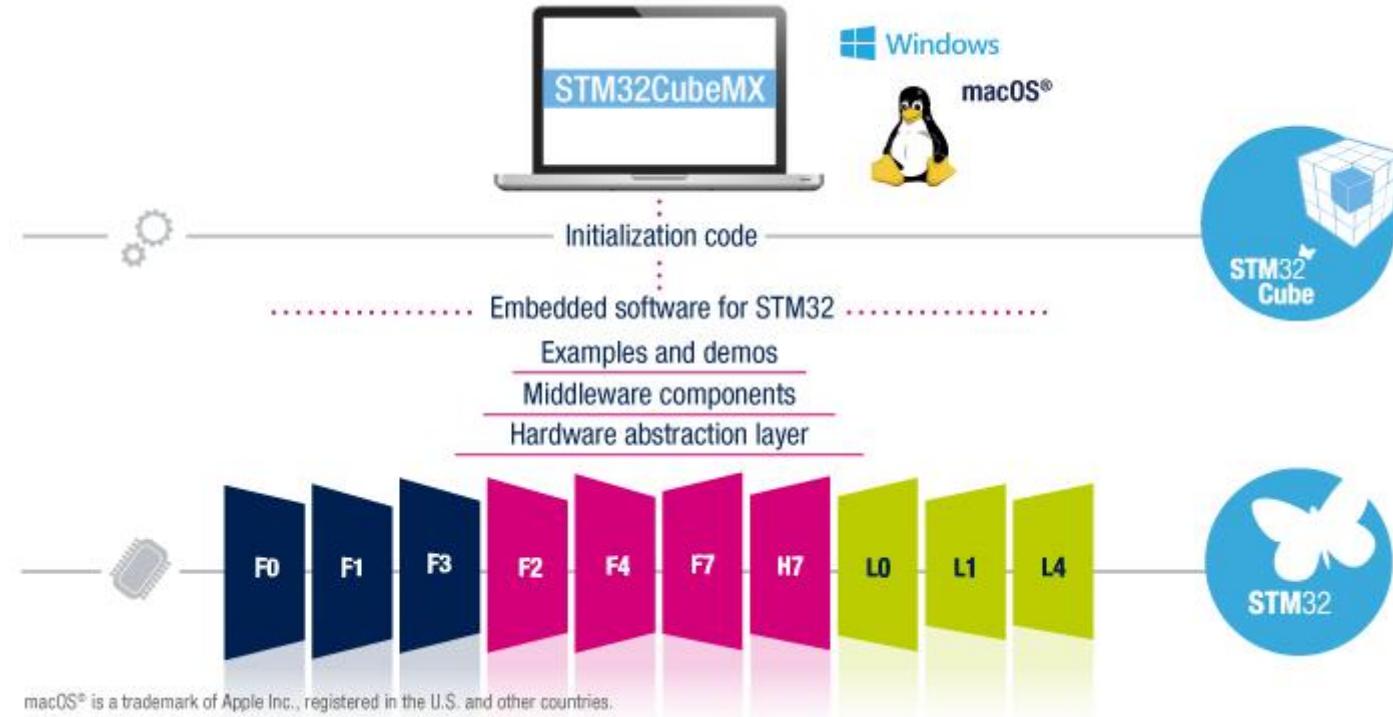


# STM32Cube Overview

# STM32Cube™ Introduction

2

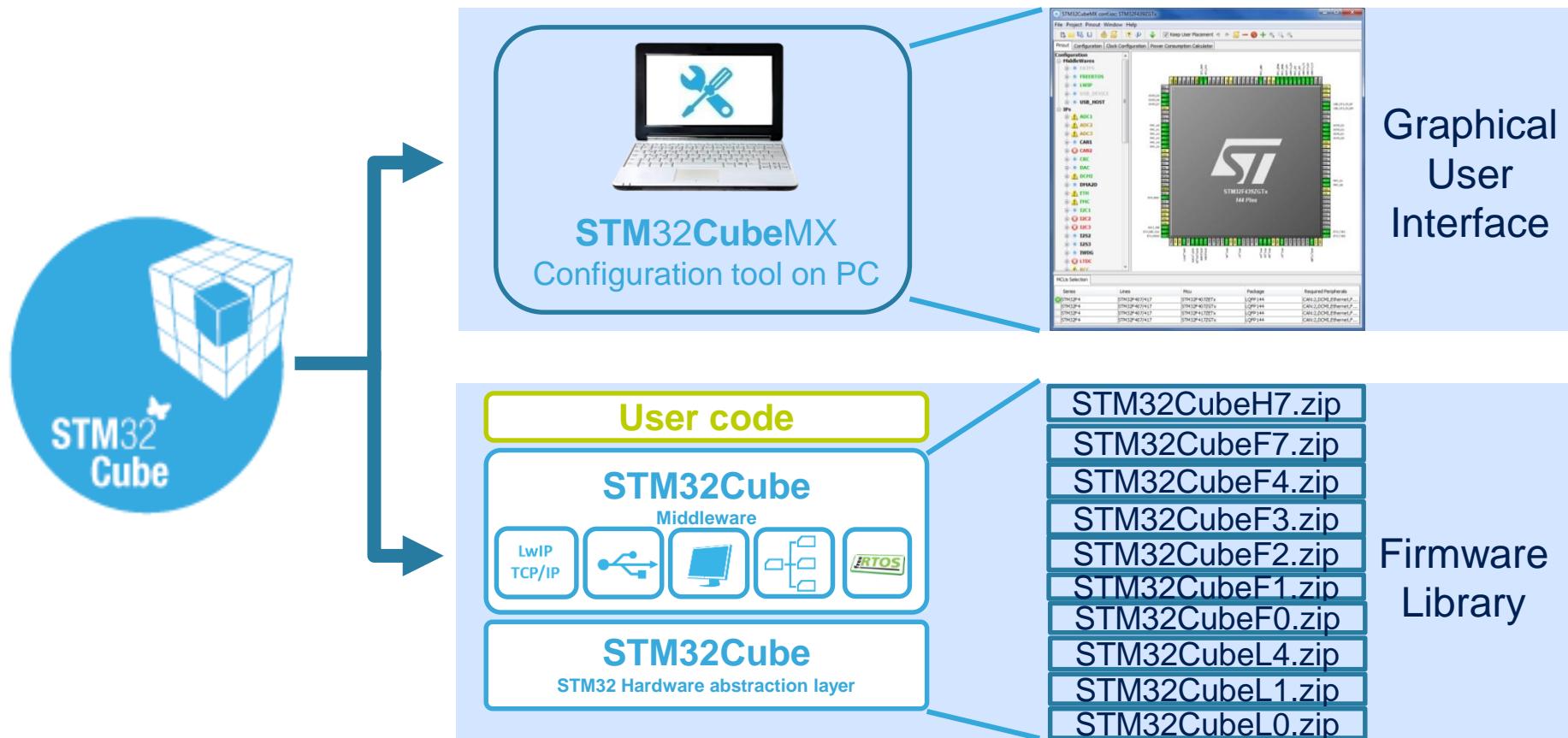
- STMCube™ is a 100% free solution by STMicroelectronics to ease developers life by reducing development efforts, time and cost.



# STM32Cube™ Overview

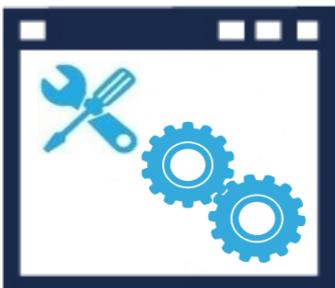
3

- STM32Cube™ is a software development platform that combines
  - A *PC software configuration tool* called **STM32CubeMX**
  - *STM32 embedded software* bricks called **STM32Cube MCU Package**



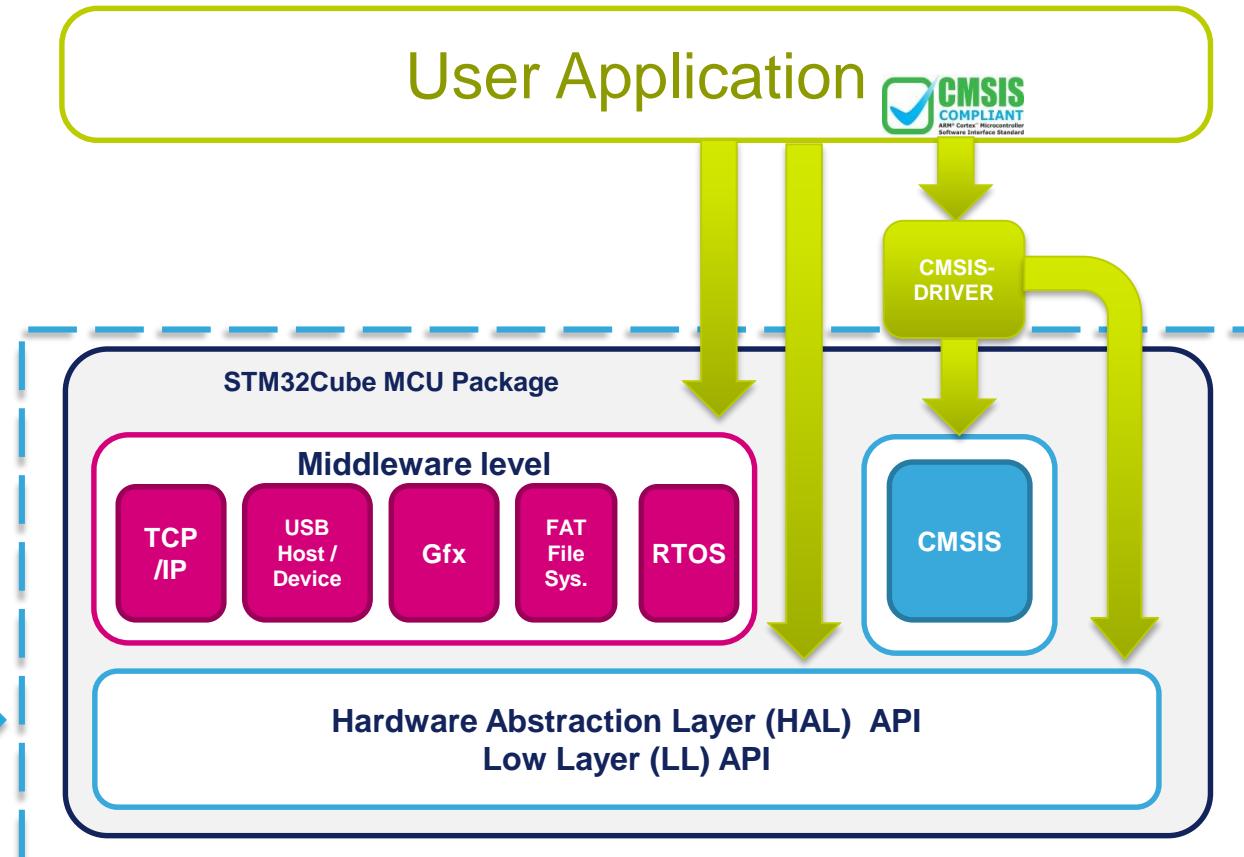
# STM32Cube Work Flow

## STM32CubeMX Configuration tool on PC



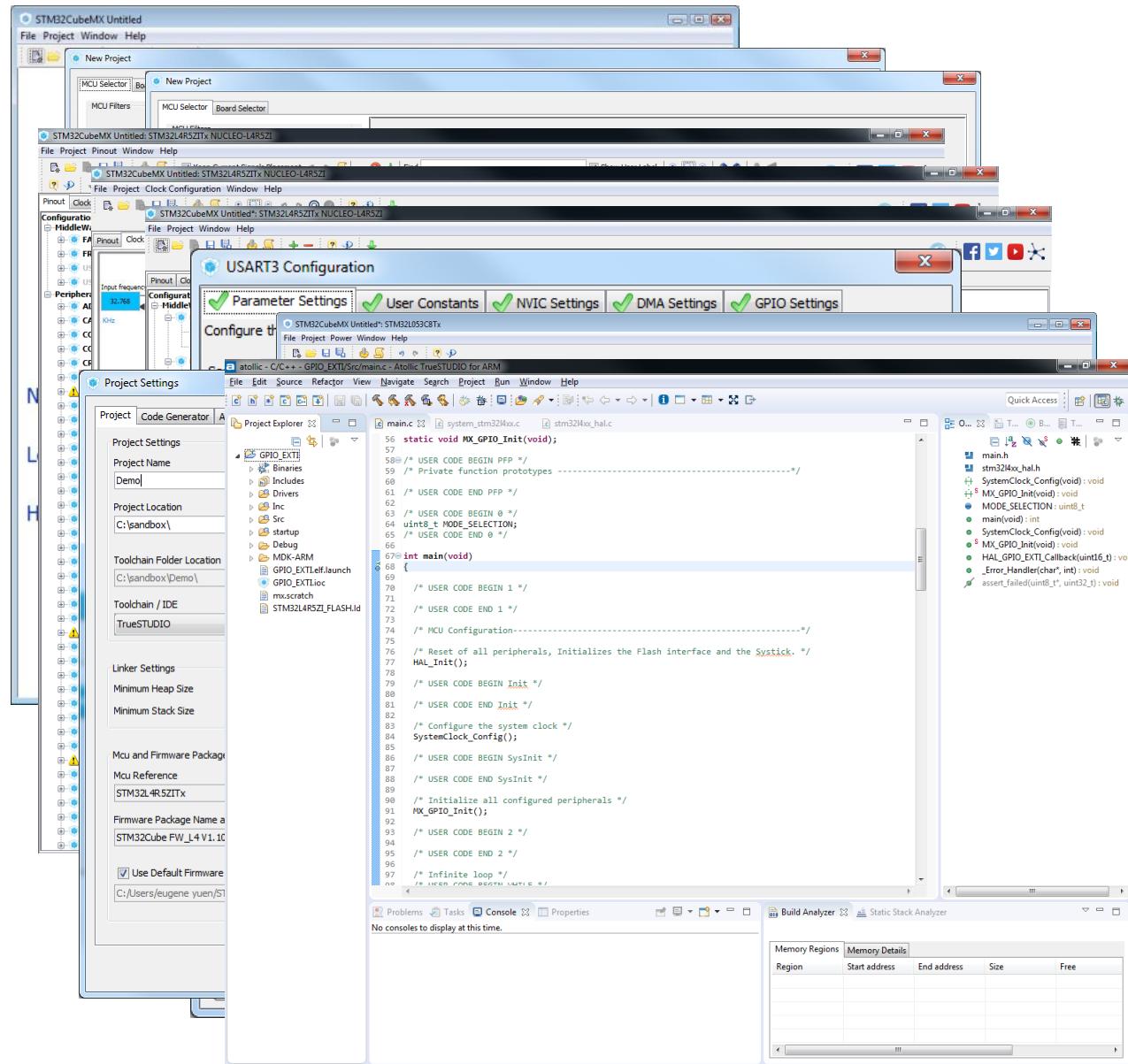
STM32F0    STM32F1    STM32F2  
STM32F3    STM32F4    STM32F7  
STM32L0    STM32L1    STM32L4  
STM32H7

C code generation  
for initialization,  
depending on user  
choices



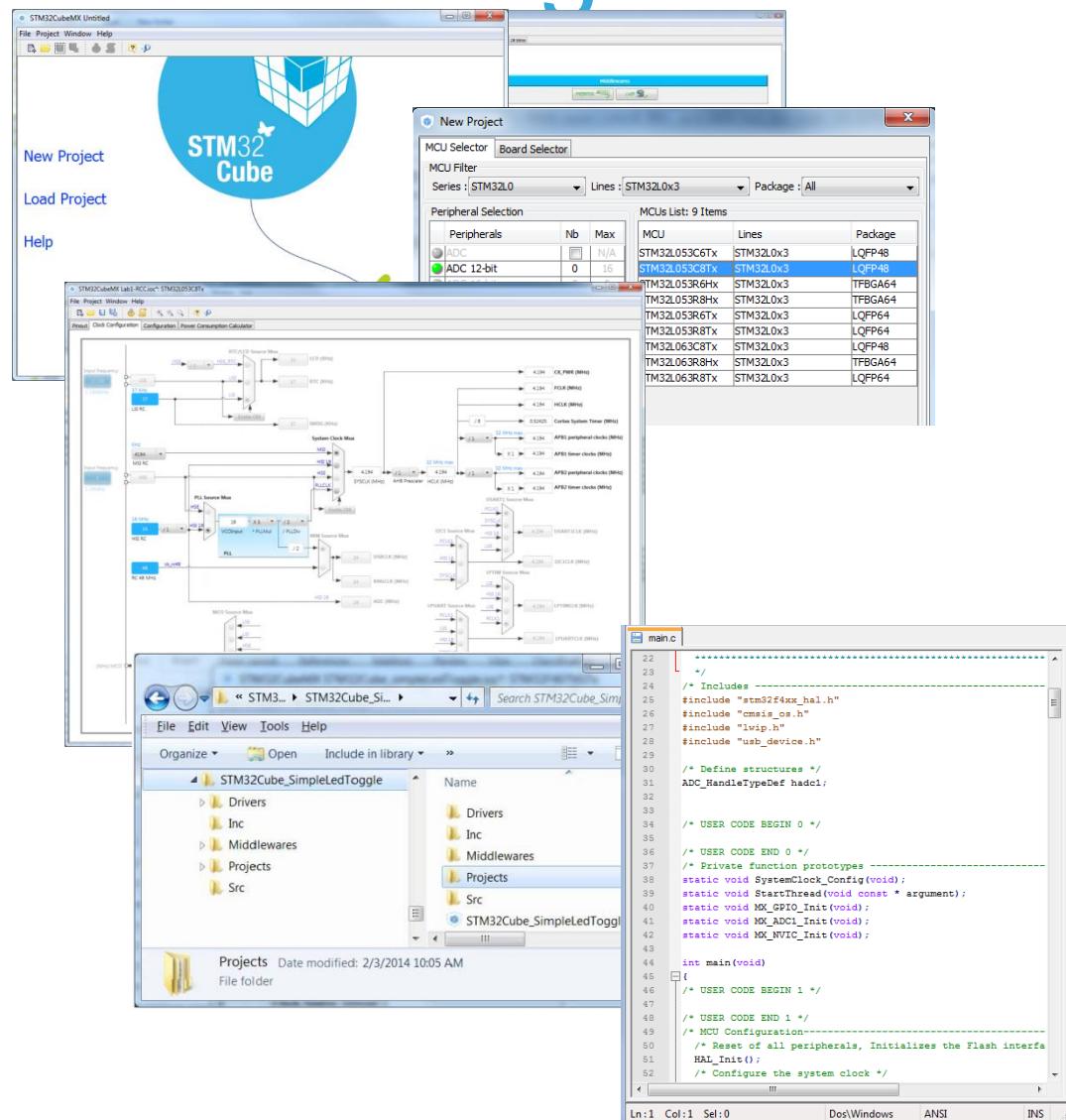
# STM32CubeMX

- STM32CubeMX is a graphical software configuration tool that allows generating C initialization code using graphical wizards.
- Step 1: Select the microcontroller
  - Through easy filtering capabilities
- Step 2: Configure the microcontroller
  - Pin out wizard
  - Clock tree wizard
  - Peripherals and Middleware wizards
  - Power consumption wizard (optional)
- Step 3: Initialization code generation
  - Generates code for your favorite IDE !

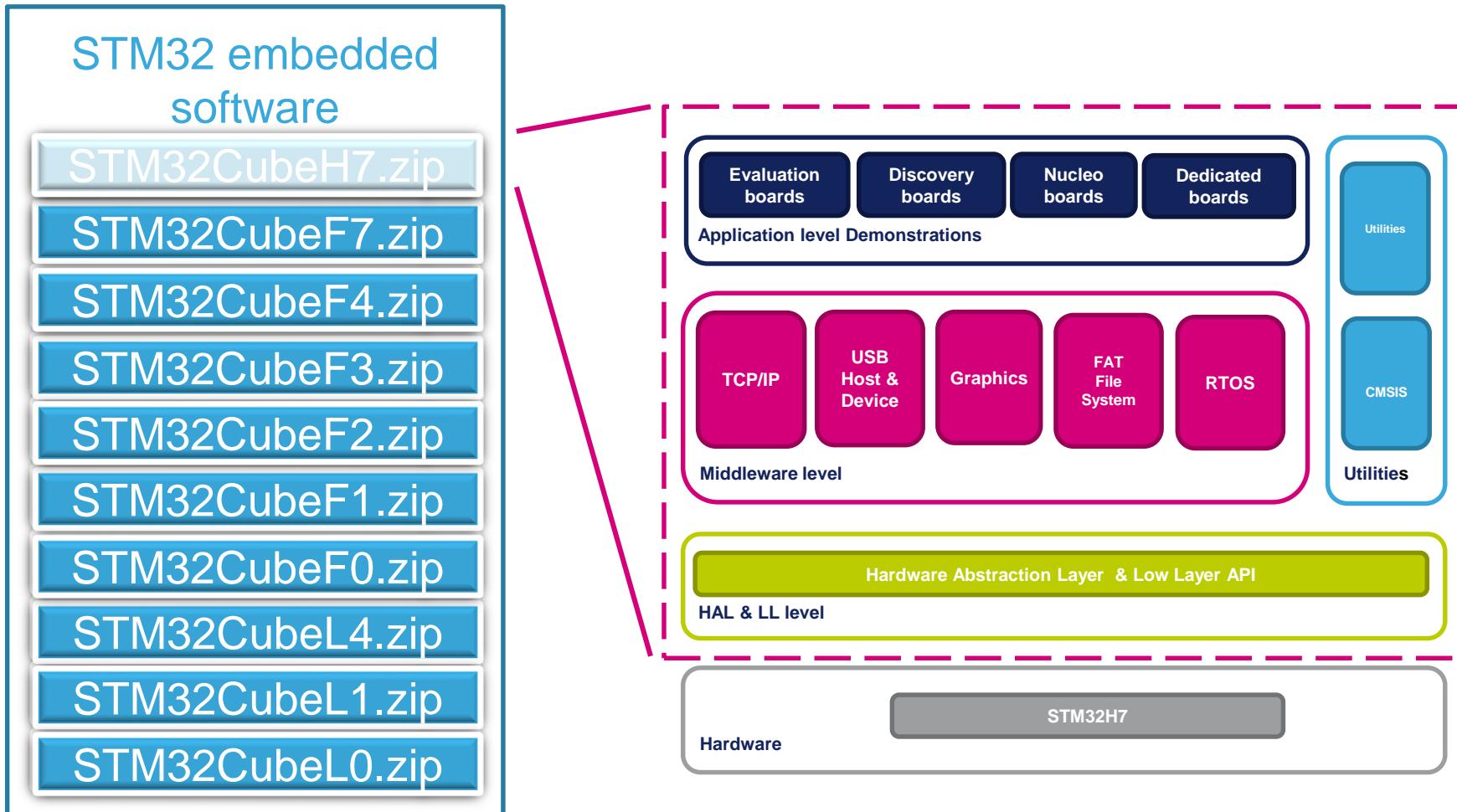


# STM32CubeMX for STM32 configuration and initialization C code generation

- The STM32CubeMX, a graphical software configuration tool that allows to generate C initialization C code using graphical wizards.
- STM32CubeMX has the following key features:
  - Easy microcontroller selection covering whole STM32 portfolio.
  - Board selection from a list of STMicroelectronics boards.
  - Easy microcontroller configuration (pins, clock tree, peripherals, middleware) and generation of the corresponding initialization C code.
  - Generation of configuration reports.
  - Generation of IDE ready projects for a selection of integrated development environment tool chains.
  - STM32CubeMX projects include the generated initialization C code, STM32 HAL drivers, the middleware stacks required for the user configuration, and all the relevant files needed to open and build the project in the selected IDE.
  - Power consumption calculation for a user-defined application sequence.
  - Self-updates allowing the user to keep the STM32CubeMX up-to-date.
  - Downloading and updating STM32Cube™ firmware packages allowing the download from [www.st.com](http://www.st.com) of the MCU firmware package required for the development of the user application



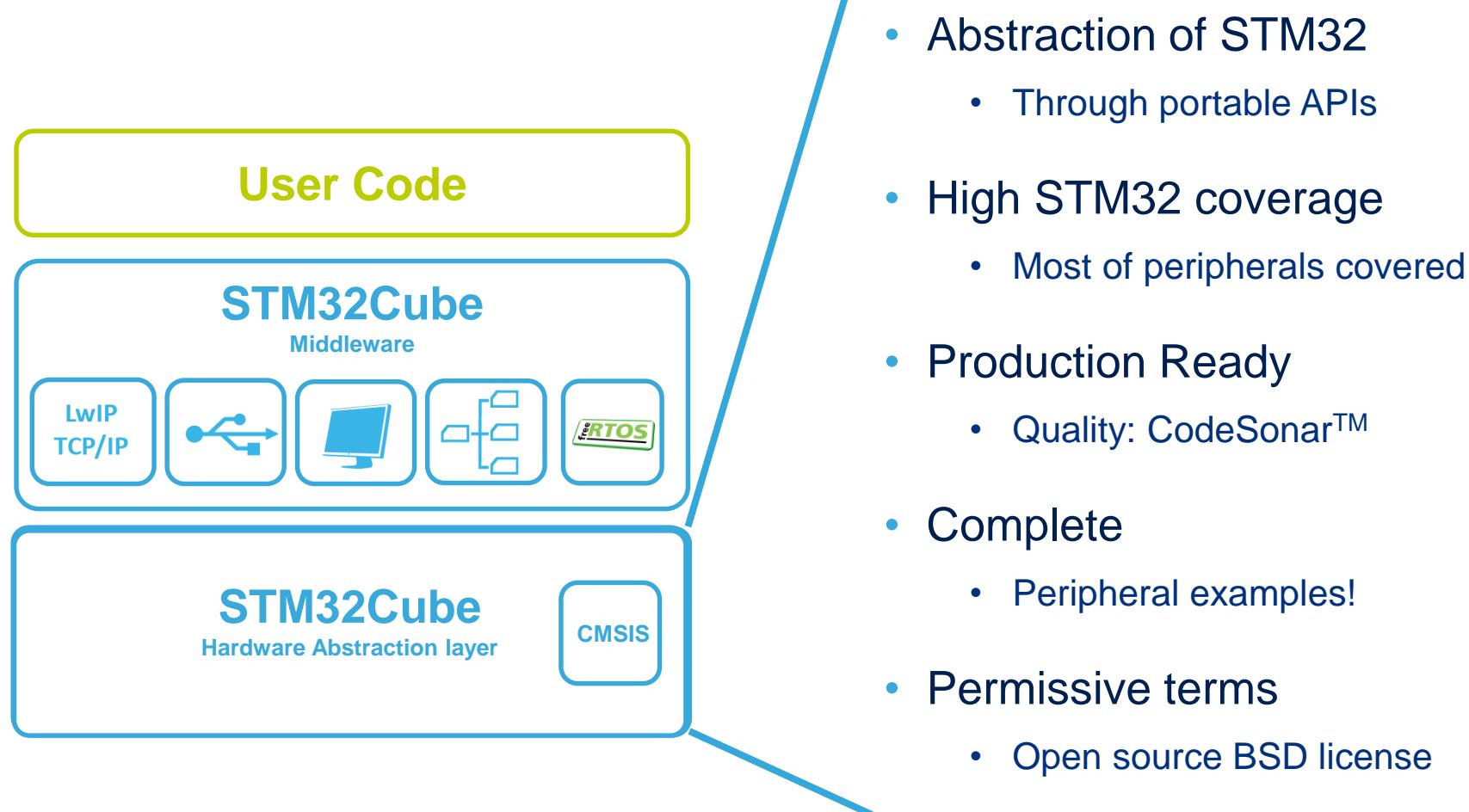
# STM32Cube MCU Package



Downloadable manually from [www.st.com/stm32cube](http://www.st.com/stm32cube)  
or via **STM32CubeMX** download libraries menu

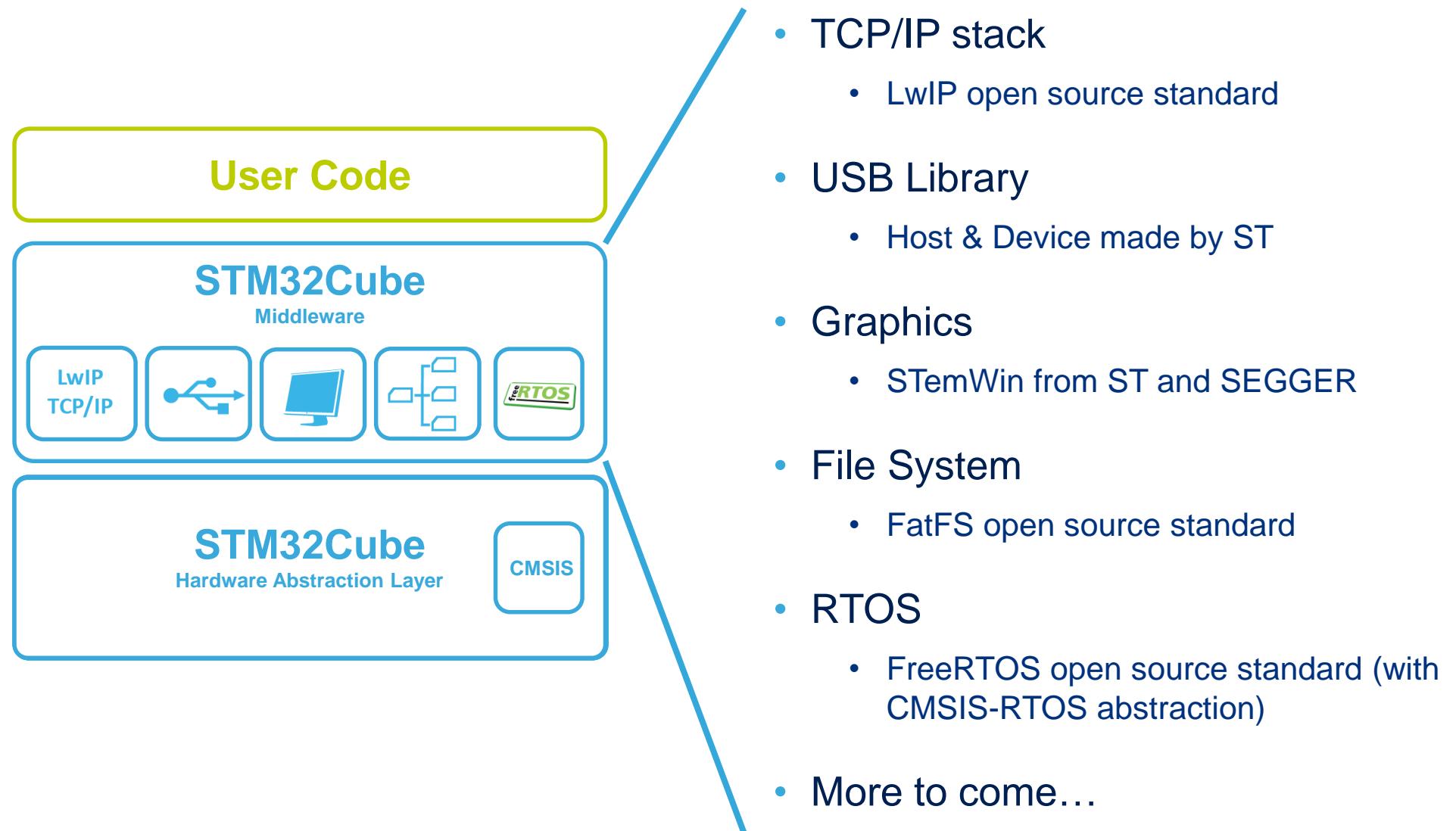
# STM32Cube V1

## STM32Cube MCU Package



# STM32Cube V1

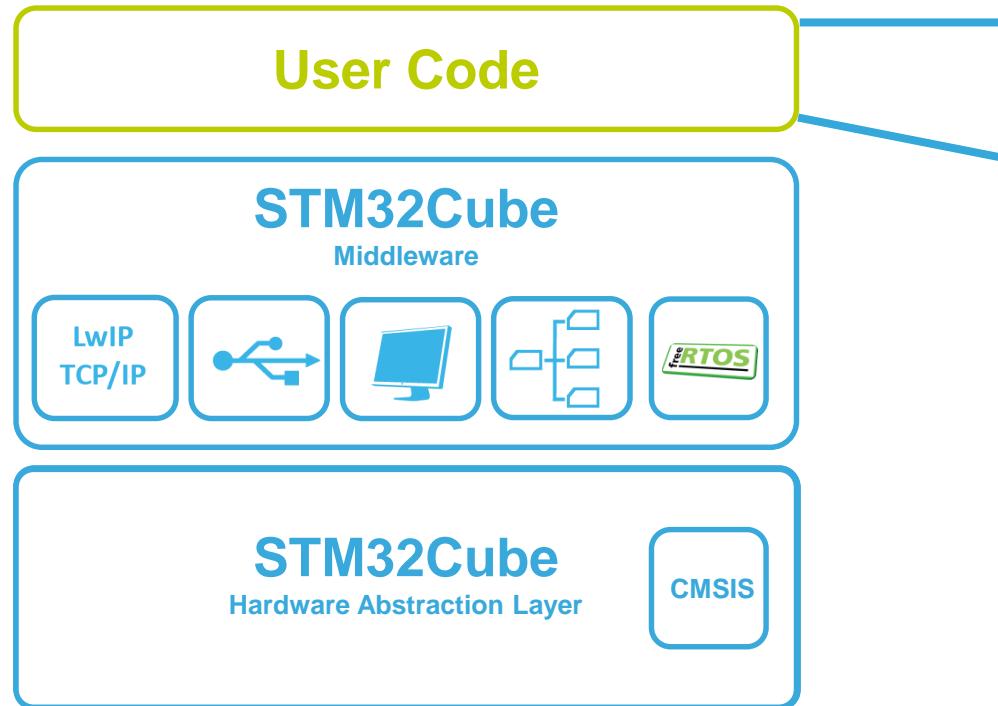
## STM32Cube MCU Package



# STM32Cube V1

## STM32Cube MCU Package

10



- Customer focusing on his application differentiation

# STM32CubeTM V1 – Key Benefits

- Simplify and Speed up Application Development for STM32!

- Through STM32CubeMX:

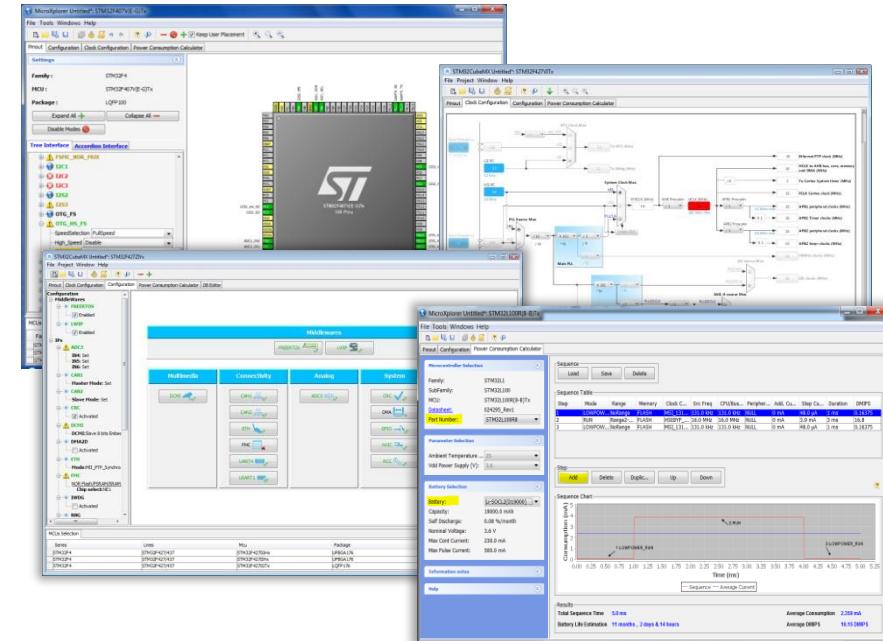
- MCU Selector
- Graphical Peripheral Configuration
- Power Consumption Wizard
- Peripheral Initialization Code Generation
- Automatic update feature

Ensuring the developer is aware of new versions and fixes, as well as new components

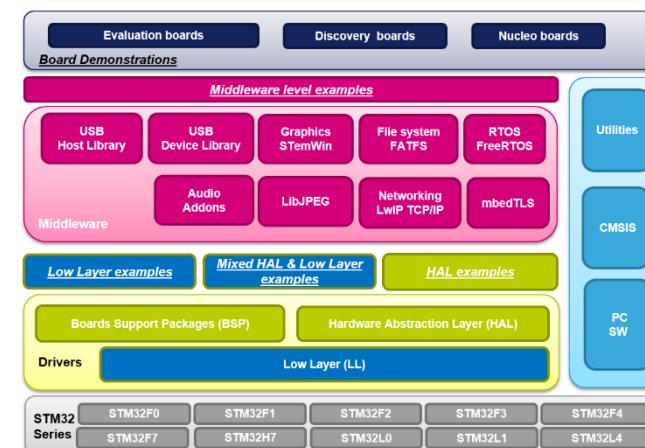
- Through extensive set of “ready-to-run” peripheral examples and application examples, with ready project files for IAR, Keil and GCC included in the STM32Cube MCU firmware packages

- More than Cost-friendly !

- 100% FREE** embedded software!
- 100% FREE** software tool !
- ST-branded, ST-supported !
- Users gain time with initialization code generation, and remain focused on their key application code



## User Application

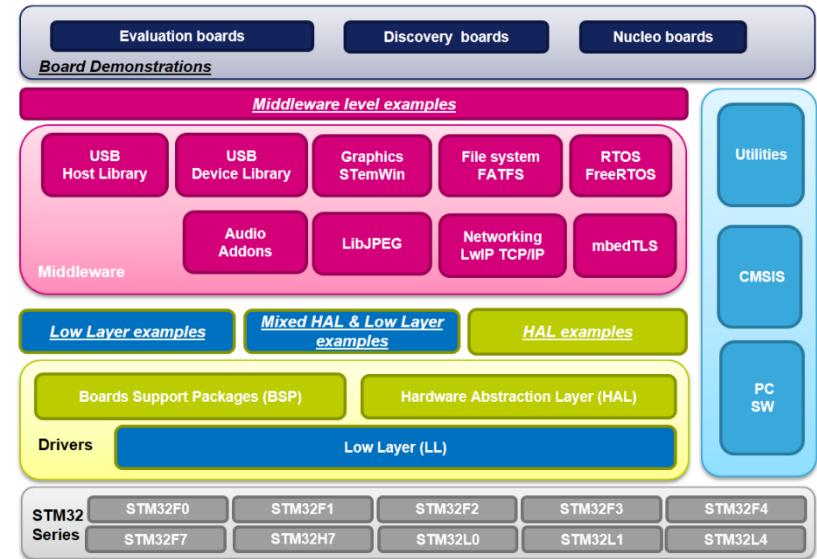




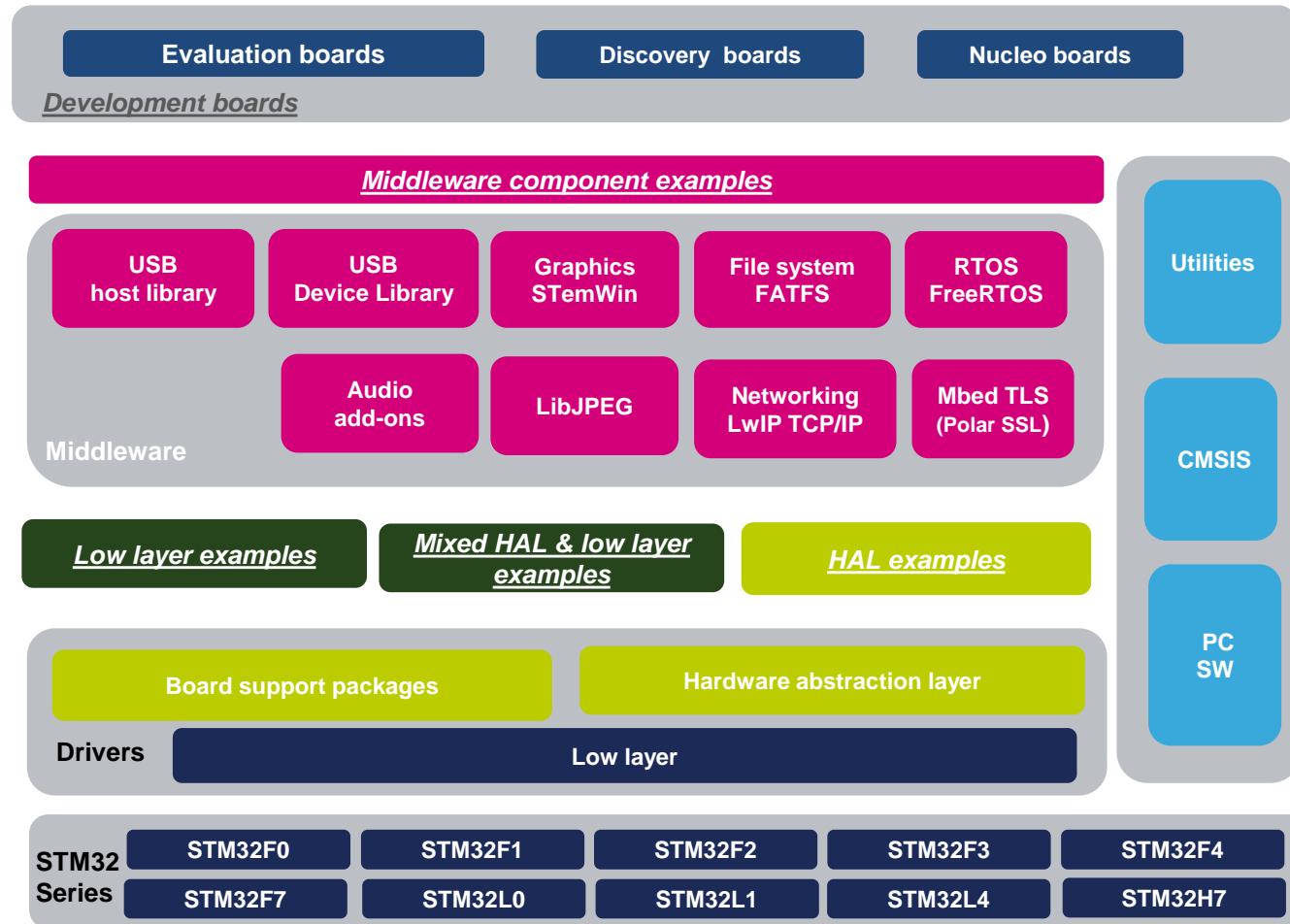
# STM32Cube MCU Package

# STM32Cube MCU Package Features

- STM32Cube gathers together, in a single package, all the generic and highly portable embedded software components required to develop an application on STM32 microcontrollers.
- The package includes a low level hardware abstraction layer (HAL) that covers the microcontroller hardware, together with an extensive set of examples running on STMicroelectronics boards.
- It also contains a set of middleware components (\*) with the corresponding examples. They come with very permissive license terms:
  - Full USB Host and Device stack supporting many classes.
    - Host Classes: HID, MSC, CDC, Audio, MTP
    - Device Classes: HID, MSC, CDC, Audio, DFU
  - Graphics
    - STemWin, a professional graphical stack solution available in binary format and based on the emWin solution from ST's partner SEGGER
    - LibJPEG, an open source implementation on STM32 for JPEG images encoding and decoding.
  - CMSIS-RTOS implementation with FreeRTOS open source solution
  - FAT File system based on open source FatFS solution
  - TCP/IP stack based on open source LwIP solution
  - SSL/TLS secure layer based on open source mbed TLS (formerly PolarSSL)
- A demonstration implementing all these middleware components is also provided

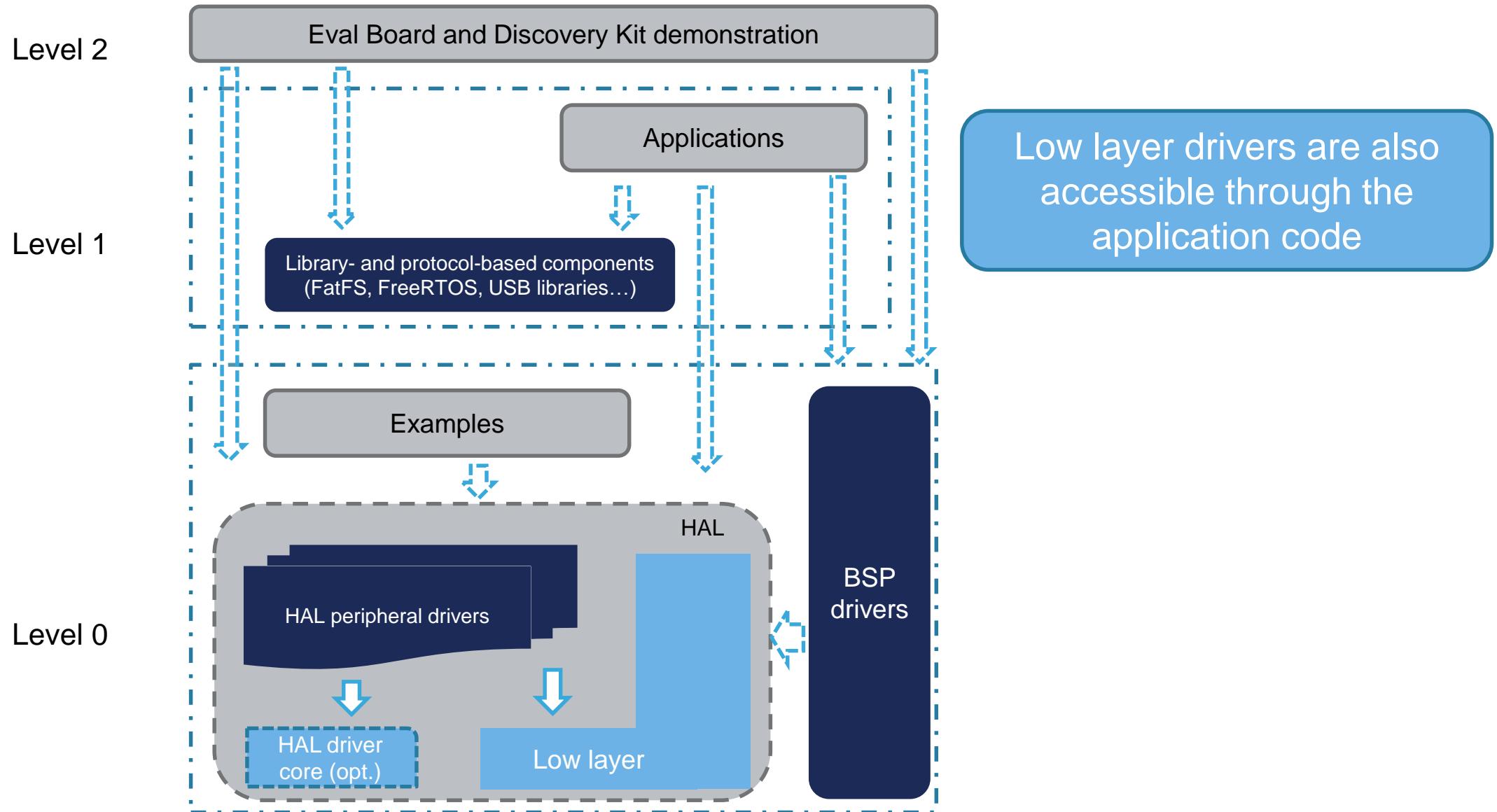


# STM32Cube MCU package

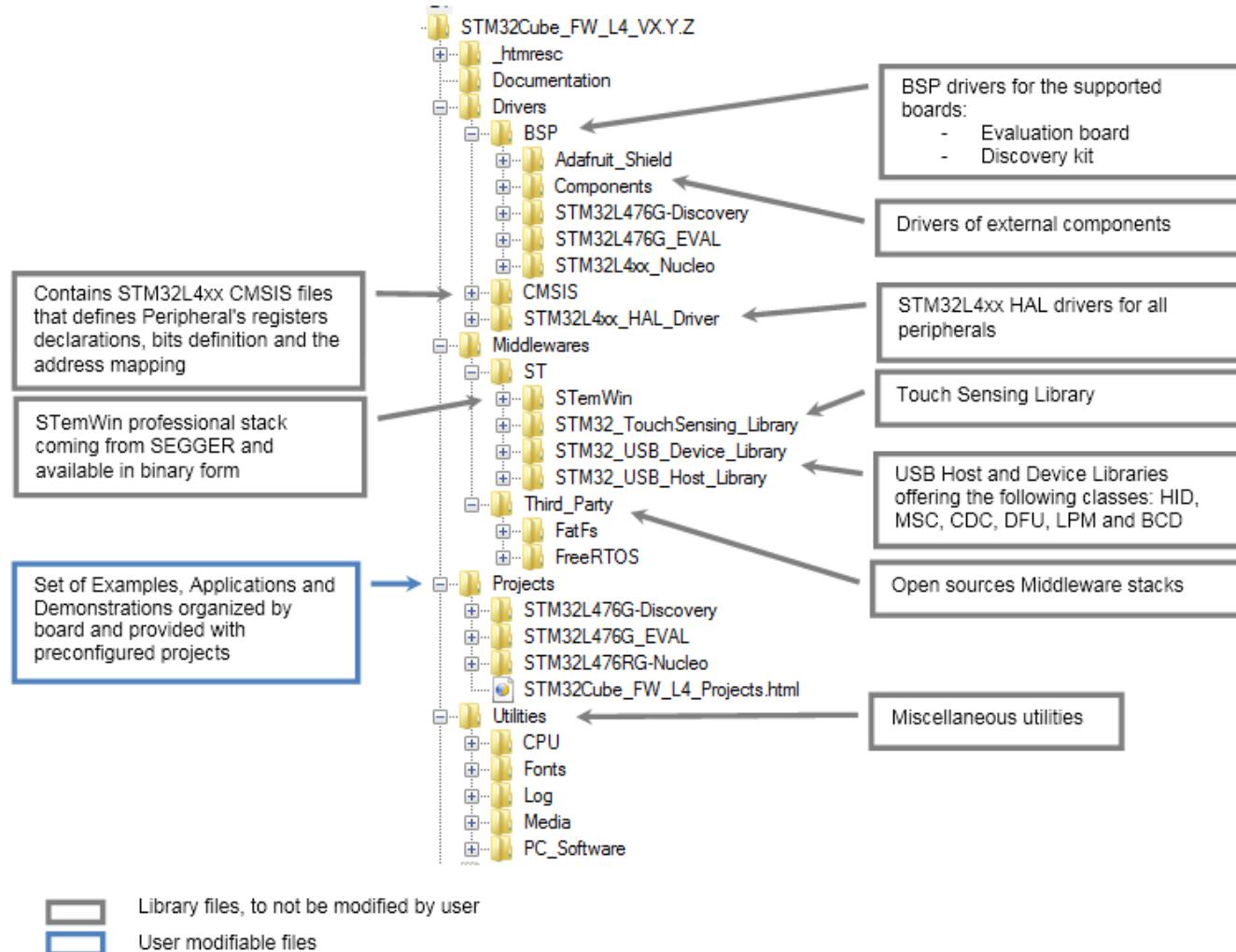


# Architecture with LL

15



# Firmware package structure



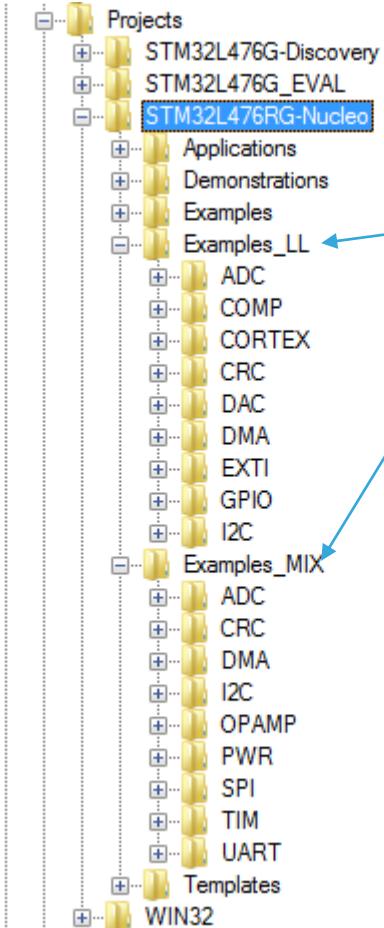
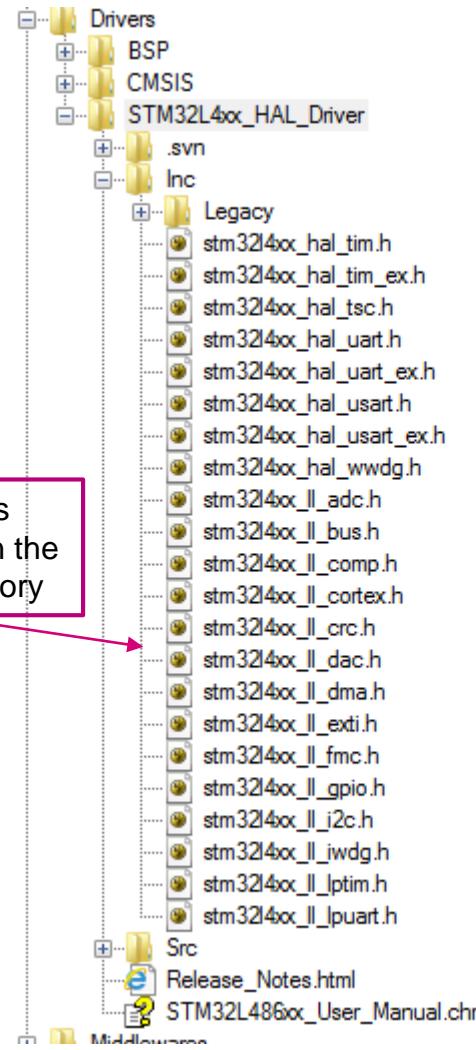
Library files, to not be modified by user



User modifiable files

# Firmware package structure with low layer

17



Examples based on LL are proposed in Nucleo board projects

Two new sets of examples available in Examples\_LL and Examples\_MIX

Examples\_LL contains projects relying only on LL function calls

Examples\_MIX contains projects relying only both on HAL and LL function calls

# STM32CubeL4 documentation

18

- The STM32Cube documentation vary from one STM32 series to another.
- More high-end MCUs(e.g. STM32F4) are supported by more middleware libraries, like for STemWin graphics library and LwIP TCP/IP Stack.

## Technical Documentation

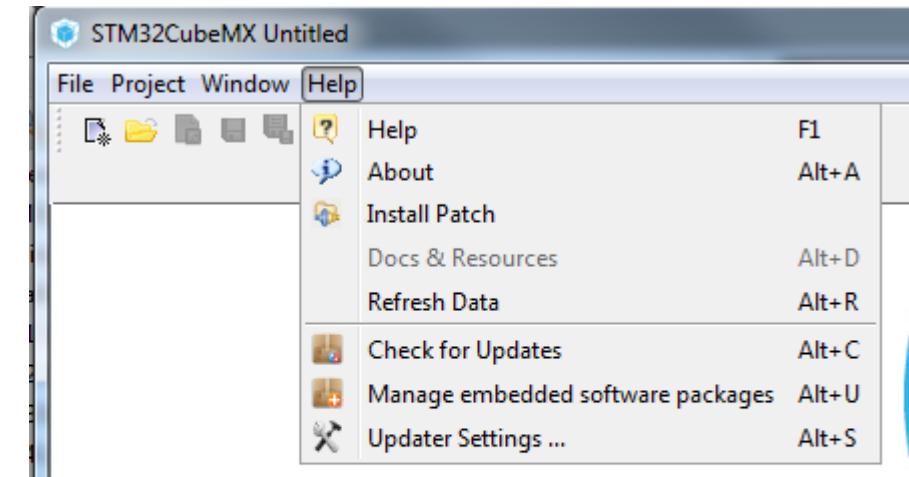
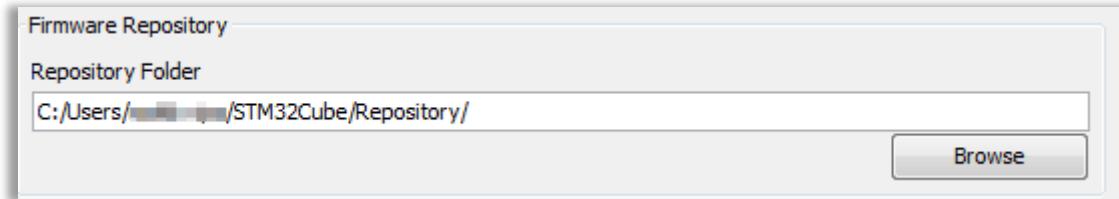
Product Specifications			
Description	Version	Size	
 DB2602: STM32Cube MCU Package for STM32L4 Series and STM32L4+ Series with HAL, low-layer drivers and dedicated middleware	4.0	149 KB	
Application Notes			
Description	Version	Size	
 AN4726: STM32Cube firmware examples for STM32L4 Series and STM32L4+ Series	11.0	493 KB	
User Manuals			
Description	Version	Size	
 UM1884: Description of STM32L4/L4+ HAL and Low-layer drivers	7.0	49 MB	
 UM1721: Developing Applications on STM32Cube with FatFs	2.5	516 KB	
 UM1722: Developing Applications on STM32Cube with RTOS	2.5	710 KB	
 UM1913: Developing applications on STM32Cube with STMTouch® touch sensing library	5.0	2 MB	



# STM32CubeMX

# Prerequisites and settings

- STM32CubeMX needs Java RE
  - Check release notes of the particular version for additional requirements
- After installation, hit Alt+S to configure the updater – not only for the GUI but also for Cube firmware libraries.
- Select the software library placement.

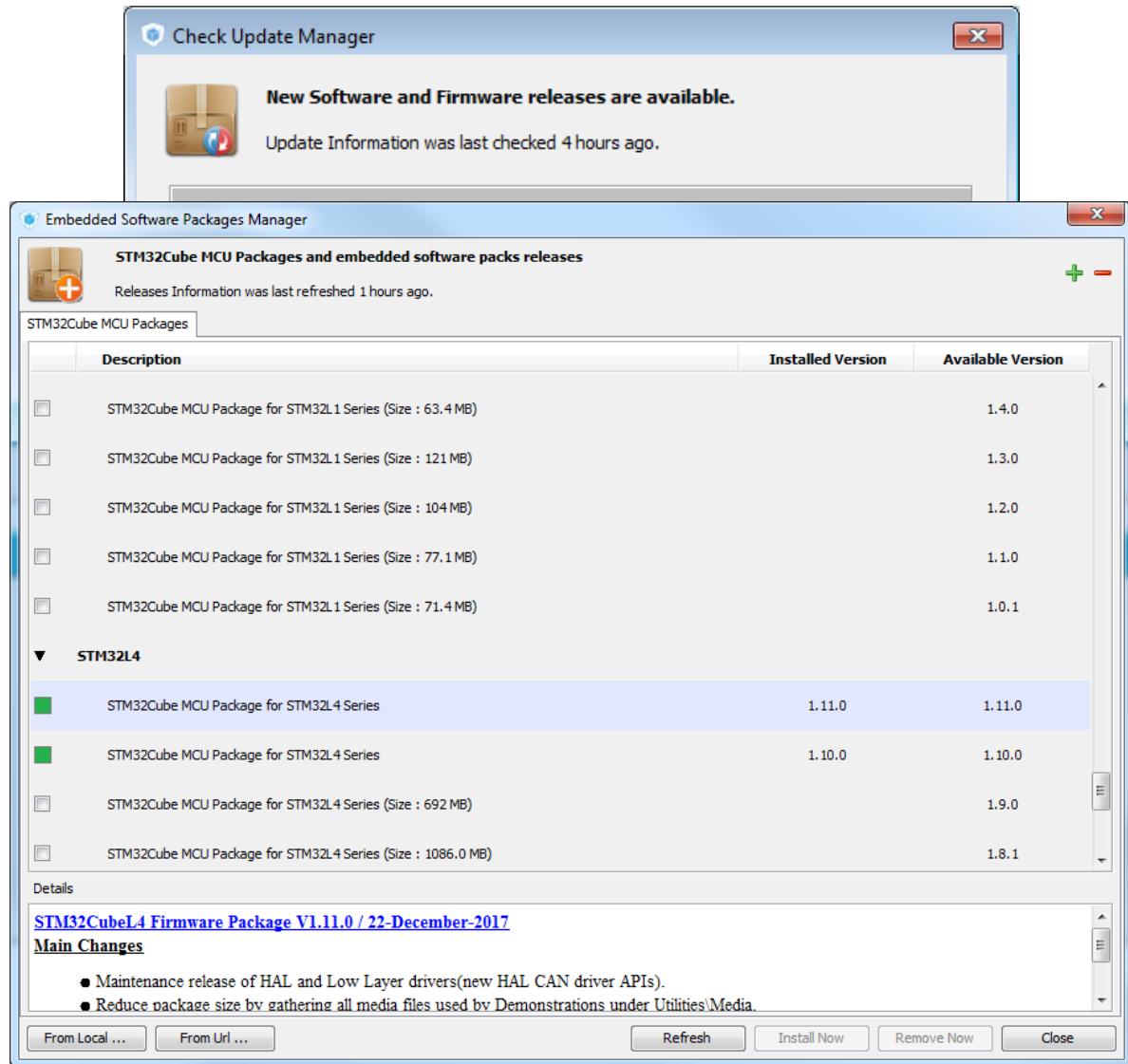


New Project

# Updater

21

- Updates are accessible from the Help menu
- The tool updater can detect new releases of the tool and the associated STM32Cube library.
- Use “Manage embedded software packages” to download new library packages.

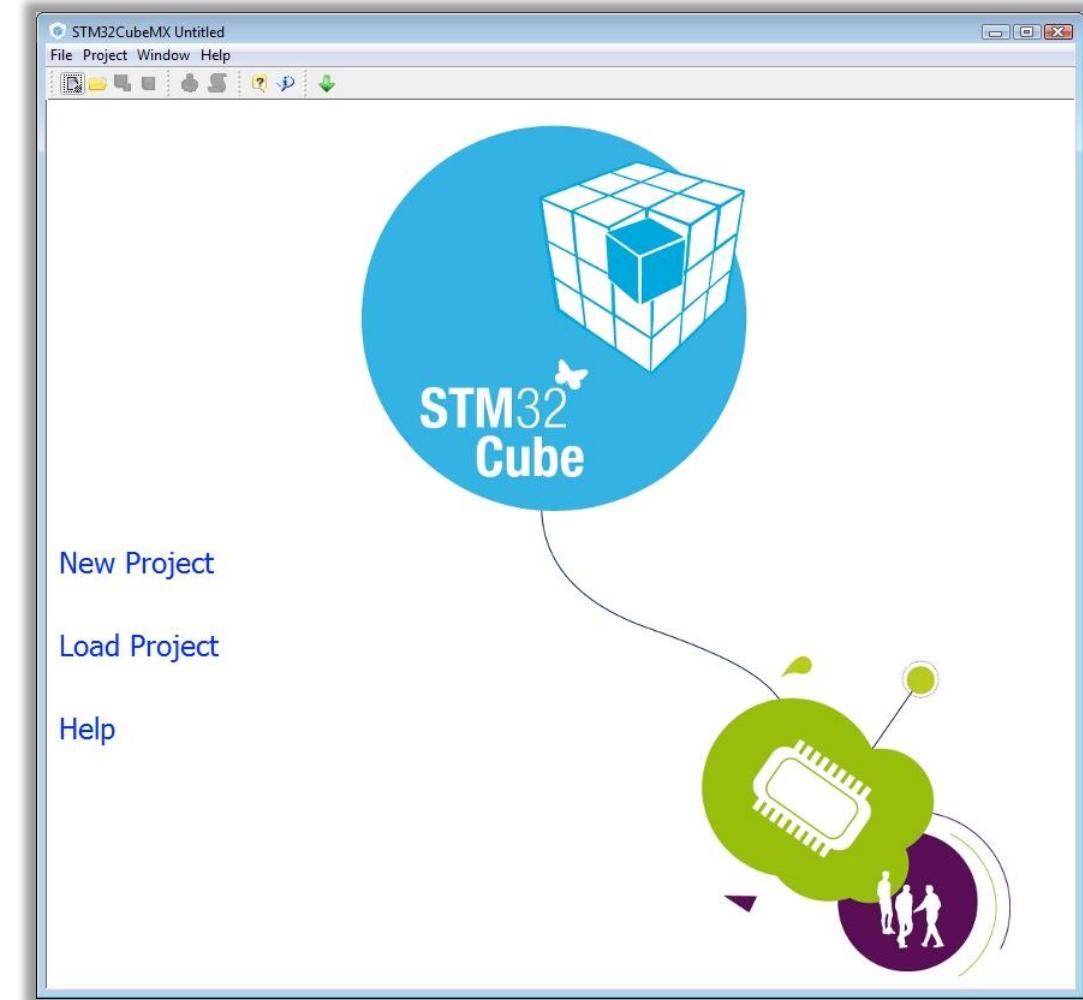


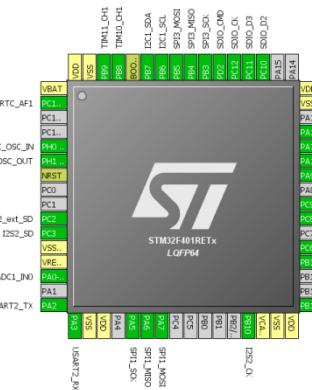
# STM32Cube: STM32CubeMX

22

Step by step:

- MCU selector
- Pinout configuration
- Clock tree initialization
- Peripherals and middleware parameters
- Code generation
- Power consumption calculator





### Pinout Wizard

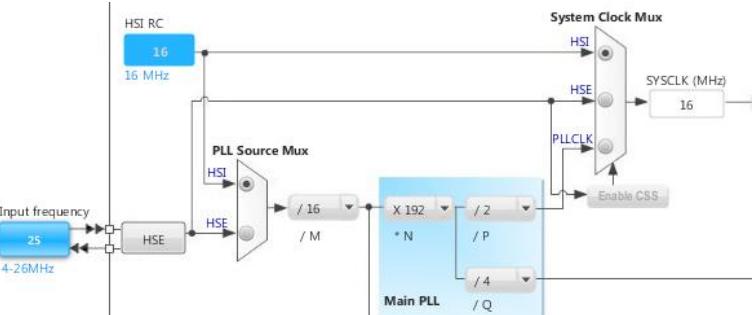
# STM3CubeMX



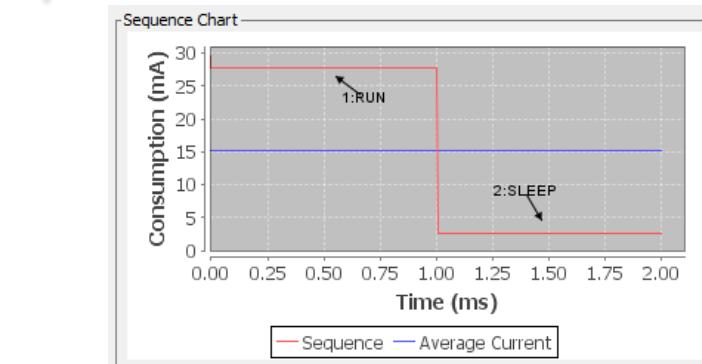
Basic Parameters	
Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples
Baud Rate	
BaudRate must be between <b>110 Bits/s</b> and <b>10.5 MBits/s</b> .	

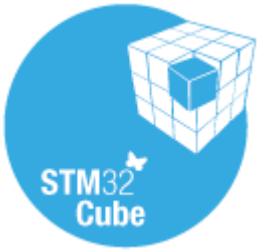
### Peripherals & Middleware Wizard

### Clock Tree wizard

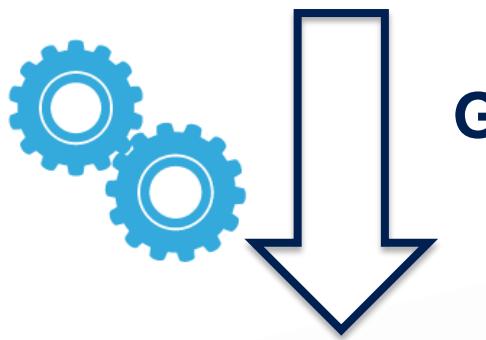


### Power Consumption Wizard





## STM32CubeMX



**Generates Initialization C Code  
based on user choices !**

# MCU Selector

25

- Several hundred products in database
- Find MCU by name ...
  - Quickly locate by Series and Lines
- ... or application needs
  - Package (pin count)
  - RAM size
  - NV memory requirements
  - Embedded peripherals
  - Number and type of interfaces

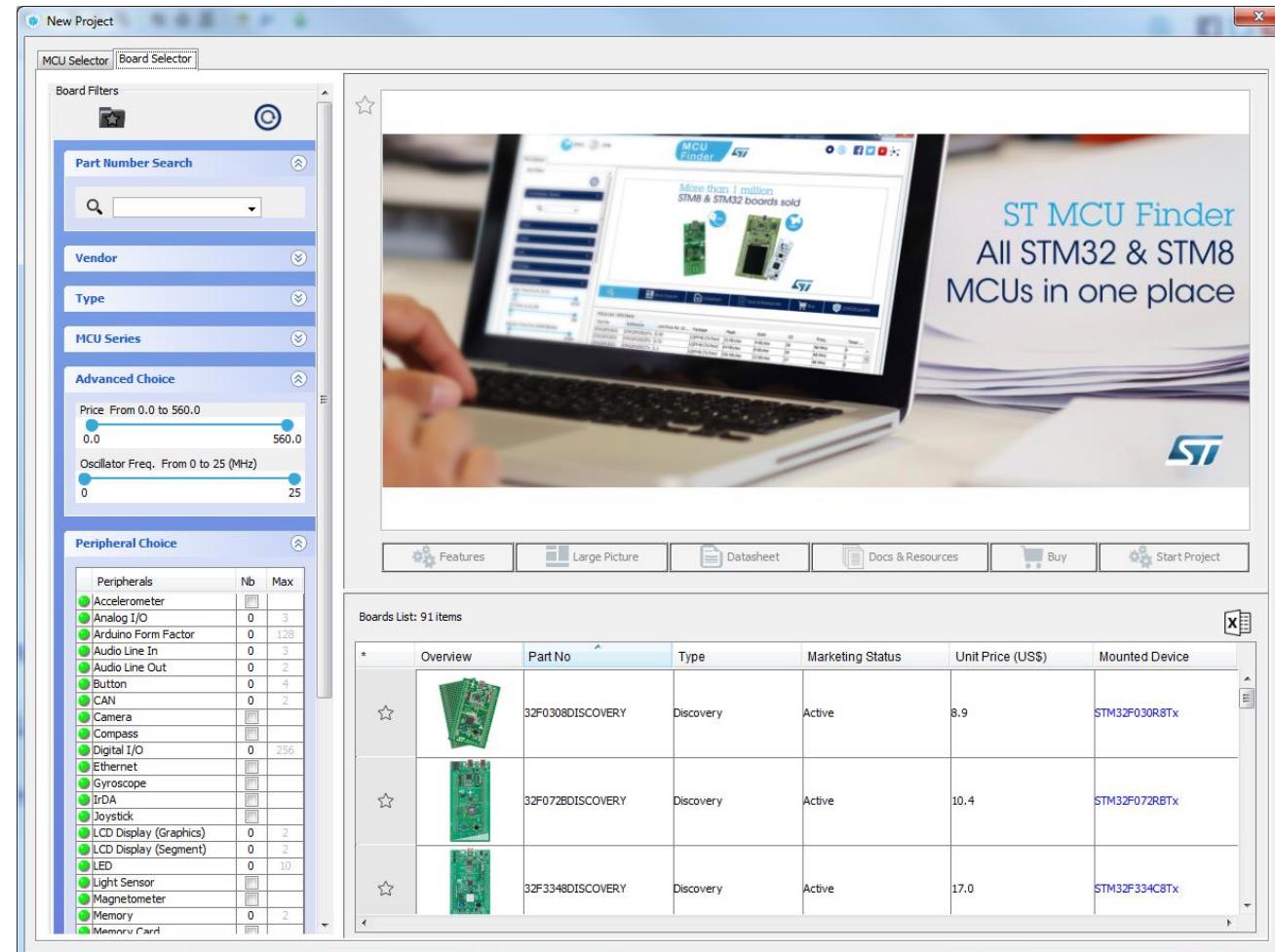
The image shows the ST MCU Finder software interface. On the left, a window titled 'New Project' displays 'MCU Selector' and 'Board Selector' tabs. The 'MCU Selector' tab is active, showing a 'MCU Filters' panel with sections for Part Number Search, Core, Series, Line, and Package. Below these are 'Advanced Choice' filters for Price (0.0 to 12.214), IO (11 to 168), Eeprom (0 to 16384 Bytes), Flash (8 to 2048 kBytes), Ram (2 to 1056 kBytes), and Freq. (24 to 400 MHz). A 'Peripheral Choice' section lists Peripherals (ADC 12-bit, ADC 16-bit, AES, CAN, COMP) with their counts (Nb) and maximum values (Max). To the right, a large window shows a laptop displaying the 'ST MCU Finder' website, which features a search bar, a sidebar with navigation links like 'Features', 'Block Diagram', 'Datasheet', 'Docs & Resources', 'Buy', and 'Start Project', and a main area with a product card for the STM32F030C6Tx. The card includes a photo of the board, a brief description, and technical specifications: Price \$0.597, LQFP48 package, 32 kBbytes flash, 4 kBbytes RAM, 39 IO pins, and 48 MHz frequency. Below the website window is a table titled 'MCUs List: 1129 items' with columns for Part No, Reference, Marketing Status, Unit Price for 10kU (US\$), Board, Package, Flash, RAM, IO, and Freq.

* Part No	Reference	Marketing Status	Unit Price for 10kU (US\$)	Board	Package	Flash	RAM	IO	Freq.
STM32F030C6	STM32F030C6Tx	Active	0.597		LQFP48	32 kBbytes	4 kBbytes	39	48 MHz
STM32F030C8	STM32F030C8Tx	Active	0.723		LQFP48	64 kBbytes	8 kBbytes	39	48 MHz
STM32F030CC	STM32F030CCTx	Active	1.1		LQFP48	256 kBbytes	32 kBbytes	37	48 MHz
STM32F030F4	STM32F030F4Tx	Active	0.424		TSSOP20	16 kBbytes	4 kBbytes	15	48 MHz
STM32F030K6	STM32F030K6Tx	Active	0.519		LQFP32	32 kBbytes	4 kBbytes	25	48 MHz
STM32F030R8	STM32F030R8Tx	Active	0.754		LQFP64	64 kBbytes	8 kBbytes	55	48 MHz
STM32F030RC	STM32F030RCTx	Active	1.21		LQFP64	256 kBbytes	32 kBbytes	51	48 MHz
STM32F031C4	STM32F031C4Tx	Active	0.971		LQFP48	16 kBbytes	4 kBbytes	39	48 MHz
STM32F031C6	STM32F031C6Tx	Active	1.014		LQFP48	32 kBbytes	4 kBbytes	39	48 MHz
STM32F031E6	STM32F031E6Yx	Active	0.777		WL CSP25	32 kBbytes	4 kBbytes	20	48 MHz
STM32F031F4	STM32F031F4Px	Active	0.711		TSSOP20	16 kBbytes	4 kBbytes	15	48 MHz
STM32F031F6	STM32F031F6Px	Active	0.756		TSSOP20	32 kBbytes	4 kBbytes	15	48 MHz

# MCU selector

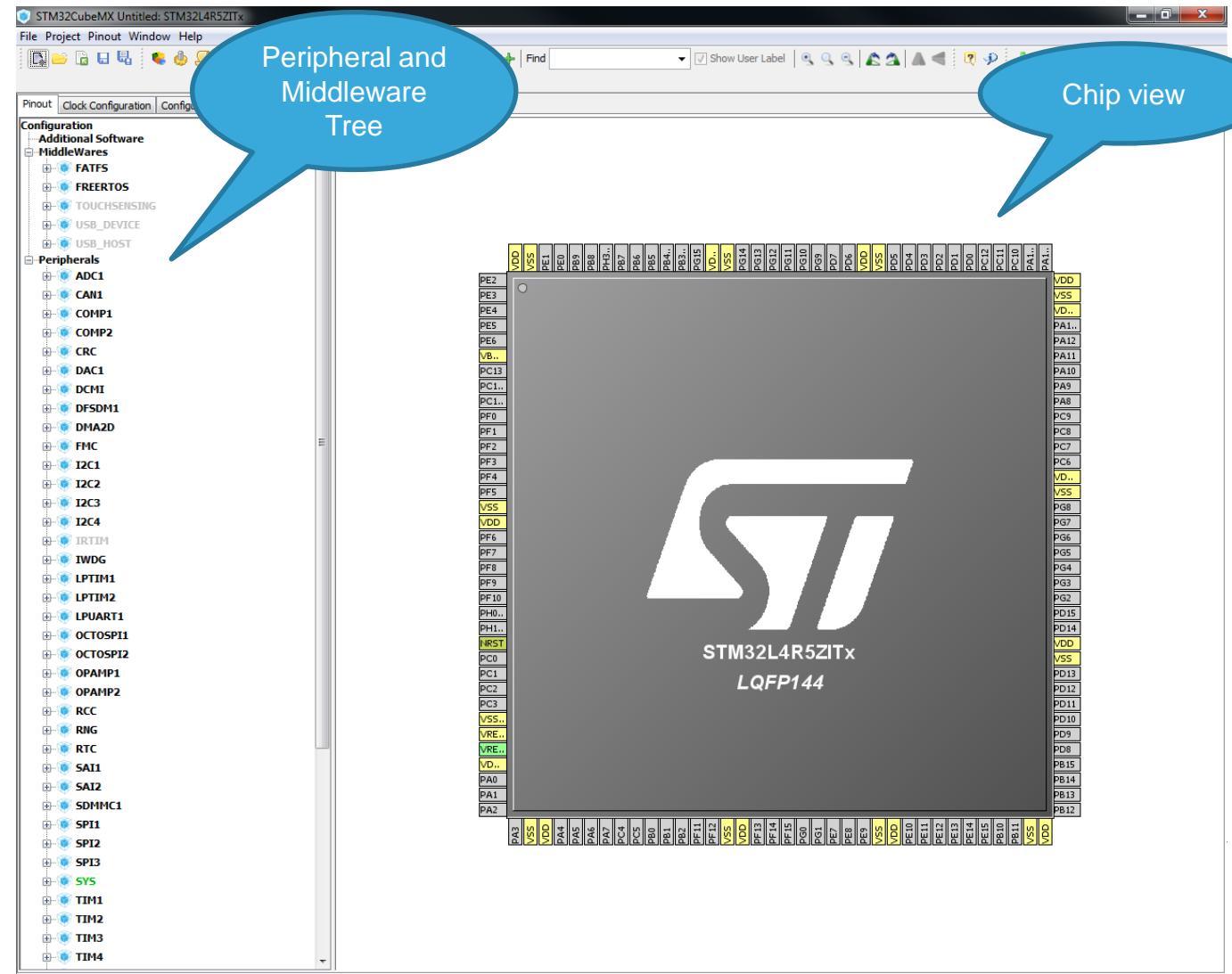
26

- Second tab provides shortcuts to predefined boards equipped with STM32 MCU.
- Predefined boards come with pinouts already assigned to use the connections and features of the particular board.
- Alternative board configurations are not covered.

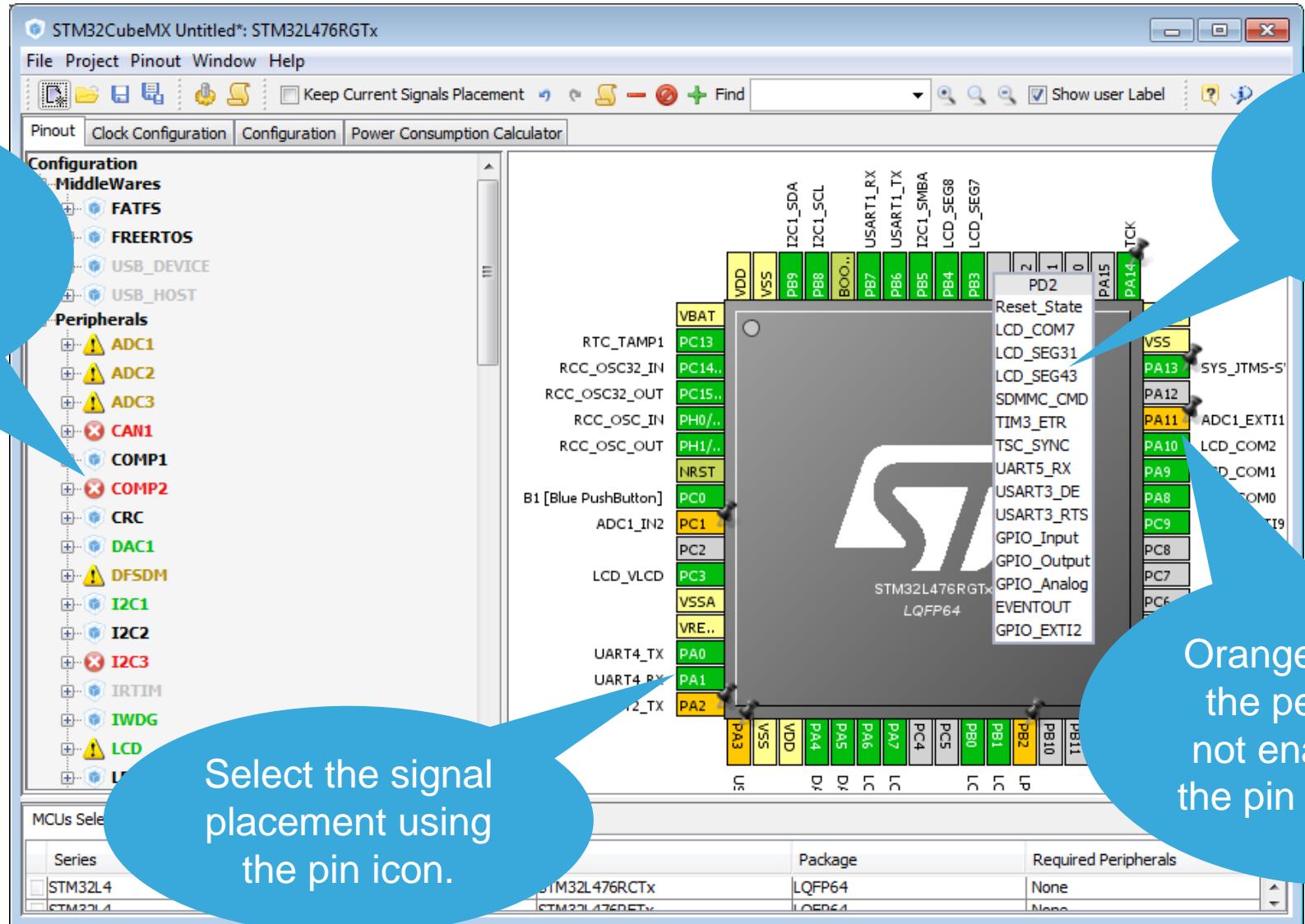


# Pin assignment

- Two way to configure microcontroller
  - Peripheral and Middleware tree
  - Chip view
- Automatic signal remapping
- Management of dependencies between peripherals and/or middleware (FatFS, FREERTOS, USB, ...)



# Pin assignment



# Pin assignment (cont.)

29

- Different possible states for peripheral modes.
  - Dimmed: the mode is not available because it requires another mode to be set.
  - Yellow : The mode is available with limitations.
  - Red : Signals required for this mode can't be mapped to the pinout.
- Signals can be set/moved directly from the pinout view.
  - Click on the pin to see the list of possible signals and select one.
  - To see alternate pins for a signal, CTRL+click on the signal and drag it elsewhere.
  - Ignore unused pins since the code generator can set them to power-saving analog mode.

- Pinning & labelling signals on pins
  - In “Chip” view, right click on signals to pin/unpin
  - From [Pinout->Pins/Signals Options...] dialog box
- Keep Current Signals Placement
  - Allocated (mapped) pins function can't be moved automatically by STM32CubeMX to another pin once it has been allocated
  - Unchecked/disabled by default

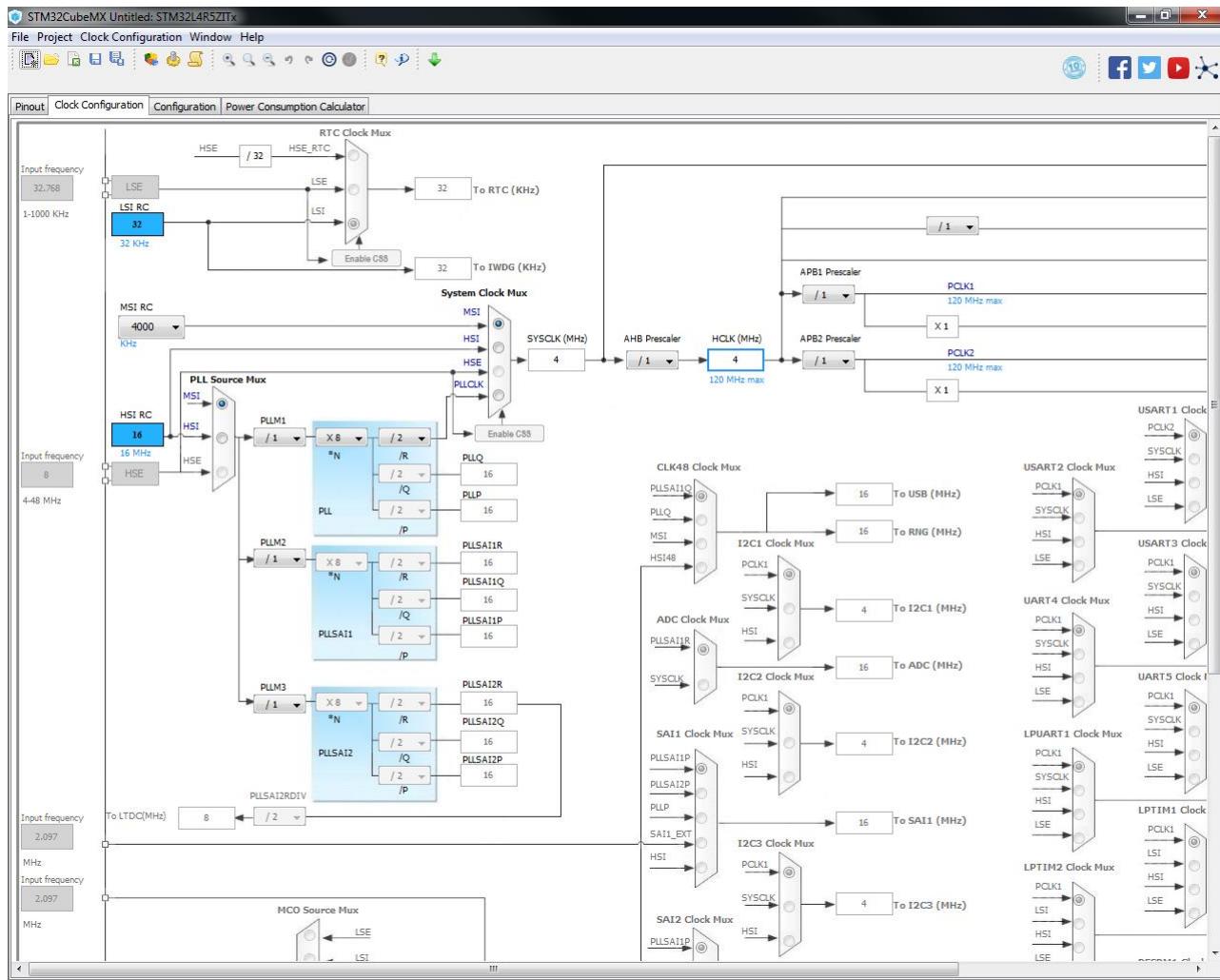
Example in STM32L4R5ZI,

1. LPUART1 asynchronous mode is enabled (PC0 and PC1 will be selected)
2. Checked and Unchecked “Keep Current Signals Placement”
  - Observed ADC1 IN1 and IN2

# Clock configuration

31

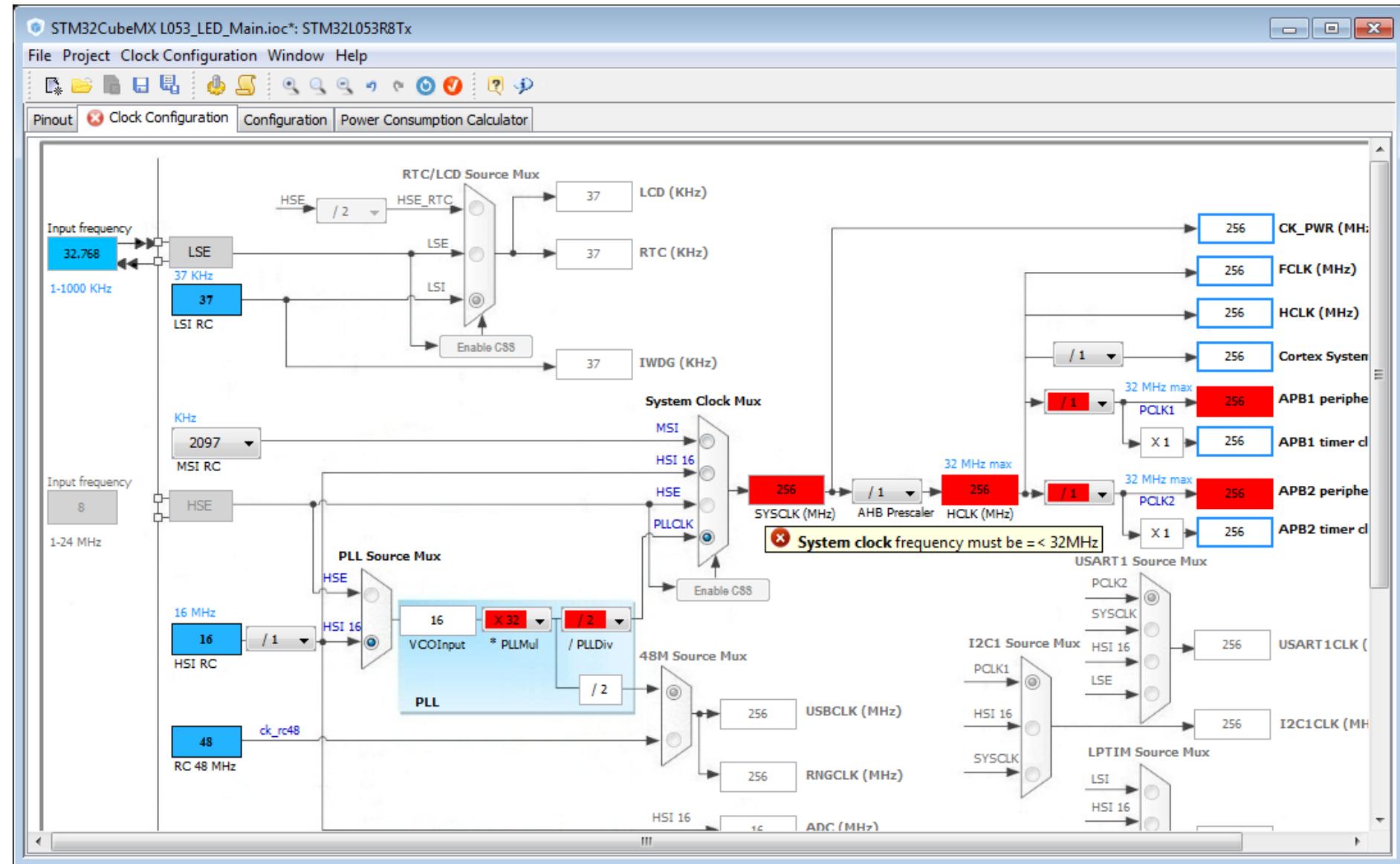
- Immediate display of all clock values
- Management of all clock constraints
- Highlight of errors
- Enter desired clock and let tool adjust the divider and multiplier



# Clock configuration

32

- Highlight of errors – instantly turns red.
- Enter the value in the blue frame and let the tool adjust the dividers and multipliers.
- Lock a value to prevent the tool from modifying it.



# Peripheral and middleware configuration

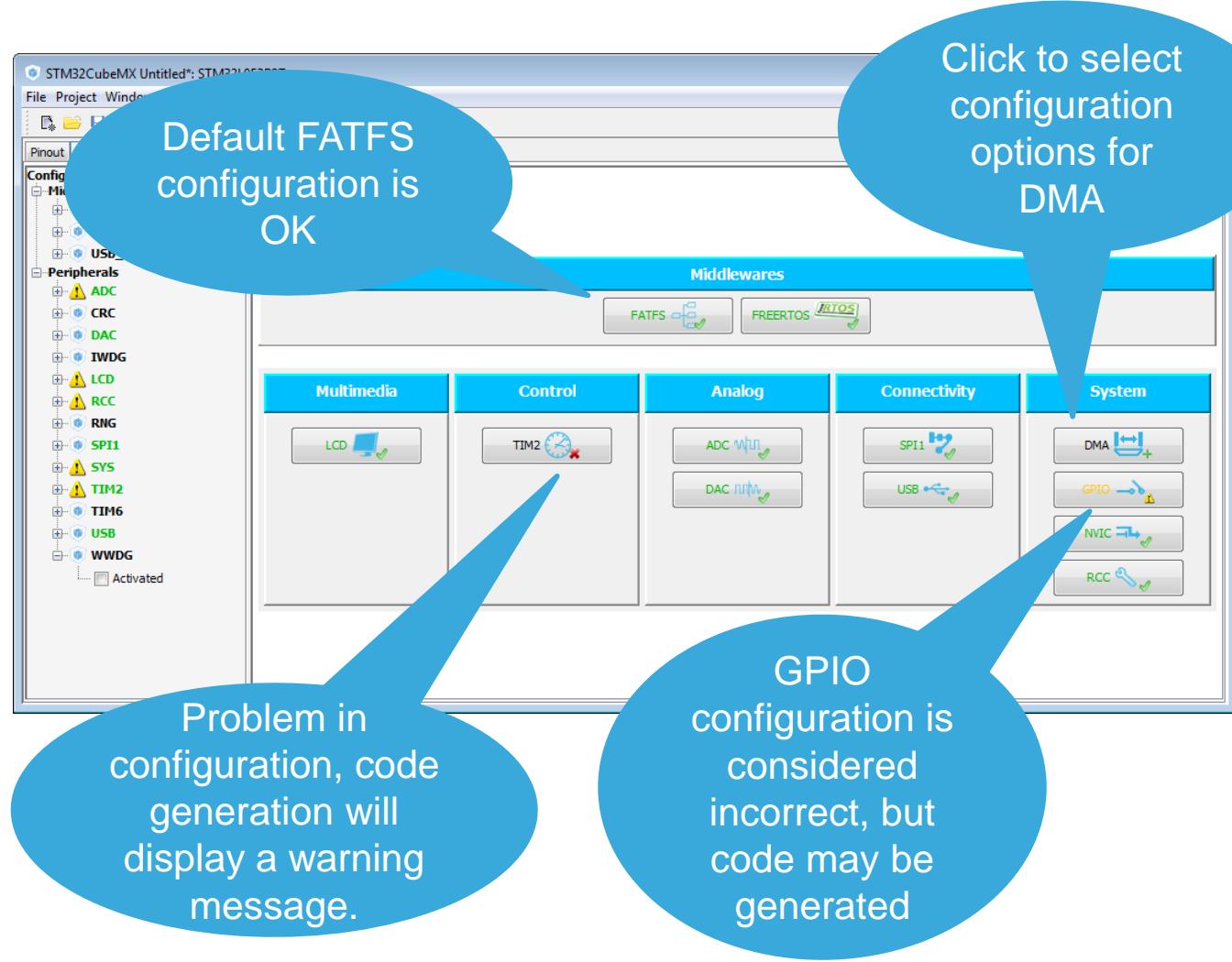
33

- Global view of used peripherals and middleware

- Highlight of configuration errors

- + Not configured
- ✓ OK
- ⚠ Non-blocking problem
- ✗ Error

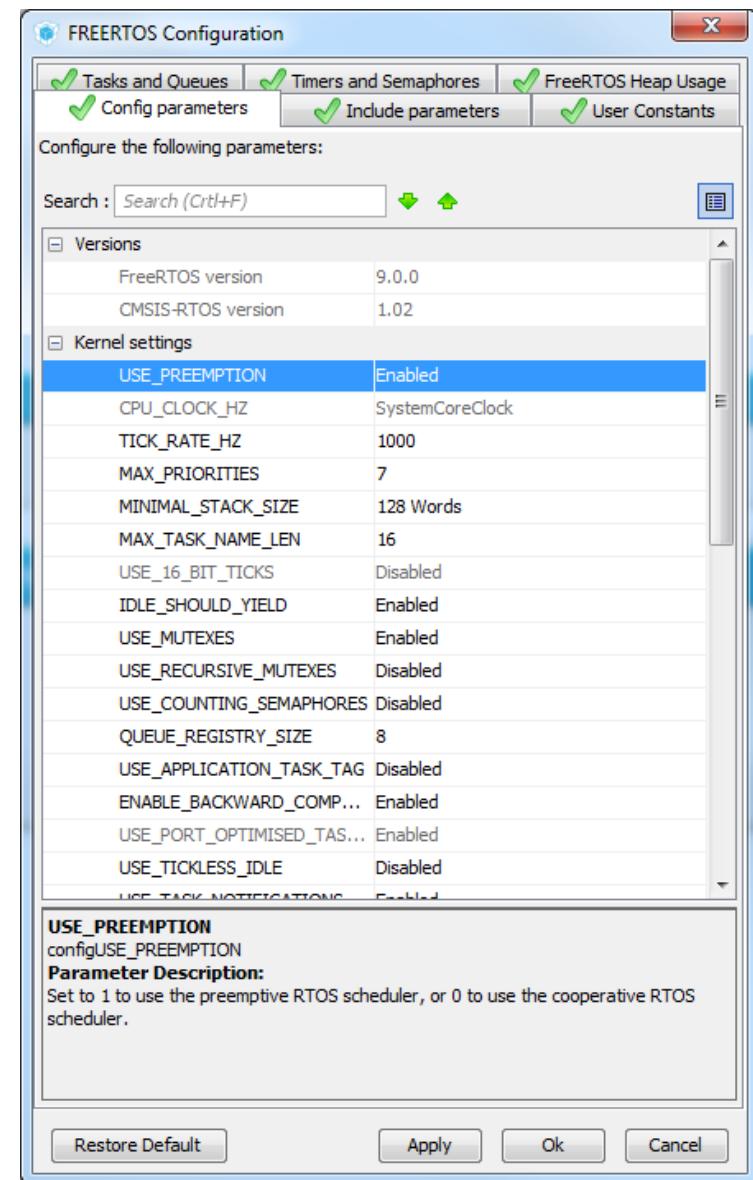
- Read only tree view on the left with access to IPs / Middleware having no impact on the pinout



# Middleware configuration

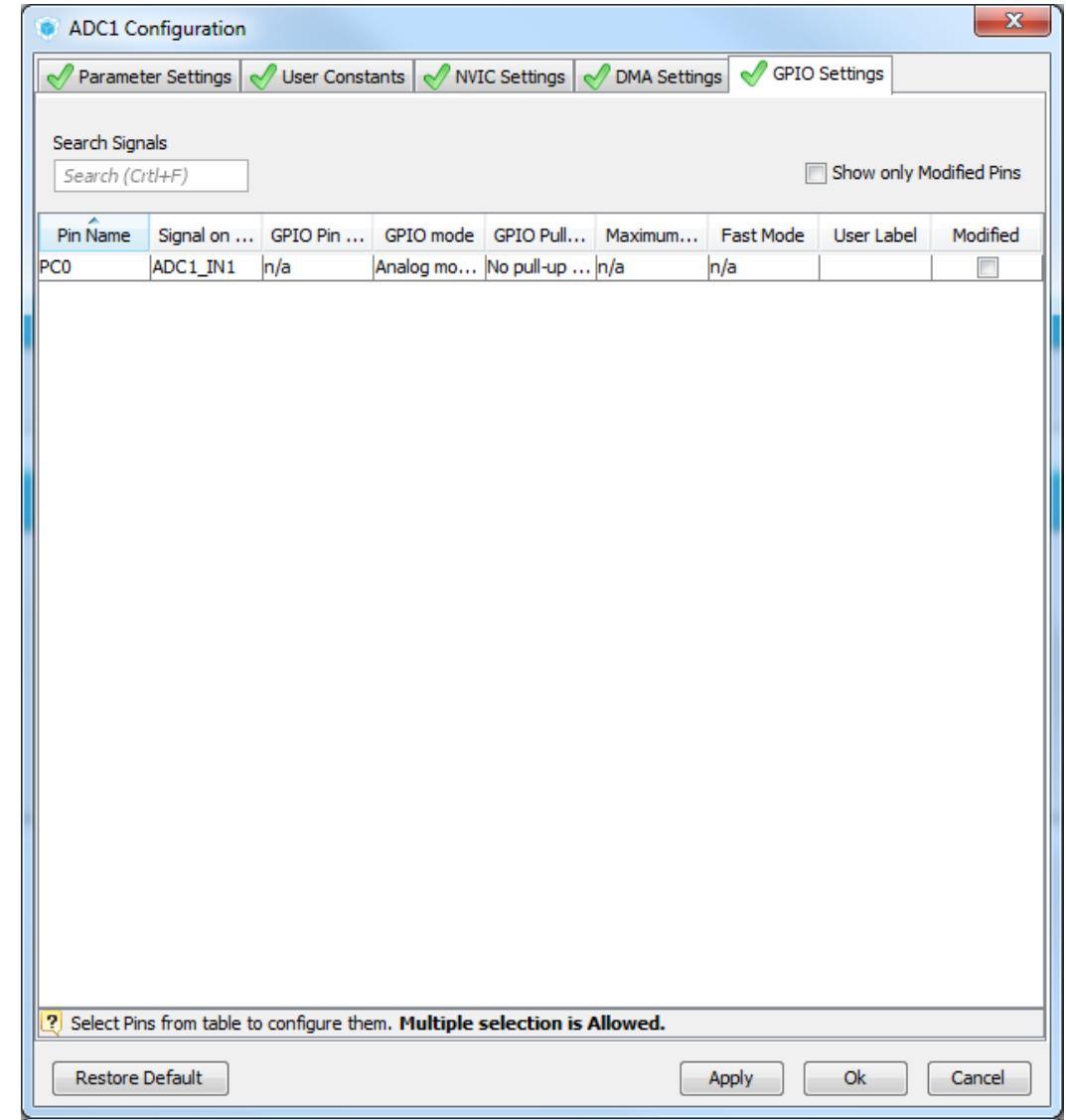
34

- Presents options specific to each supported software component.
- All settings are organized in logical groups.
- Description and constraints are available for quick reference.



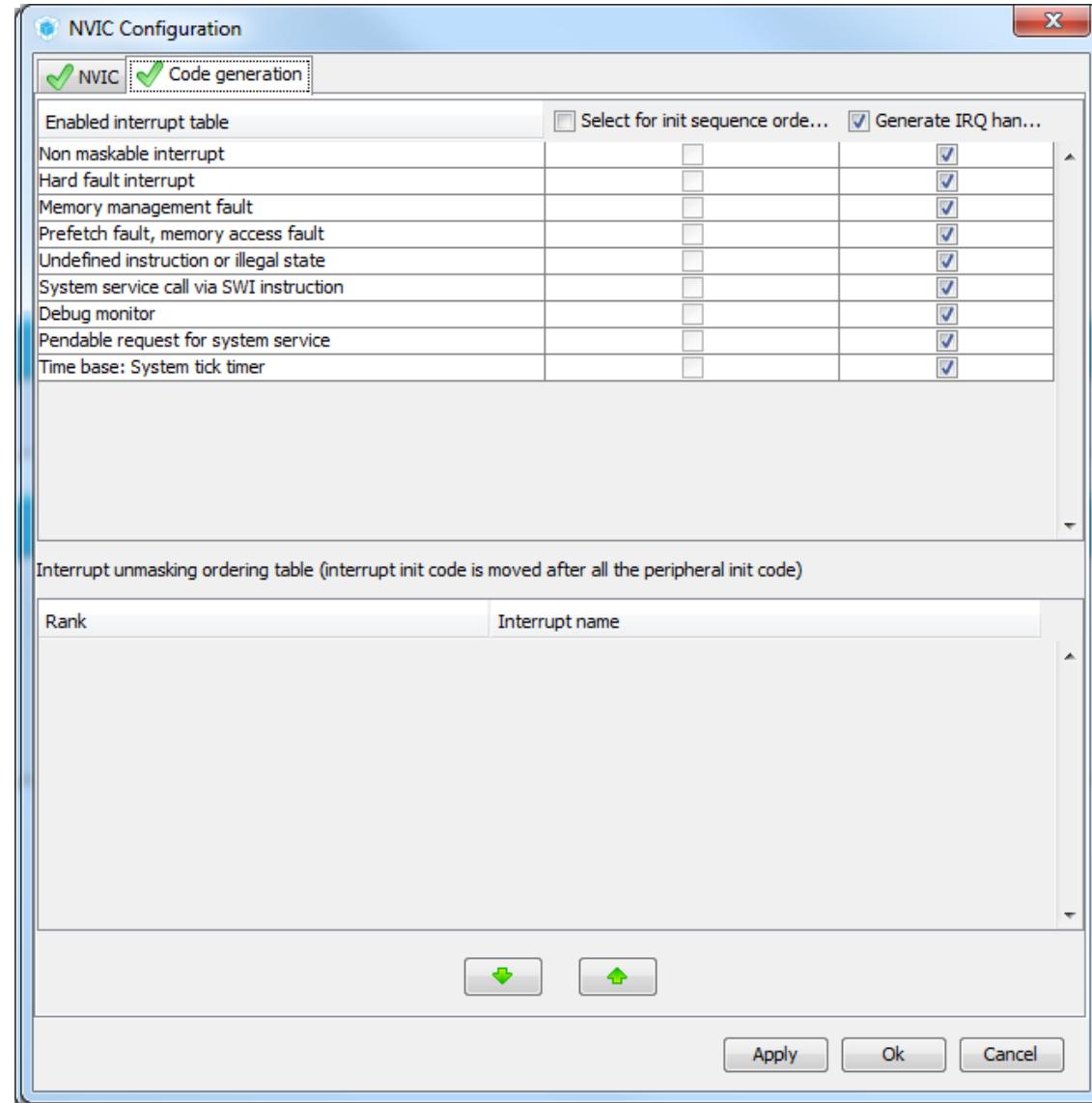
# Peripheral configuration

- All available initialization parameters are presented with short description and options.
- Interrupts may be assigned to peripherals.
- DMA may be associated, where applicable.
- GPIO settings for peripherals with input and/or output.



# NVIC configuration panel

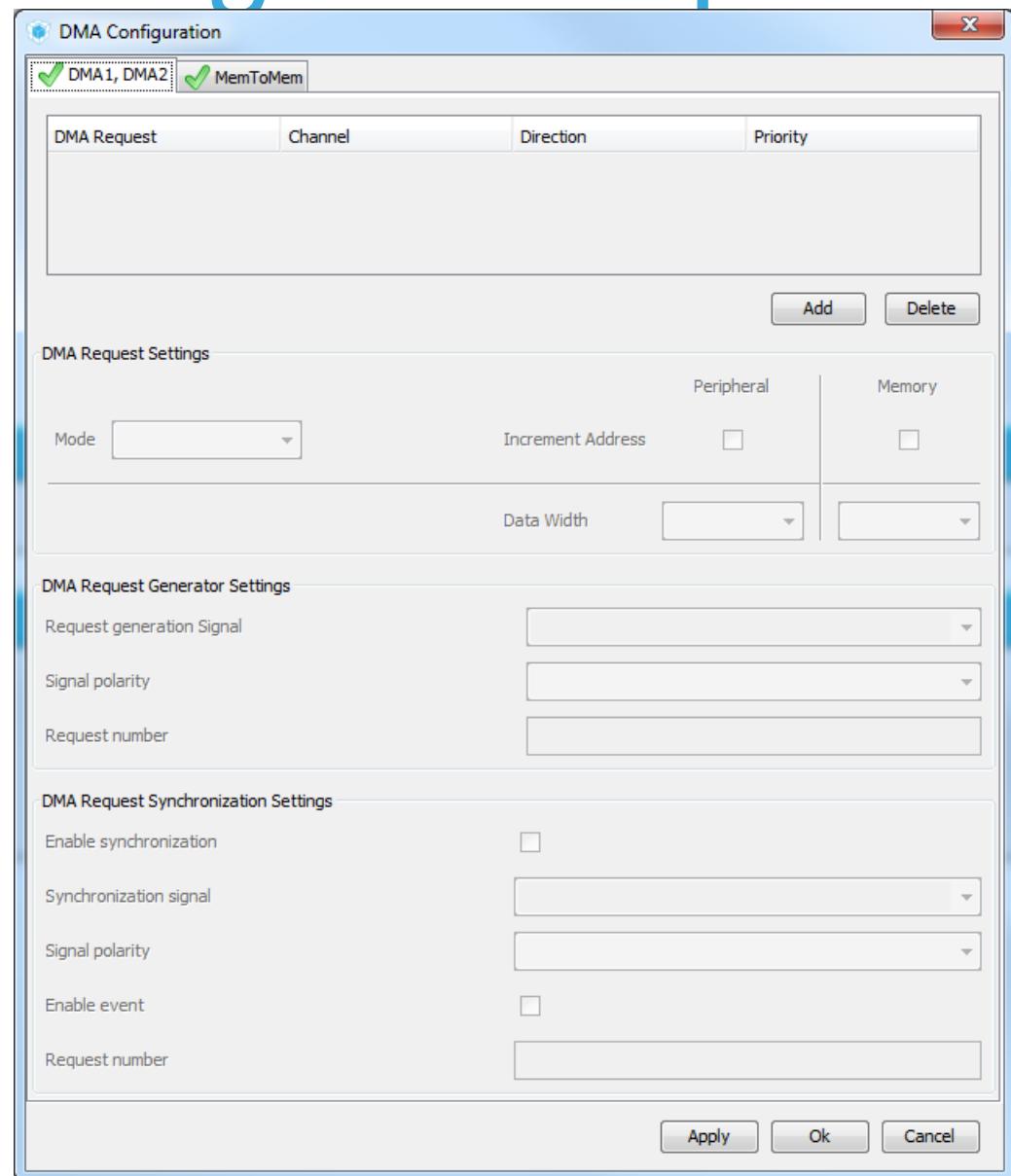
- Single control panel for all interrupts.
- Manage priorities and sub-priorities.
- Interface for searching, filtering and sorting interrupts in the list.



# DMA configuration panel

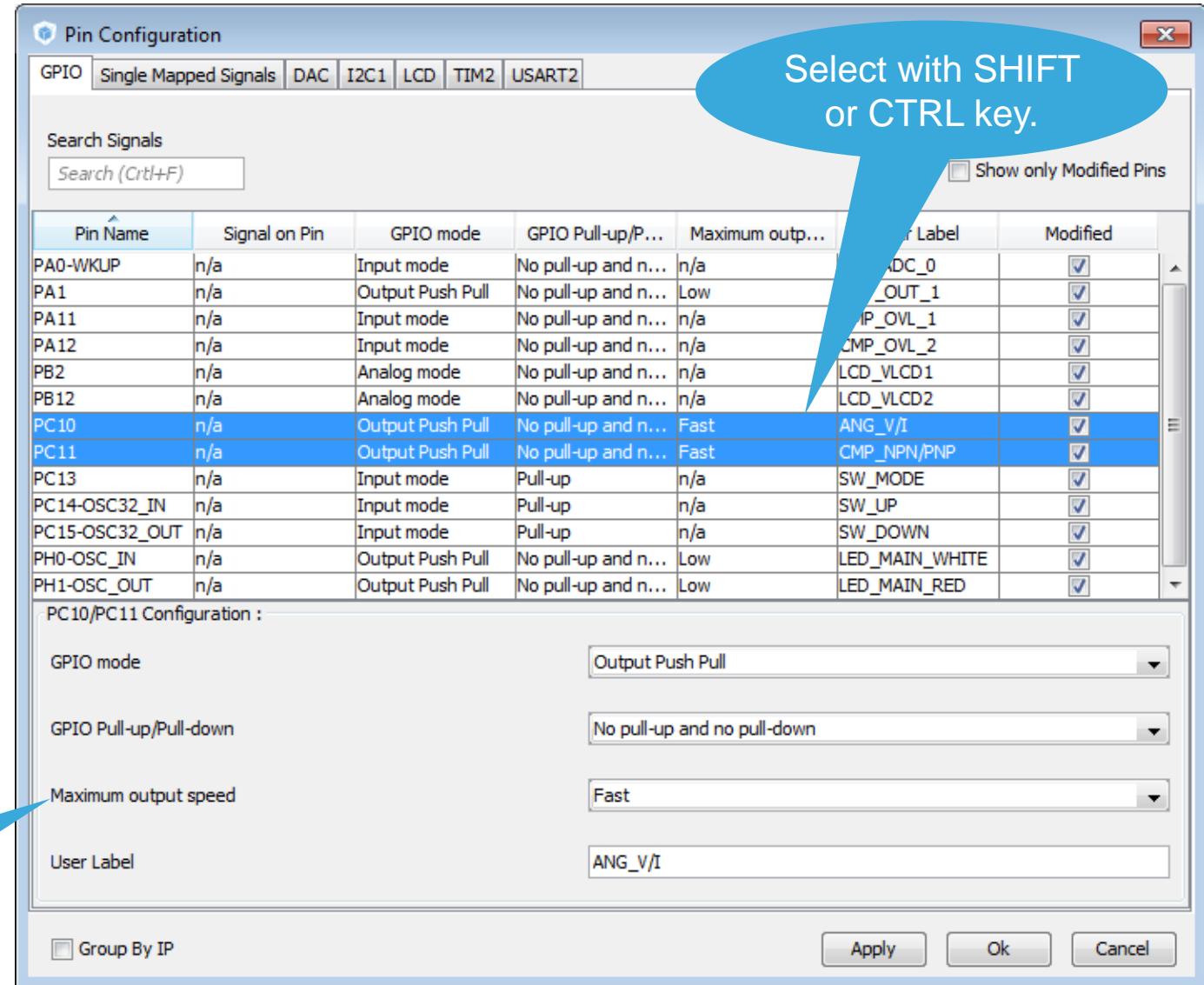
37

- Manages all DMA requests including memory to memory.
- Configure direction, priority and other settings.



# GPIO configuration panel

- The application attempts to set sensible default values to most GPIO parameters.
- Default values are chosen conservatively, as low speed and no pull-up.
- Multiple pins can be selected simultaneously for the same configuration.



# Code generation

39

- Generates all the initialization code in C.
- Generates project file for any supported development toolchain.
- User code can be added in dedicated sections and will be kept upon regeneration.
- Option to use the latest library version or keep the same even if re-generating.

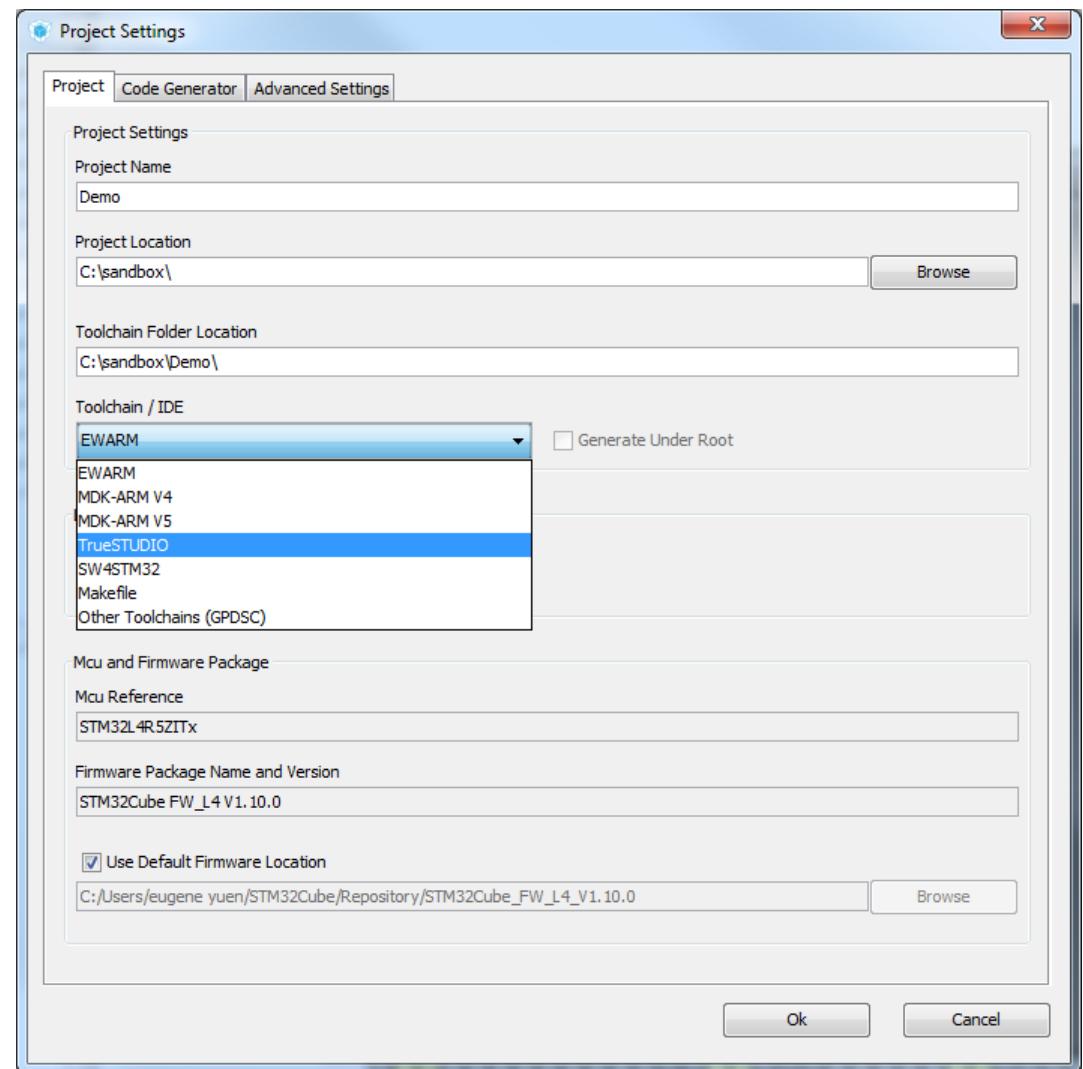
The screenshot shows a code editor window with the file "main.c" open. The code is a C program that includes various header files and defines structures for ADC handling. It features several sections of code marked with comments like /\* USER CODE BEGIN \*/ and /\* USER CODE END \*/. A blue speech bubble is overlaid on the right side of the editor, containing the text: "Write your code here to keep option to regenerate the project." This indicates that users can add their own code to these sections while maintaining the ability to regenerate the entire project.

```
22  ****
23  */
24  /* Includes -----
25  #include "stm32f4xx_hal.h"
26  #include "cmsis_os.h"
27  #include "lwip.h"
28  #include "usb_device.h"
29
30  /* Define structures */
31  ADC_HandleTypeDef hadc1;
32
33
34  /* USER CODE BEGIN 0 */
35
36  /* USER CODE END 0 */
37  /* Private function prototypes -----
38  static void SystemClock_Config(void);
39  static void StartThread(void const * argument);
40  static void MX_GPIO_Init(void);
41  static void MX_ADC1_Init(void);
42  static void MX_NVIC_Init(void);
43
44  int main(void)
45  {
46  /* USER CODE BEGIN 1 */
47
48  /* USER CODE END 1 */
49  /* MCU Configuration-----
50  /* Reset of all peripherals, Initializes the Flash interface
51  HAL_Init();
52  /* Configure the system clock */
```

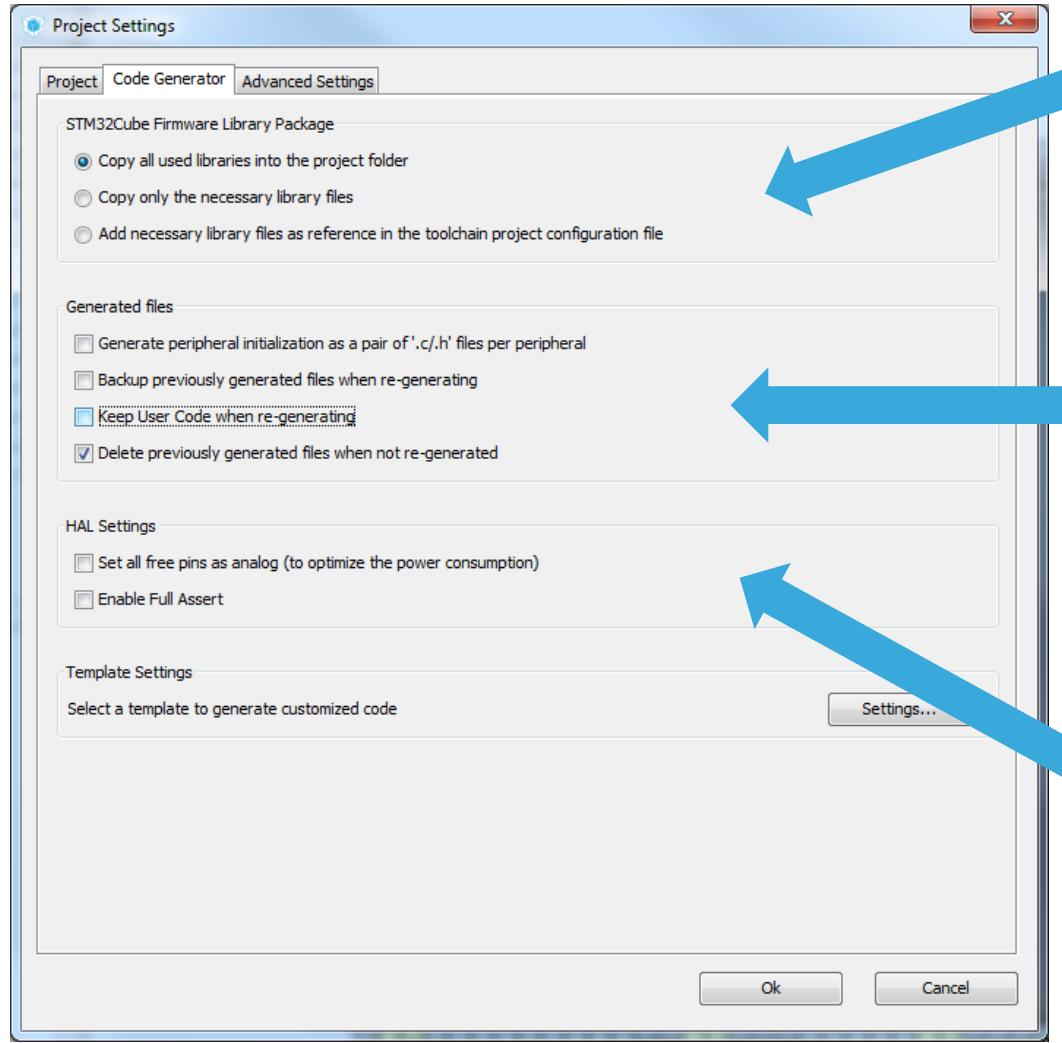
# Project settings

40

- Name your project when saving
- Browse for project location
- Select the preferred toolchain
  - EWARM, MDK ARM, TrueSTUDIO, SW4STM32
- Review the exact MCU type and library version
  - Specific version of STM32Cube Firmware library can be specified



# Code generation options



- STM32Cube Firmware Library package
  - Copy entire library or just the required part to the generated project folder.
  - Or keep the library in its original location and refer to it from all projects.
- Generated files
  - Each peripheral is initialized in separate file or in common source file (main.c)
  - Options for handling old files.
  - **The option to keep user code intact is here.**
- HAL settings
  - Setting free pins to analog leads to lower power consumption, but be careful to **explicitly select SWD/JTAG in Pinout tab**.
  - Full assert is useful for debugging.

# STM32CubeMX C Code generation overview

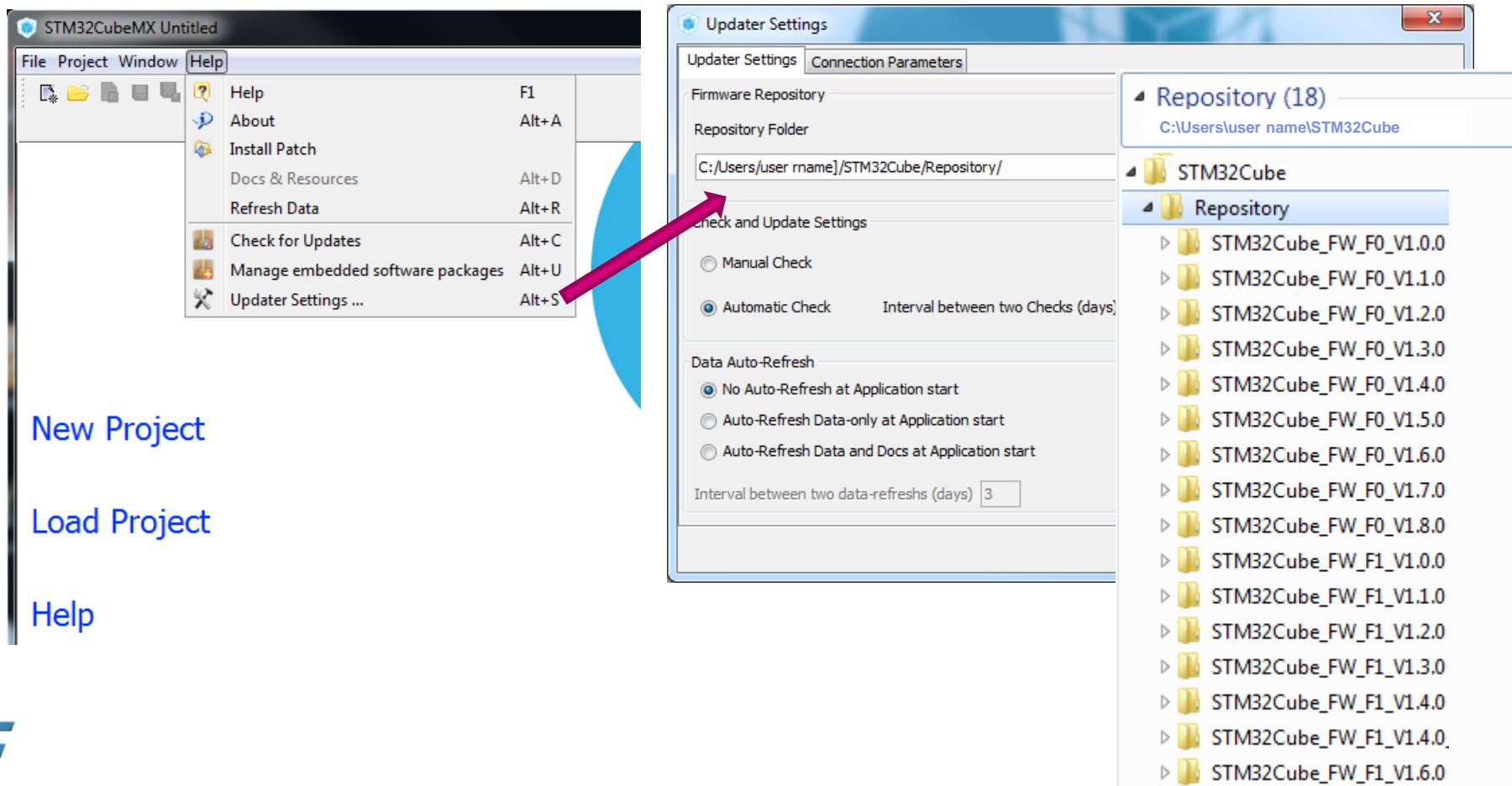
42

- During the C code generation process, STM32CubeMX performs the following actions:
  - It downloads the relevant STM32Cube firmware package if it is missing from the STM32CubeMX repository.
  - It copies from the firmware package, the relevant files in Drivers/CMSIS and Drivers/STM32xx\_HAL\_Driver folders and in the Middleware folder if a middleware was selected.
  - It generates a Projects folder that contains the toolchain specific files that match the user project settings.
  - It generates the initialization C code ( .c/.h files) corresponding to the user MCU configuration and stores it in the Inc and Src folders. By default, the following files are included:

Files	Description
<b>stm32l4xx_hal_conf.h</b>	this file defines the enabled HAL modules and sets some parameters (e.g. External High Speed oscillator frequency) to pre-defined default values or according to user configuration (clock tree).
<b>stm32l4xx_hal_msp.c</b> (MSP=MCU Support package)	this file defines all initialization functions to configure the IP instances according to the user configuration (pin allocation, enabling of clock, use of DMA and Interrupts).
<b>main.c</b>	is in charge of: <ul style="list-style-type: none"><li>- Resetting the MCU to a known state by calling the HAL_init() function that resets all peripherals, initializes the Flash memory interface and the SysTick.</li><li>- Configuring and initializing the system clock.</li><li>- Configuring and initializing the GPIOs that are not used by IPs.</li><li>- Defining and calling, for each configured IP, an IP initialization function that defines a handle structure that will be passed to the corresponding IP HAL init function which in turn will call the IP HAL MSP initialization function.</li></ul>

# STM32CubeMX Repository

- Downloaded software and firmware releases will be stored in the Repository folder. The Default folder is defined in STM32CubeMX->Help->Updater Settings->Repository Folder.



# STM32CubeMX Documentation

44

## Technical Documentation

Product Specifications		
Description	Version	Size
DB2163: STM32 configuration and initialization C code generation	8.0	267 KB
User Manuals		
Description	Version	Size
UM1718: STM32CubeMX for STM32 configuration and initialization C code generation	23.0	21 MB
Release Notes		
Description	Version	Size
RN0094: STM32CubeMX release 4.23.0	33.0	603 KB
Legal		
License Agreement		
Description	Version	Size
SLA0047: Image V2 - SOFTWARE LICENSE AGREEMENT	1.14	99 KB



<http://www.st.com/stm32cube>

Embedded Software		
MCUS EMBEDDED SOFTWARE	Part Number	Manufacturer
STM32CubeF0	ST	STM32Cube MCU Package for STM32F0 series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, File system, RTOS, Touch Sensing - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeF1	ST	STM32Cube MCU Package for STM32 F1 series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, File system, RTOS, Touch Sensing, Graphic - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeF2	ST	STM32Cube MCU Package for STM32 F2 series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, File system, RTOS, Touch Sensing, Graphic - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeF3	ST	STM32Cube MCU Package for STM32F3 series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, File system, RTOS, Graphic - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeF4	ST	STM32Cube MCU Package for STM32F4 series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, TCP/IP, File system, RTOS, Graphic - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeF7	ST	STM32Cube MCU Package for STM32F7 series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, TCP/IP, File system, RTOS, Graphic - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeH7	ST	STM32Cube MCU Package for STM32H7 series (HAL low level drivers, USB, TCP/IP, File system, RTOS, Graphic - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeL0	ST	STM32Cube MCU Package for STM32L0 series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, File system, RTOS, Touch Sensing - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeL1	ST	STM32Cube MCU Package for STM32 L1 series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, File system, RTOS, Touch Sensing, Graphic - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)
STM32CubeL4	ST	STM32Cube MCU Package for STM32L4 series and STM32L4 Plus series (HAL, Low-Layer APIs and CMSIS (CORE, DSP, RTOS), USB, TouchSensing, File system, RTOS, Graphic - coming with examples running on ST boards: STM32 Nucleo, Discovery kits and Evaluation boards)

## Links to various STM32Cube offering

d CMSIS with ition

d CMSIS with

# Warning and disclaimer

45

- Universal design for the entire STM32 family range at times prevents the tool from focusing on specific features of a particular product.
- **The STM32CubeMX GUI tool is not a replacement for the reference manual or datasheet**
  - Always refer to written documentation for further information!
  - Important features are often available on the product or in the HAL but not in the GUI.
- The GUI helps start a project and initialize a working starting configuration – but the configuration can be dynamically changed at runtime (i.e. GPIO, NVIC priority or clock settings).



# Atollic TrueSTUDIO®

# 2017 Christmas Present

47

ST acquires Atollic

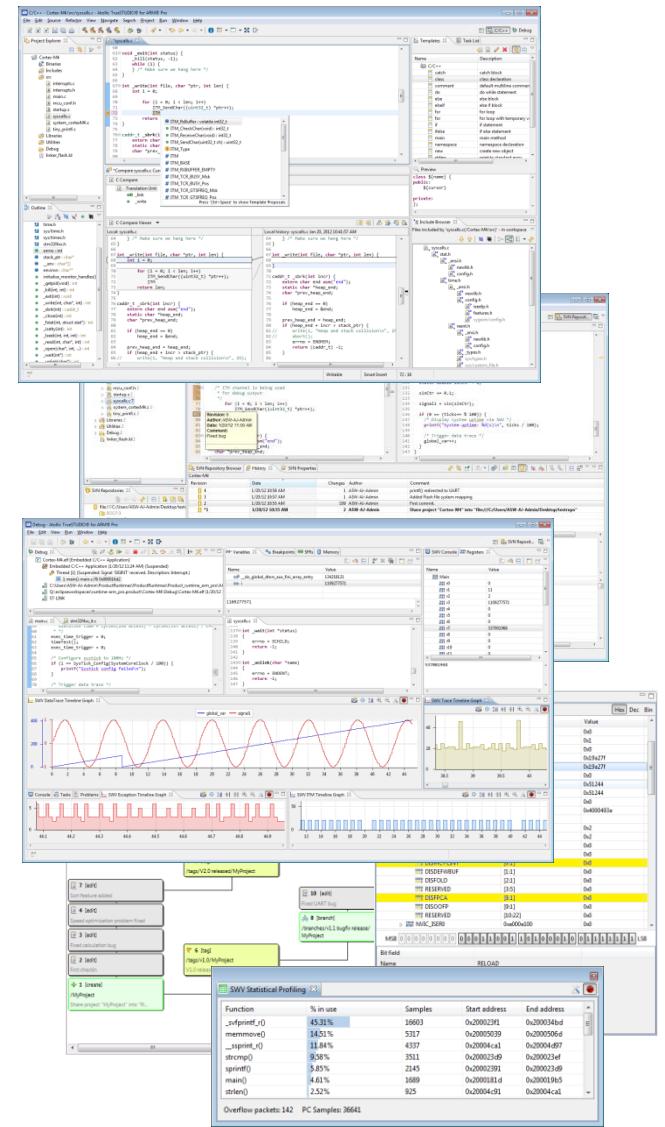
TrueSTUDIO®  
will become  
part of the  
STM32  
ecosystem



Pro version of TrueSTUDIO® made available to STM32 developers at no charge.

# TrueSTUDIO® for STM32

- Free to download and use even for commercial projects
- Built on **Eclipse**, **GCC** and **GDB** but polished and simplified
- Edit, compile & debug with no code size limitation
- Project wizards, configuration interface and sample projects
- Project management and team collaboration tools
- Extensive STM32 target support
- STM32 CubeMX integrates and runs nicely inside TrueSTUDIO!
- Kernel-aware RTOS debugging for **FreeRTOS/OpenRTOS**, **ThreadX**, **Micrium uC-OS III**, **Segger embOS**, **RTXC Quadros**
- Supports ST Link and Segger **J-Link/J-Trace** debug probes



# Advance debug feature

- Build Analyzer
  - View RAM and FLASH memory utilization
- Static Stack Analyzer
  - View call-graph and worse case stack usage scenario
- Crash Analyzer
  - Instantly see the cause and location of hard faults
- Data Trace
  - Visualize and log data events in real time during debug
- Performance Analysis
  - Real-time system performance optimization
- Exception/Interrupt trace
  - Visualize and analyze interrupt performance
- RTOS aware debugging

Pro version features now  
available free

- Atollic TrueSTUDIO® website
  - <https://atollic.com/>
- TrueSTUDIO® Application notes
  - <http://info.atollic.com/resources/application-notes>
- User forum/discussion boards
  - <https://forum.atollic.com/>
- Videos
  - <https://www.youtube.com/user/AtollicTools>