

STM32 Graphics



PC & mobile user experience revolution

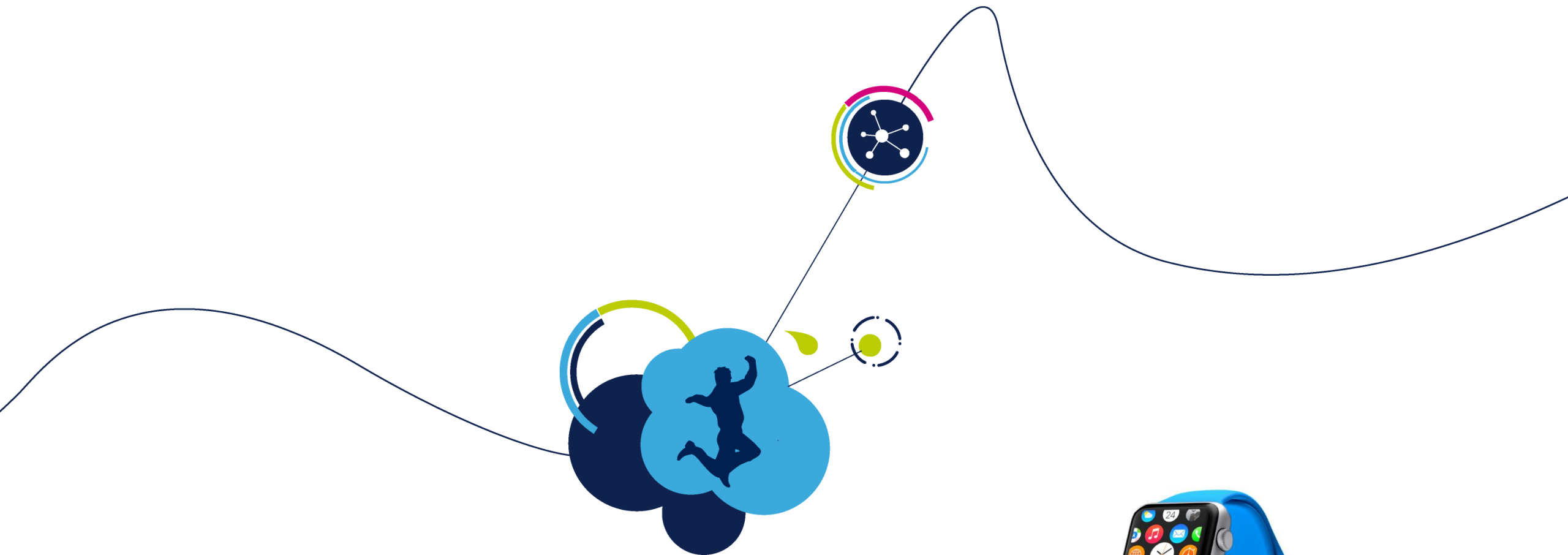
2

Comes to everyday products



Towards enhanced user experience

- Smarter and richer application
- **Advanced User interfaces**
- Extended connectivity
- Security



The right products

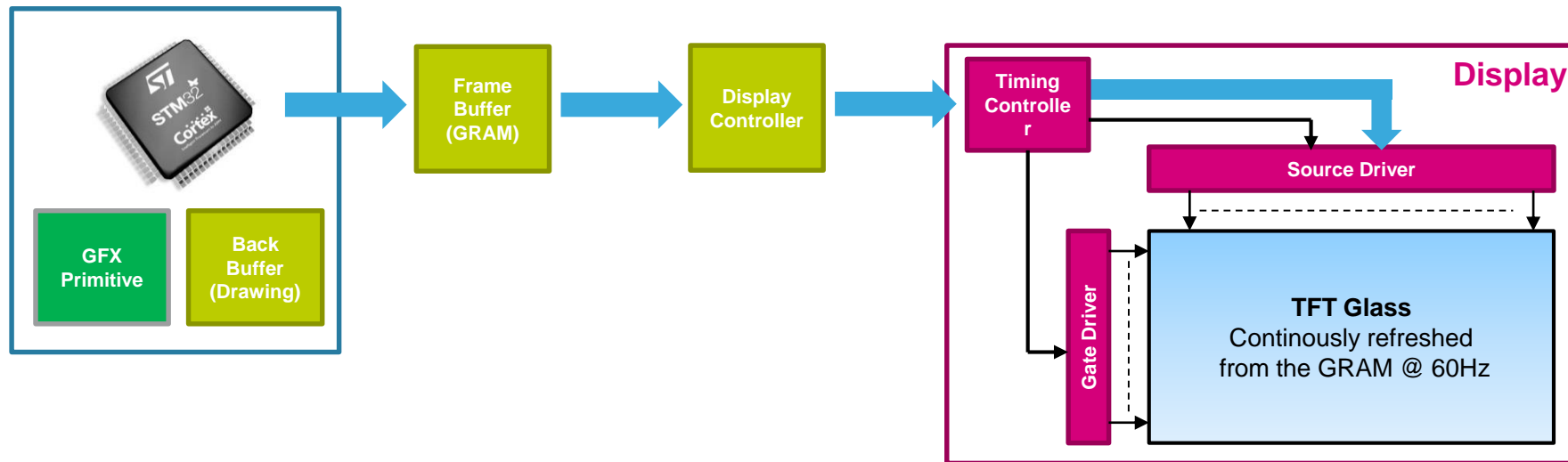
Hardware setup

4

Microcontroller setup – Low cost, low power consumption and low complexity

- **Graphical system consists of 4 elements**

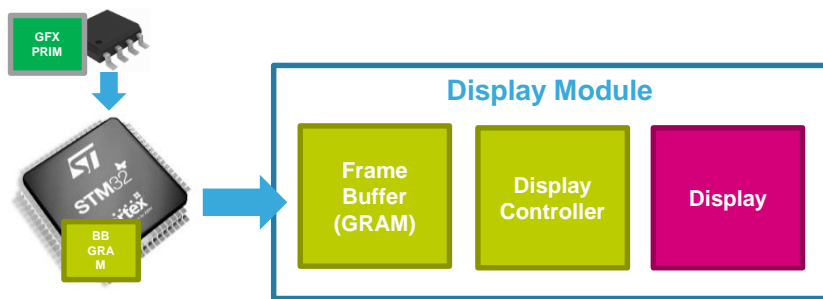
- A frame buffer (or GRAM) is a linear chunk of memory containing the desired pixel values
- The display glass must be continuously updated (~50/60 Hz typ.) even if the pixels do not change



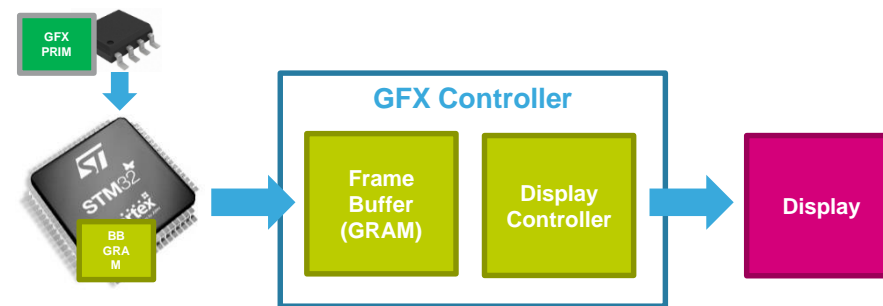
Display hardware setup

5

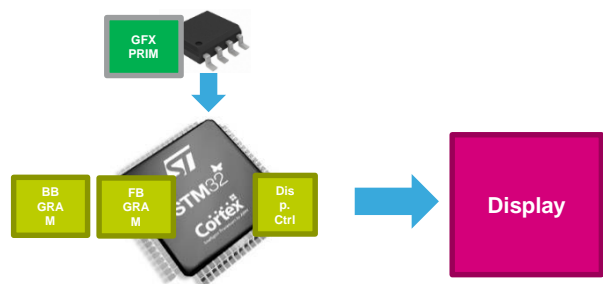
Four hardware configurations for display



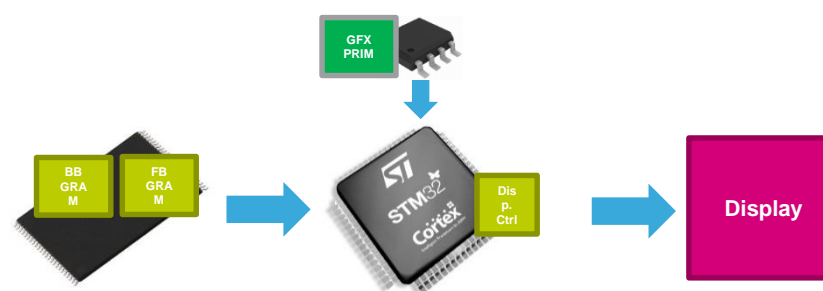
STM32 + Display Module + 1 NOR Flash (opt.)



STM32 + GFX Controller + Display + 1 NOR Flash (opt.)



STM32 (GRAM + Disp.Ctrl) + Display + 1 NOR Flash (opt.)



STM32 + (Disp.Ctrl) + PSRAM/SDRAM + Display + 1 NOR Flash

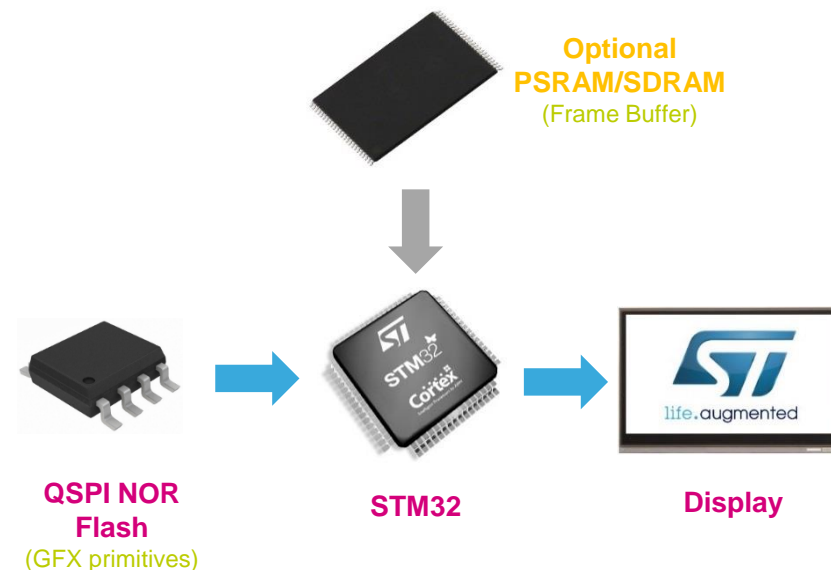
MCU Ressources

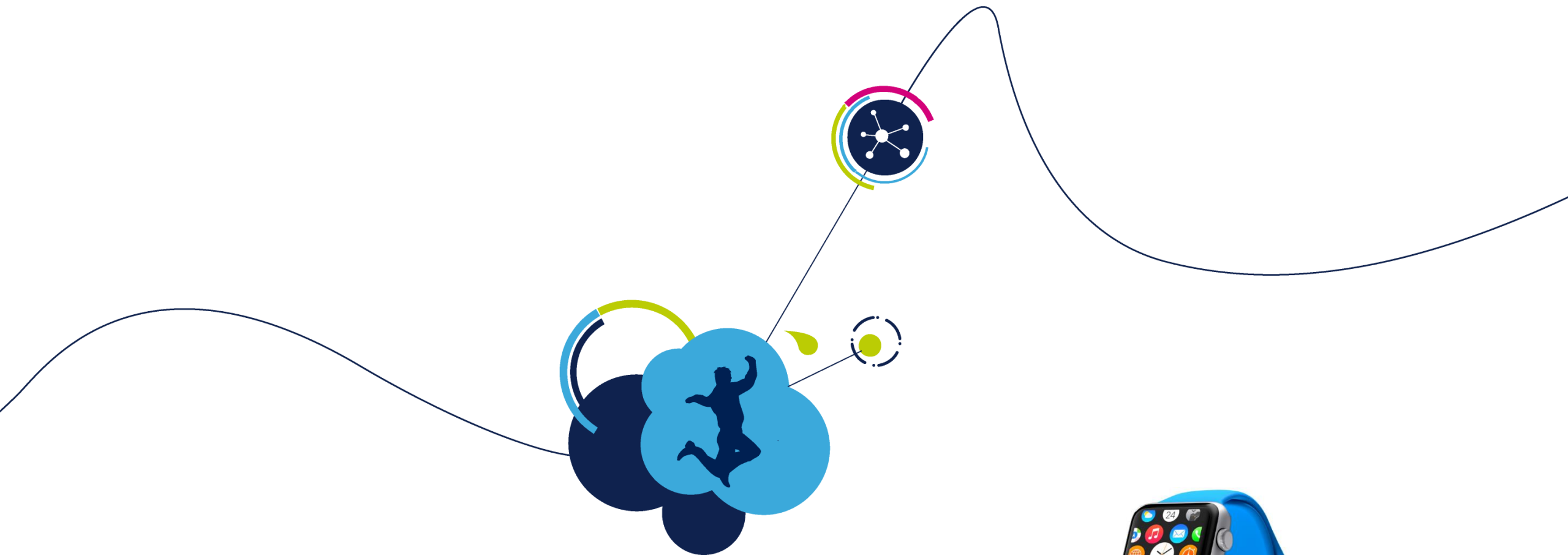
- **Microcontroller**

- Internal LCD-TFT Controller
- Internal Flash (< 2048 kB)
- Internal RAM (< 512/1024 kB)
- External NOR Flash (~8/32 MB)
- External PSRAM or SDRAM (~1-2/4 MB)

- **Frame buffer**

- LCD-TFT Controller updates the display with a constant frequency(~17ms / 60Hz)
- Effective frame rate depends on the rendering time
- Single vs. double buffering
- **Size:** #pixels x color depth x #buffers
- **Example (WQVGA):** 480x272 x 2 Bytes x 2 = 510KB





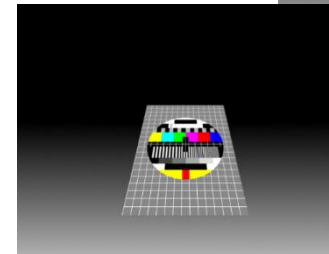
The right libraries

Evaluation of library on the market

8

HMI oriented benchmark

- **Menu bench (HMI with animation)**
 - Checks level of DMA2D support (All operation can be performed w/ DMA2D)
 - Checks level of integration of DMA2D in the code (shall be 0 CPU load)
 - Check font rendering methods
- **Needle bench (tachometer)**
 - Check rotation speed (all done by SW)
 - Check rasterisation algorithm
- **Coverflow**
 - Advanced HMI with geometrical transformations



Select the best solutions

9

Two clear technical positioning

Entry Level

First target is MCU, optimized ressource usage

TouchGFX 



expresslogic

High-End

Derivated from MPU, huge ressource usage



Crank
software inc.

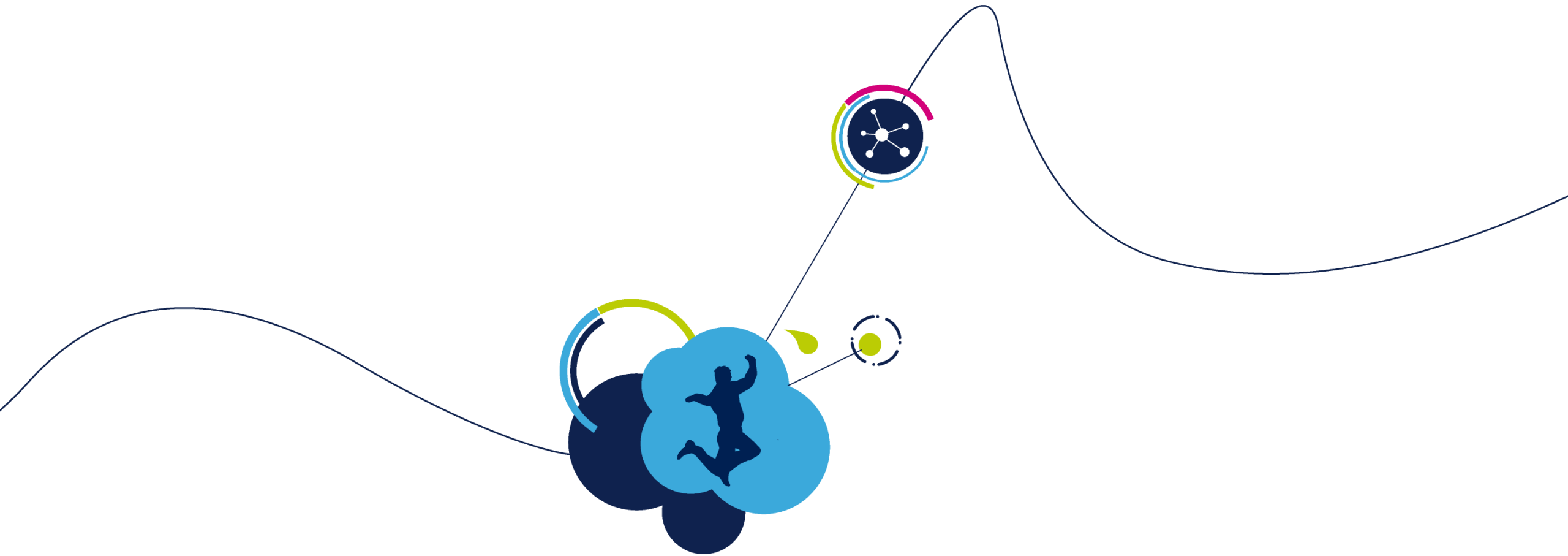
New STM32CubeMX Version 4.25

10

- Graphics integration in STM32CubeMX
 - Code generation for the supported stacks:
 - To choose the graphics stack to use
 - To configure the parameters and generate the project
 - Graphics Selector:
 - To select the right MCU basing on customer graphics criteria
 - Fully integrated in the current MCU Selector
 - Graphics Simulator:
 - To simulate graphics configurations using a set of relevant parameters
 - Evaluate performance of simulation graphic configuration
 - To apply the simulation data to the current user configuration in STM32CubeMX



STM32CubeMX with Graphics support: STemWIN now, TouchGFX & Embedded Wizard soon

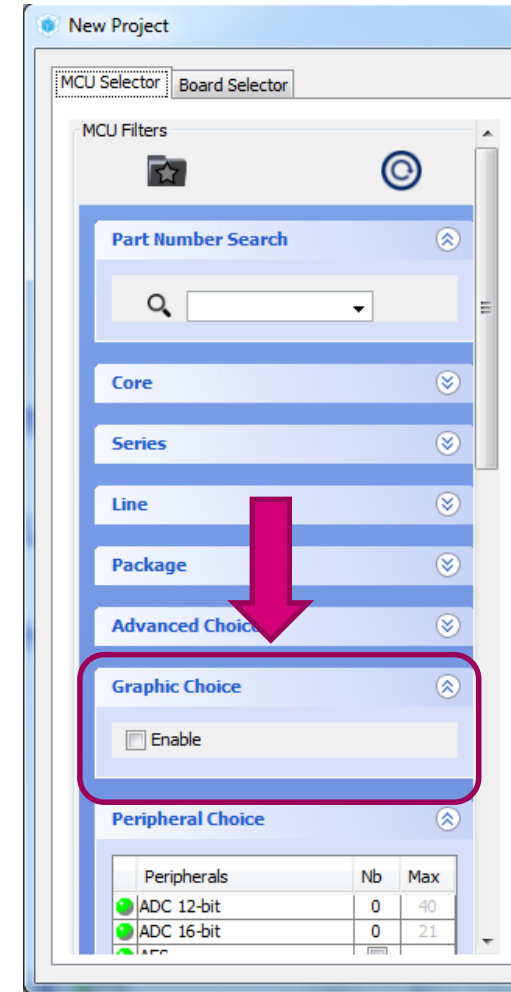


STM32CubeMX Graphics Selector

Graphics Selector main features (1/2)

12

- Fully integrated in MCU Selector
- New filter added “Graphics Choice”
- Click on “Enable” checkbox to display the list of graphics criteria
- Graphics criteria can be combined with other MCU Selector Criteria



Graphics Selector main features (2/2)

13

The screenshot shows the 'New Project' window of the ST Graphics Selector tool. The interface is divided into several sections:

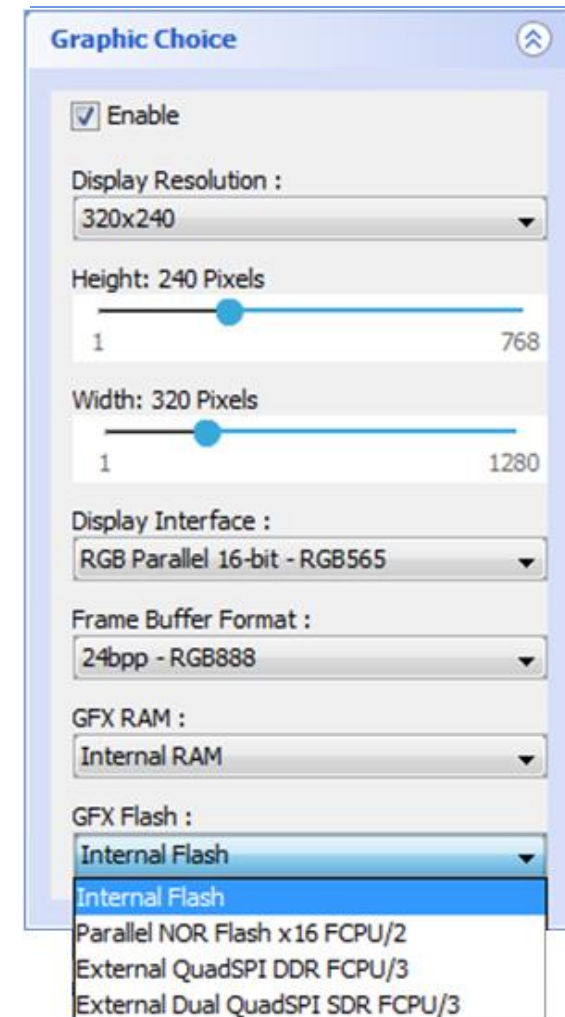
- MCU Selector / Board Selector:** Located at the top left, with a search bar and a list of components.
- Core:** A section for selecting the microcontroller core.
- Package:** A section for selecting the package type.
- Advanced Choice:** A section for selecting advanced options.
- Graphic Choice:** A section for selecting graphics-related options, including:
 - ☒ Enable
 - Display Resolution: 320x240
 - Height: 240 Pixels (slider from 1 to 768)
 - Width: 320 Pixels (slider from 1 to 1280)
 - Display Interface: RGB Parallel 16-bit - RGB565
 - Frame Buffer Format: 24bpp - RGB888
 - GFX RAM: Internal RAM
 - GFX Flash: Internal Flash
- Graphics summary - main data:** A section on the right showing a summary of the selected configuration, including:
 - Requirement Pixel Clock (MHz): 5.53
 - Requirement Graphic Ram (KBytes): 450.0
 - Maximum System Pixel clock (MHz): 134.27 to 149.18
 - Graphic Performance Score (GFXMark): 562.8 to 626.9
- MCUs List:** A table at the bottom showing 16 items, with columns for Part No, Reference, Marketing, Unit Price, Board, Package, Flash, RAM, IO, Freq., and GFX S... (Graphics Score).

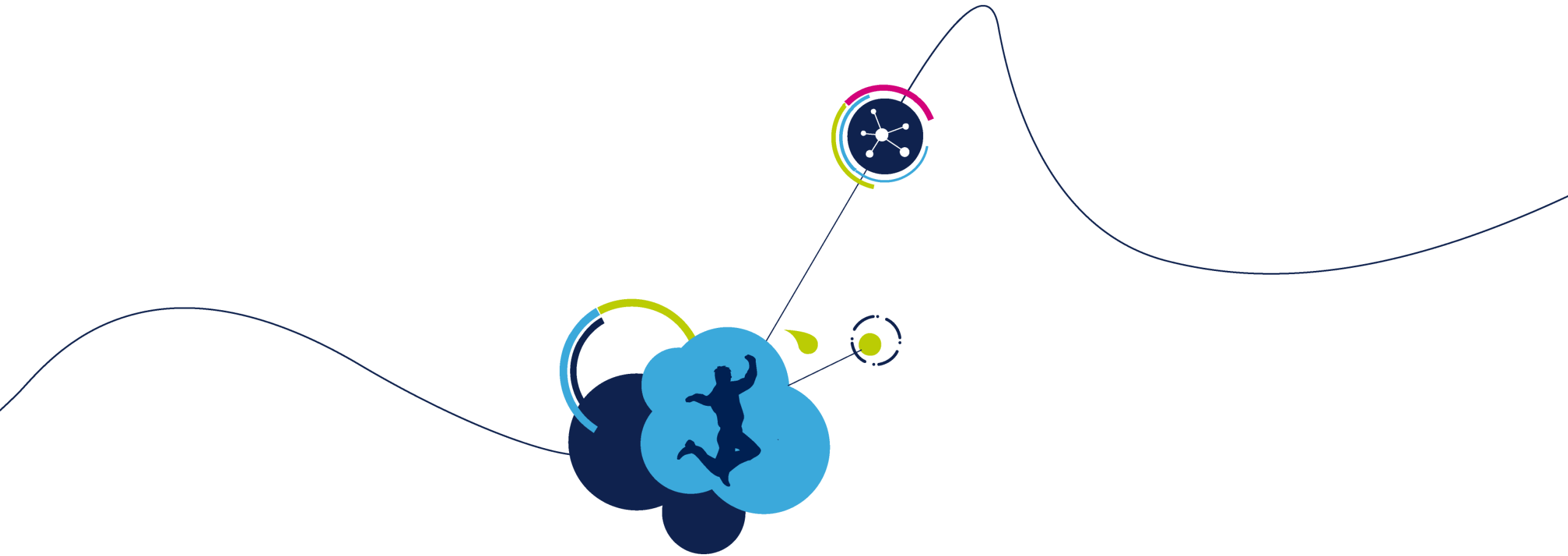
Annotations highlight key features:

- List of criteria specific for Graphics:** Points to the 'Graphic Choice' section.
- Graphics summary - main data:** Points to the 'Graphics summary' section.
- Graphics Score helps user to choose the best performance:** Points to the 'GFX S...' column in the 'MCUs List' table.

* Part No	Reference	Marketing...	Unit Price fo...	Board	Package	Flash	RAM	IO	Freq.	GFX S...
STM32F429II	STM32F429...	Active	8.777		LQFP176	2048 kBytes	256 kBytes	140	180 MHz	562.8
STM32F429NI	STM32F429...	Active	9.054	STM32429I-E	TFBGA216	2048 kBytes	256 kBytes	168	180 MHz	562.8
STM32F429VI	STM32F429...	Active	8.083		LQFP100	2048 kBytes	256 kBytes	82	180 MHz	562.8
STM32F429ZI	STM32F429...	Active	6.88						180 MHz	562.8
STM32F469NI	STM32F469...	Active	9.051						180 MHz	562.8
STM32F469VI	STM32F469...	Active	9.514						180 MHz	562.8
STM32F469ZI	STM32F469...	Active	8.357						180 MHz	562.8
STM32F469II	STM32F469...	Active	8.82		LQFP144	2048 kBytes	384 kBytes	106	180 MHz	562.8
STM32F469IG	STM32F469...	Active	8.24		LQFP176	1024 kBytes	320 kBytes	140	216 MHz	626.9
STM32F74...	STM32F74...	Active	8.703	3... S...	TFBGA216	1024 kBytes	320 kBytes	168	216 MHz	626.9
STM32F74...	STM32F74...	Active	7.43		LQFP100	1024 kBytes	320 kBytes	82	216 MHz	626.9

- Display Resolution
 - Predefined size
 - User defined size with width & height size in slider below
- Display interface
- Product Memories
 - GFX RAM
 - GFX Flash





Graphics stacks integration in STM32CubeMX

Graphics stacks integration (1/2)

16

- Graphics module is represented in STM32cubeMX as a middleware that integrates three different graphics frameworks: **STemWin**, **ST-EmbeddedWizard** and **ST-TouchGFX**.



Now



Coming

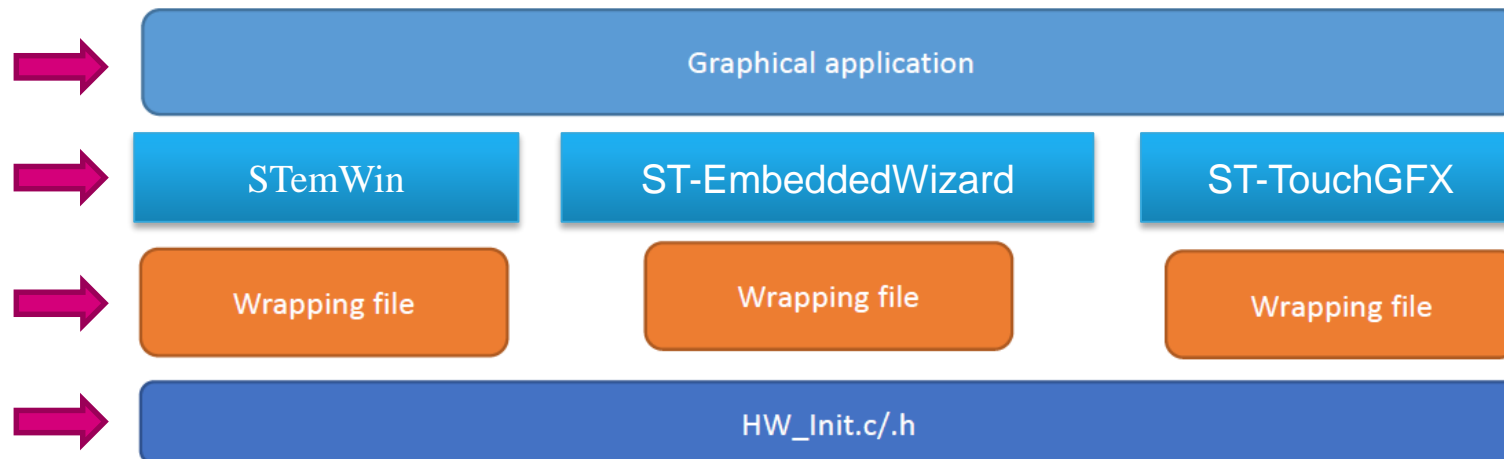


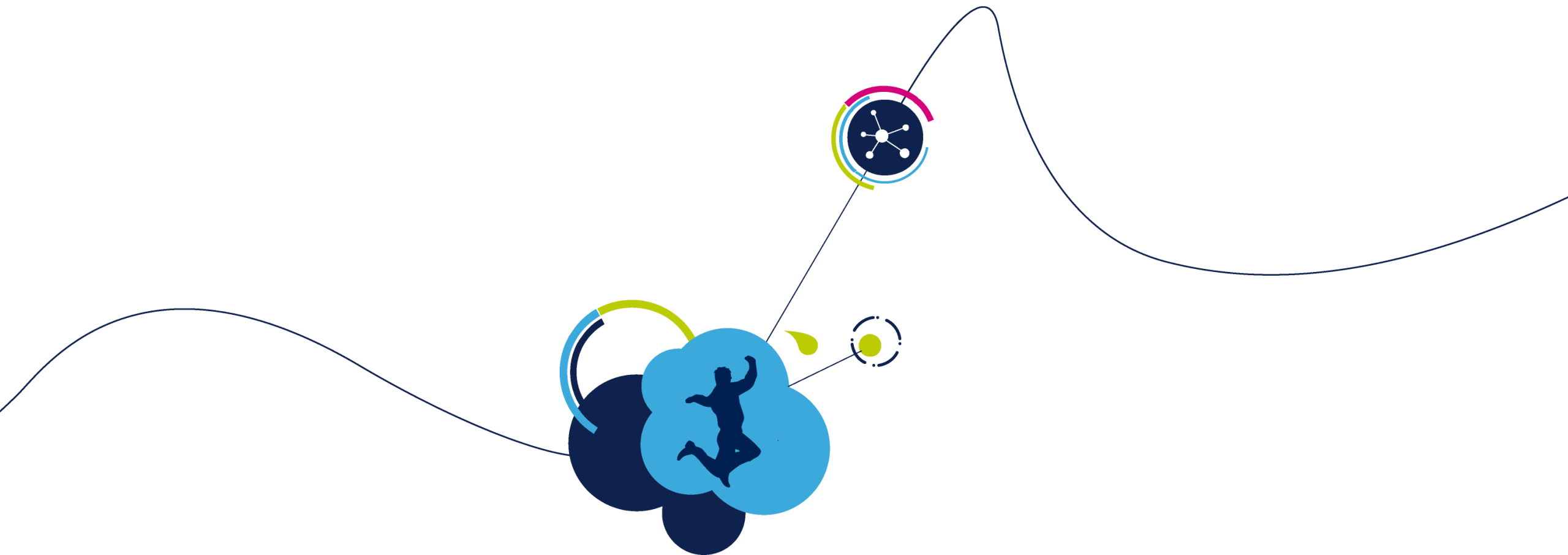
Coming

Graphics stacks integration (2/2)

17

- The goal is to have only one hardware initialization file for all the graphics frameworks with a wrapper file for each stack.
 - Hardware initialization (HW_Init)
 - Add the configuration files (STemWin_wrapper..) for the project to ensure a specific settings for each stack
 - Initialize the graphic library
 - Generate the final project





STemWin

2

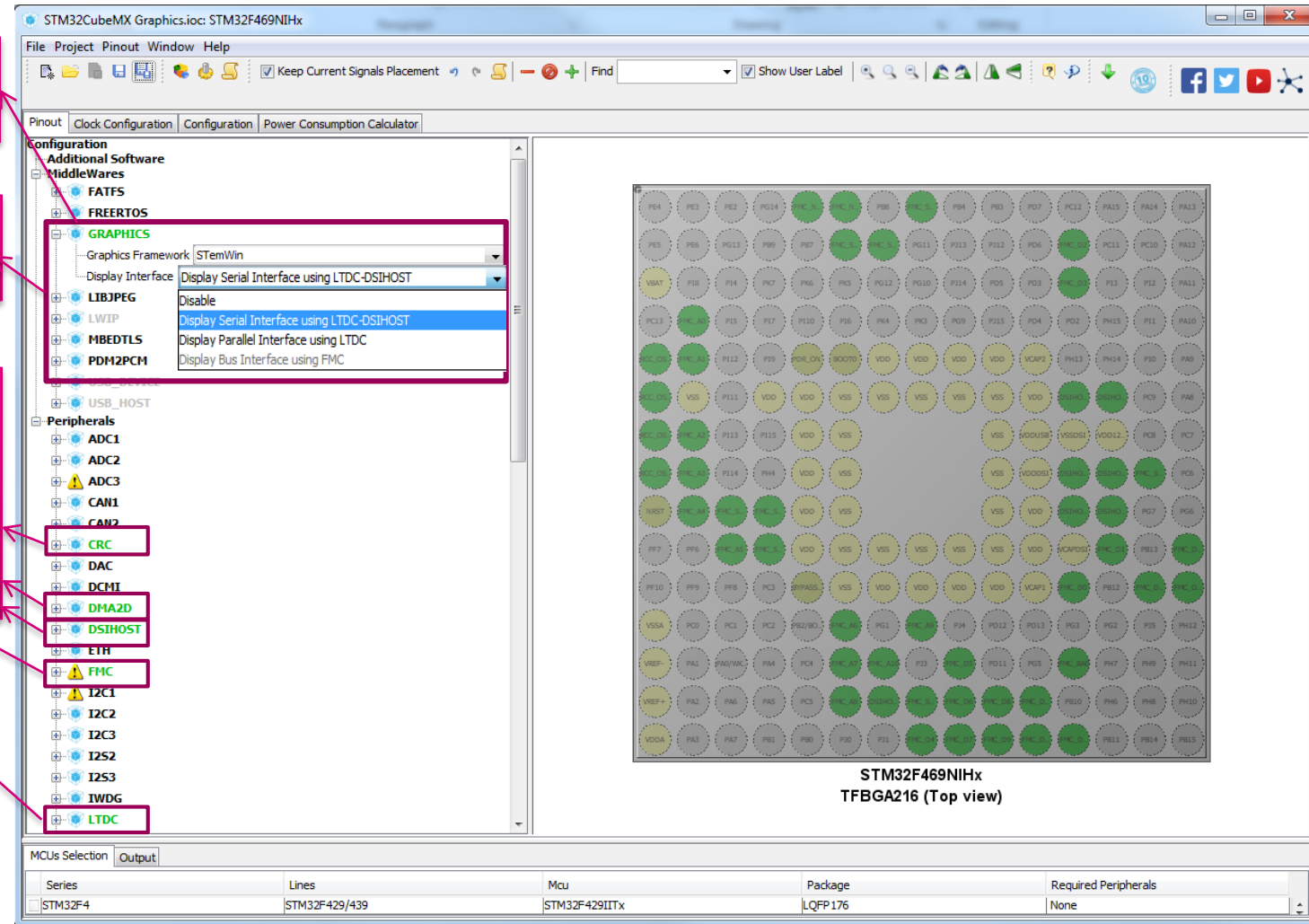
Select the STemWin stack from Graphics MiddleWares.

3

Choose one of the display interface type.

1

Enable the IPs (CRC , DMA2D in case of use the Graphics acceleration ,FMC (SDRAM/SRAM) , LTDC and DSIHOST) that will be used for the connection with the display interface.



Configuration View

20

The image shows the STMicroelectronics configuration tool interface. On the left, the 'GRAPHICS Configuration' dialog is open, displaying various settings for the STM32F429IITx. The 'Parameter Settings' tab is selected, showing parameters like Stack Name and Version, External Tool, Number of Layers, Physical Display Size, Display Driver, Multiple Buffers - Virtual Screens, Frame Buffer, and GUI Memory size. A pink box highlights the 'Parameter Settings' tab. A pink arrow points from a text box to the 'GRAPHICS' button in the 'Middlewares' section of the main configuration view.

Through this interface you will be able to configure all parameters used for the STemWin UI applications.

4

Middlewares

GRAPHICS

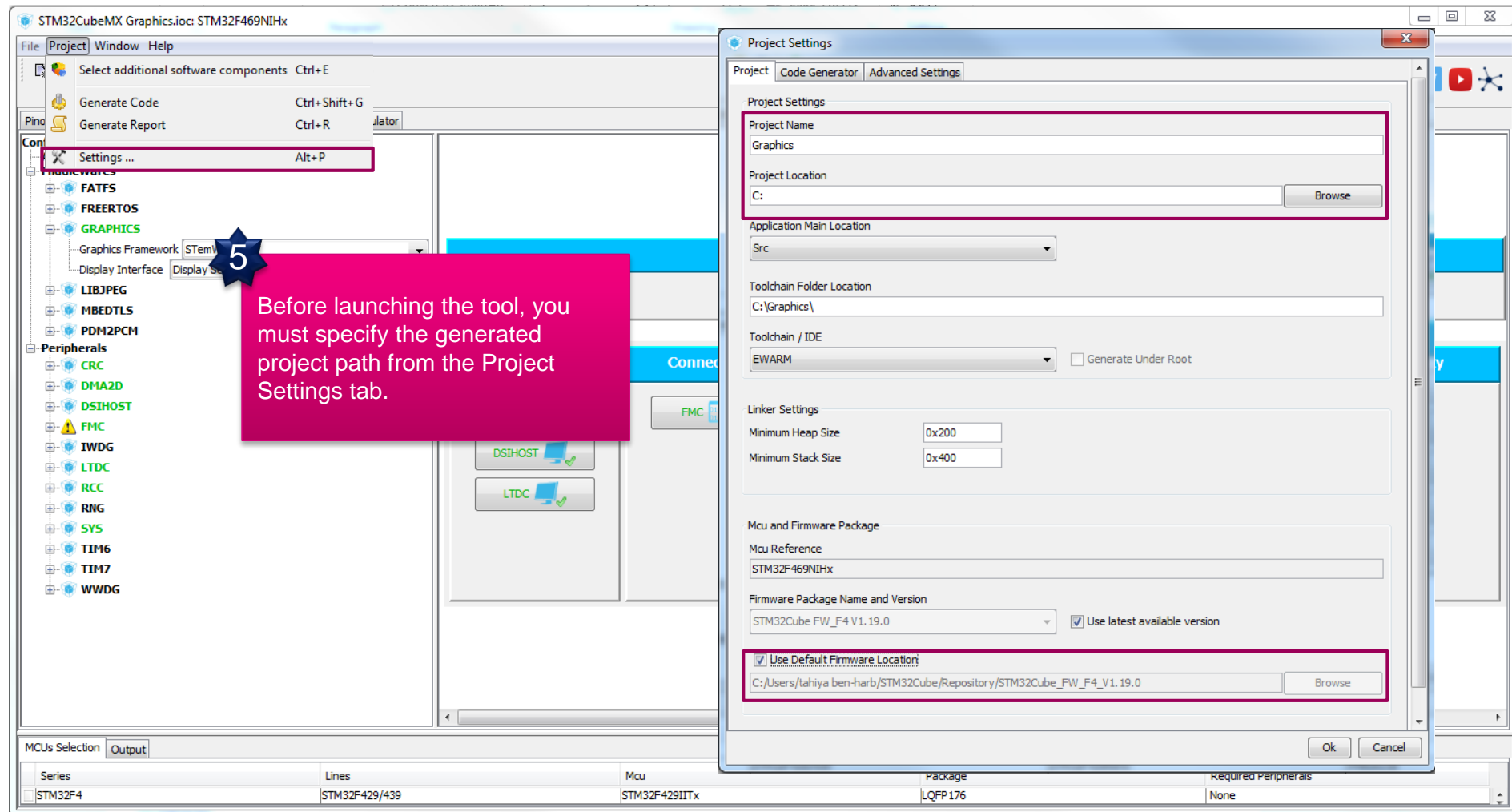
Analog	System	Control	Security
	CRC ✓		
	DMA +		
	GPIO ✓		
	NVIC ✓		
	RCC ✓		

MCUs Selection

Series	Lines	McU	Package	Required Peripherals
STM32F4	STM32F429/439	STM32F429IITx	LQFP 176	None

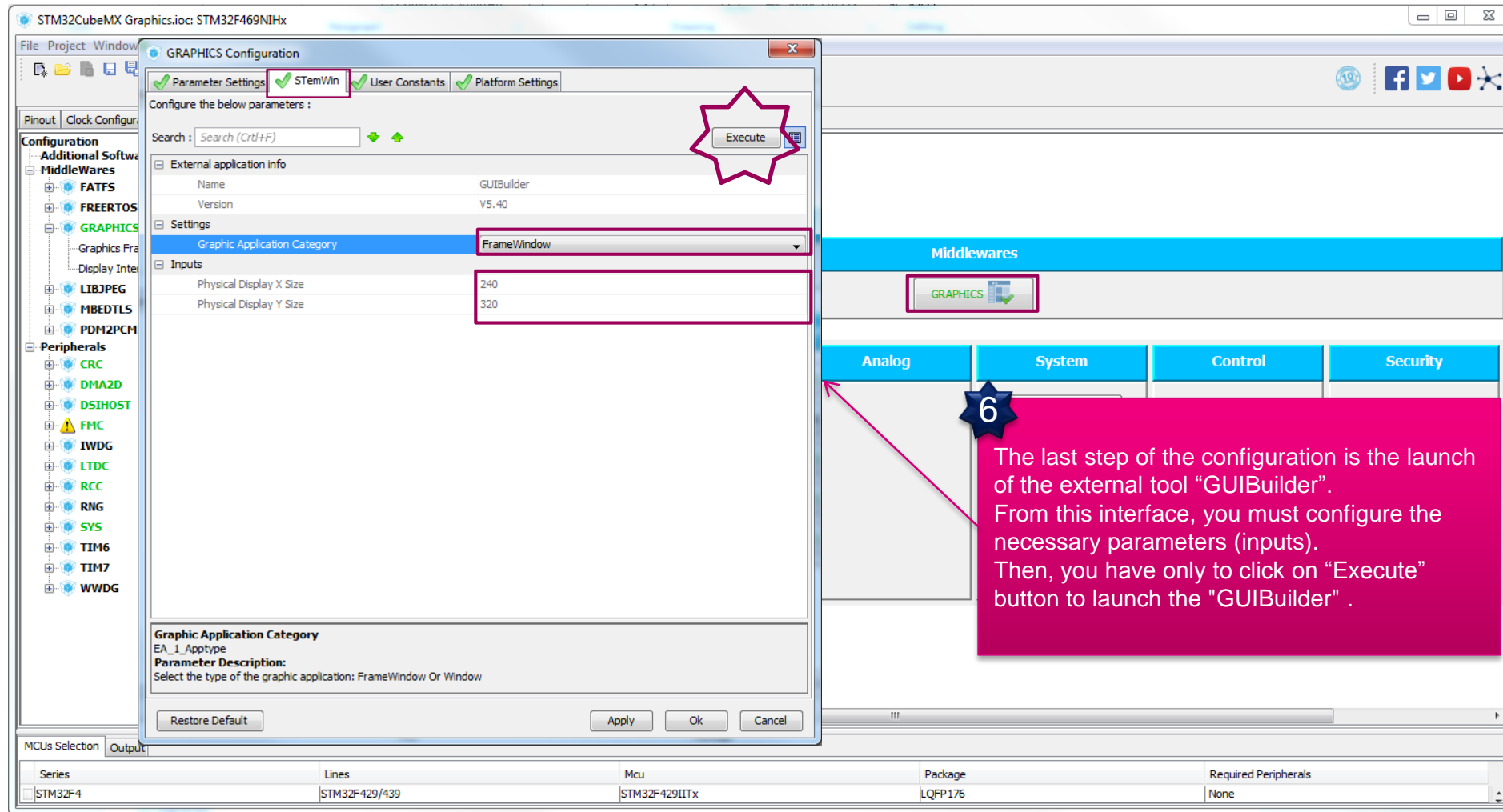
Project Settings

21

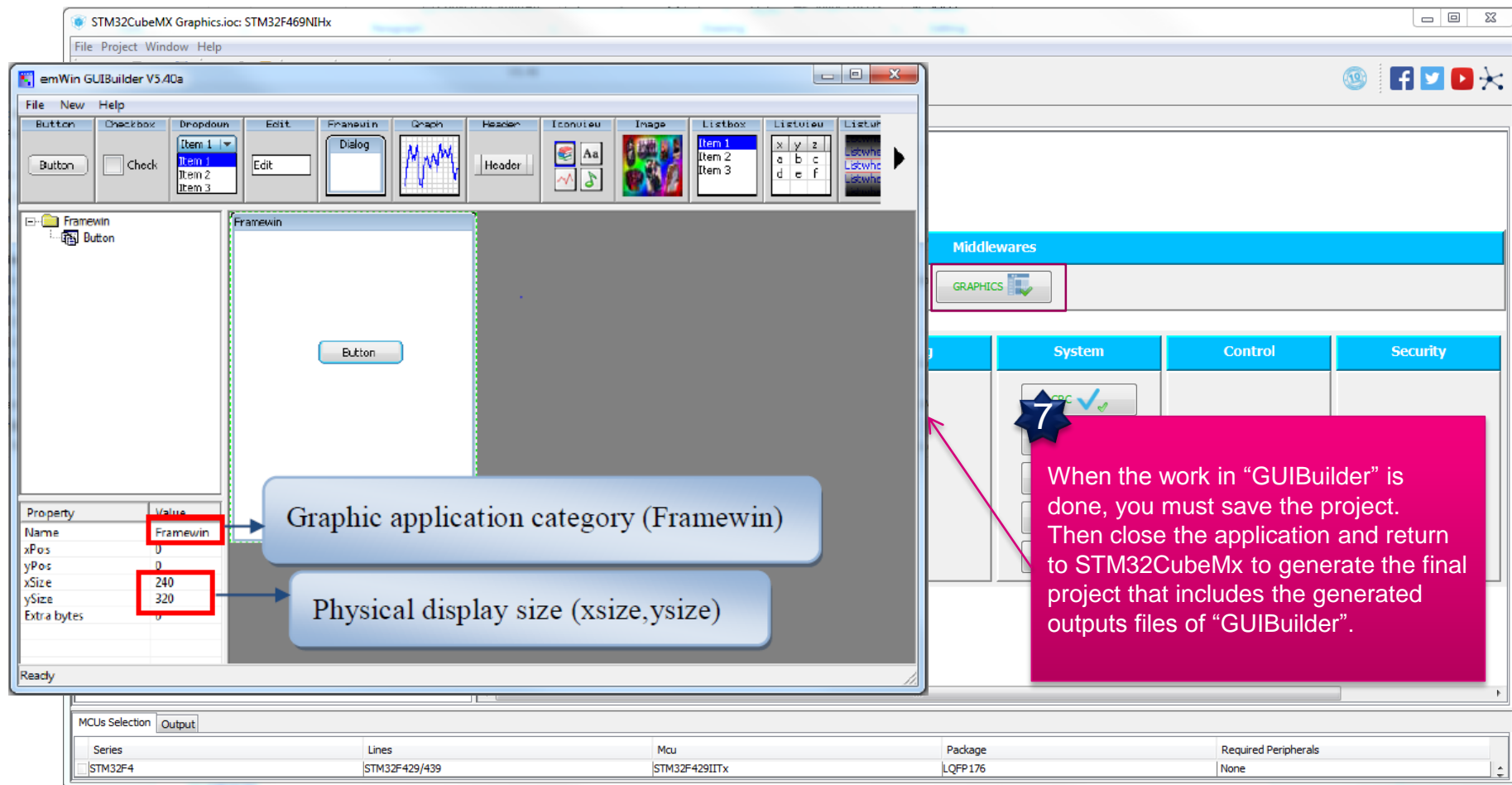


STemWin Configuration View

22



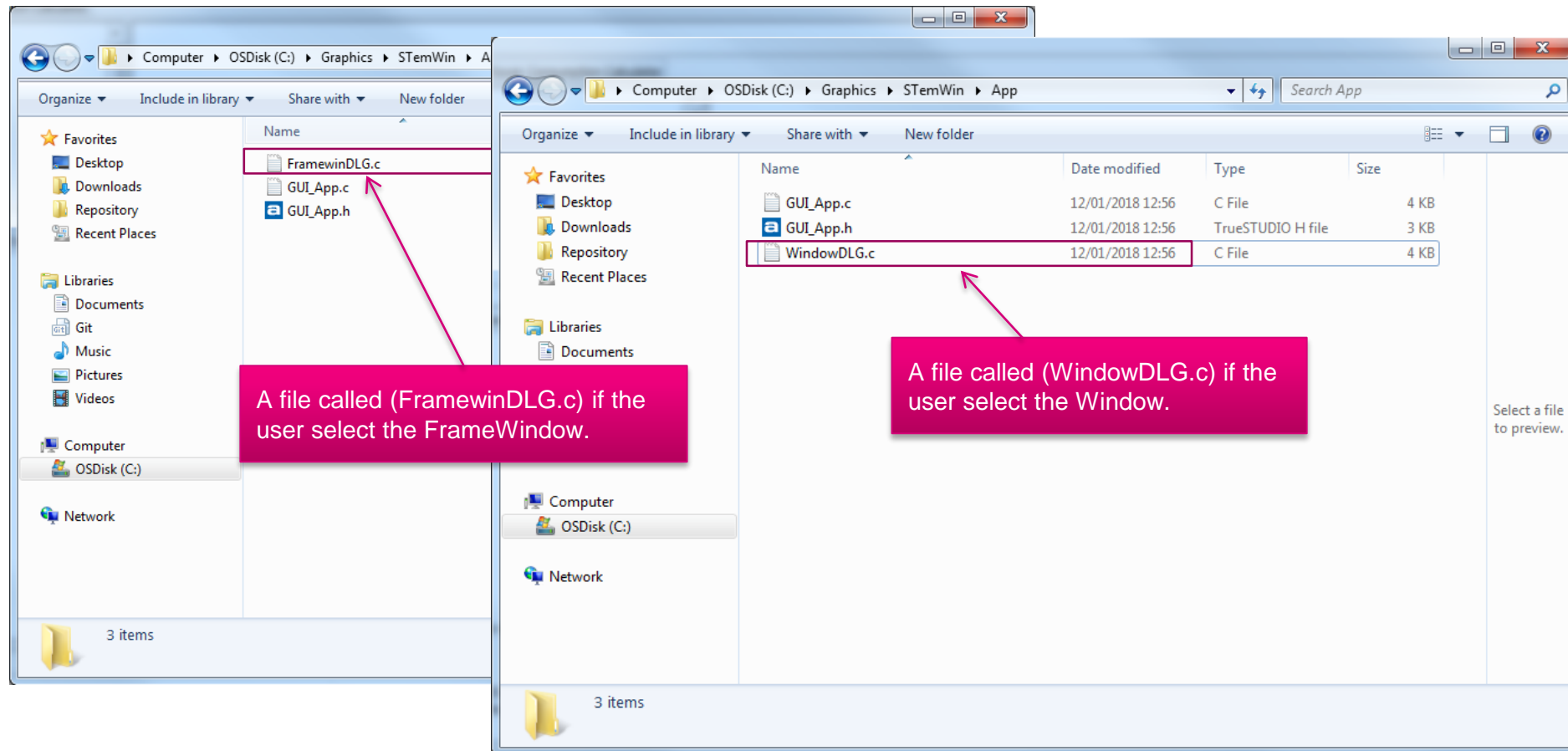
The GUIBuilder will start with the same values of the parameters “Graphical application category (FrameWindow or Window)” and “Physical Display Size (xsize, ysize)” configured in STM32CubeMx .



Project Generation (1/2)

24

The GUIBuilder will generate a file that can be either customized or integrated into the final project generated by STM32CubeMx according to your configuration.



Project Generation (2/2)

25

```
#define XSIZE_PHYS 240
#define YSIZE_PHYS 320

#define ZONES 2 /* expected value 2,4 */
#define HACT XSIZE_PHYS/ZONES /* SCREEN DIVIDED INTO TWO AREAS */

#define NUM_BUFFERS 1 /* Number of multiple buffers to be used */
#define NUM_VSCREENS 1 /* Number of virtual screens to be used */

#define COLOR_CONVERSION_0 GUICC_665
#define DISPLAY_DRIVER_0 GUIDRV_LIN_16

#define COLOR_CONVERSION_1 GUICC_888
#define DISPLAY_DRIVER_1 GUIDRV_LIN_32

#define LCD_LAYER0_FRAME_BUFFER ((uint32_t)0x0) /* LTDC Layer 0 frame buffer */
#define LCD_LAYER1_FRAME_BUFFER ((uint32_t)0x0) /* LTDC Layer 0 frame buffer */

#define __DSI_MASK_TE() (GPIOJ->AFR[0] &= (0xFFFF0FFF)) /* Mask DSI TearingEffect Pin*/
#define __DSI_UNMASK_TE() (GPIOJ->AFR[0] |= (uint32_t)(GPIO_AF13_DSI) << 8) /* UnMask DSI TearingEffect Pin*/

extern LTDC_HandleTypeDef hltcd;
extern DSI_HandleTypeDef hdsi;
extern volatile GUI_TIMER_TIME OS_TimeMS;

uint8_t pPage[] = {0x00, 0x00, 0x01, 0xDF}; /* 0 -> 479 */

/* Constant .. To be generated with OTMS009 LCD driver */
uint8_t pCols[ZONES][4] =
{
    #if (ZONES == 4)
    {0x00, 0x00, 0x00, 0xC7}, /* 0 -> 199 */
    {0x00, 0xC8, 0x01, 0x8F}, /* 200 -> 399 */
    {0x01, 0x90, 0x02, 0x57}, /* 400 -> 599 */
    {0x02, 0x58, 0x03, 0x1F}, /* 600 -> 799 */
    #elif (ZONES == 2)
    {0x00, 0x00, 0x01, 0x8F}, /* 0 -> 399 */
    {0x01, 0x90, 0x03, 0x1F}
    #endif
};

static LCD_LayerPropTypeDef layer_prop[GUI_NUM_LAYERS];
volatile int32_t LCD_ActiveRegion = 1;
volatile int32_t LCD_Refreshing = 0;
volatile char TransferInProgress = 0;

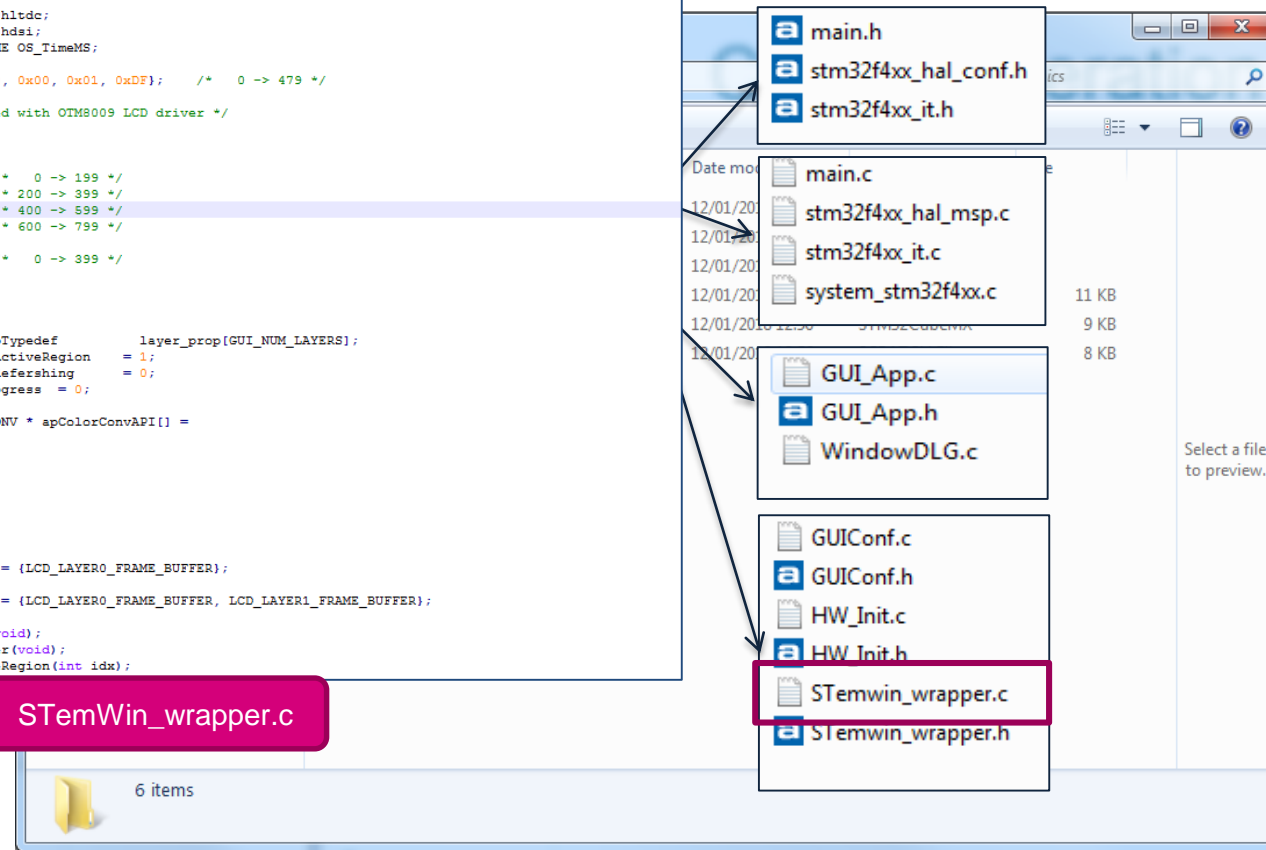
static const LCD_API_COLOR_CONV * apColorConvAPI[] =
{
    COLOR_CONVERSION_0,
    #if GUI_NUM_LAYERS > 1
    COLOR_CONVERSION_1,
    #endif
};

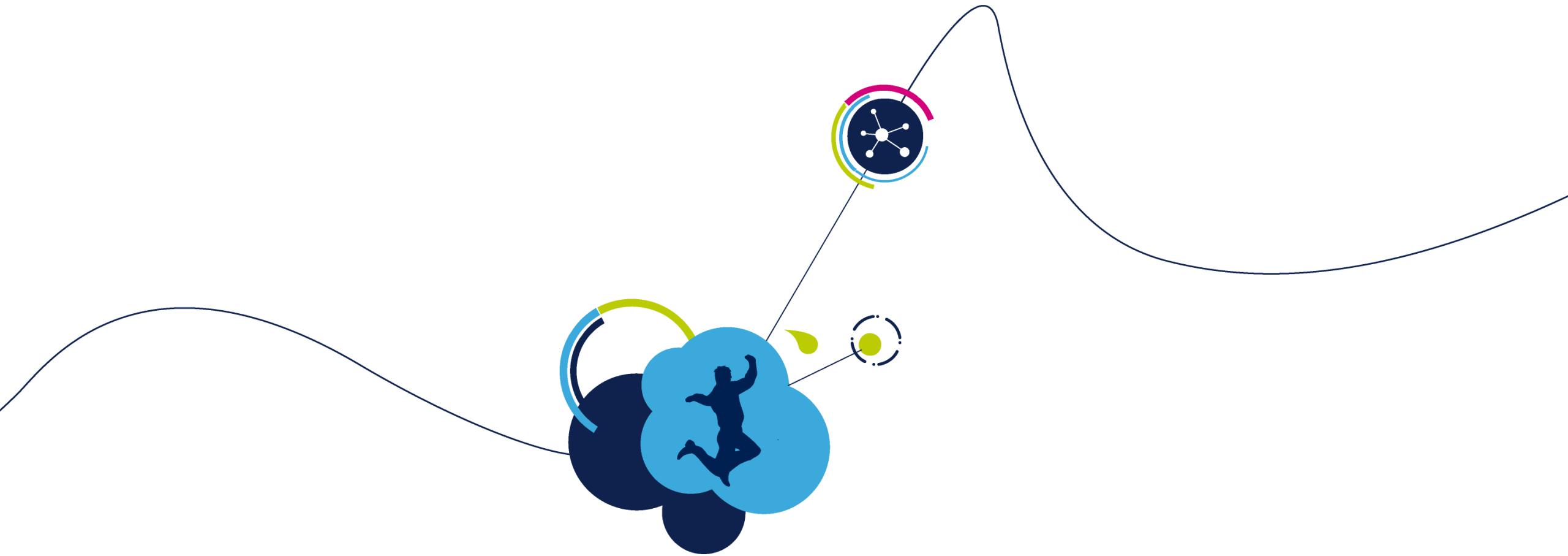
#if GUI_NUM_LAYERS < 2
U32 LCD_Addr[GUI_NUM_LAYERS] = {LCD_LAYER0_FRAME_BUFFER};
#else
U32 LCD_Addr[GUI_NUM_LAYERS] = {LCD_LAYER0_FRAME_BUFFER, LCD_LAYER1_FRAME_BUFFER};
#endif

void LCD_ReqTear(void);
void DSI_IRQHandler(void);
void LCD_SetUpdateRegion(int idx);
```

STemWin_wrapper.c

at contains the configuration of STemWin wrapper.h/c", the hardware initialization in those generated by the GUIBuilder).



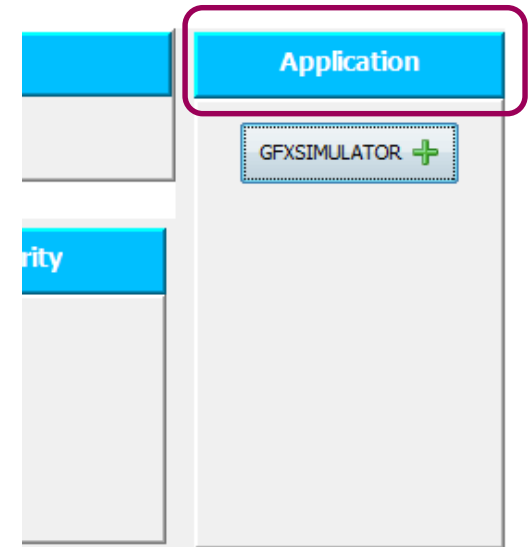


STM32CubeMX Graphics Simulator

STM32CubeMX Graphics Simulator

27

- Fully integrated in STM32CubeMX for F429/F469/F746/F769 .
- New button added “GFXSimulator” on configuration tab
 - Add new vertical section “Application”
- Customer able Now to :
 - Simulate graphics configurations using a set of parameters
 - Evaluate performance of simulation graphic configuration



Current Configuration Benchmark Results Window

28

Simulator Configuration

Import Settings

Current Configuration

Main features

FAHB (MHz)168

Frame buffer format (bit/pixel)16 BPP

SDRAMSDRAM 32

SDRAM (Processing buffers memory location)

SDRAM StateEnable

SDRAM Clock Prescaler2

Row to column delay1

Common row precharge delay1

CAS latency1

Display

Screen Size in pixel per line (x)640

Screen Size in pixel per column (y)480

Graphic object in pixel per line (x)388

Graphic object in pixel per row (y)120

Flash - 128-bit (Graphic object memory location)

Main features

FAHB (MHz)16

Frame buffer format (bit/pixel)16 BPP

SDRAMSDRAM 32

SDRAM (Processing buffers memory location)

SDRAM StateEnable

SDRAM Clock Prescaler2

Row to column delay16

Common row precharge delay16

CAS latency1

Display

Screen Size in pixel per line (x)640

Screen Size in pixel per column (y)480

Graphic object in pixel per line (x)388

Graphic object in pixel per row (y)120

Flash - 128-bit line (Graphic object memory location)

Current Configuration Benchmark Results Window

Result Of Simulator and Current Configuration

Image building using internal Flash and external SDRAM - 16 bpp frame buffer

SDRAM Type32

Frame buffer format (bit/pixel)16

Background drawing with format conversion

Number of cycles (system clock) per pixel3.31

Total processing time (ms)6.06

Number of mega pixels per second (Mpx/s)50.72

Graphic object drawing with transparency and format conversion

Number of cycles (system clock) per pixel5.06

Total processing time (ms)1.4

Number of mega pixels per second (Mpx/s)33.19

Image building and displaying using internal Flash and external SDRAM - 16 bpp frame buffer

SDRAM Type32

Frame buffer format (bit/pixel)16

Background drawing and displaying with format conversion

Image building using internal Flash and external SDRAM - 16 bpp frame buffer

SDRAM Type32

Frame buffer format (bit/pixel)16

Background drawing with format conversion

Number of cycles (system clock) per pixel2.06

Total processing time (ms)39.6

Number of mega pixels per second (Mpx/s)7.76

Graphic object drawing with transparency and format conversion

Number of cycles (system clock) per pixel7.56

Total processing time (ms)22.01

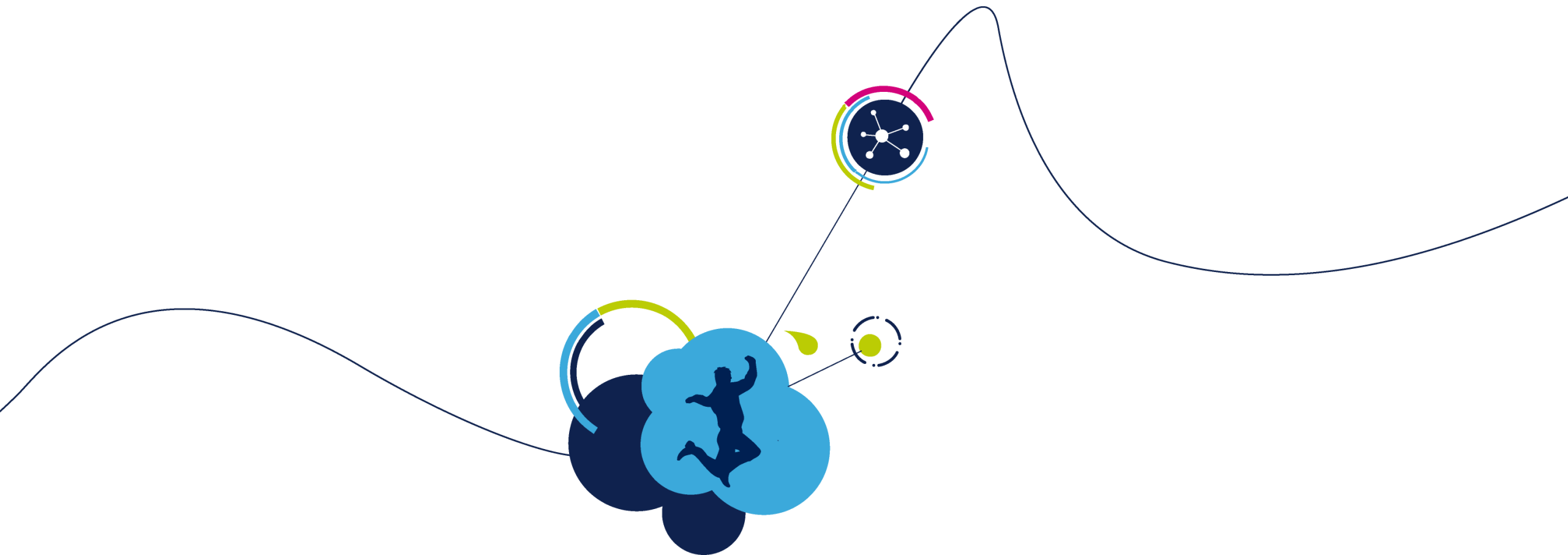
Number of mega pixels per second (Mpx/s)2.12

Image building and displaying using internal Flash and external SDRAM - 16 bpp frame buffer

SDRAM Type32

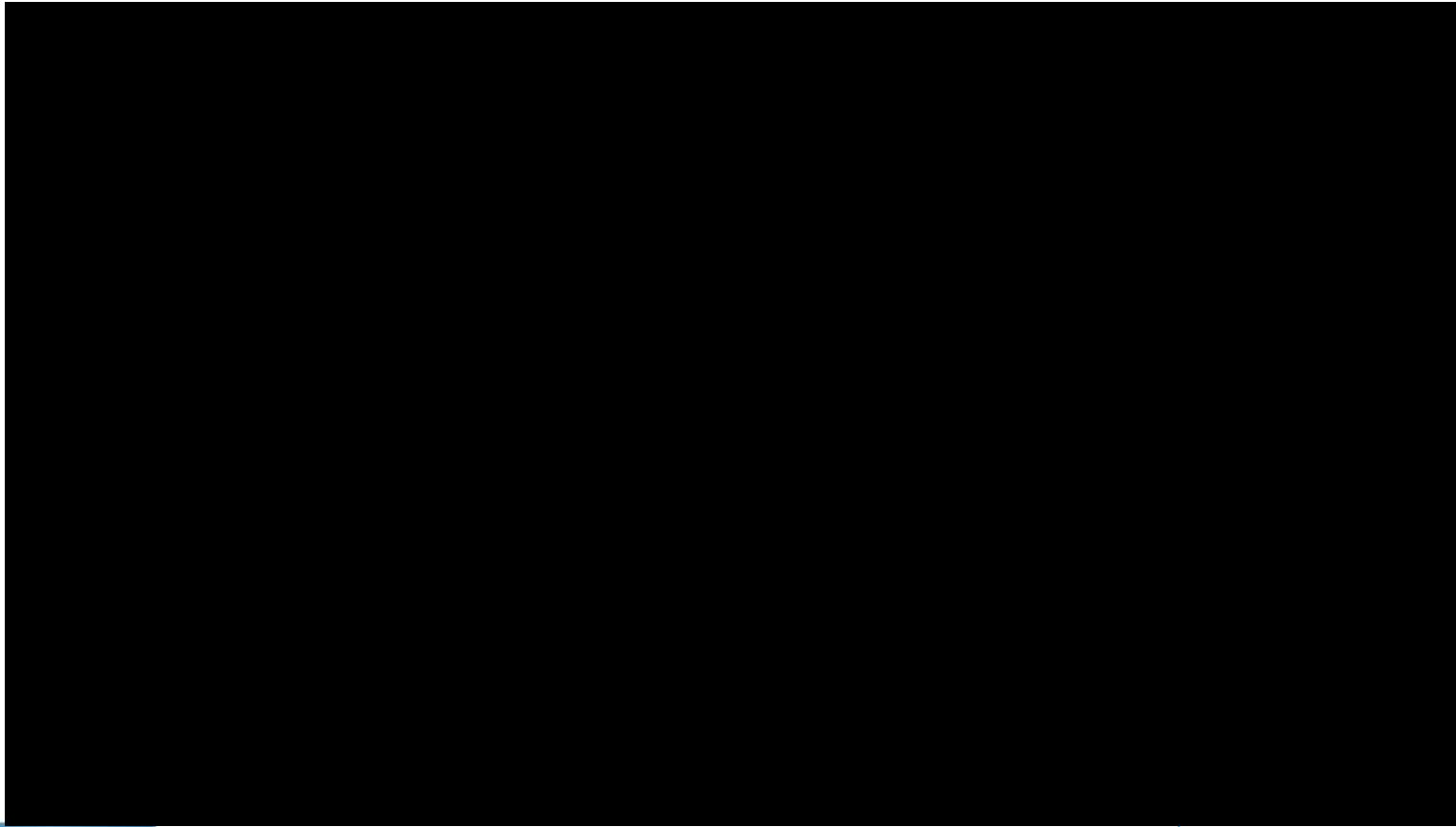
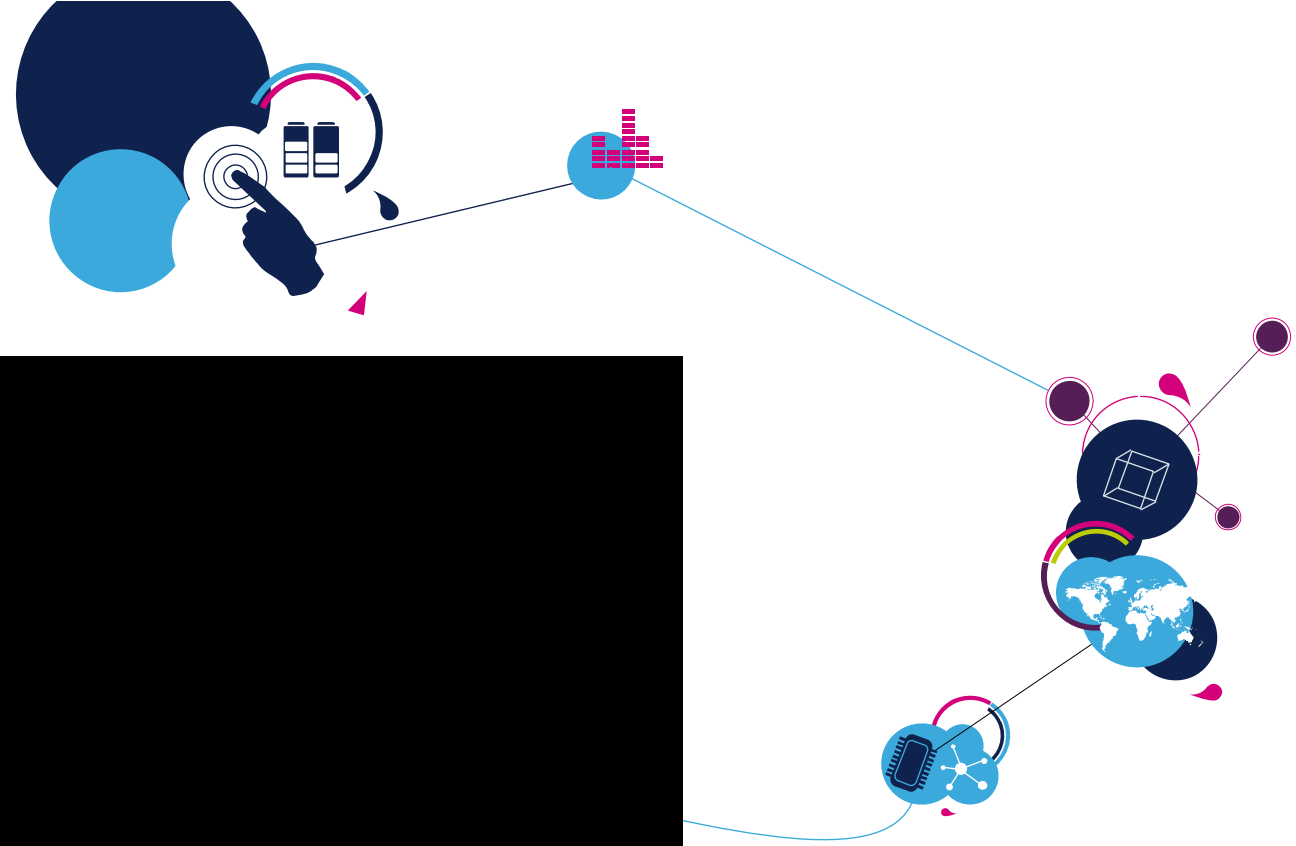
Frame buffer format (bit/pixel)16

Background drawing and displaying with format conversion

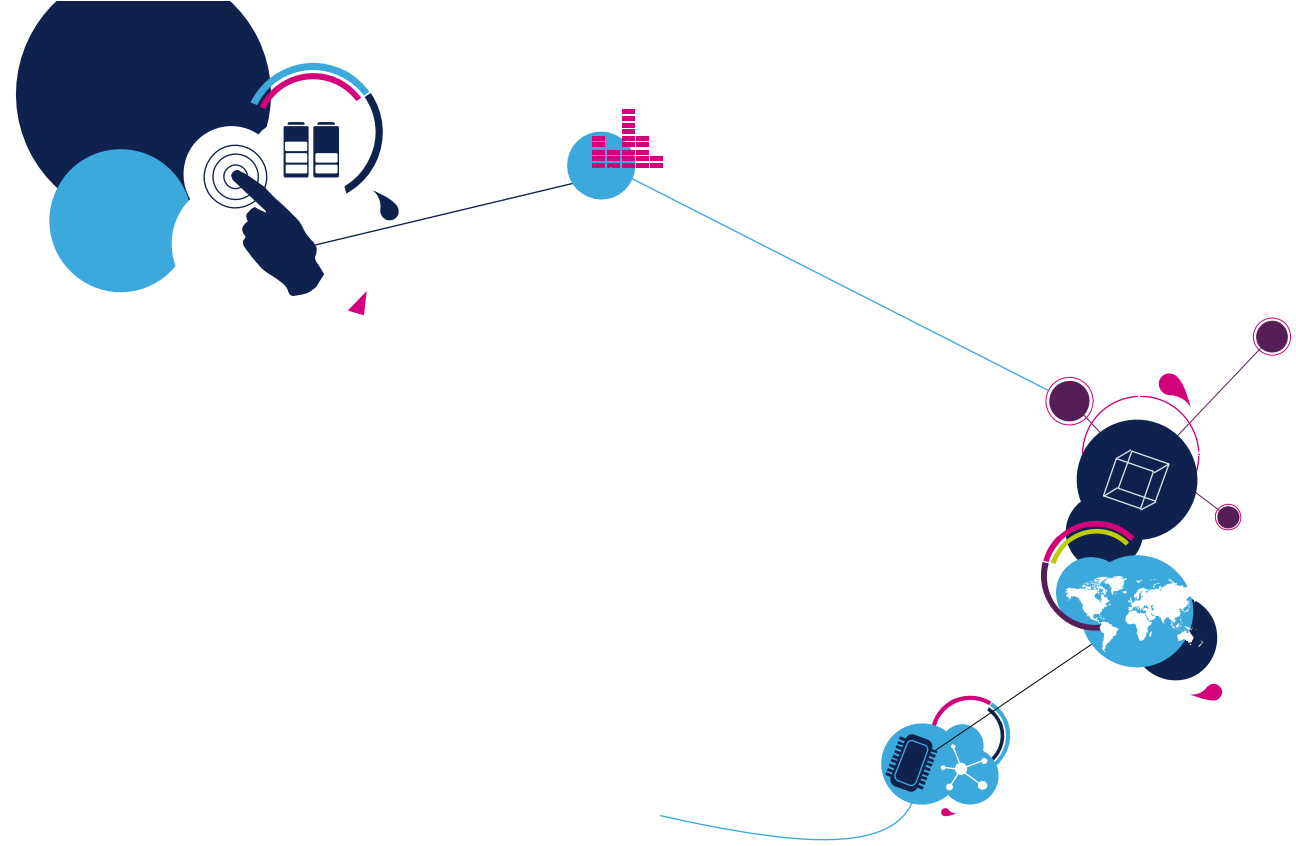


Demo

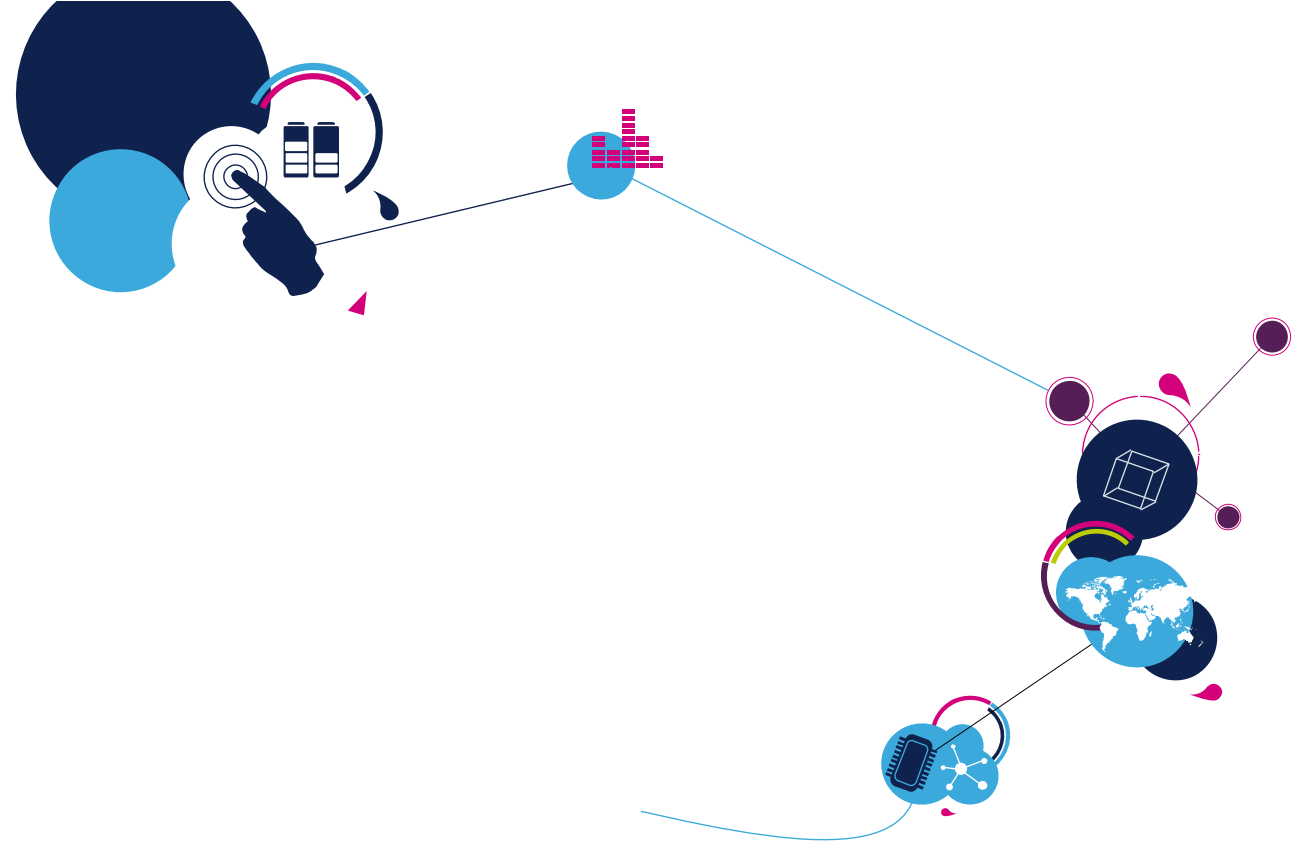
STemWin Graphics Demo



TouchGFX Graphics Demo



Embedded Wizard Graphics Demo



Further examples based on STemWin



- “STemWin Getting started” application note : AN4323
- STemWin user and reference guide:

STM32Cube_FW\Middlewares\ST\STemWin\Documentation\STemWin540.pdf

- STemWin demonstration in STM32Cube MCU firmware package

STM32Cube_FW\Projects\STM32469I-Discovery\Demonstration

- STemWin applications in STM32Cube MCU firmware package

STM32Cube_FW\Projects\STM32469I-Discovery\Applications\STemWin

- Online support <http://my.st.com>