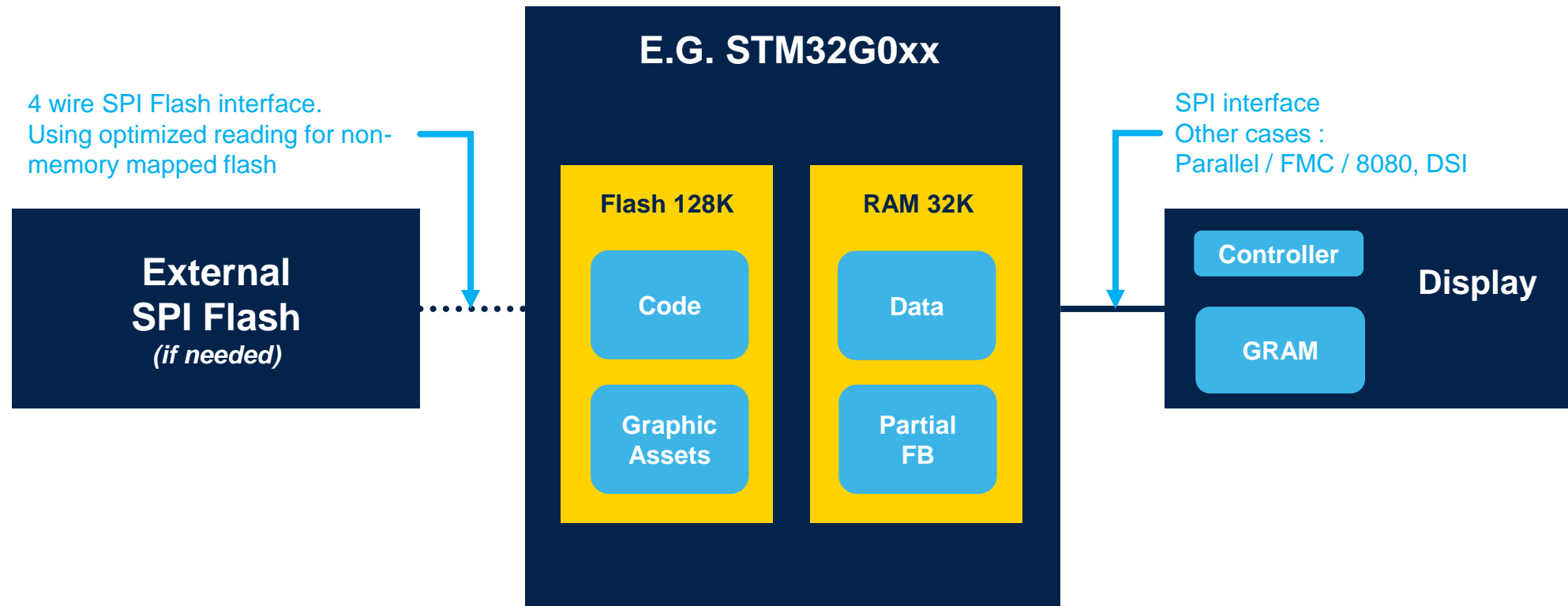
State-of-the-art graphic software and tools



TouchGFX embedded library

Optimized and Hardware accelerated graphic library

- UI memory setup on STM32G0
Using Partial Framebuffer and SPI flash





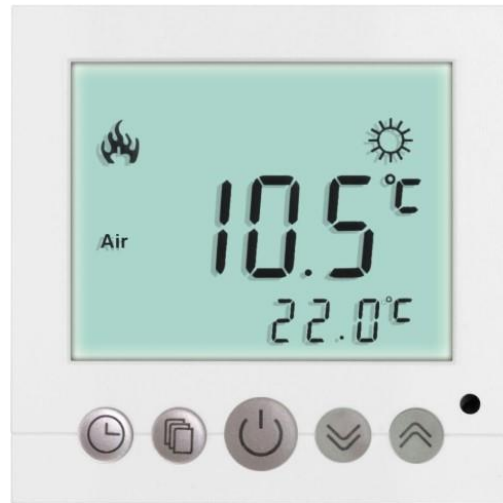
State-of-the-art graphic software and tools



TouchGFX embedded library

Optimized and Hardware accelerated graphic library

- VISION for STM32 Entry-Level Solution



Classic Low-Cost
Segment display

- Replace traditional segment display with modern graphical display solutions
- Smartphone-inspired User Experience
- Low Power / Long Battery Life
- BOM cost remains at the same level

TouchGFX



Modern Low-Cost
Graphical color display

Reference designs and worldwide support



Reference designs



32F429IDISCO
STM32F429
320x240
QVGA LCD
64 Mbits SDRAM



32F469IDISCO
STM32F469
800x480
WVGA LCD
128 Mbits SDRAM
128 Mbit QSPI Flash
Arduino Uno



32F769IDISCO
STM32F769
800x480
WVGA LCD
128 Mbits SDRAM
512 Mbit QSPI Flash
Arduino Uno



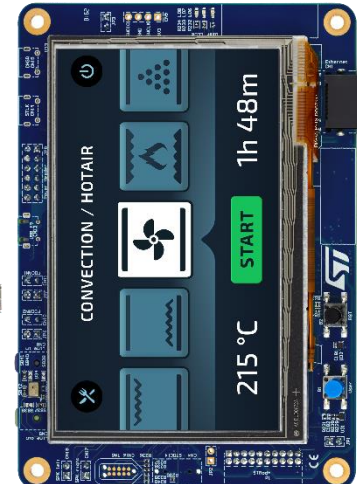
32L4R9IDISCOVERY
STM32L4R9
390x390
AMOLED
16 Mbits PSRAM
512 Mbit OctoSPI Flash
Arduino Uno



32F750GDISCO
STM32F750
480x272
WQVGA LCD
64 Mbits SDRAM
128 Mbit QSPI Flash
Arduino Uno



STM32H747IDISCOVERY
STM32H747XIH6U
800*480
WVGA LCD
256 Mbit SDRAM
2*512 Mbit QSPI NOR Flash
Arduino Uno



STM32H750BDISCOVERY
STM32H750
480*272
WQVGA LCD
128 Mbit SDRAM
2*512 Mbit QSPI NOR Flash
4-Gbyte on-board eMMC
Arduino Uno



STM32H7B3I-DK
STM32H7B3
480*272
WQVGA LCD
128 Mbit SDRAM
512 Mbit OctoSPI Flash
Arduino Uno



Reference designs



STM32429I-EVAL

- STM32F429
- 480x272 WQVGA LCD
- 256 MB SDRAM
- 128 MB NOR Flash



STM32439I-EVAL

- STM32F439
- 640x480 VGA LCD
- 256 MB SDRAM
- 128 MB NOR Flash



STM32469I-EVAL

- STM32F469
- 800x480 WVGA LCD
- 256 MB SDRAM
- 128 MB NOR Flash
- 512 MB QSPI Flash



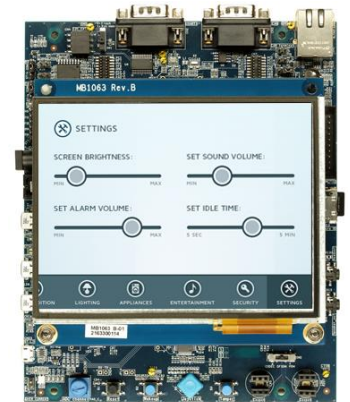
STM32756G-EVAL

- STM32F756
- 640x480 VGA LCD
- 256 MB SDRAM
- 128 MB NOR Flash
- 512 MB QSPI Flash



STM32769G-EVAL

- STM32F769
- 800x480 WVGA LCD
- 256 MB SDRAM
- 128 MB NOR Flash
- 512 MB QSPI Flash

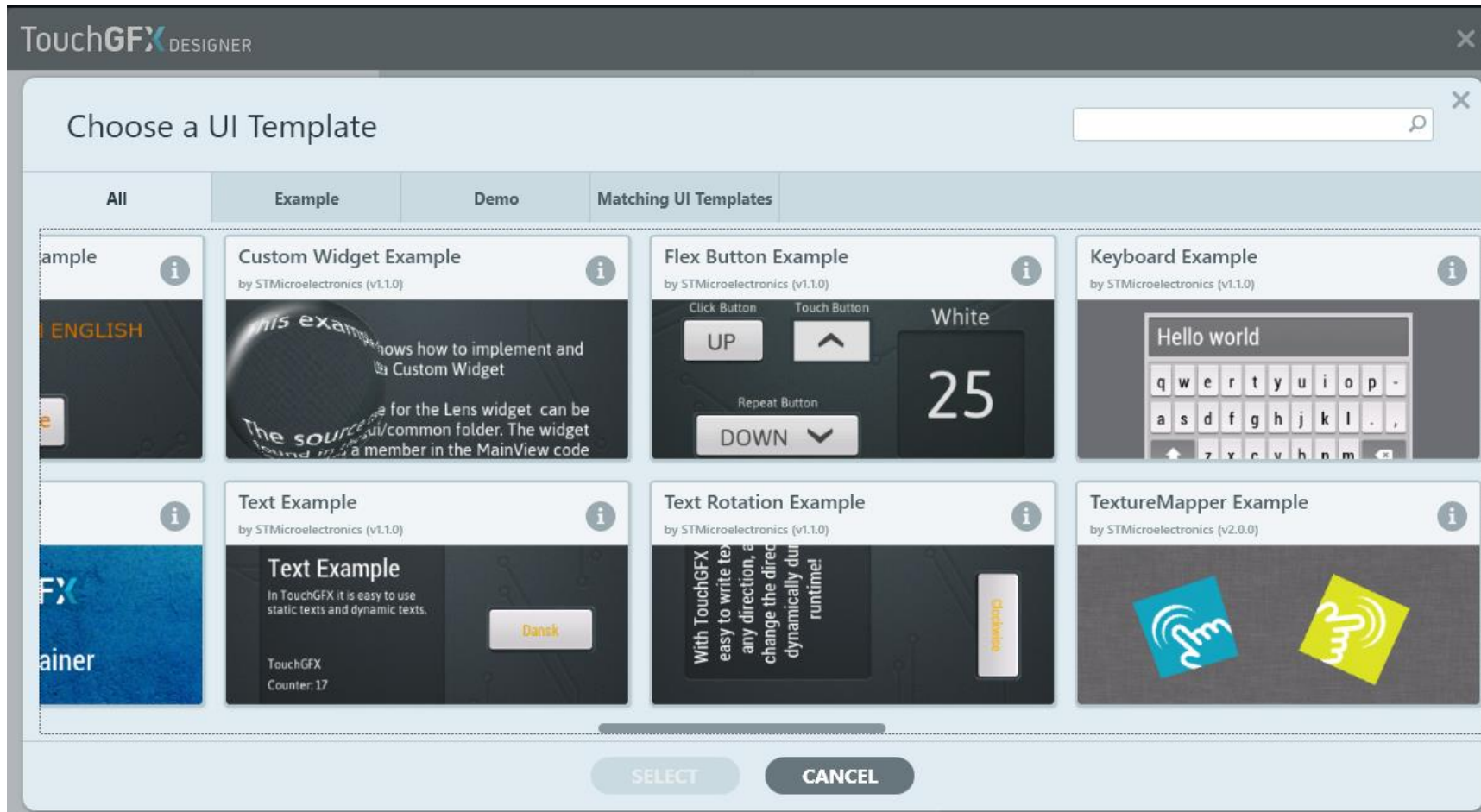


STM32743I-EVAL

- STM32H743
- 800x480 WVGA LCD
- 256 MB SDRAM
- 128 MB NOR Flash
- 512 MB QSPI Flash



Reference designs



TouchGFX Designer – Examples & Demo



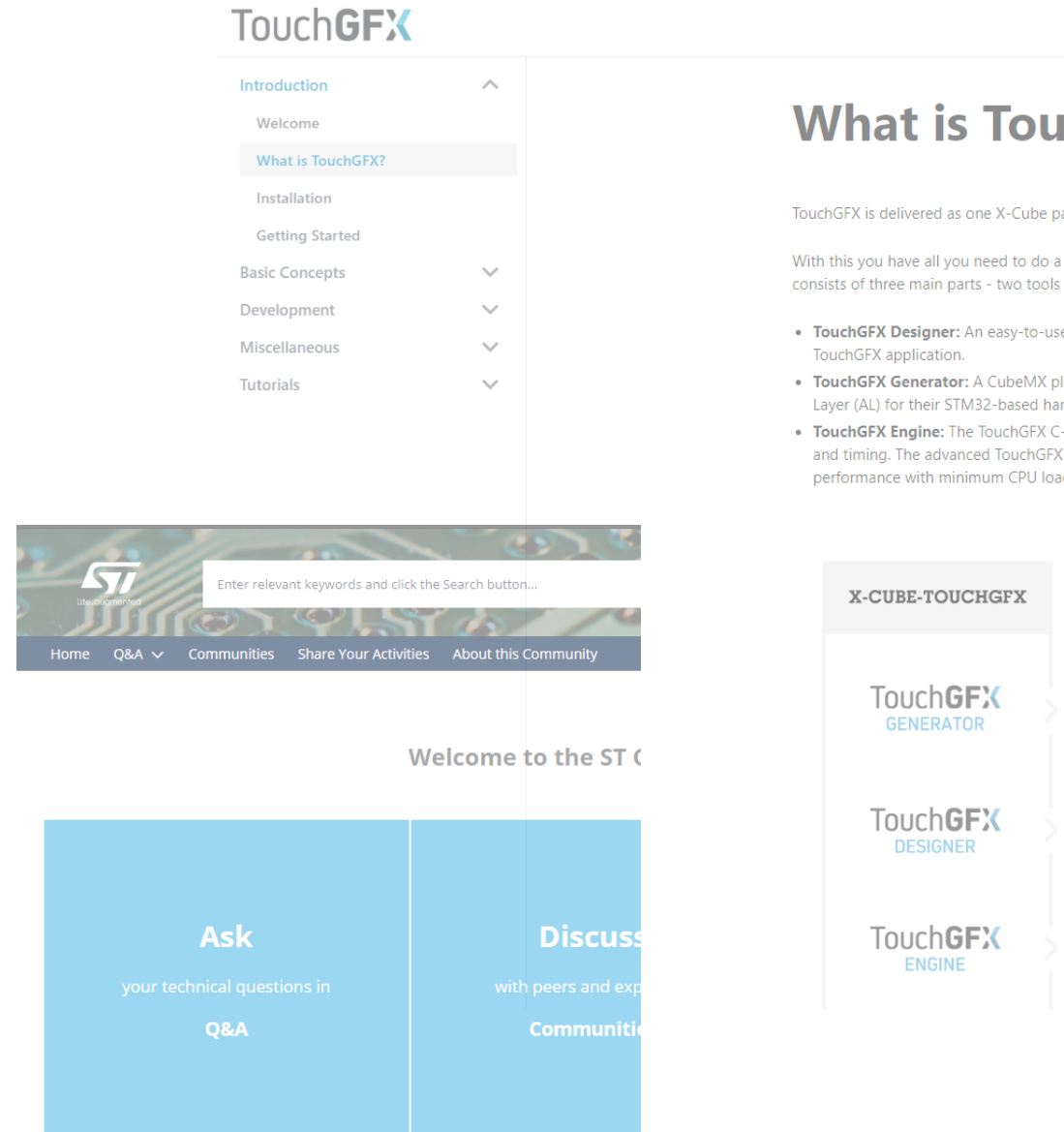
Worldwide support

- **Documentation**

- TouchGFX Documentation
<https://support.touchgfx.com>

- **Q&A**

- ST Community
<https://community.st.com>



The screenshot displays two web pages. The top page is the TouchGFX website, featuring a navigation menu with items like 'Introduction', 'Welcome', 'What is TouchGFX?', 'Installation', 'Getting Started', 'Basic Concepts', 'Development', 'Miscellaneous', and 'Tutorials'. The main content area is titled 'What is Tou' and includes a list of features: 'TouchGFX Designer', 'TouchGFX Generator', and 'TouchGFX Engine'. The bottom page is the ST Community Q&A page, showing a search bar and navigation links for 'Home', 'Q&A', 'Communities', 'Share Your Activities', and 'About this Community'. The main content area is titled 'Welcome to the ST C' and features two large blue buttons: 'Ask your technical questions in Q&A' and 'Discuss with peers and experts in Communities'.

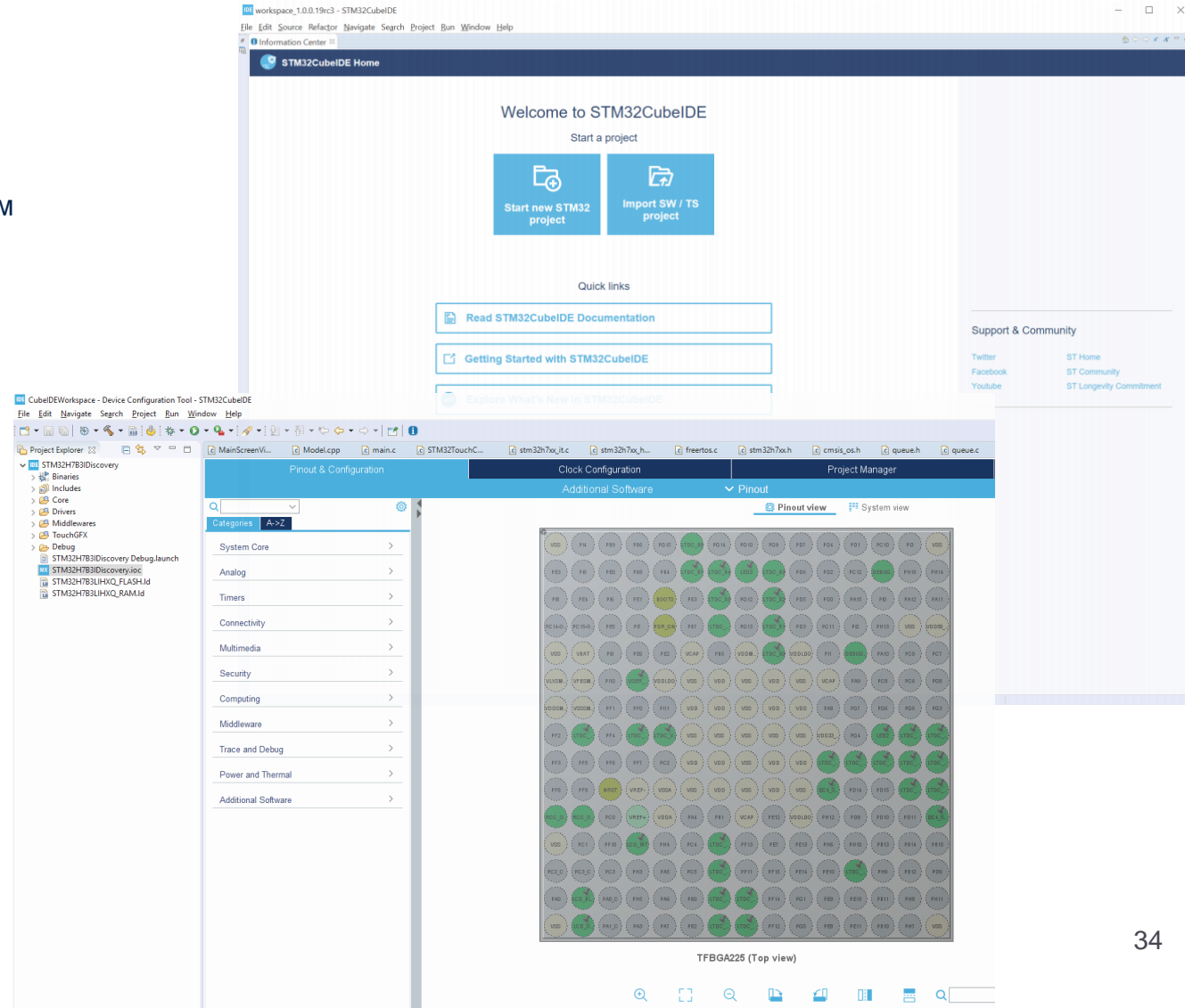
How to design TouchGFX with STM32CubeIDE



S/W tools

- **STM32CubeIDE**

- STM32 dedicated IDE integrated with STM32CubeMX
- Based on ECLIPSE™/CDT, with support of ECLIPSE™ add-ons, GNU C/C++ for Arm® toolchain and GDB debugger
- Multi-OS support
Windows®, Linux®, and MacOS®, 64-bit only
- Download
www.st.com/stm32cubeide





• TouchGFXDesigner

- Drag-and-Drop-based graphic-building PC tool
- GUI development
- WYSIWYG simulator
- Generate and maintain performant C++ code
- Include various examples and demos
- Support of several IDEs (IAR, KEIL, GCC-based)
- Download

<https://www.st.com/en/development-tools/touchgfxdesigner.html>





• STM32H7B3I-DK

- STM32H7B3LI ARM® Cortex-M7
- 2 MB Flash memory / 1.4MB RAM
- 4.3" (480 x 272 pixel) TFT color LCD including a capacitive touch panel with RGB interface
- Wi-Fi® module compliant with 802.11 b/g/n
- USB OTG HS
- Audio codec
- 512-Mbit Octo-SPI NOR Flash memory
- 128-Mbit SDRAM
- More information ...

https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-discovery-kits/stm32h7b3i-dk.html



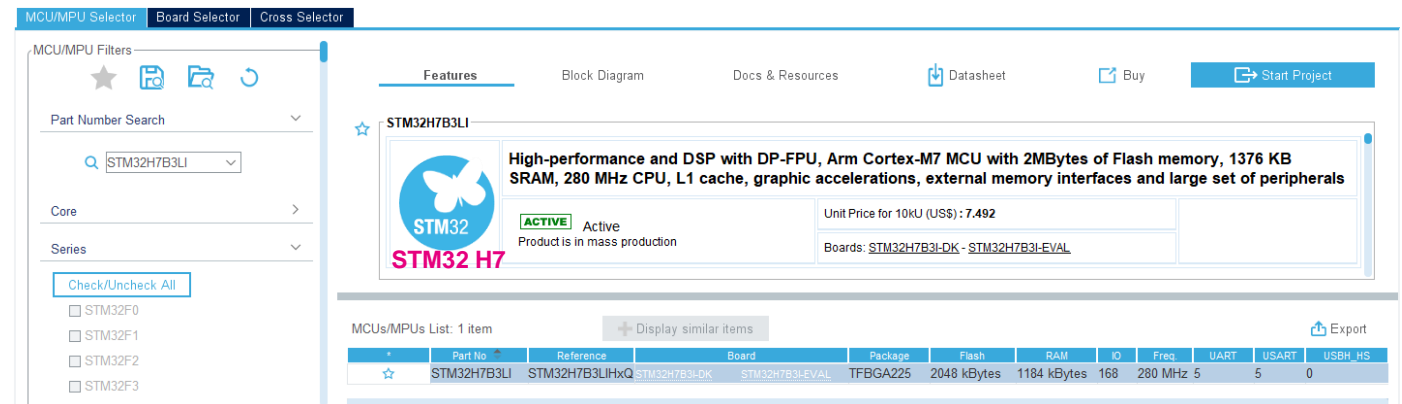
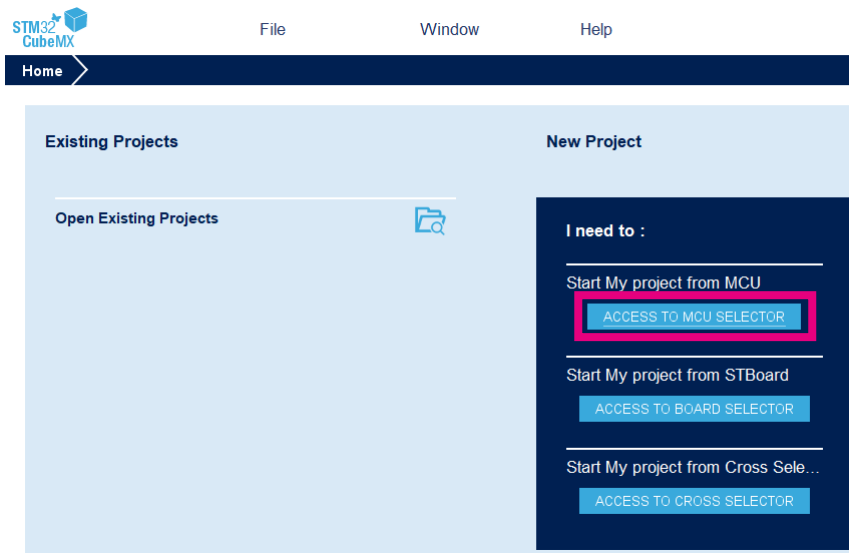
STM32 H7





Start STM32CubeMX

- At the beginning, it is needed to create new project based on MCU via STM32CubeMX





STM32CubeMX configuration

- IP's parameters based on your target should be properly configured.
 - Note, It is mandatory to enable **CRC** for X-CUBE-TOUCHGFX

The screenshot displays the STM32CubeMX configuration interface for a TFPGA225 target. The main window is titled "Pinout & Configuration" and shows the "LTDC Mode and Configuration" settings. The "Mode" section is set to "RGB888 (24 bits)". The "Configuration" section includes tabs for "User Constants", "NVIC Settings", "GPIO Settings", "Parameter Settings", and "Layer Settings". The "Layer Settings" tab is active, showing the following parameters:

- Number of Layers: 1 layer
- Windows Position:
 - Layer 0 - Window Horizontal Start: 0
 - Layer 0 - Window Horizontal Stop: 480
 - Layer 0 - Window Vertical Start: 0
 - Layer 0 - Window Vertical Stop: 272
- Pixel Parameters:
 - Layer 0 - Pixel Format: RGB888
- Blending:
 - Layer 0 - Alpha constant for blending: 255
 - Layer 0 - Default Alpha value: 0
 - Layer 0 - Blending Factor1: Alpha constant
 - Layer 0 - Blending Factor2: Alpha constant
- Frame Buffer:
 - Layer 0 - Color Frame Buffer Start Address: 0
 - Layer 0 - Color Frame Buffer Line Length (...): 480
 - Layer 0 - Color Frame Buffer Number of Li...: 272
- Background Color:
 - Layer 0 - Blue: 255
 - Layer 0 - Green: 0
 - Layer 0 - Red: 0

The right side of the interface shows a "Pinout view" of the TFPGA225 (Top view) board. The pinout diagram displays various pins, including VSS, VBAT, VCAP, VDD, VDD0, VDD1, VDD2, VDD3, VDD4, VDD5, VDD6, VDD7, VDD8, VDD9, VDD10, VDD11, VDD12, VDD13, VDD14, VDD15, VDD16, VDD17, VDD18, VDD19, VDD20, VDD21, VDD22, VDD23, VDD24, VDD25, VDD26, VDD27, VDD28, VDD29, VDD30, VDD31, VDD32, VDD33, VDD34, VDD35, VDD36, VDD37, VDD38, VDD39, VDD40, VDD41, VDD42, VDD43, VDD44, VDD45, VDD46, VDD47, VDD48, VDD49, VDD50, VDD51, VDD52, VDD53, VDD54, VDD55, VDD56, VDD57, VDD58, VDD59, VDD60, VDD61, VDD62, VDD63, VDD64, VDD65, VDD66, VDD67, VDD68, VDD69, VDD70, VDD71, VDD72, VDD73, VDD74, VDD75, VDD76, VDD77, VDD78, VDD79, VDD80, VDD81, VDD82, VDD83, VDD84, VDD85, VDD86, VDD87, VDD88, VDD89, VDD90, VDD91, VDD92, VDD93, VDD94, VDD95, VDD96, VDD97, VDD98, VDD99, VDD100, VDD101, VDD102, VDD103, VDD104, VDD105, VDD106, VDD107, VDD108, VDD109, VDD110, VDD111, VDD112, VDD113, VDD114, VDD115, VDD116, VDD117, VDD118, VDD119, VDD120, VDD121, VDD122, VDD123, VDD124, VDD125, VDD126, VDD127, VDD128, VDD129, VDD130, VDD131, VDD132, VDD133, VDD134, VDD135, VDD136, VDD137, VDD138, VDD139, VDD140, VDD141, VDD142, VDD143, VDD144, VDD145, VDD146, VDD147, VDD148, VDD149, VDD150, VDD151, VDD152, VDD153, VDD154, VDD155, VDD156, VDD157, VDD158, VDD159, VDD160, VDD161, VDD162, VDD163, VDD164, VDD165, VDD166, VDD167, VDD168, VDD169, VDD170, VDD171, VDD172, VDD173, VDD174, VDD175, VDD176, VDD177, VDD178, VDD179, VDD180, VDD181, VDD182, VDD183, VDD184, VDD185, VDD186, VDD187, VDD188, VDD189, VDD190, VDD191, VDD192, VDD193, VDD194, VDD195, VDD196, VDD197, VDD198, VDD199, VDD200, VDD201, VDD202, VDD203, VDD204, VDD205, VDD206, VDD207, VDD208, VDD209, VDD210, VDD211, VDD212, VDD213, VDD214, VDD215, VDD216, VDD217, VDD218, VDD219, VDD220, VDD221, VDD222, VDD223, VDD224, VDD225, VDD226, VDD227, VDD228, VDD229, VDD230, VDD231, VDD232, VDD233, VDD234, VDD235, VDD236, VDD237, VDD238, VDD239, VDD240, VDD241, VDD242, VDD243, VDD244, VDD245, VDD246, VDD247, VDD248, VDD249, VDD250, VDD251, VDD252, VDD253, VDD254, VDD255, VDD256, VDD257, VDD258, VDD259, VDD260, VDD261, VDD262, VDD263, VDD264, VDD265, VDD266, VDD267, VDD268, VDD269, VDD270, VDD271, VDD272, VDD273, VDD274, VDD275, VDD276, VDD277, VDD278, VDD279, VDD280, VDD281, VDD282, VDD283, VDD284, VDD285, VDD286, VDD287, VDD288, VDD289, VDD290, VDD291, VDD292, VDD293, VDD294, VDD295, VDD296, VDD297, VDD298, VDD299, VDD300, VDD301, VDD302, VDD303, VDD304, VDD305, VDD306, VDD307, VDD308, VDD309, VDD310, VDD311, VDD312, VDD313, VDD314, VDD315, VDD316, VDD317, VDD318, VDD319, VDD320, VDD321, VDD322, VDD323, VDD324, VDD325, VDD326, VDD327, VDD328, VDD329, VDD330, VDD331, VDD332, VDD333, VDD334, VDD335, VDD336, VDD337, VDD338, VDD339, VDD340, VDD341, VDD342, VDD343, VDD344, VDD345, VDD346, VDD347, VDD348, VDD349, VDD350, VDD351, VDD352, VDD353, VDD354, VDD355, VDD356, VDD357, VDD358, VDD359, VDD360, VDD361, VDD362, VDD363, VDD364, VDD365, VDD366, VDD367, VDD368, VDD369, VDD370, VDD371, VDD372, VDD373, VDD374, VDD375, VDD376, VDD377, VDD378, VDD379, VDD380, VDD381, VDD382, VDD383, VDD384, VDD385, VDD386, VDD387, VDD388, VDD389, VDD390, VDD391, VDD392, VDD393, VDD394, VDD395, VDD396, VDD397, VDD398, VDD399, VDD400, VDD401, VDD402, VDD403, VDD404, VDD405, VDD406, VDD407, VDD408, VDD409, VDD410, VDD411, VDD412, VDD413, VDD414, VDD415, VDD416, VDD417, VDD418, VDD419, VDD420, VDD421, VDD422, VDD423, VDD424, VDD425, VDD426, VDD427, VDD428, VDD429, VDD430, VDD431, VDD432, VDD433, VDD434, VDD435, VDD436, VDD437, VDD438, VDD439, VDD440, VDD441, VDD442, VDD443, VDD444, VDD445, VDD446, VDD447, VDD448, VDD449, VDD450, VDD451, VDD452, VDD453, VDD454, VDD455, VDD456, VDD457, VDD458, VDD459, VDD460, VDD461, VDD462, VDD463, VDD464, VDD465, VDD466, VDD467, VDD468, VDD469, VDD470, VDD471, VDD472, VDD473, VDD474, VDD475, VDD476, VDD477, VDD478, VDD479, VDD480, VDD481, VDD482, VDD483, VDD484, VDD485, VDD486, VDD487, VDD488, VDD489, VDD490, VDD491, VDD492, VDD493, VDD494, VDD495, VDD496, VDD497, VDD498, VDD499, VDD500, VDD501, VDD502, VDD503, VDD504, VDD505, VDD506, VDD507, VDD508, VDD509, VDD510, VDD511, VDD512, VDD513, VDD514, VDD515, VDD516, VDD517, VDD518, VDD519, VDD520, VDD521, VDD522, VDD523, VDD524, VDD525, VDD526, VDD527, VDD528, VDD529, VDD530, VDD531, VDD532, VDD533, VDD534, VDD535, VDD536, VDD537, VDD538, VDD539, VDD540, VDD541, VDD542, VDD543, VDD544, VDD545, VDD546, VDD547, VDD548, VDD549, VDD550, VDD551, VDD552, VDD553, VDD554, VDD555, VDD556, VDD557, VDD558, VDD559, VDD560, VDD561, VDD562, VDD563, VDD564, VDD565, VDD566, VDD567, VDD568, VDD569, VDD570, VDD571, VDD572, VDD573, VDD574, VDD575, VDD576, VDD577, VDD578, VDD579, VDD580, VDD581, VDD582, VDD583, VDD584, VDD585, VDD586, VDD587, VDD588, VDD589, VDD590, VDD591, VDD592, VDD593, VDD594, VDD595, VDD596, VDD597, VDD598, VDD599, VDD600, VDD601, VDD602, VDD603, VDD604, VDD605, VDD606, VDD607, VDD608, VDD609, VDD610, VDD611, VDD612, VDD613, VDD614, VDD615, VDD616, VDD617, VDD618, VDD619, VDD620, VDD621, VDD622, VDD623, VDD624, VDD625, VDD626, VDD627, VDD628, VDD629, VDD630, VDD631, VDD632, VDD633, VDD634, VDD635, VDD636, VDD637, VDD638, VDD639, VDD640, VDD641, VDD642, VDD643, VDD644, VDD645, VDD646, VDD647, VDD648, VDD649, VDD650, VDD651, VDD652, VDD653, VDD654, VDD655, VDD656, VDD657, VDD658, VDD659, VDD660, VDD661, VDD662, VDD663, VDD664, VDD665, VDD666, VDD667, VDD668, VDD669, VDD670, VDD671, VDD672, VDD673, VDD674, VDD675, VDD676, VDD677, VDD678, VDD679, VDD680, VDD681, VDD682, VDD683, VDD684, VDD685, VDD686, VDD687, VDD688, VDD689, VDD690, VDD691, VDD692, VDD693, VDD694, VDD695, VDD696, VDD697, VDD698, VDD699, VDD700, VDD701, VDD702, VDD703, VDD704, VDD705, VDD706, VDD707, VDD708, VDD709, VDD710, VDD711, VDD712, VDD713, VDD714, VDD715, VDD716, VDD717, VDD718, VDD719, VDD720, VDD721, VDD722, VDD723, VDD724, VDD725, VDD726, VDD727, VDD728, VDD729, VDD730, VDD731, VDD732, VDD733, VDD734, VDD735, VDD736, VDD737, VDD738, VDD739, VDD740, VDD741, VDD742, VDD743, VDD744, VDD745, VDD746, VDD747, VDD748, VDD749, VDD750, VDD751, VDD752, VDD753, VDD754, VDD755, VDD756, VDD757, VDD758, VDD759, VDD760, VDD761, VDD762, VDD763, VDD764, VDD765, VDD766, VDD767, VDD768, VDD769, VDD770, VDD771, VDD772, VDD773, VDD774, VDD775, VDD776, VDD777, VDD778, VDD779, VDD780, VDD781, VDD782, VDD783, VDD784, VDD785, VDD786, VDD787, VDD788, VDD789, VDD790, VDD791, VDD792, VDD793, VDD794, VDD795, VDD796, VDD797, VDD798, VDD799, VDD800, VDD801, VDD802, VDD803, VDD804, VDD805, VDD806, VDD807, VDD808, VDD809, VDD810, VDD811, VDD812, VDD813, VDD814, VDD815, VDD816, VDD817, VDD818, VDD819, VDD820, VDD821, VDD822, VDD823, VDD824, VDD825, VDD826, VDD827, VDD828, VDD829, VDD830, VDD831, VDD832, VDD833, VDD834, VDD835, VDD836, VDD837, VDD838, VDD839, VDD840, VDD841, VDD842, VDD843, VDD844, VDD845, VDD846, VDD847, VDD848, VDD849, VDD850, VDD851, VDD852, VDD853, VDD854, VDD855, VDD856, VDD857, VDD858, VDD859, VDD860, VDD861, VDD862, VDD863, VDD864, VDD865, VDD866, VDD867, VDD868, VDD869, VDD870, VDD871, VDD872, VDD873, VDD874, VDD875, VDD876, VDD877, VDD878, VDD879, VDD880, VDD881, VDD882, VDD883, VDD884, VDD885, VDD886, VDD887, VDD888, VDD889, VDD890, VDD891, VDD892, VDD893, VDD894, VDD895, VDD896, VDD897, VDD898, VDD899, VDD900, VDD901, VDD902, VDD903, VDD904, VDD905, VDD906, VDD907, VDD908, VDD909, VDD910, VDD911, VDD912, VDD913, VDD914, VDD915, VDD916, VDD917, VDD918, VDD919, VDD920, VDD921, VDD922, VDD923, VDD924, VDD925, VDD926, VDD927, VDD928, VDD929, VDD930, VDD931, VDD932, VDD933, VDD934, VDD935, VDD936, VDD937, VDD938, VDD939, VDD940, VDD941, VDD942, VDD943, VDD944, VDD945, VDD946, VDD947, VDD948, VDD949, VDD950, VDD951, VDD952, VDD953, VDD954, VDD955, VDD956, VDD957, VDD958, VDD959, VDD960, VDD961, VDD962, VDD963, VDD964, VDD965, VDD966, VDD967, VDD968, VDD969, VDD970, VDD971, VDD972, VDD973, VDD974, VDD975, VDD976, VDD977, VDD978, VDD979, VDD980, VDD981, VDD982, VDD983, VDD984, VDD985, VDD986, VDD987, VDD988, VDD989, VDD990, VDD991, VDD992, VDD993, VDD994, VDD995, VDD996, VDD997, VDD998, VDD999, VDD1000.

TFPGA225 (Top view)



TouchGFX Generator

- Add TouchGFX Generator

Pinout & Configuration | Clock Configuration | Additional Software | Pinout

LTDC Mode and Configuration

Additional Software Components selection

Filters

Search

Pack Vendor

- ARM
- STMicroelectronics

Software Component Class

- Artificial Intelligence
- Board Component
- Board Extension
- Board Support
- CMSIS
- CMSIS Driver
- Data Exchange
- Device
- Graphics
- Motion Library
- RTOS
- Wireless

Packs

Pack / Bundle / Component	Version	Selection
> ARM.CMSIS	5.6.0	
> STMicroelectronics.X-CUBE-AI	5.1.2	
> STMicroelectronics.X-CUBE-BLE1	5.0.0	
> STMicroelectronics.X-CUBE-BLE2	1.0.0	
> STMicroelectronics.X-CUBE-GNSS1	4.1.0	
> STMicroelectronics.X-CUBE-MEMS1	7.2.0	
> STMicroelectronics.X-CUBE-NFC4	1.5.2	
> STMicroelectronics.X-CUBE-SUBG2	1.1.0	
✓ STMicroelectronics.X-CUBE-TOUCHGFX	4.13.0	
✓ Graphics Application	4.13.0	
Application		TouchGFX Generator

Component dependencies

Bundle Application (from pack STMicroelectronics.X-CUBE-TOUCHGFX.4.13.0)
Cannot show dependencies at bundle level

Details and warnings

Bundle details

Pack STMicroelectronics.X-CUBE-TOUCHGFX.4.13.0

Name Application

Version 4.13.0

Class Graphics

★ Add to favorites

Description

STM32 Graphics framework, TouchGFX Generator

Documents

License

Show/hide filters | Show/hide details | Show/hide dependencies

Ok Cancel



TouchGFX Generator configuration

- Configure TouchGFX Generator
 - Display
 - Interface
 - Parallel RGB (LTDC)
 - Custom (For DSI, FMC and SPI interface)
These interface drivers must be implemented by developers themselves.
 - Framebuffer Strategy
 - Single Buffer – Use only one application frame buffer.
 - Double Buffer – Use two frame buffers.
 - Partial Buffer – Use one or more user defined chunks of memory as the frame buffer. This is targeted at low cost solutions.
 - Buffer Location
 - By Allocation – Lets the linker place frame buffer memory according to linker script.
 - By Address – Allow the user to define one or two frame buffer addresses

Pinout & Configuration | Clock Configuration | Additional Software

STMicroelectronics.X-CUBE-TOUCHGFX.4.13.0 Mode and Configuration

Mode

- Graphics Application

Configuration

Reset Configuration

- TouchGFX Generator
- User Constants

Configure the below parameters :

Search (Ctrl+F)

Display

Interface	Parallel RGB (LTDC)
Framebuffer Pixel Format	RGB888
Width (LTDC)	480 pixels
Height (LTDC)	272 pixels
Framebuffer Strategy	Double Buffer
Buffer Location	By Allocation

Driver

Application Tick Source	LTDC
Graphics Accelerator	ChromART (DMA2D)
Real-Time Operating System	CMSIS_RTOS_V2



TouchGFX Generator configuration

- Configure TouchGFX Generator
 - Driver
 - Application Tick Source
 - LTDC – Drive the application forward by calling “sygnalVSync()” from LTDC interrupt handler.
 - Custom – Developer should implement a handler that drives the application forward by calling “sygnalVSync()” repeatedly.
 - Graphics Accelerator
 - None – The application uses only the CPU to draw frames.
 - Chrom-ART (DMA2D) – The application uses the Chrom-ART chip when possible to move and blend pixels, freeing up CPU cycles.
 - Buffer Location
 - No RTOS.
 - CMSIS V1 or V2
 - Custom

Pinout & Configuration | Clock Configuration | Additional Software

STMicroelectronics.X-CUBE-TOUCHGFX.4.13.0 Mode and Configuration

Mode

- Graphics Application

Configuration

Reset Configuration

- TouchGFX Generator
- User Constants

Configure the below parameters :

Search (Ctrl+F)

- Display
 - Interface: Parallel RGB (LTDC)
 - Framebuffer Pixel Format: RGB888
 - Width (LTDC): 480 pixels
 - Height (LTDC): 272 pixels
 - Framebuffer Strategy: Double Buffer
 - Buffer Location: By Allocation
- Driver
 - Application Tick Source: LTDC
 - Graphics Accelerator: ChromART (DMA2D)
 - Real-Time Operating System: CMSIS_RTOS_V2



Run TouchGFXDesigner

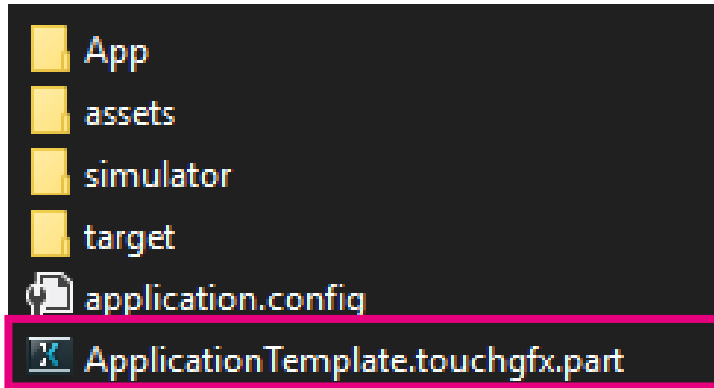
- Code Generation

The screenshot displays the TouchGFX Designer interface for the STM32H7B3I-DK. The top navigation bar includes 'Home', 'STM32H7B3LIHxQ', and 'STM32H7B3I-DK.ioc - Pinout & Configuration'. A prominent 'GENERATE CODE' button is highlighted with a red box. Below the navigation bar, the interface is divided into several sections:

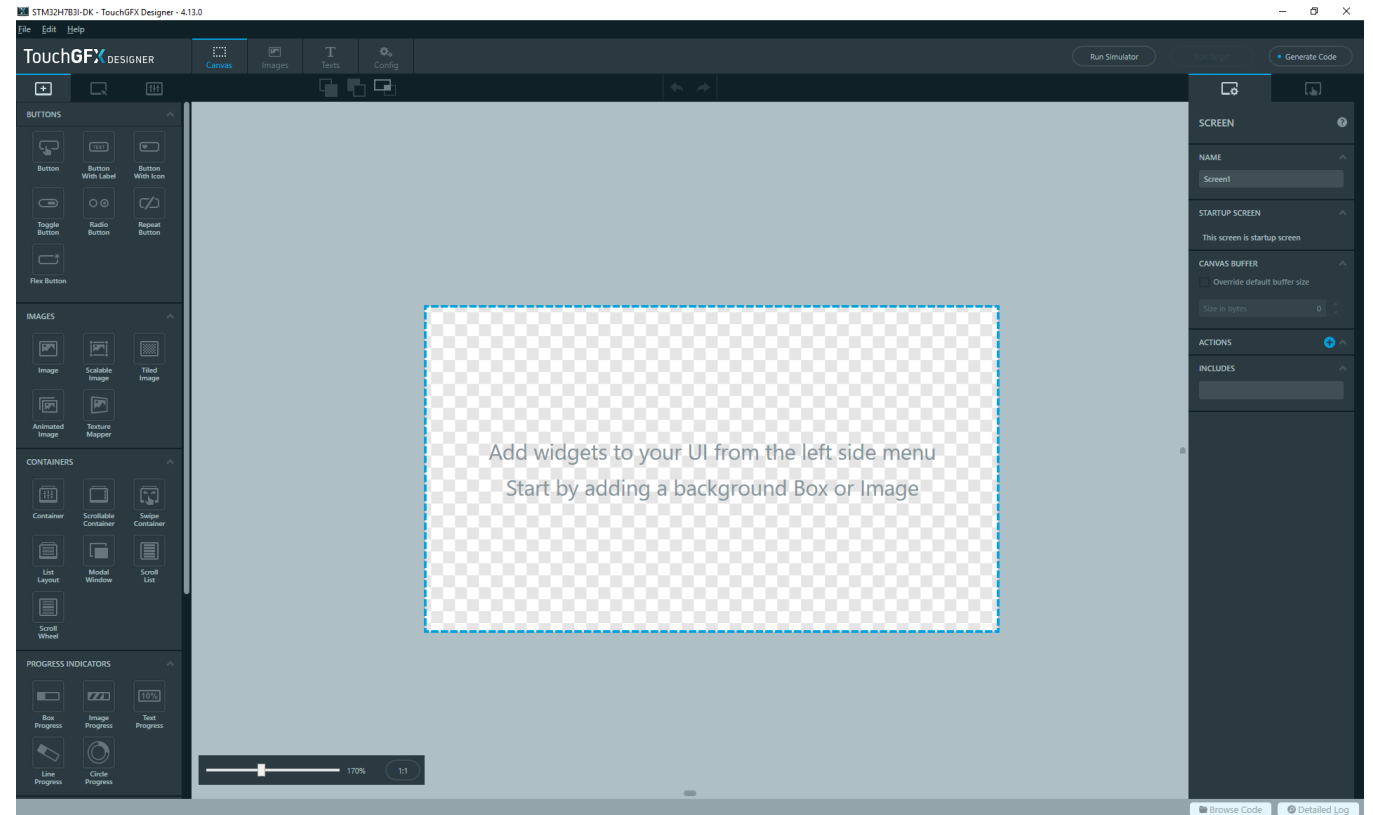
- Pinout & Configuration:** The active section, showing a search bar, a gear icon, and a category list on the left (System Core, Analog, Timers, Connectivity, Multimedia, Security, Computing, Middleware, Trace and Debug, Power and Thermal).
- Configuration:** The main area for setting parameters. It includes a 'Reset Configuration' button and two checked options: 'TouchGFX Generator' and 'User Constants'. Below this, it prompts to 'Configure the below parameters':
 - Display:**
 - Interface: Parallel RGB (LTDC)
 - Framebuffer Pixel Format: RGB888
 - Width (LTDC): 480 pixels
 - Height (LTDC): 272 pixels
 - Framebuffer Strategy: Double Buffer
 - Buffer Location: By Allocation
 - Driver:**
 - Application Tick Source: LTDC
 - Graphics Accelerator: ChromART (DMA2D)
 - Real-Time Operating System: CMSIS_RTOS_V2
- Pinout view:** A grid of pins and components, with several pins highlighted in green, indicating they are configured for the TouchGFX display.



Run TouchGFXDesigner

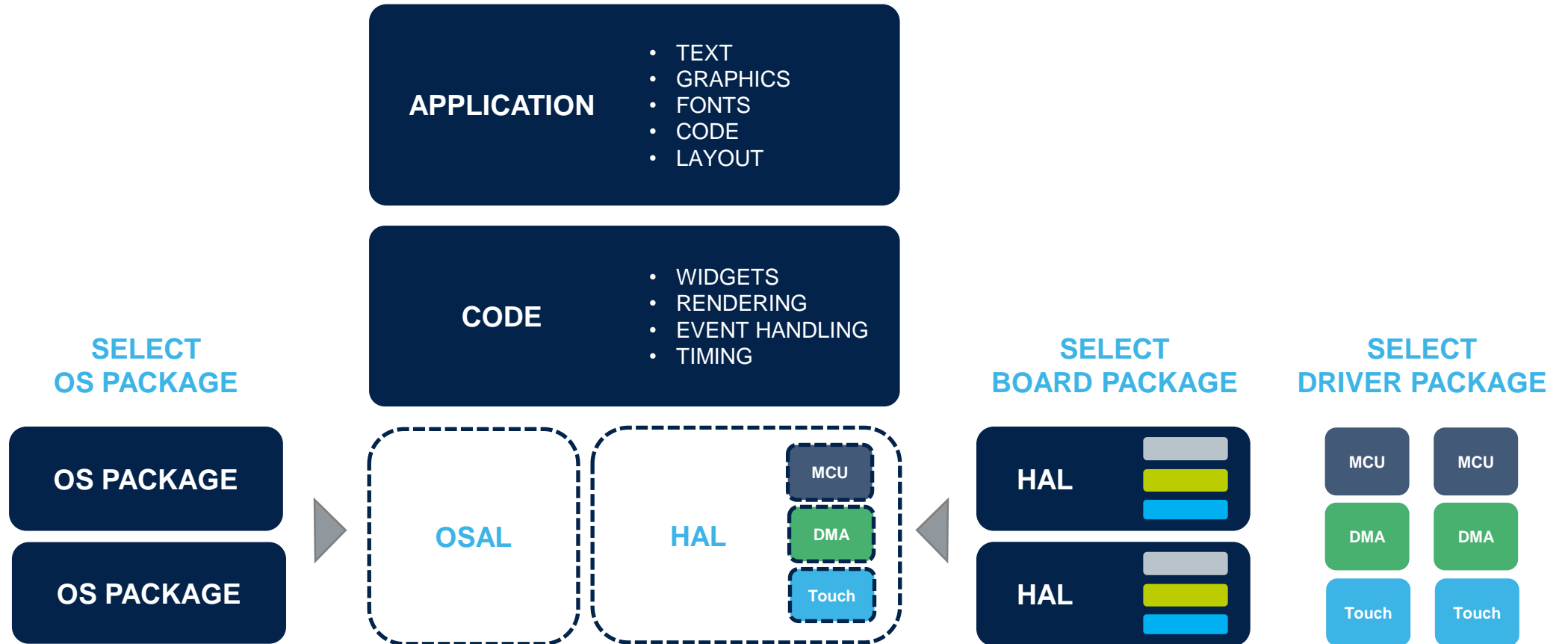


- With this, the advance preparation to design UI is ready. Let's make UI application with TouchGFX



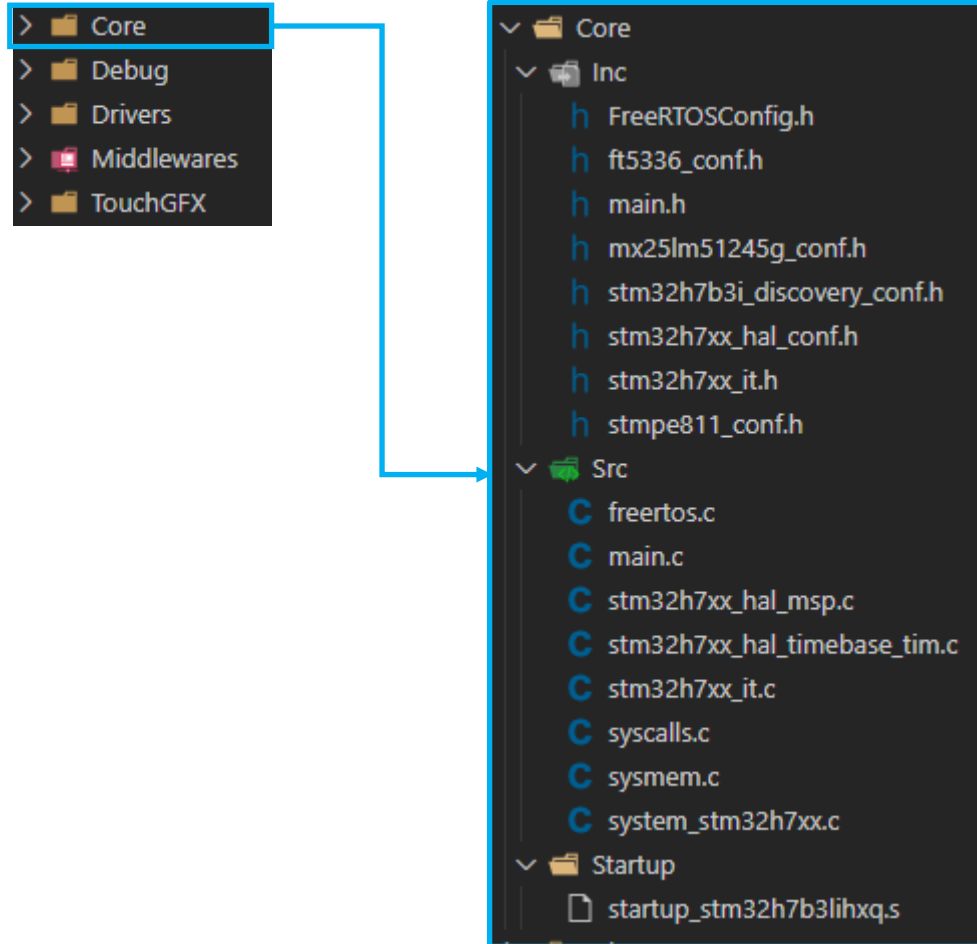


TouchGFX framework





TouchGFX project organization



../Core

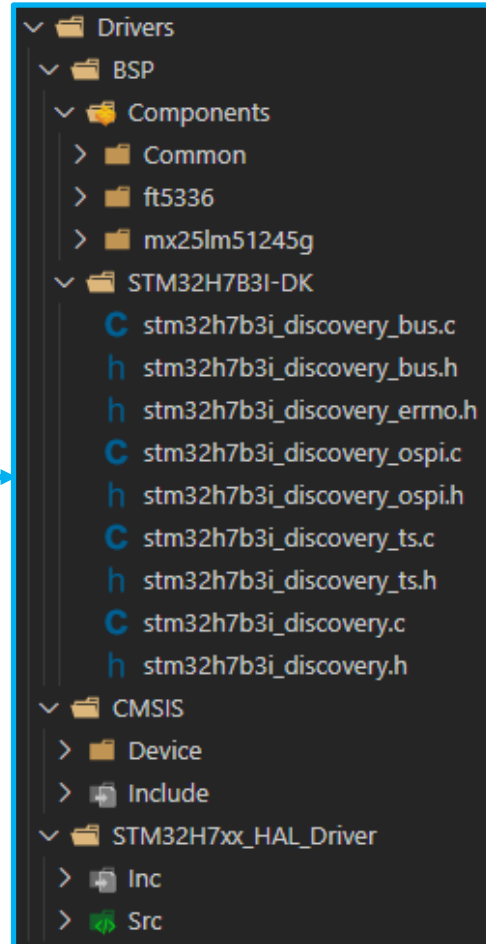
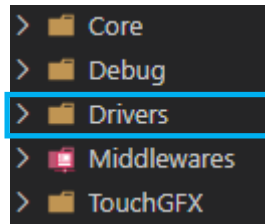
Contain CubeMX generated files for middleware, peripheral and GPIO initialization including main source and header files. Several header files related to configuration of STM32H7B3I-DK are also included.

../Startup

STM32H7B3LI Startup file for GCC based toolchain



TouchGFX project organization



../BSP/Components

Those files provide a set of functions needed to manage each components embedded on STM32H7B3I-DK (IO Expender and Touch, OctoSPI NOR Flash ...)

../BSP/STM32H7B3I-DK

Those files provide a set of firmware functions to manage the external device available on STM32H7B3I-DK board

../BSP/CMSIS

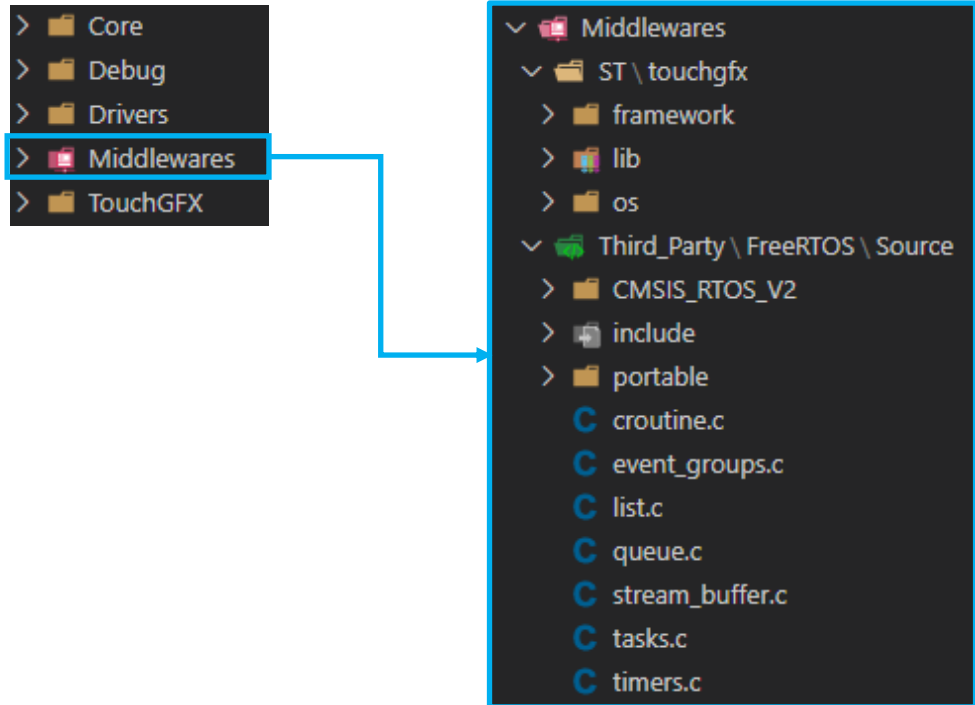
Cortex Microcontroller Software Interface Standard, a vendor-independent hardware abstraction layer

../STM32H7XX_HAL_Driver

STM32Cube Hardware abstraction layer for STM32H7



TouchGFX project organization



../ST/touchgfx

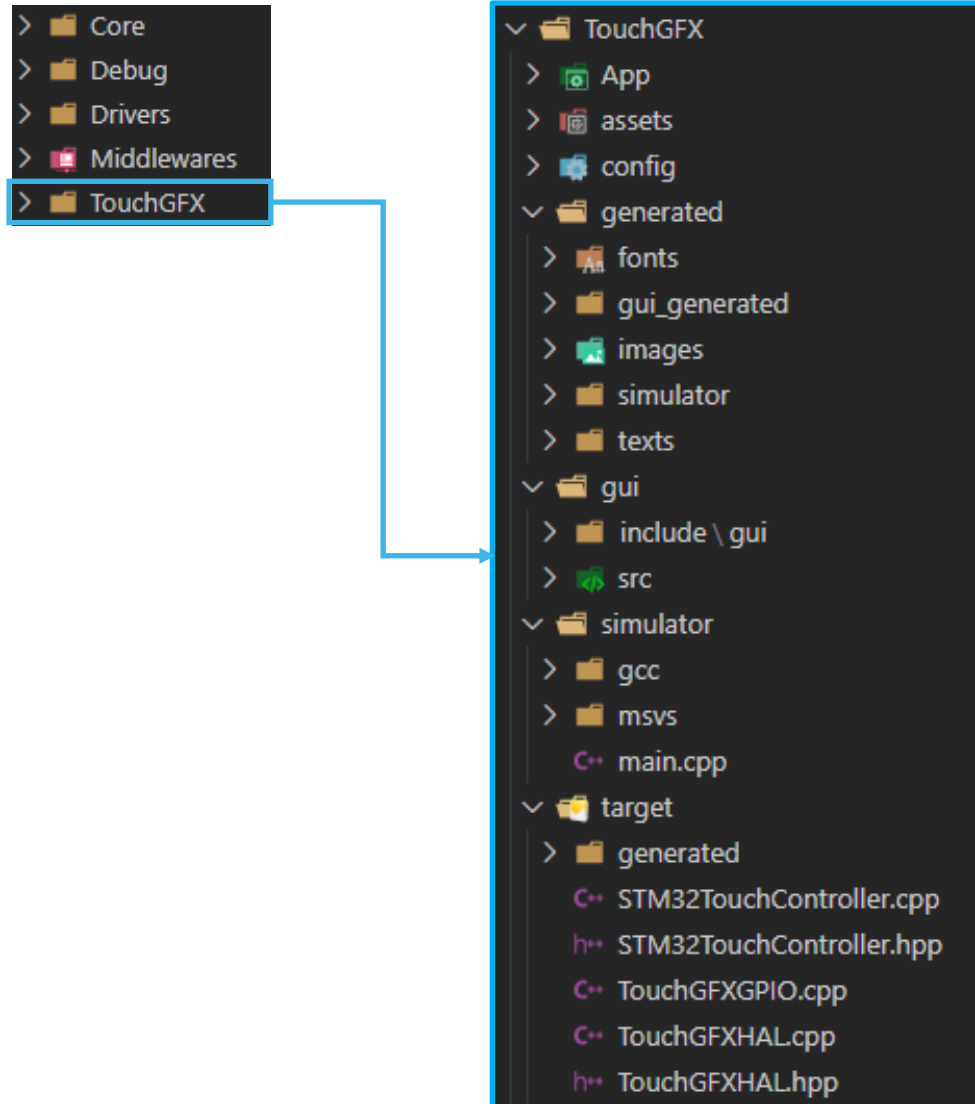
- TouchGFX framework including the widget and containers
- TouchGFX engine library

../Third_Party/FreeRTOS/Source

FreeRTOS source code(queue, task, timer...)



TouchGFX project organization



../App

To initialize TouchGFX and GUI Task, functions will call forward to TouchGFX hal class in C++ domain

../generated

Generated by TouchGFXDesigner(including UI, font, bitmap image and text). Those are basically not workspace for the user.

../gui

Consist of Model-View-Presenter source and header files so that user can write own code in order to actually implement GUI operation

../simulator

This is for simulator and includes the project file for Visual Studio.

../target

Generated by X-Cube-TouchGFX to manage the display interfaces and touch control.

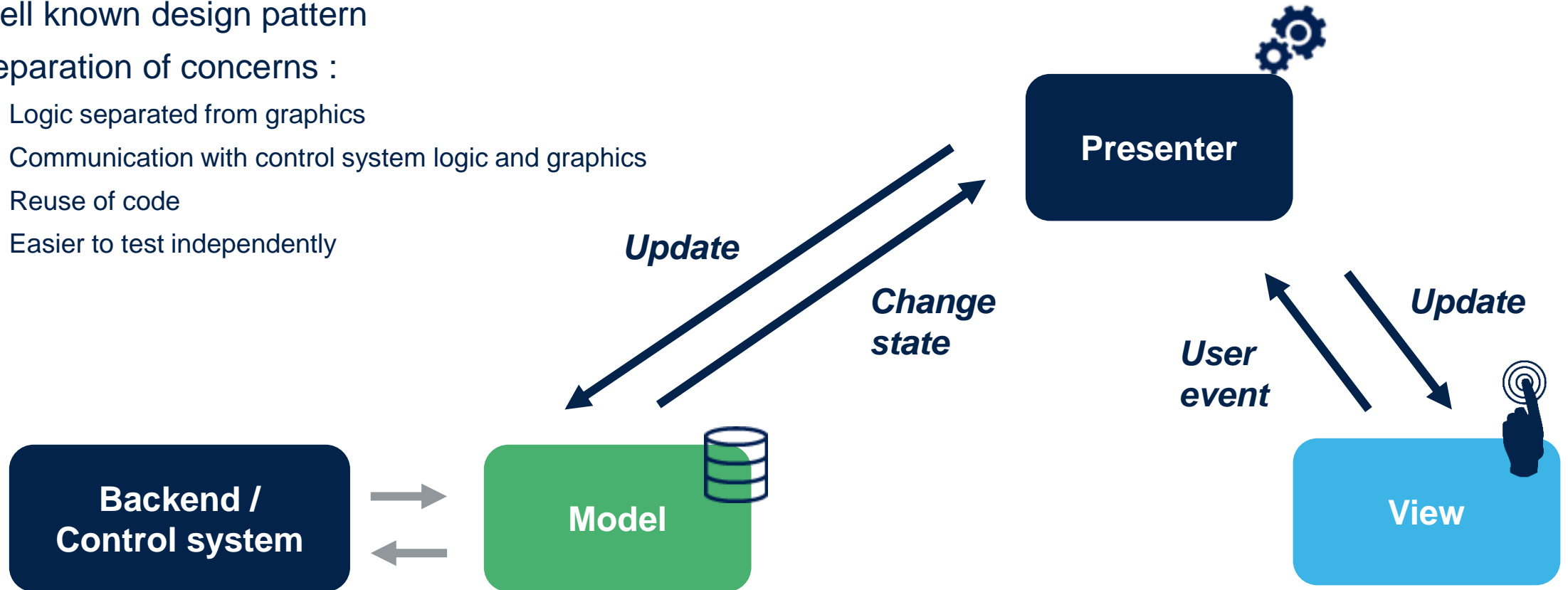
Files in “generated” folder is basically not workspace for the user but, you can overwrite the generated implementation.



MVP design pattern

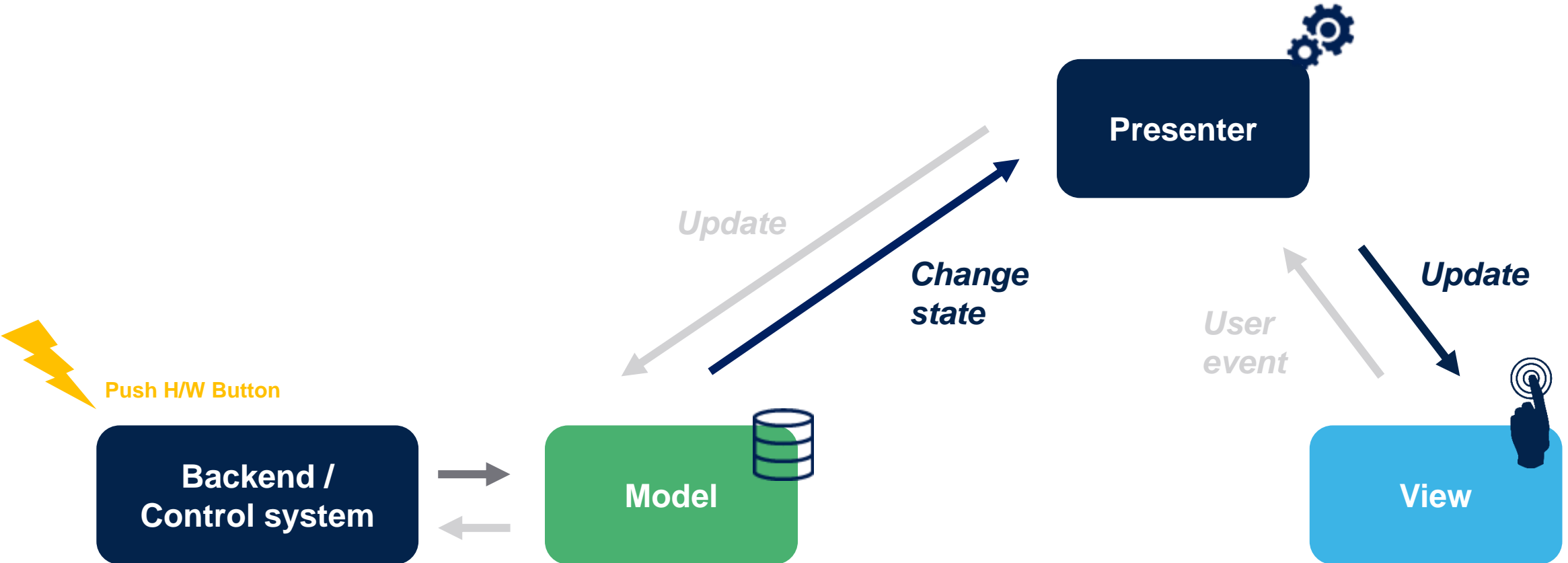
- Model-View-Presenter Software Architecture

- Well known design pattern
- Separation of concerns :
 - Logic separated from graphics
 - Communication with control system logic and graphics
 - Reuse of code
 - Easier to test independently



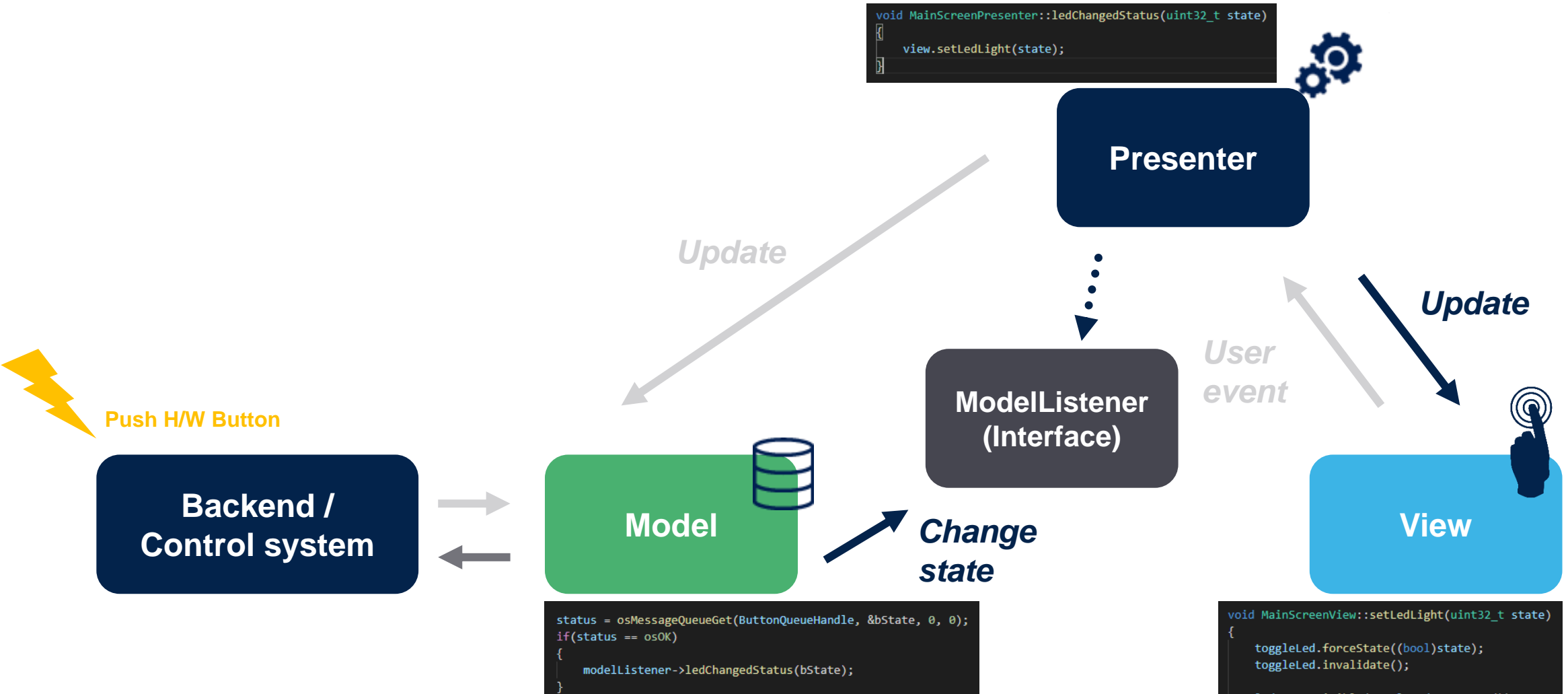


MVP design pattern





MVP design pattern



```
void MainScreenPresenter::ledChangedStatus(uint32_t state)
{
    view.setLedLight(state);
}
```

```
status = osMessageQueueGet(ButtonQueueHandle, &bState, 0, 0);
if(status == osOK)
{
    modellistener->ledChangedStatus(bState);
}
```

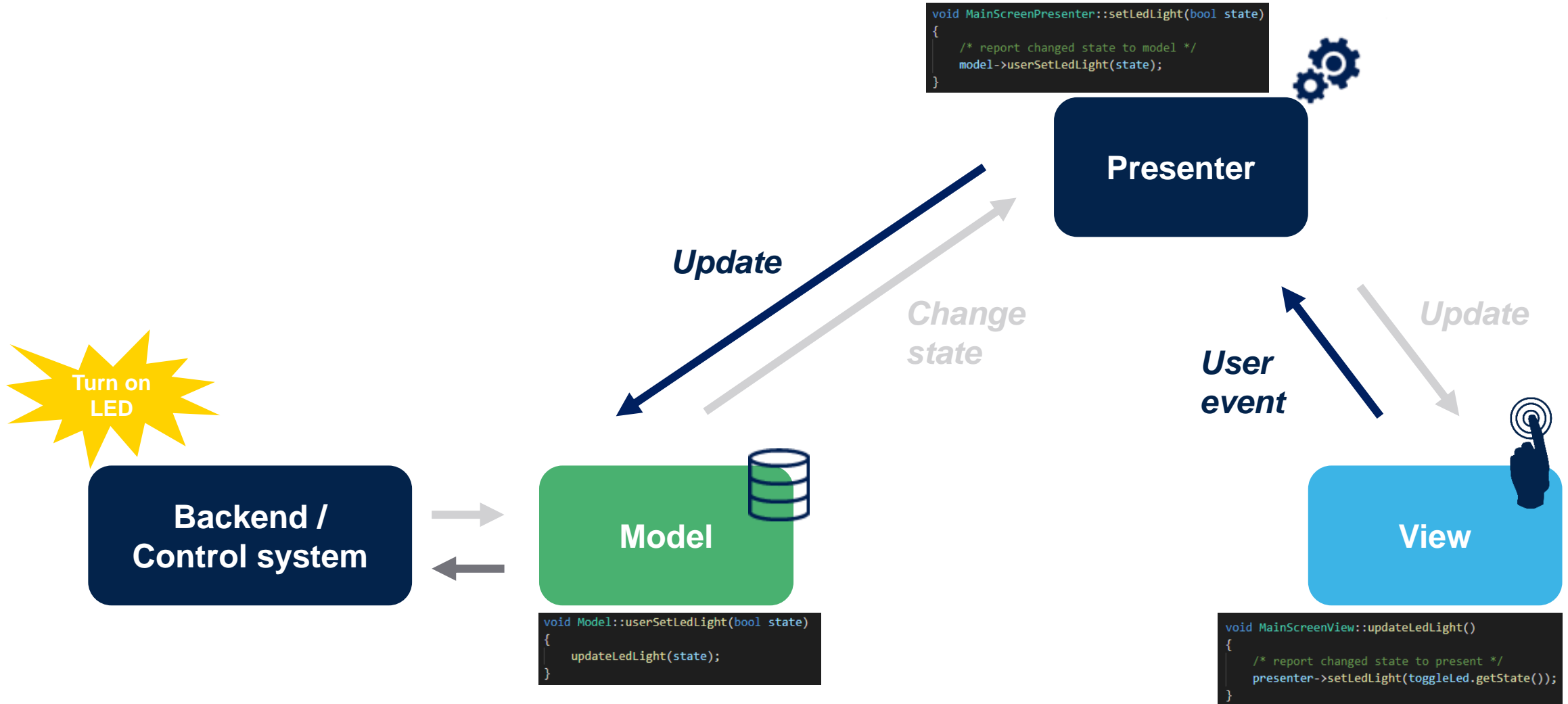
```
void MainScreenView::setLedLight(uint32_t state)
{
    toggleLed.forceState((bool)state);
    toggleLed.invalidate();

    ledOn.setVisible(toggleLed.getState());
    ledOff.setVisible(!toggleLed.getState());
    ledOn.invalidate();
    ledOff.invalidate();

    updateLedLight();
}
```



MVP design pattern



Thank you

© STMicroelectronics - All rights reserved.

The STMicroelectronics corporate logo is a registered trademark of the STMicroelectronics group of companies. All other names are the property of their respective owners.



life.augmented