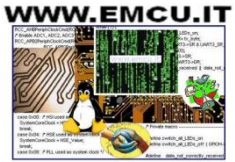
Free SW tools for STM32

life.augmented

AVNET Memec

SILICA An Avnet Company



Free SW tools for STM32

TUTORIAL

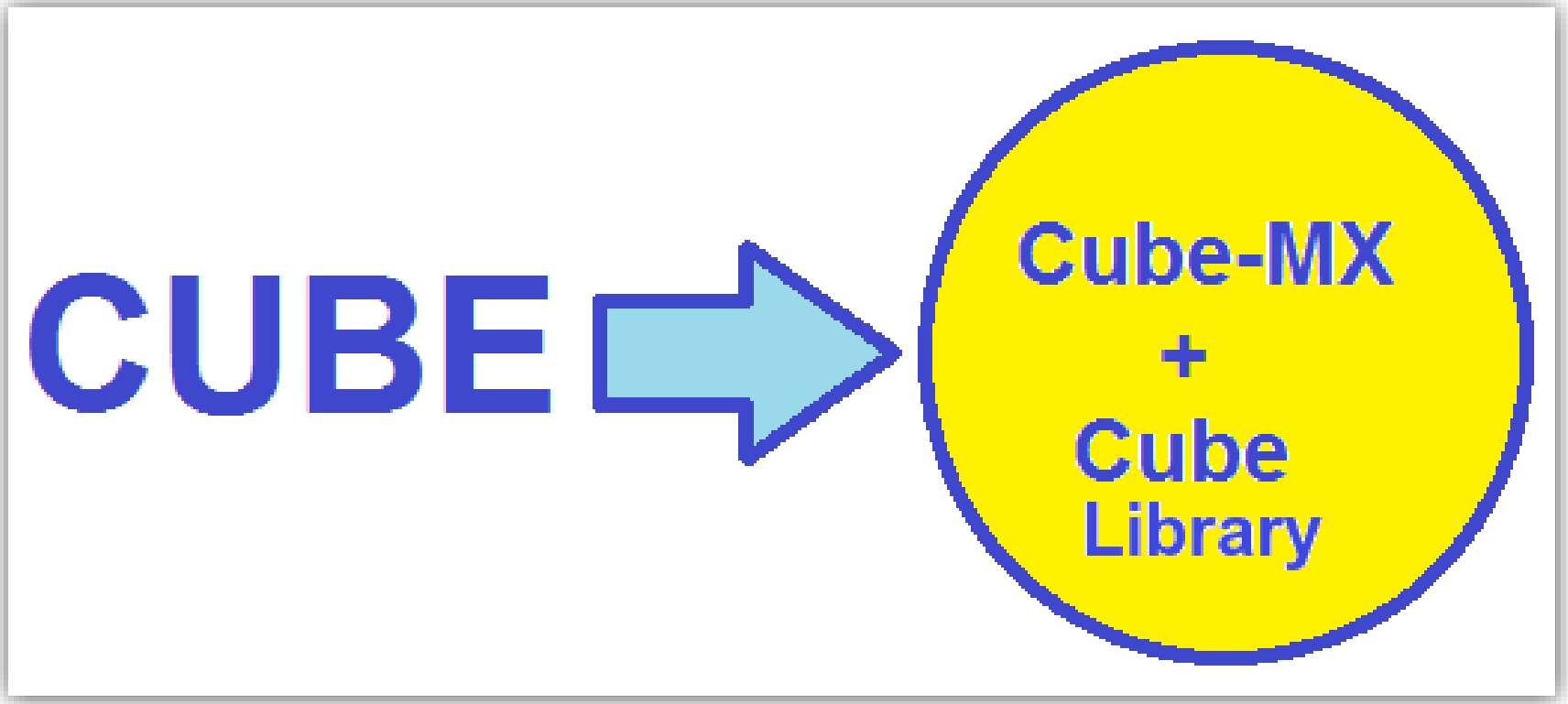
KEIL™ Tools by ARM

Free for STM32L0 and F0

GCC

atollic

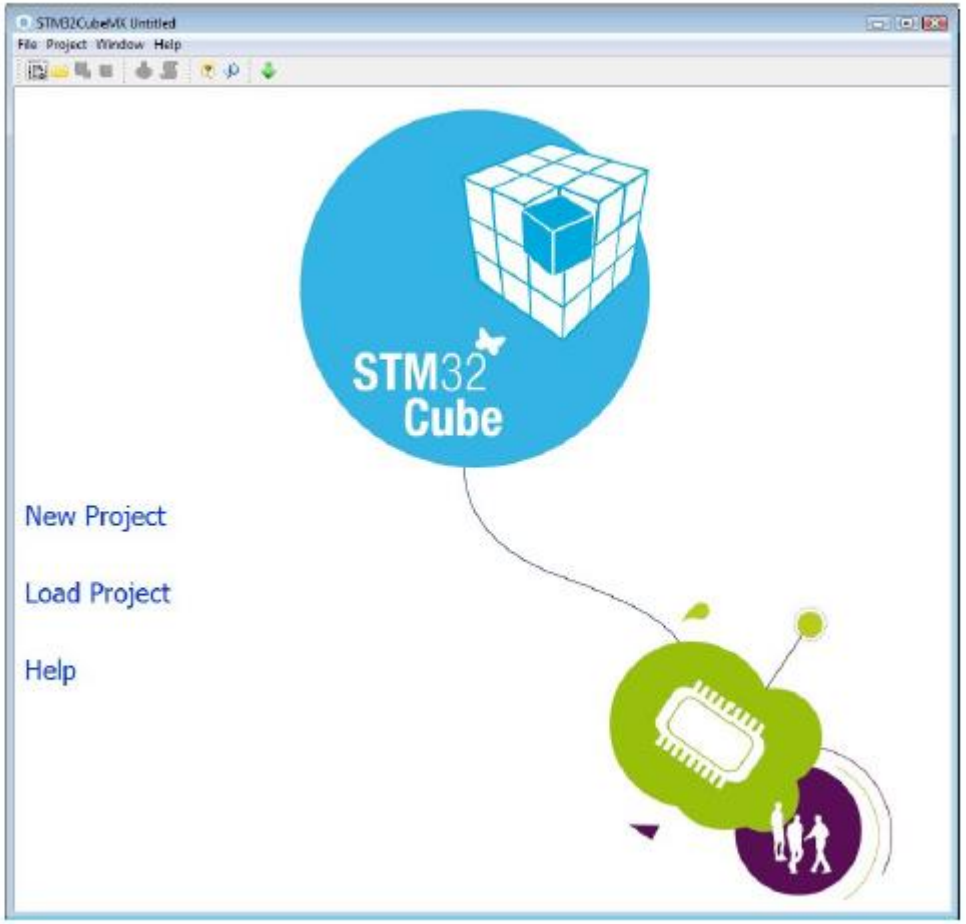
CooCox



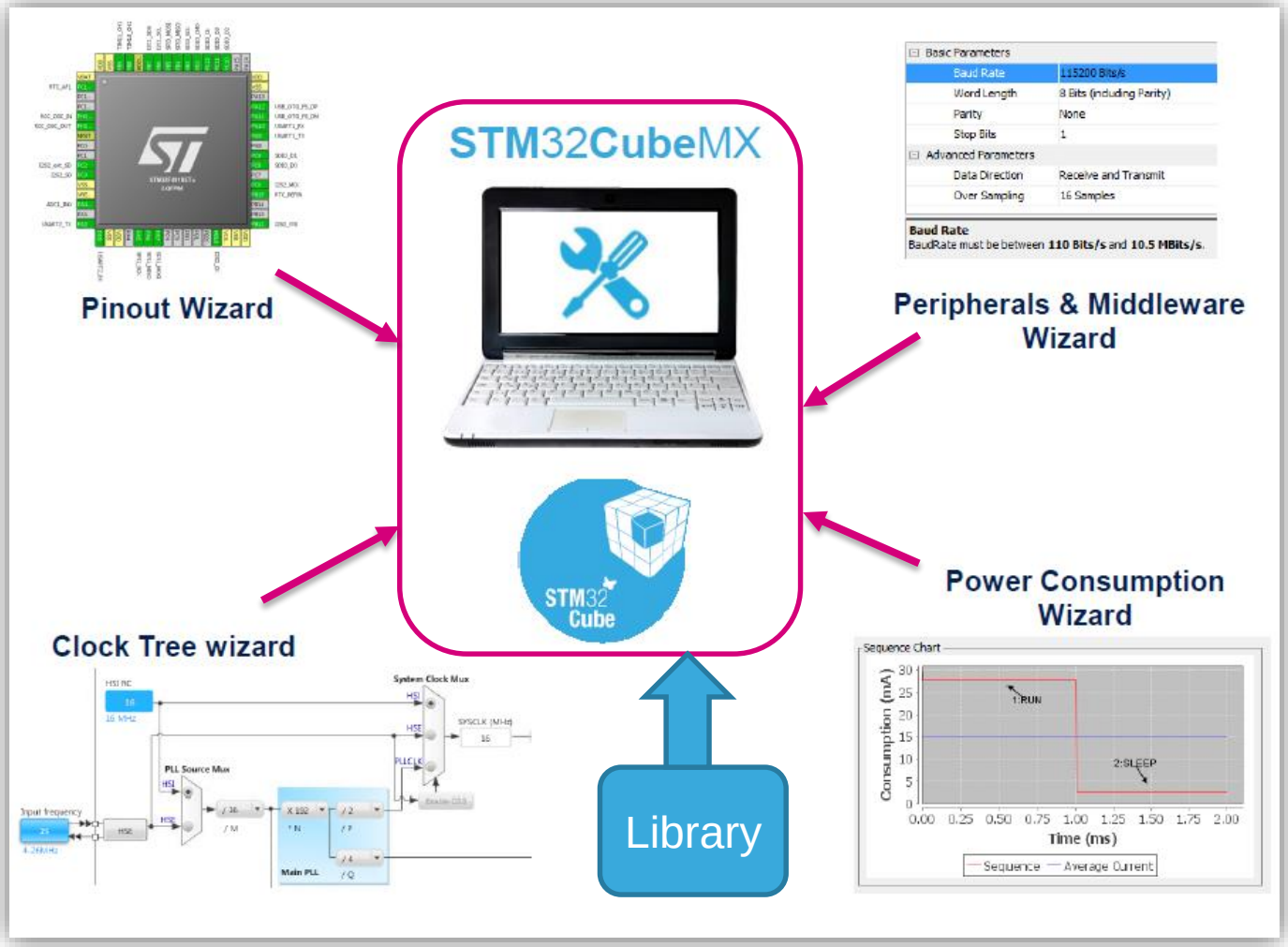
See here

CUBE-MX

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

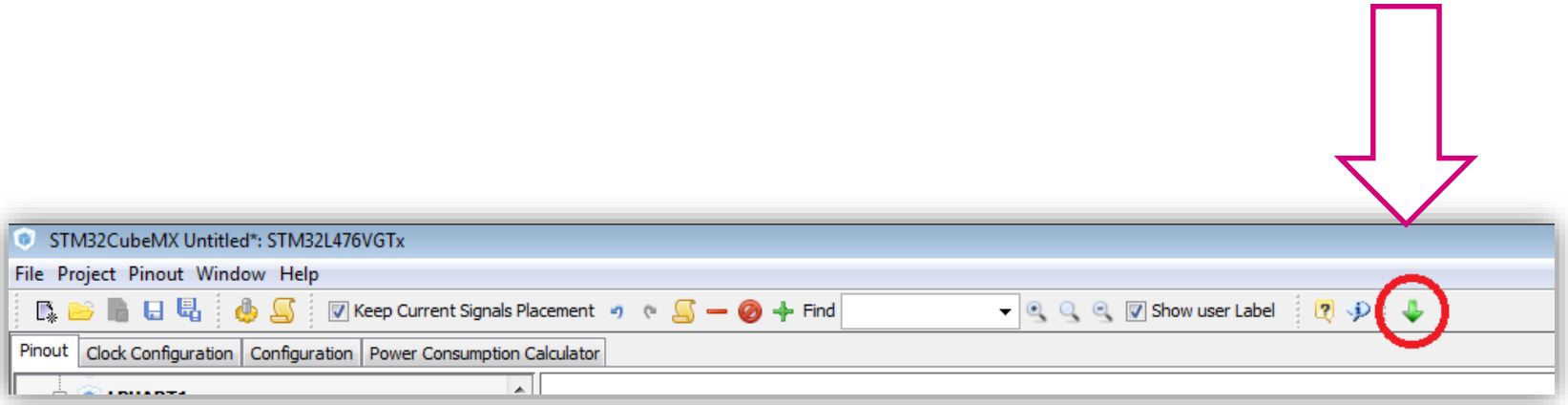


CUBE-MX

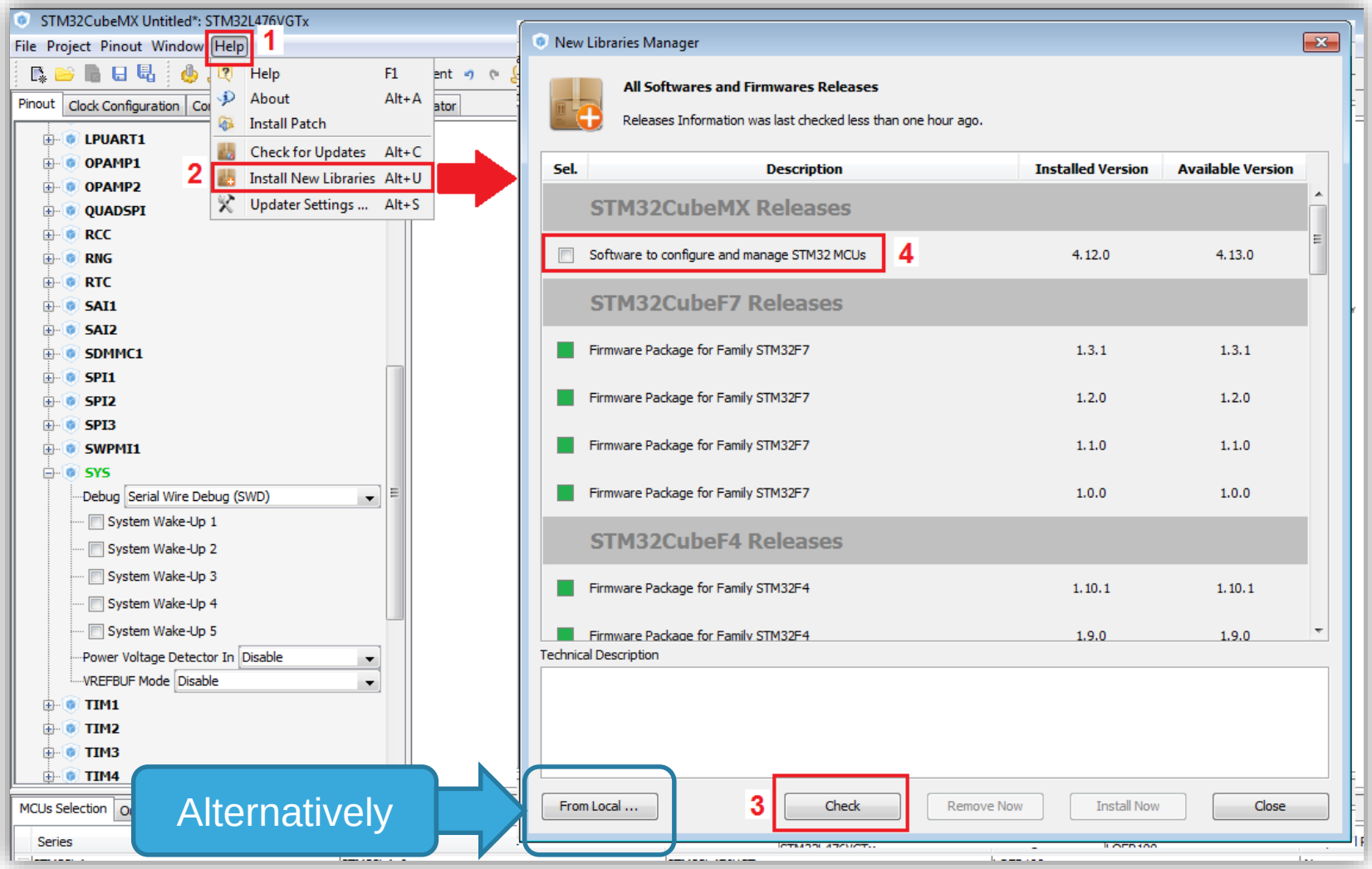


CUBE-MX - request update 1/2

The **green arrow** indicate that are presents some updates.

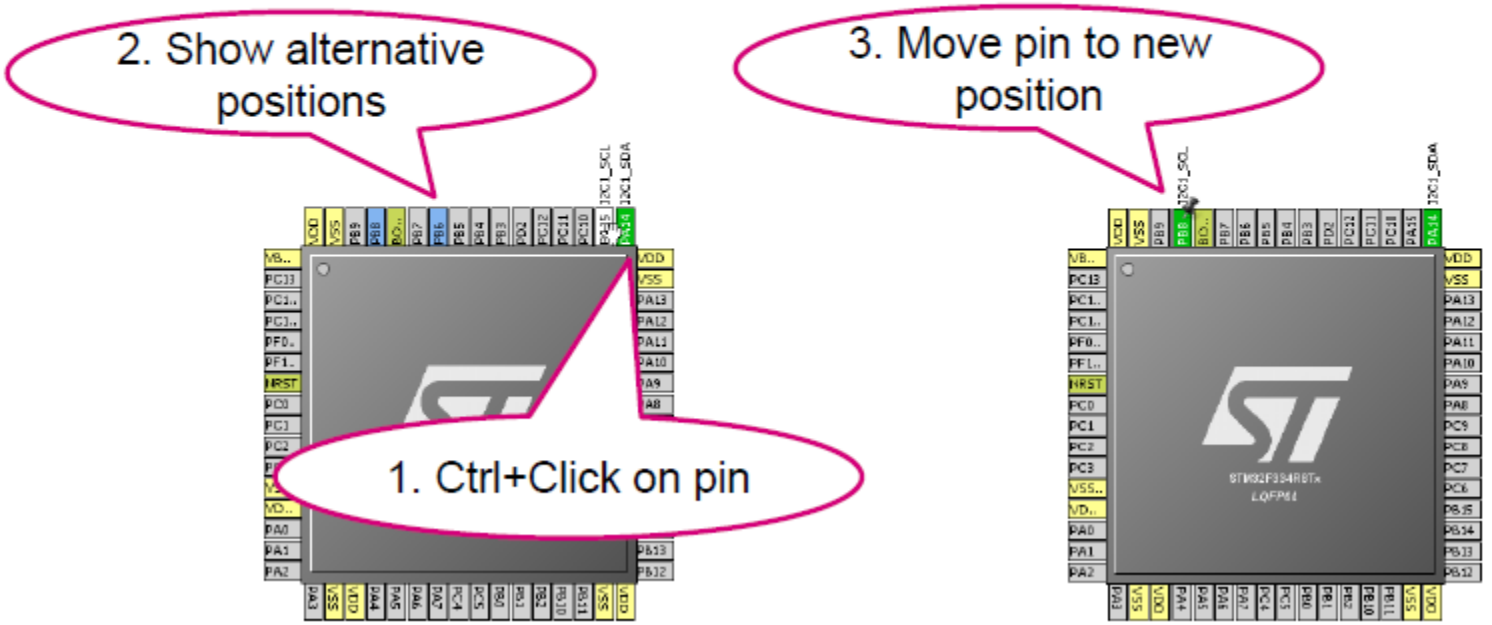


CUBE-MX - request update 2/2



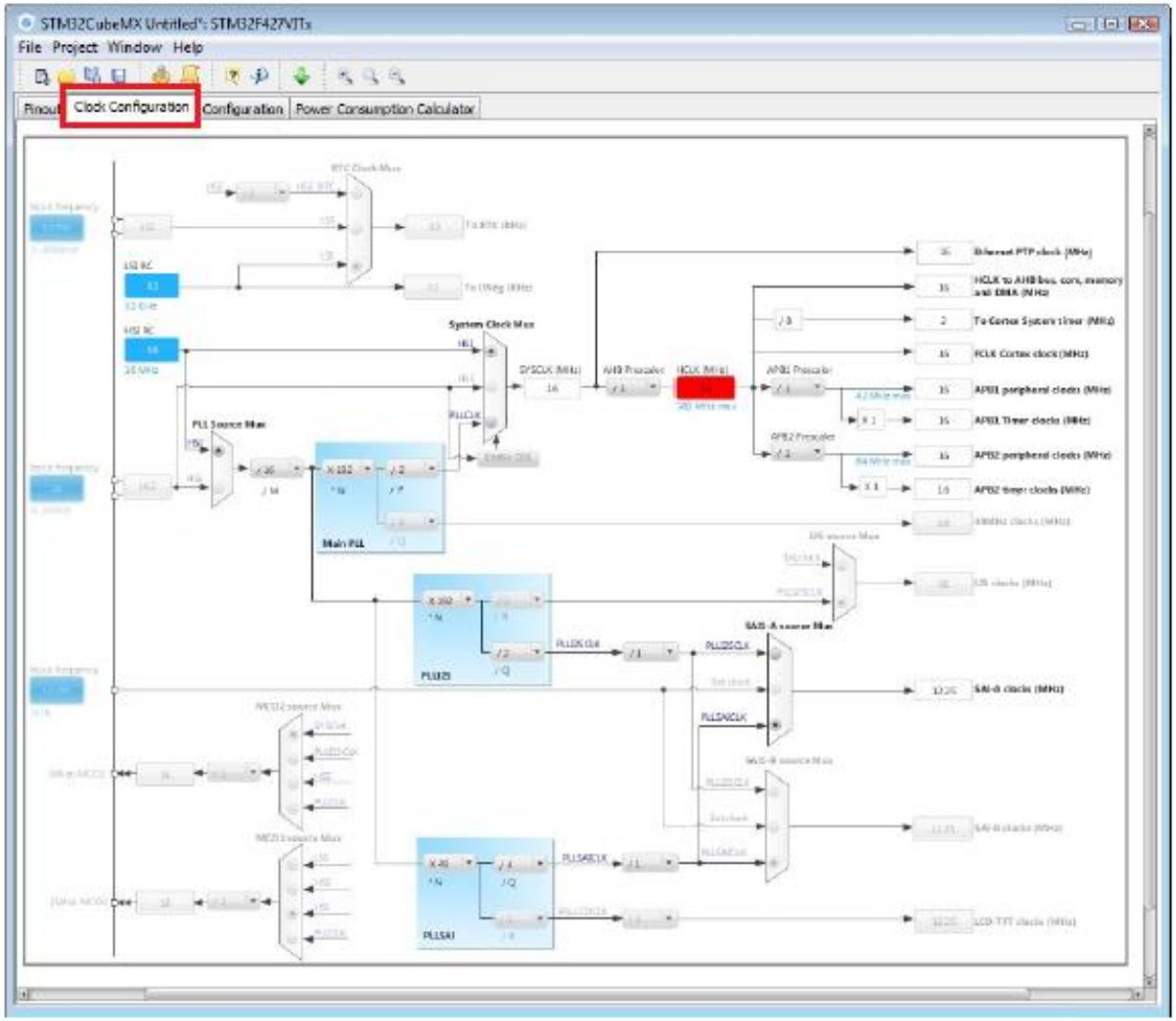
CUBE-MX - How to move the Pin Positions

- Signals can be set/moved directly from the pinout view
  - To see alternate pins for a signal Ctrl+Click on the signal, you can then drag and drop the signal to the new pin (keep pressing the Ctrl key)



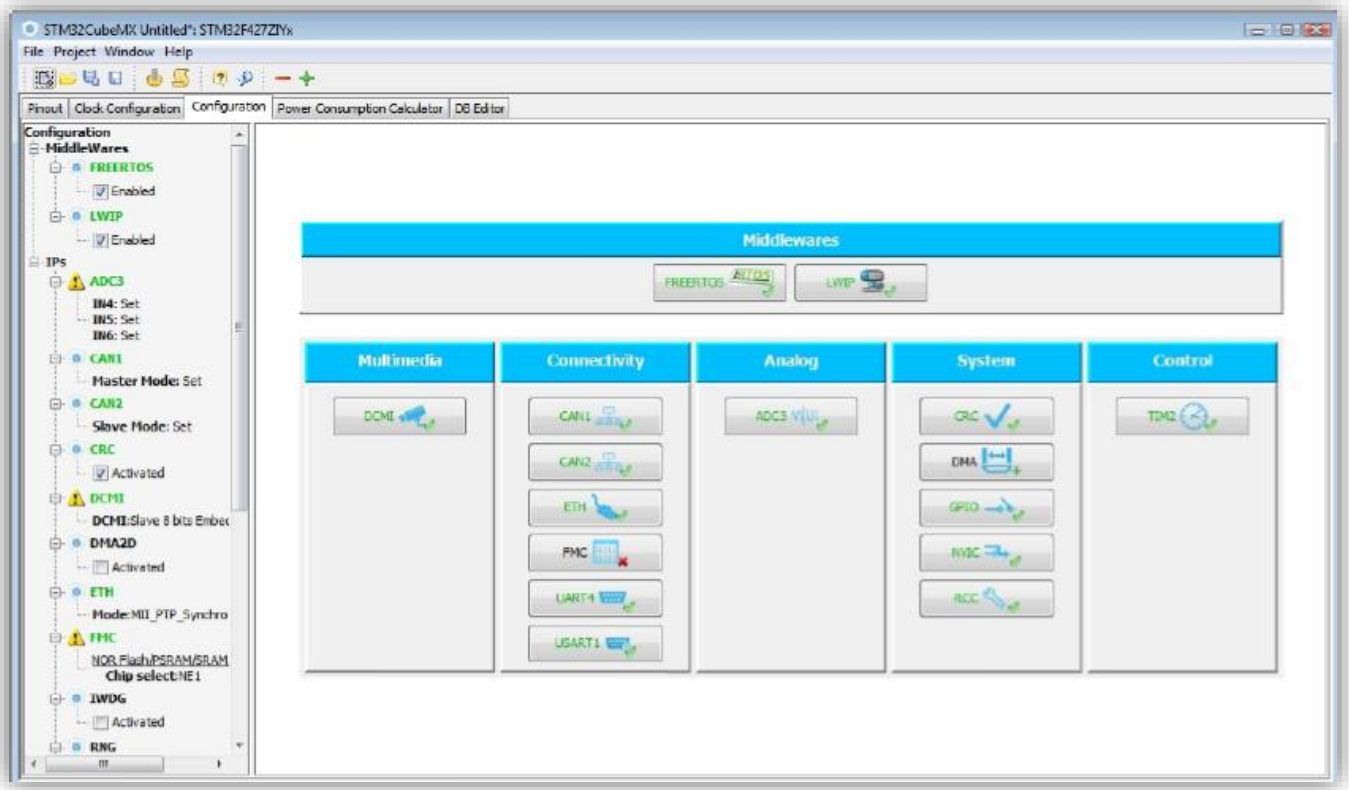
CUBE-MX - Clock tree

- Immediate display of all clock values
- Management of all clock constraints
- Highlight of errors



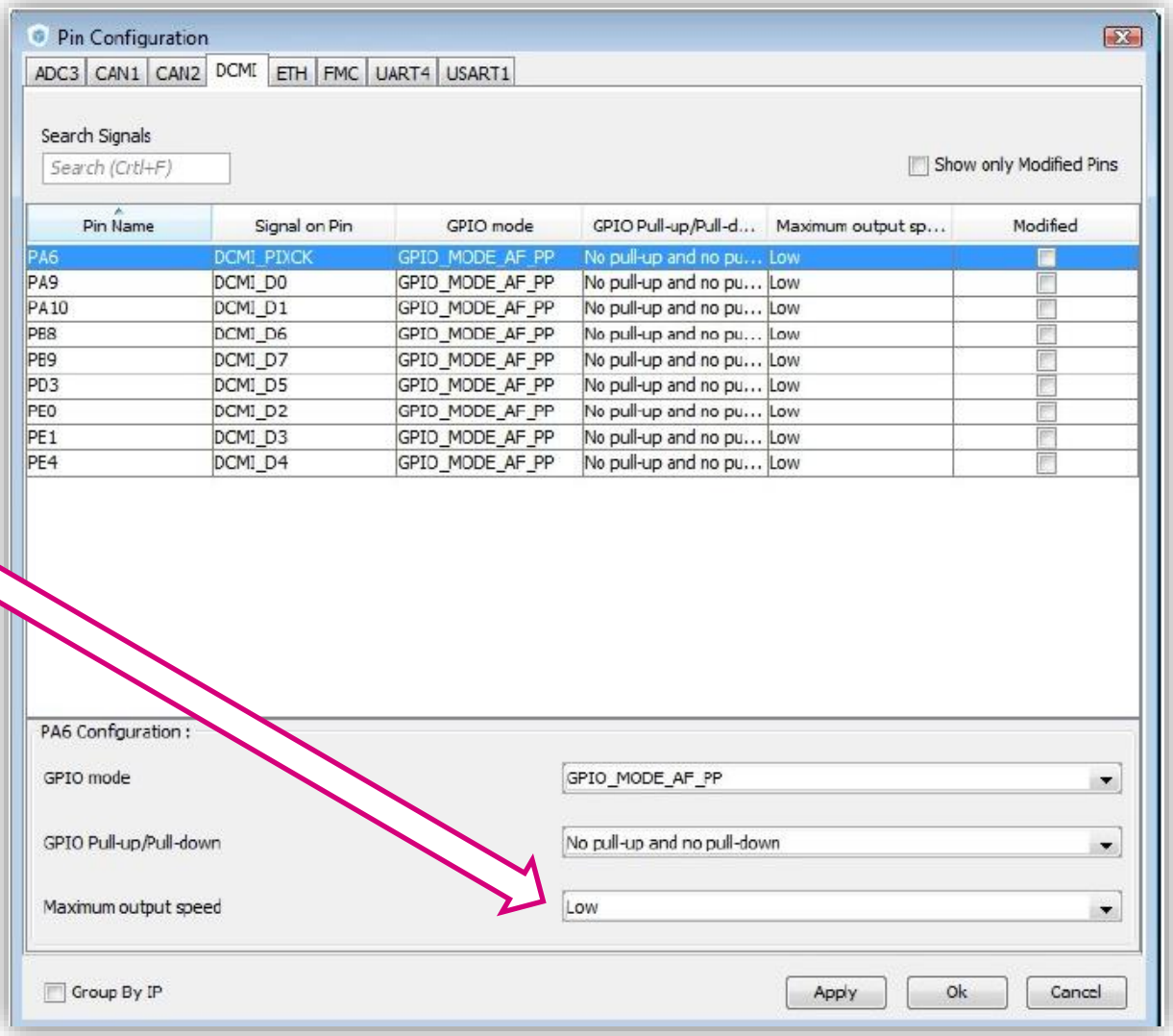
CUBE-MX - Peripheral and middleware configuration

- Global view of used peripherals and middleware
- Highlight of configuration errors
  - + Not configured
  - v OK
  - x Error
- Read only tree view on the left with access to IPs / Middleware having no impact on the pinout

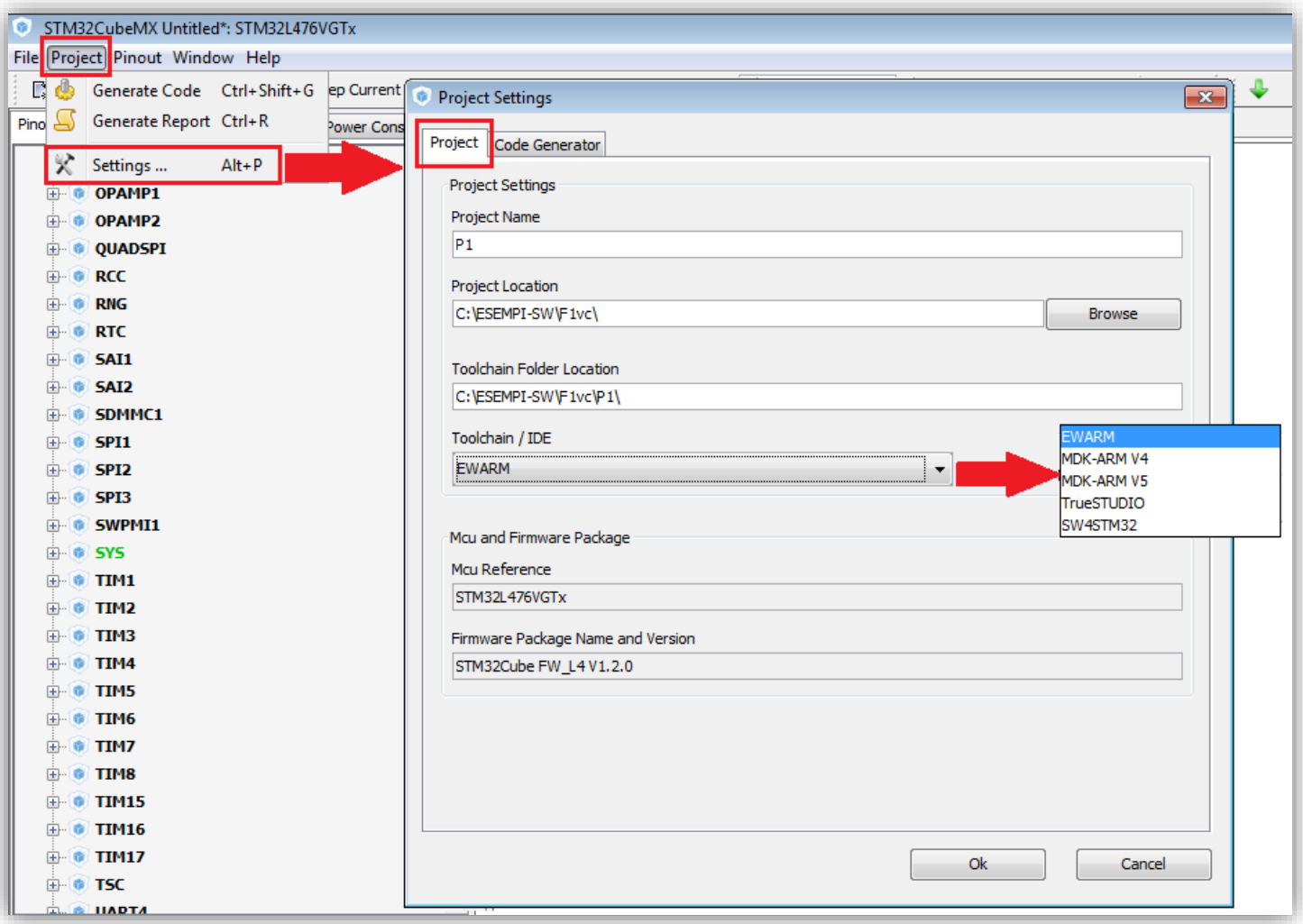


CUBE-MX - GPIO Panel

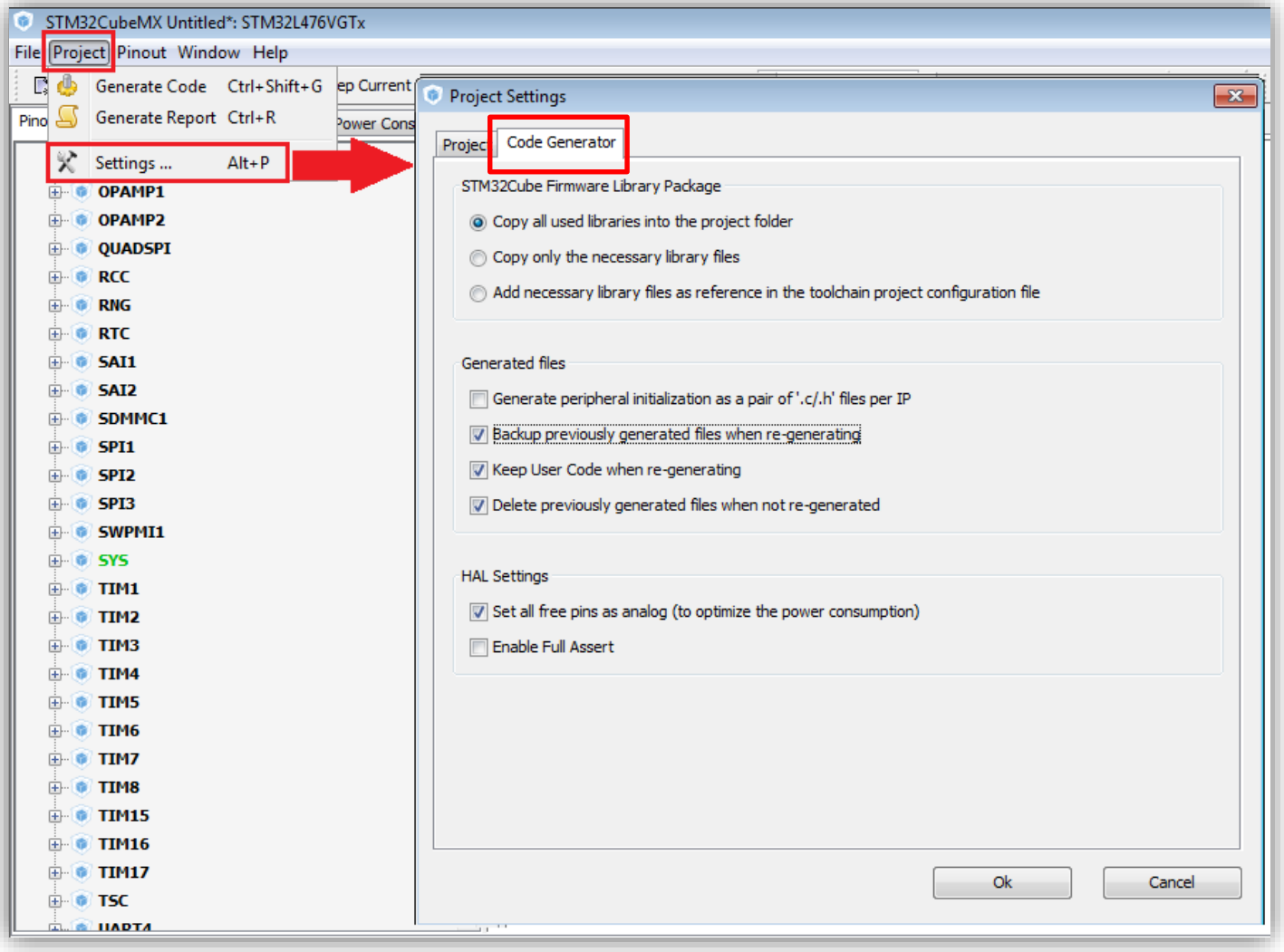
- Most of the GPIO parameters are set by default to the correct value
- You may want to change the maximum output speed
- You can select multiple pin at a time to set the same parameter



CUBE-MX - generate the code for some GUI 1/3

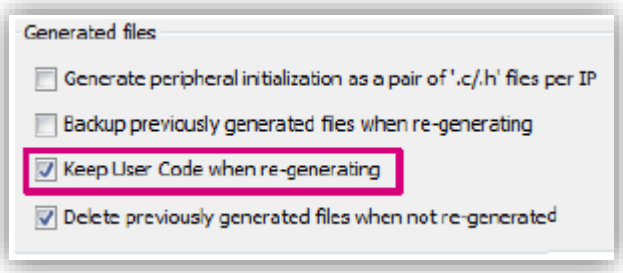


CUBE-MX - generate the code for some GUI 2/3



CUBE-MX - generate the code for some GUI 3/3

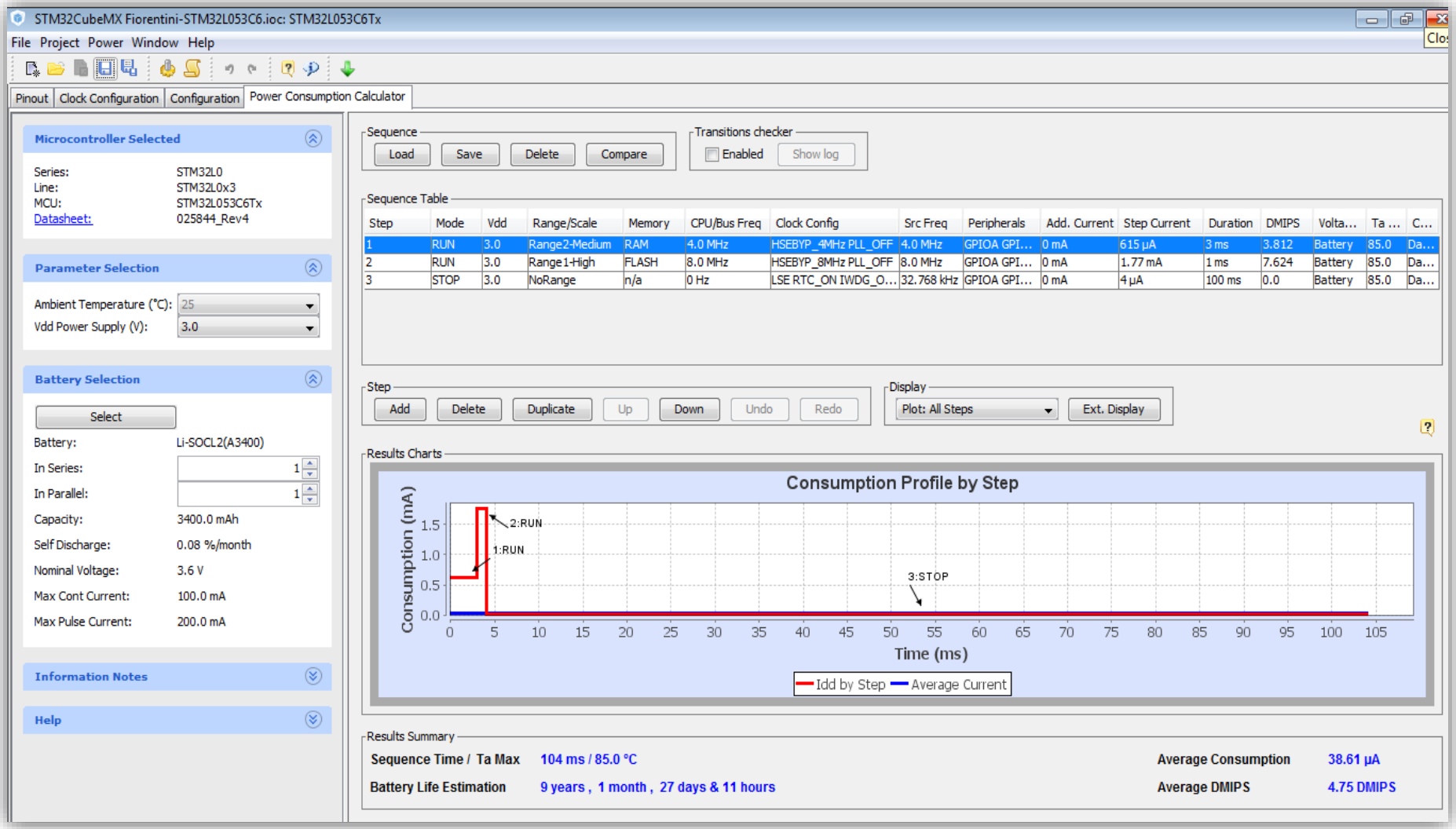
- Generation of all the C initialization code
- Automatic integration with partners toolchains
- User code can be added in dedicated sections and will be kept upon regeneration



main.c

```
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 interfa
51    HAL_Init();
52    /* Configure the system clock */
```

CUBE-MX - Power consumption calculator



life.augmented

STM32 Cube

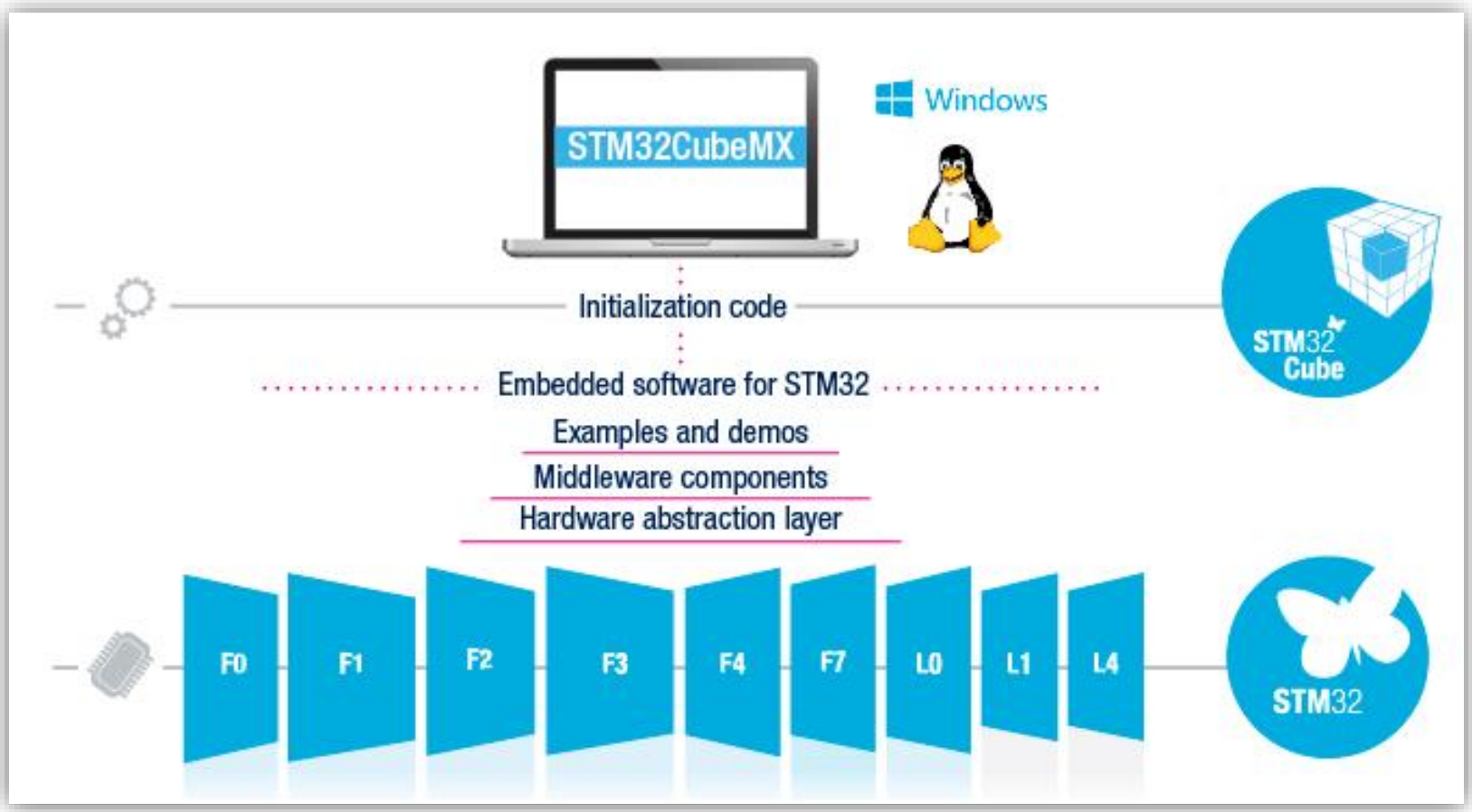
AVNET Memec

SILICA An Avnet Company

WWW.EMCU.IT

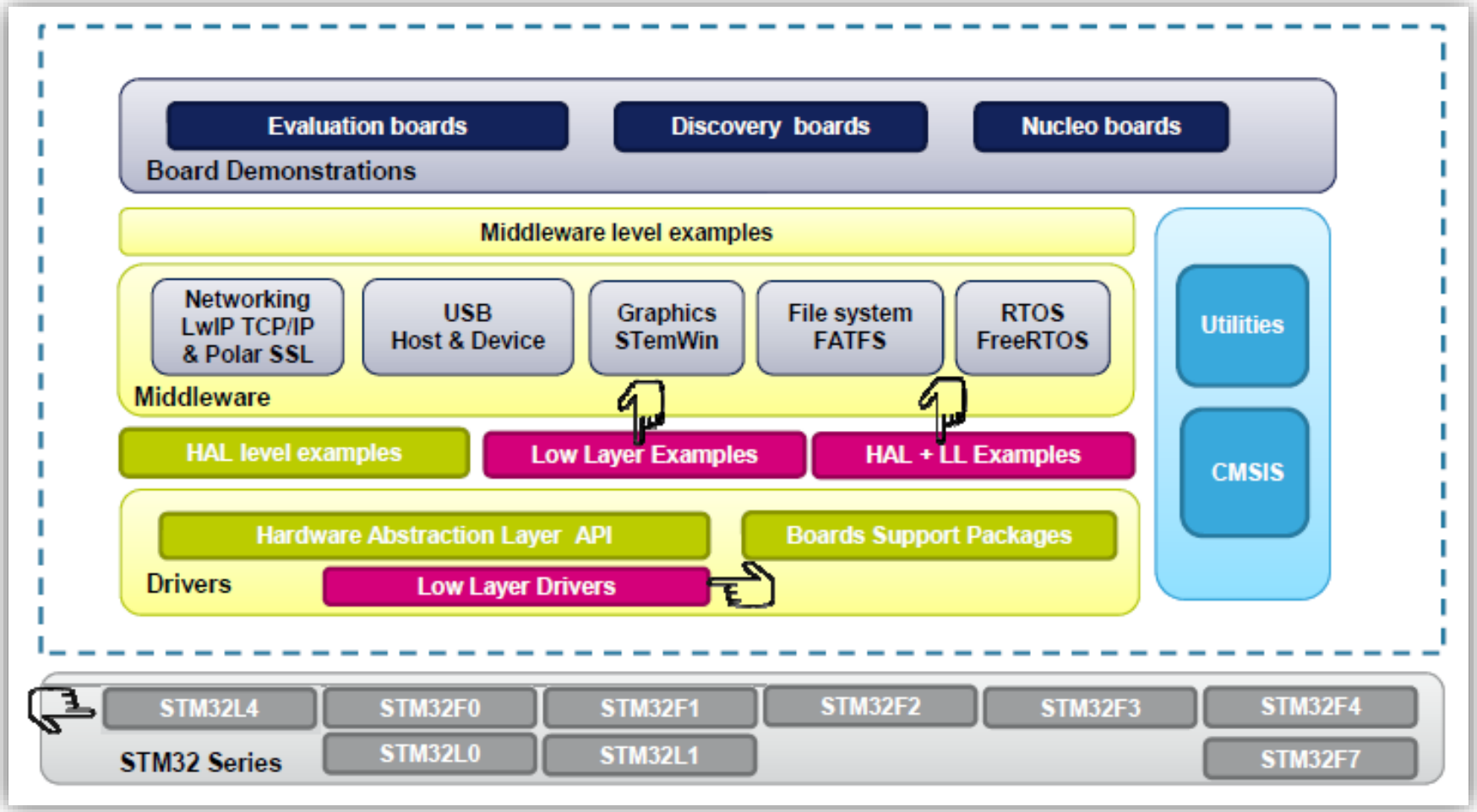
HAL library

HAL library == hardware abstraction layer



HAL library

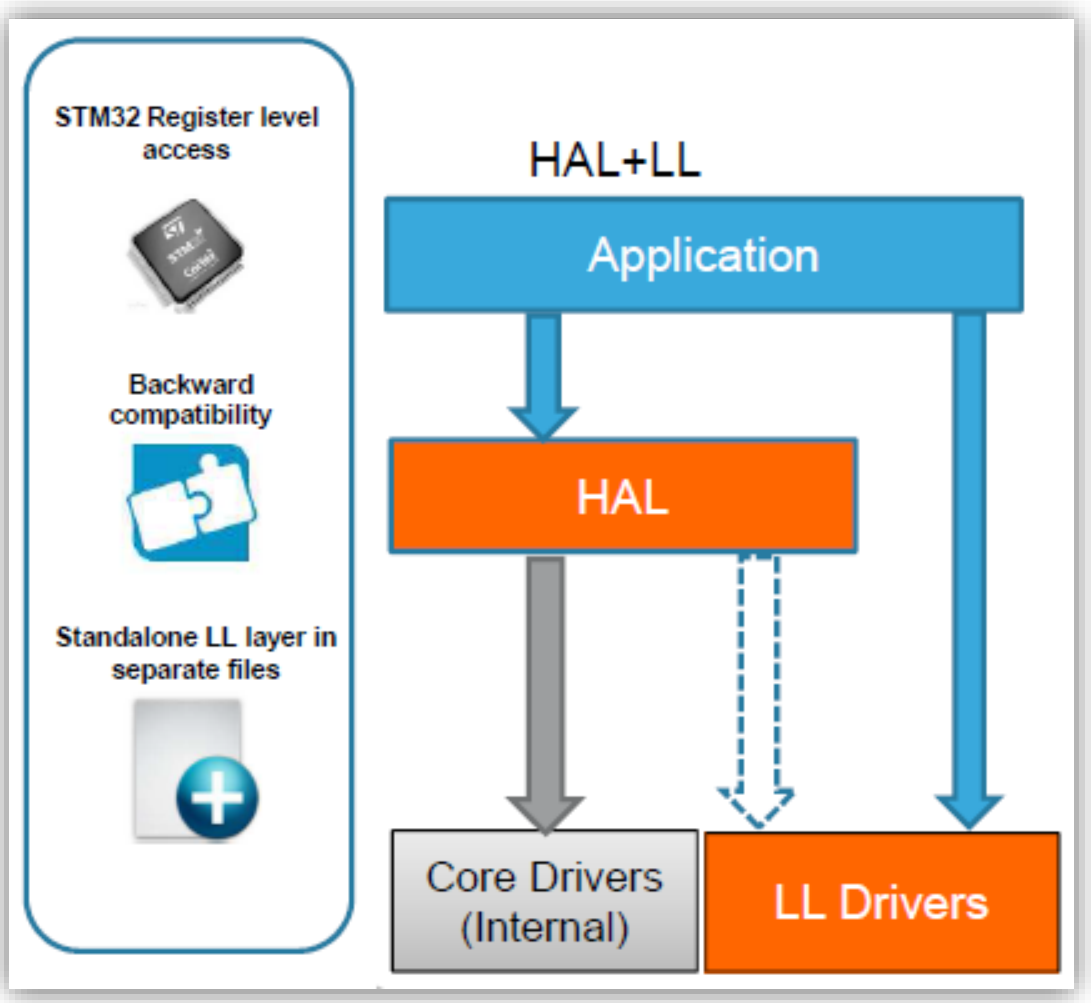
The HAL library are **here**.



HAL & LL library

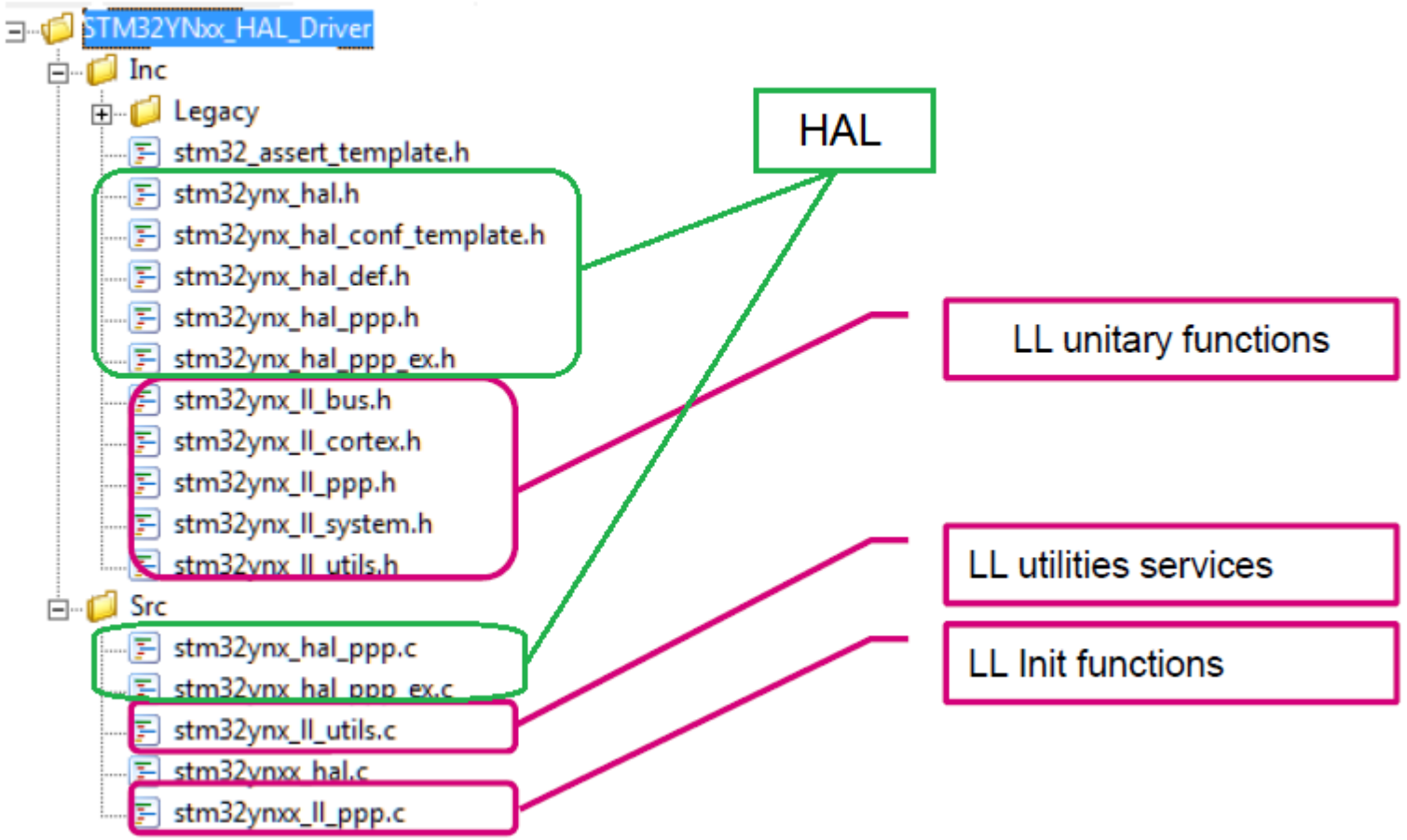
- **STM32Cube HAL & LL are complementary** and covers a wide range of applications requirements:
  - **HAL** offers **high level** and **functionalities oriented APIs**, with **high portability level** and **hide product/IPs complexity to end user**
  - **LL** offers **low level APIs at registers level,** w/ **better optimization** but **less portability** and require **deep knowledge of the product/IPs specification**
- The **new Low Layer (LL)** is offering the following services:
  - **Unitary static inline functions for direct register access** (provided in *.h files)
    - One-shot operations that can be used by the HAL drivers or from application level.
    - Independant from HAL and can be used in standalone usage (without HAL drivers)
    - Full features coverage of the supported IP
  - **Init functions** (provided in *.c files)
    - compatible with Standard peripheral library

HAL & LL library



HAL & LL library

LL drivers are located in the Src/Inc HAL Driver folders



HAL vs LL library

Covered peripherals (1/2)

Peripherals (IPs)		STM32Cube Support	
		HAL	LL
System	Flash	Yes	No (some of the Flash features need to be handled in the MISC file to prevent dependency with HAL when using LL PWR driver)
	EXTI	Yes	Yes
	GPIO	Yes	Yes
	DMAs	Yes	Yes
	PWR	Yes	Yes
	RCC	Yes	Yes
	Cortex	Yes	No (some of the cortex features added: MPU, SYSTICK, CPUID, SLEELDEEP)
	SYSCFG	Yes	Yes
Analog	ADC	Yes	Yes
	SDADC	Yes	Yes
	DAC	Yes	Yes
	COMP	Yes	Yes
	DFSDM	Yes	No
	OPAMP	Yes	Yes
Timers	RTC	Yes	Yes
	TIM	Yes	Yes
	LPTIM	Yes	Yes
	HRTIM	Yes	Yes
	WWDG	Yes	Yes
	IWDG	Yes	Yes
Cryptography	CRC	Yes	Yes
	CRYP	Yes	No
	HASH	Yes	No
	RNG	Yes	Yes

HAL vs LL library

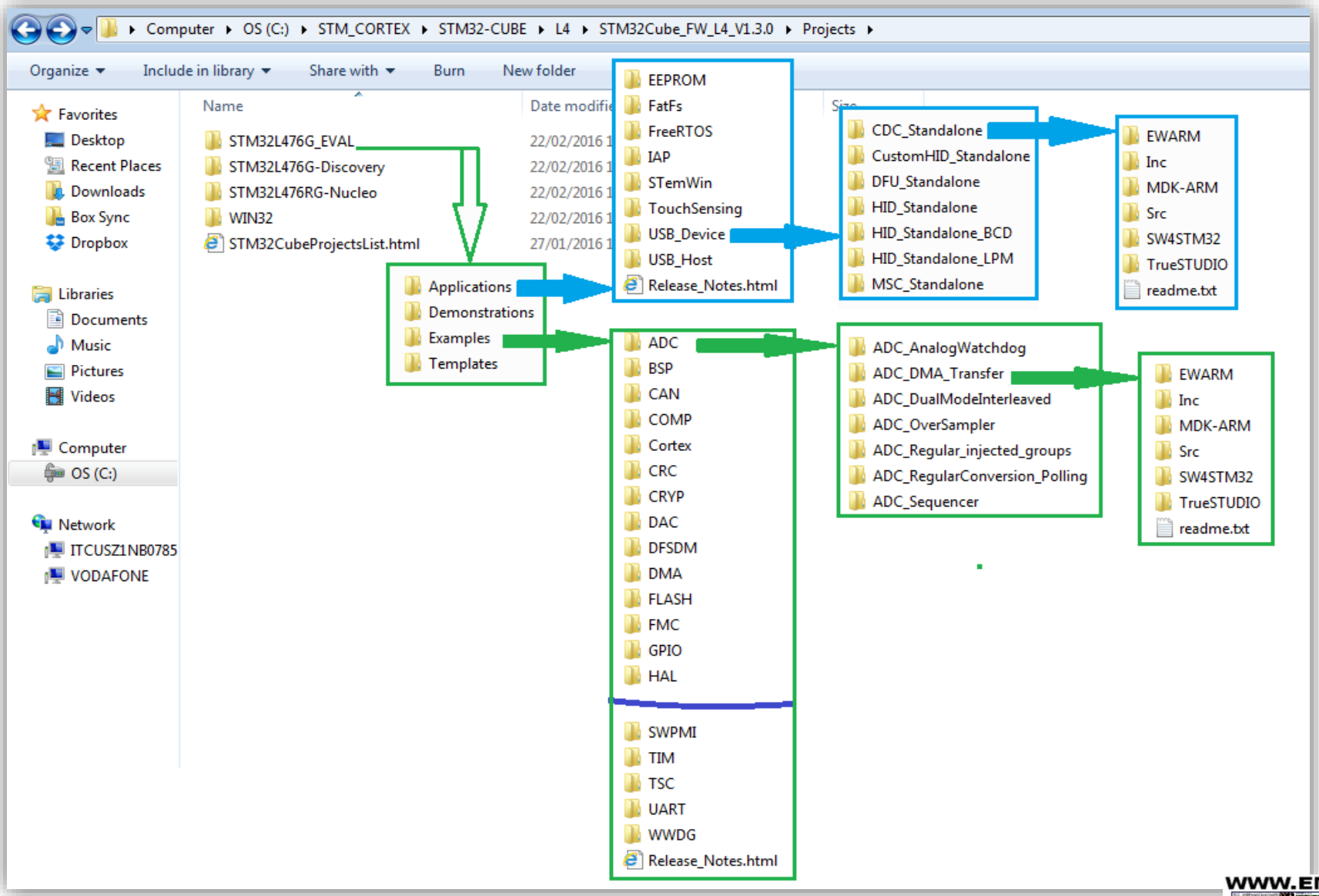
Covered peripherals (2/2)

Peripherals (IPs)		STM32Cube Support	
		HAL	LL
Basic Connectivity	I2C/SMBUS	Yes	Yes
	UART/USART/LPUART	Yes	Yes
	SWPMI	Yes	Yes
	SPI/I2S	Yes	Yes
	SDMMC(SDIO)	Yes	No
	CAN	Yes	No
	CEC	Yes	No
Advanced Connectivity	USB-FS-Device	Yes	No
	USB-OTG-FS/HS	Yes	No
	Ethernet	Yes	No
	MDIOS	Yes	No
Interface	FSMC(FMC)	Yes	No
	LCD"Glass"	Yes	No
	LTDC	Yes	No
	DSI	Yes	No
	DMA2D	Yes	Yes
	JPEG	Yes	No
	DCMI	Yes	No
	QSPI	Yes	No
	SPDIF-IN	Yes	No
	SAI	Yes	No

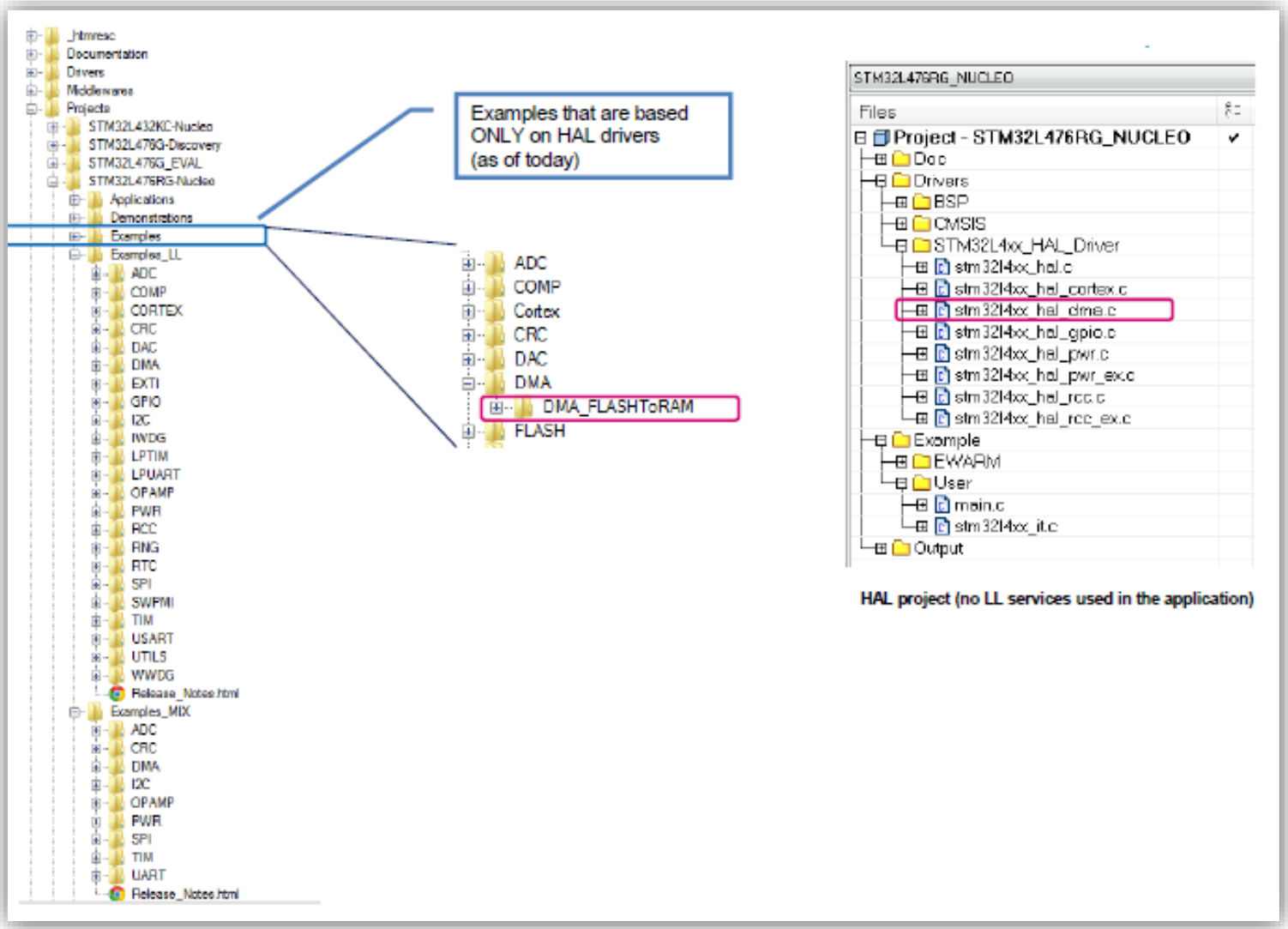
HAL vs. LL usage

- To cohabitate the HAL with the LL, user has to be aware about some HAL concepts.
- Main constraint is when the LL overwrites some registers that the content is mirorred in the HAL handles.
- The Low Layer drivers cannot be automatically used with the HAL for the same peripheral instance: mainly can't run concurrent process on the same IP using both APIs, however sequential use is allowed.
- The low layer drivers can be used without any constraint with all the HAL drivers that are not based on handle objects (RCC, Cortex, common HAL, flash and GPIO)
- The LL is intented to be used in expert mode (high knowledge on STM32 hardware aspect)

HAL library - Where to find examples ready to use ?

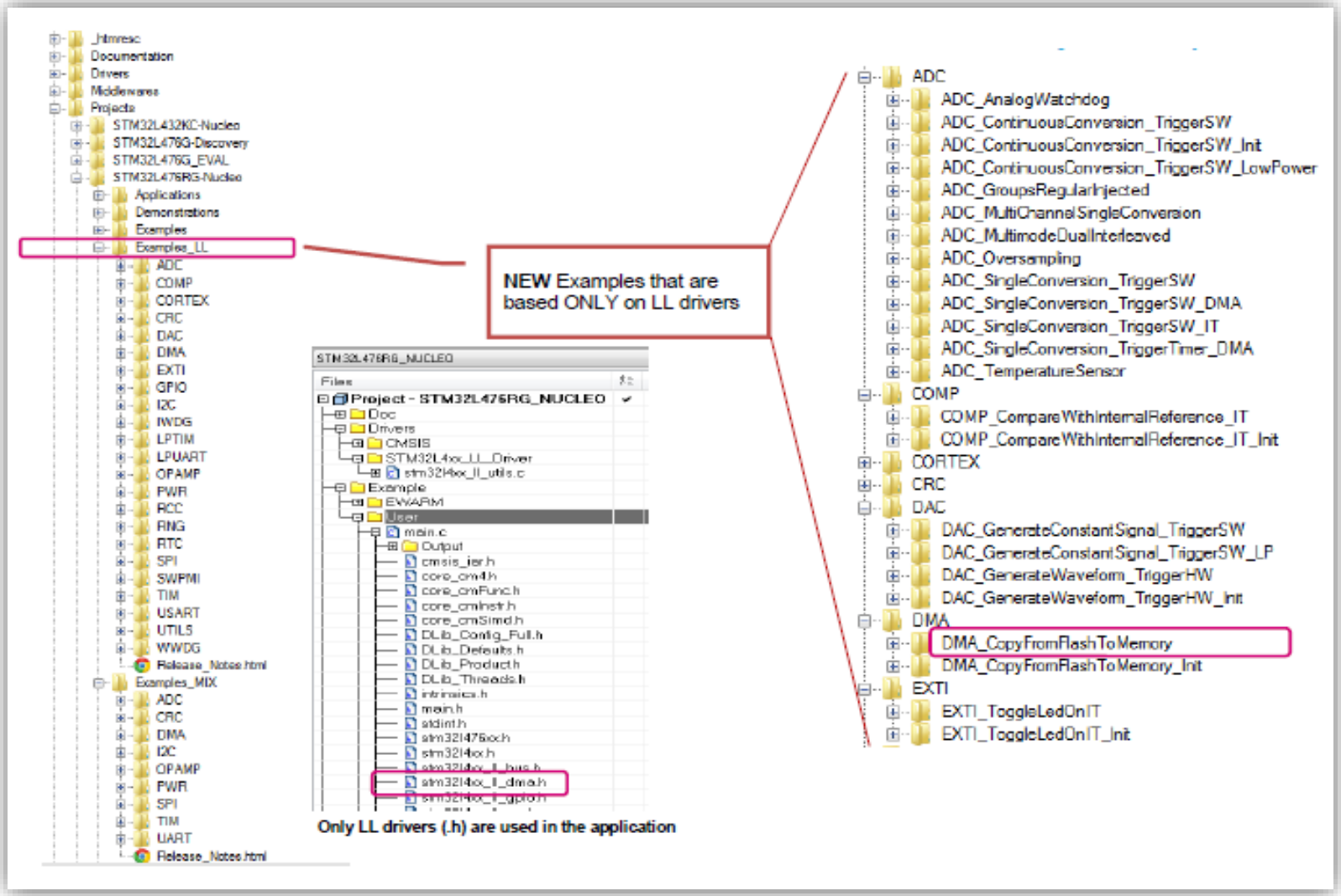


HAL library - HAL examples

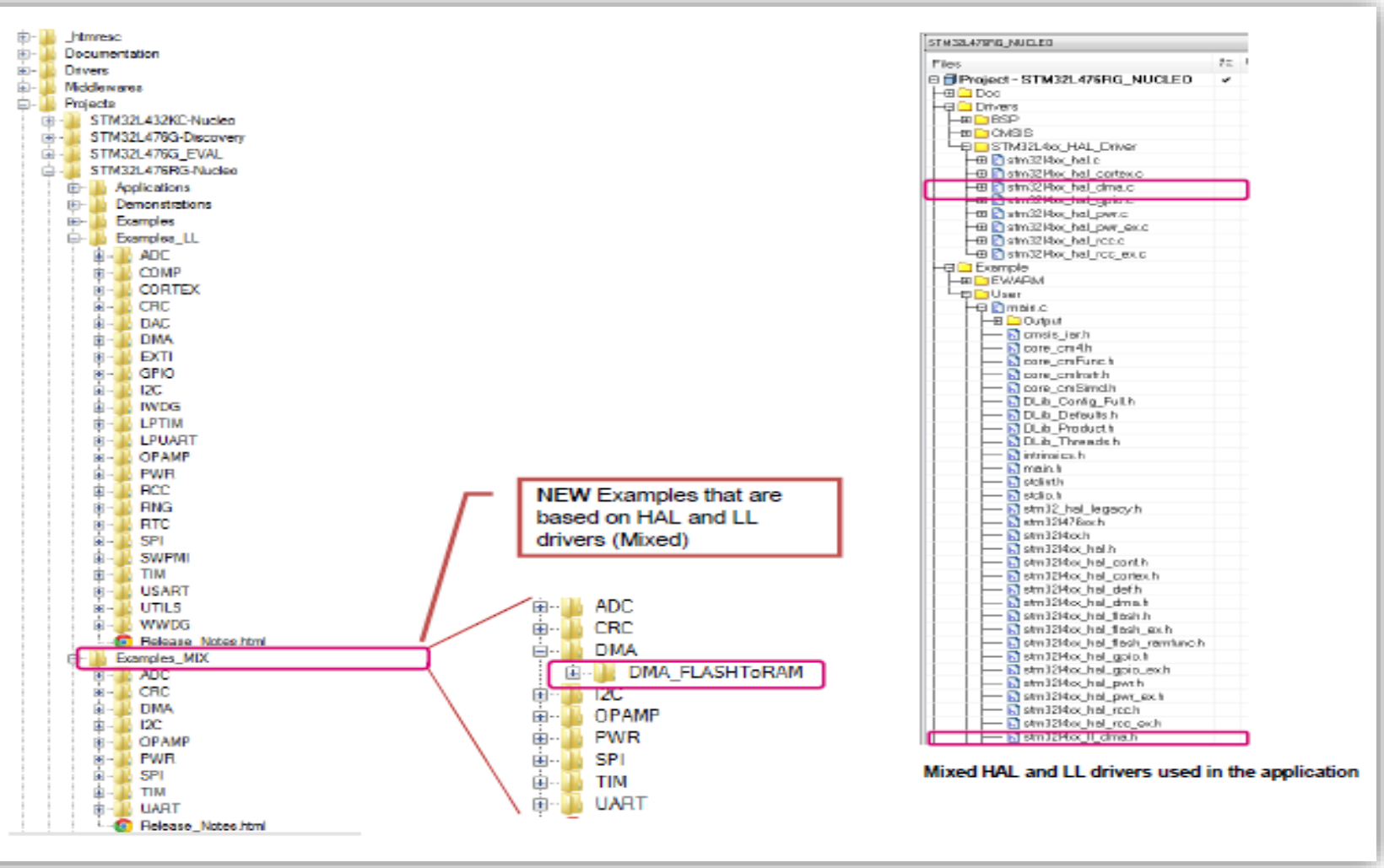


HAL project (no LL services used in the application)

HAL library - LL examples



HAL library - LL & HAL mix Example

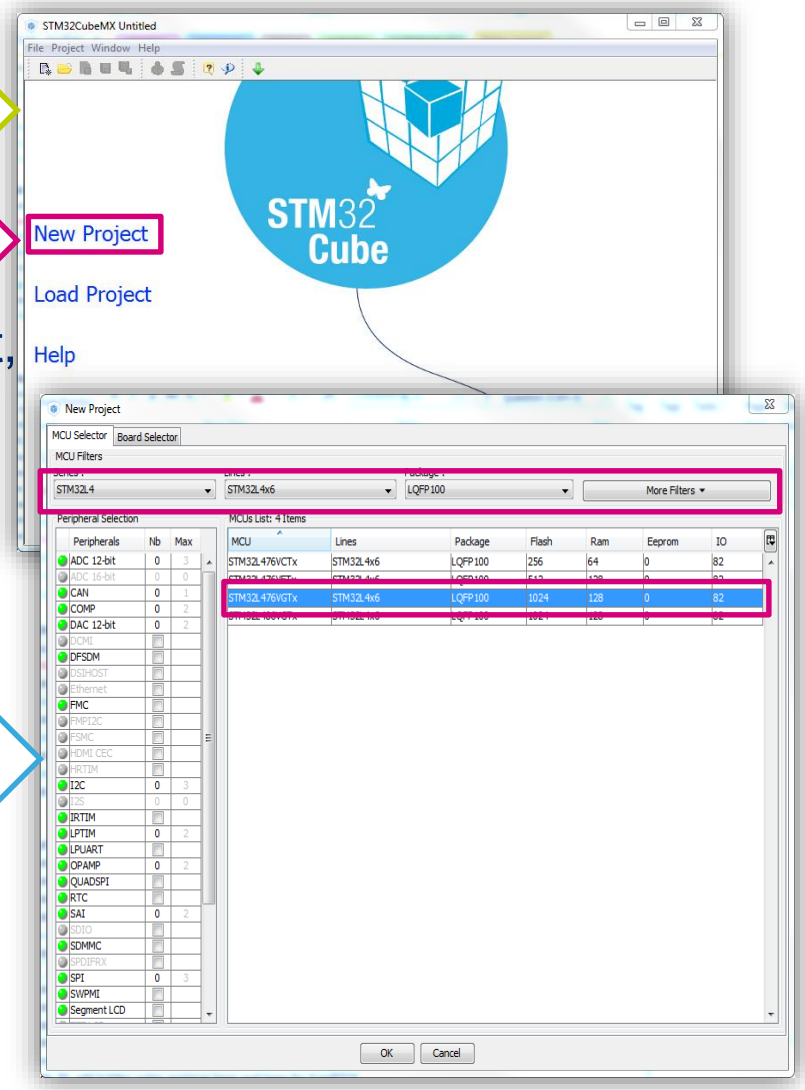


Mixed HAL and LL drivers used in the application

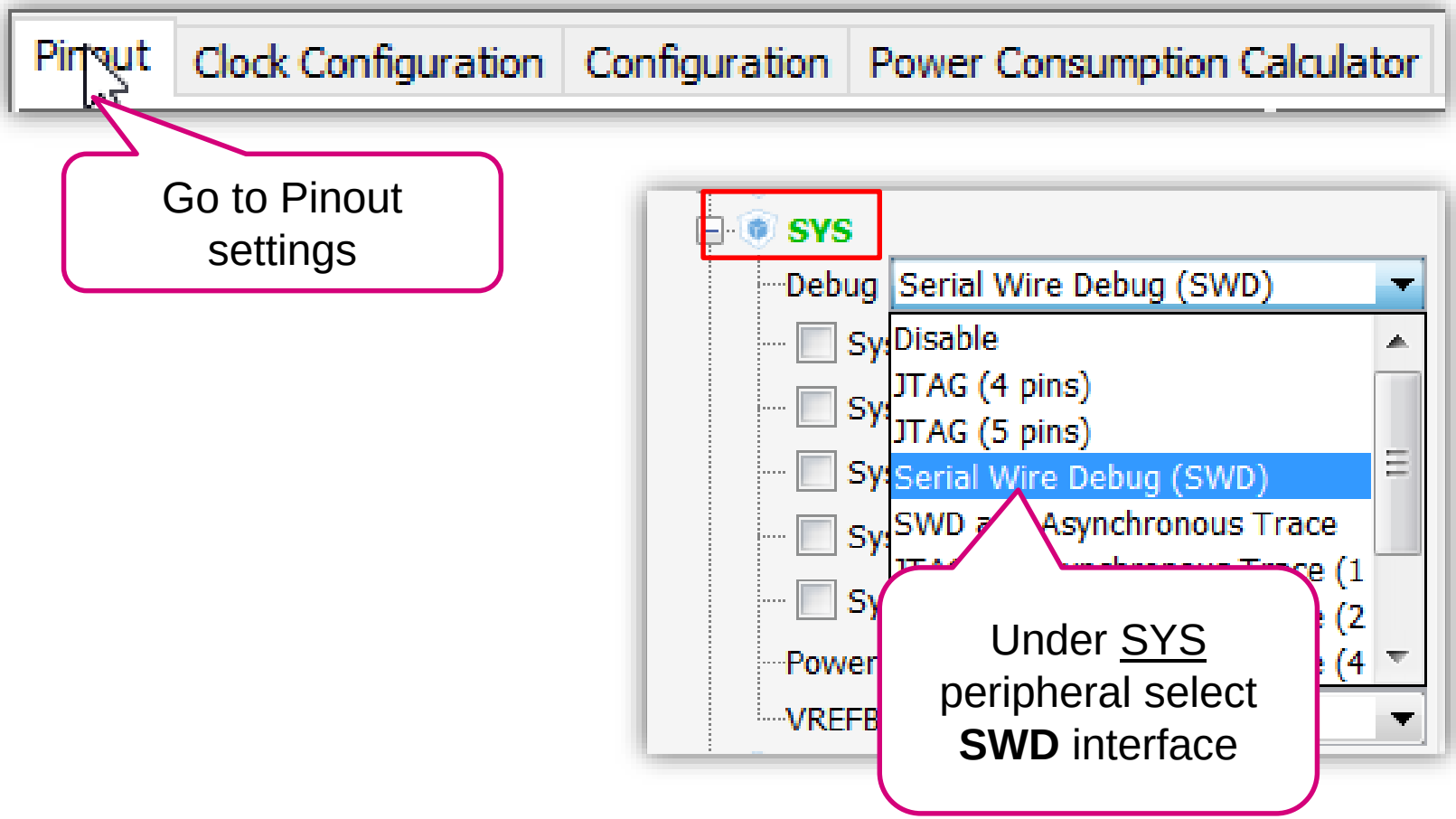
New Project

CUBE - Start new project

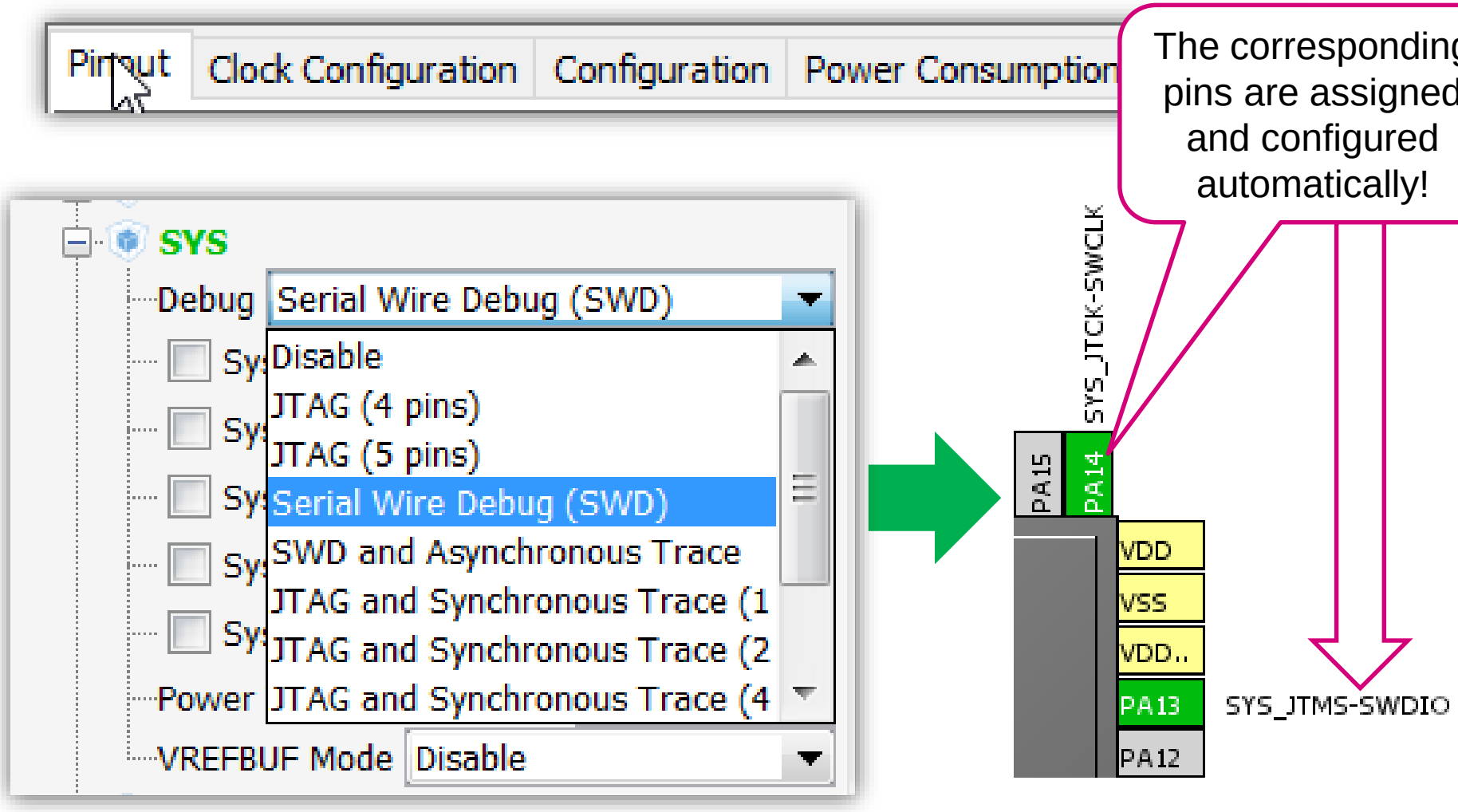
- Run CubeMX tool
- Start **new project**
  - Click “New Project” desktop shortcut, or
  - Go to “Menu->File->New Project”
- Filter:
  - Series: STM32L4
  - Line: STM32L4x6
  - Package: LQFP100
- Select: **STM32L476VGTx** (**STM32L476-DISCOVERY**)



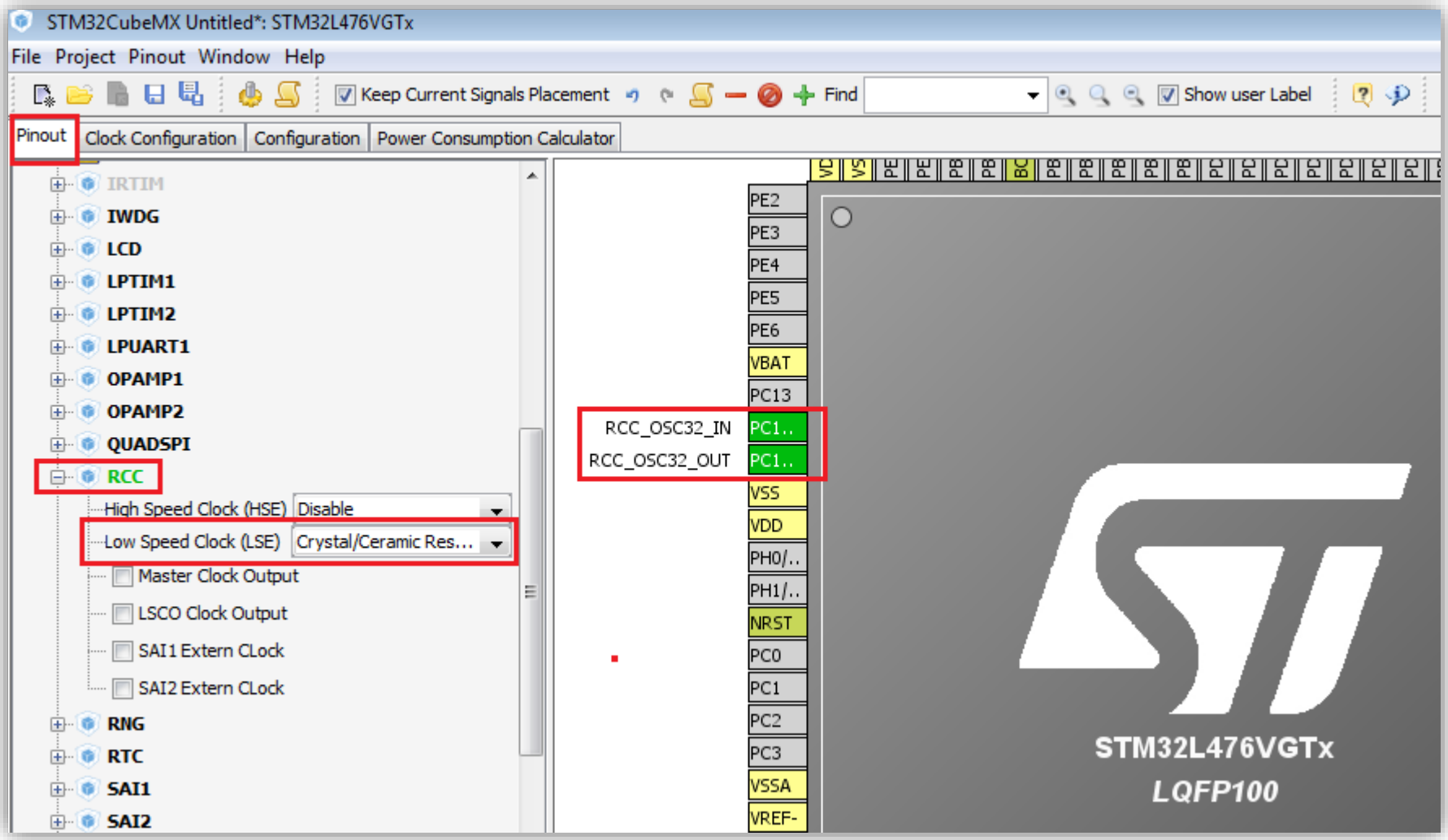
CUBE - Start new project
Configure debug interface 1/2



CUBE - Start new project
Configure debug interface 2/2

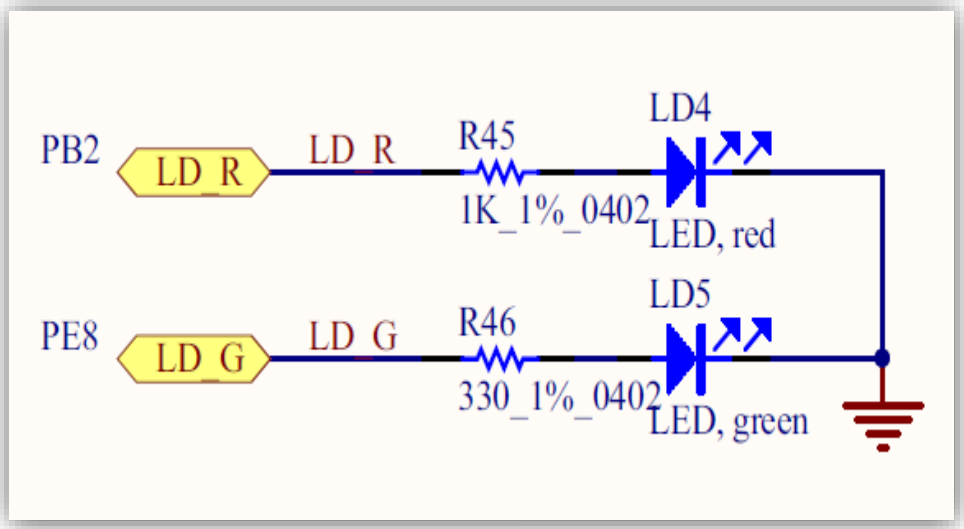


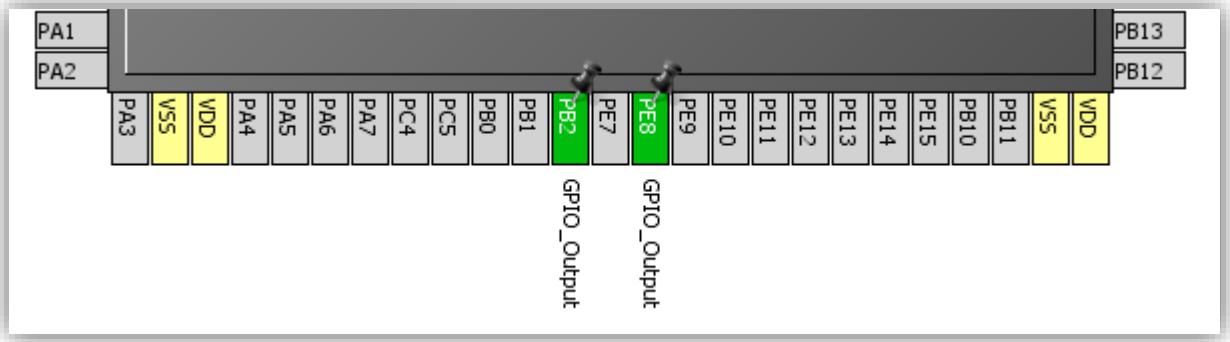
CUBE - Start new project
Configure LSE resonator (32,768 KHz)



CUBE - Start new project
Configure GPIO for LED toggling

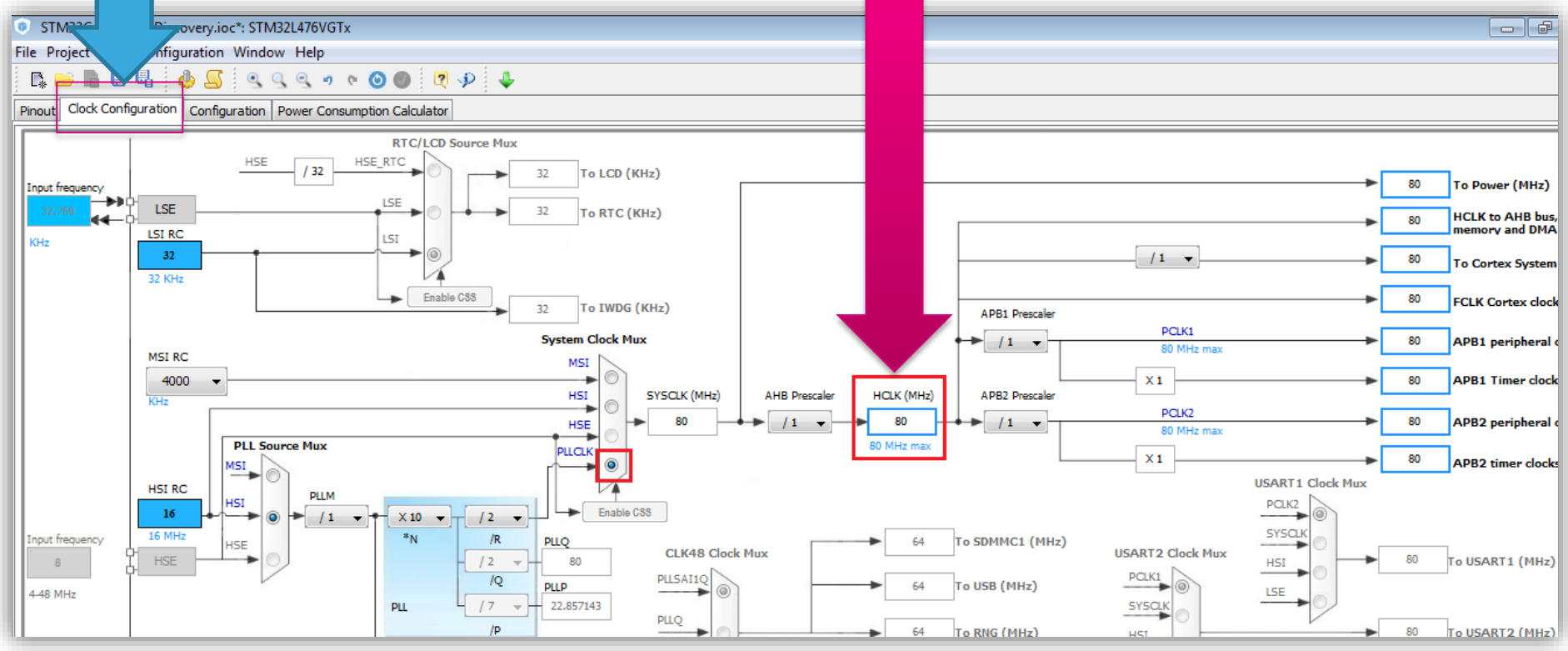
Configure LED pin as GPIO_Output



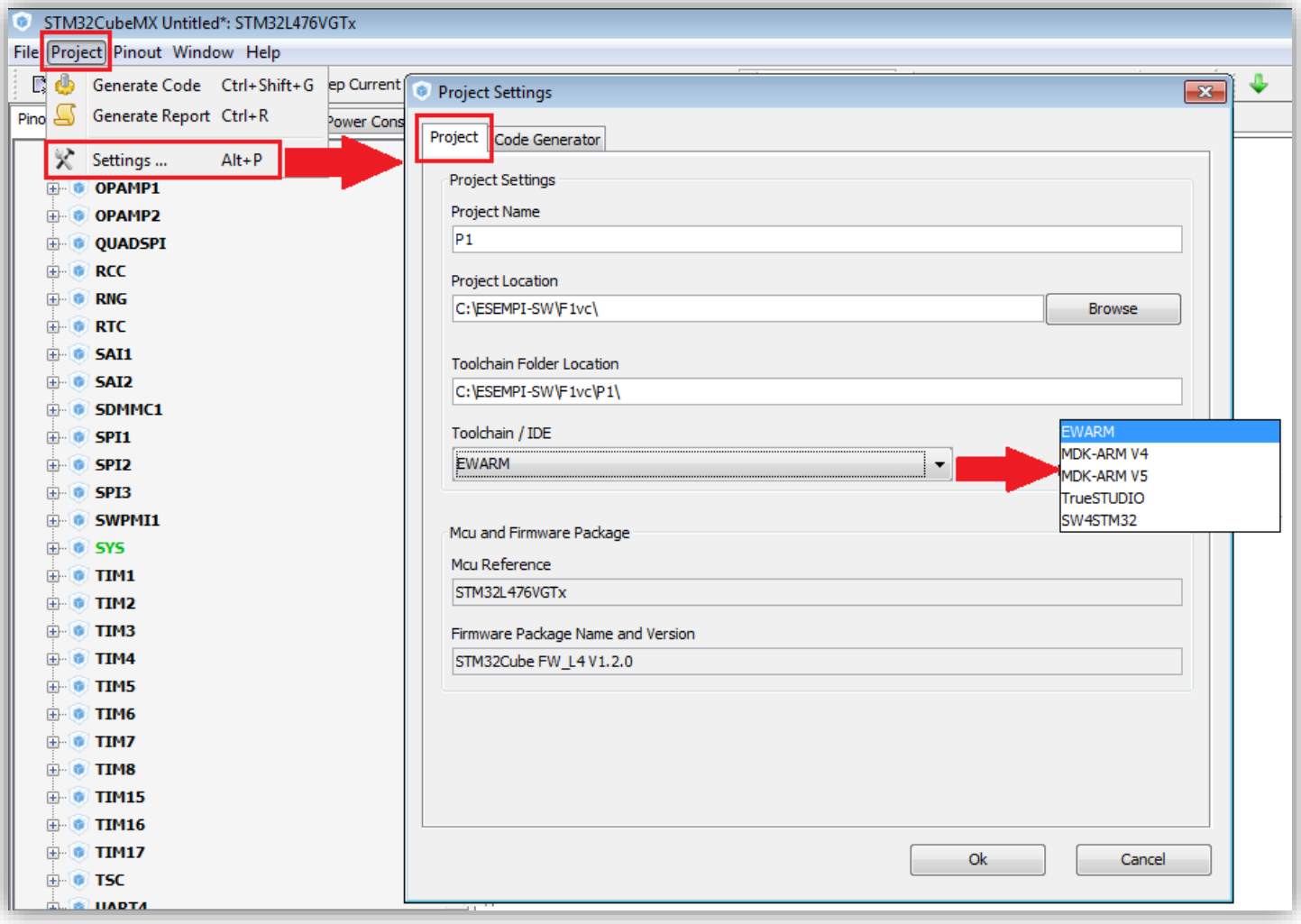


CUBE - Start new project
Clock configuration

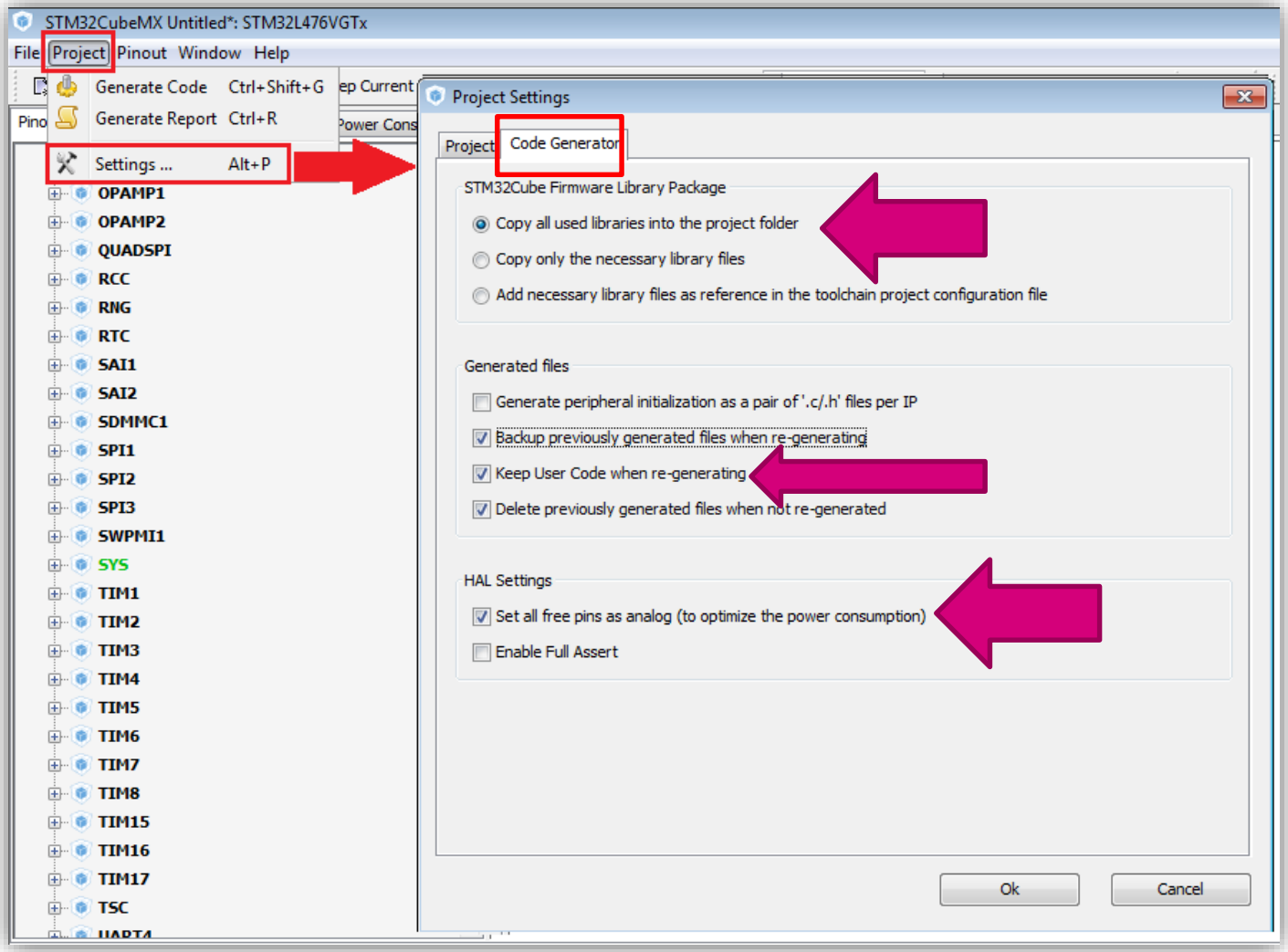
Write 80



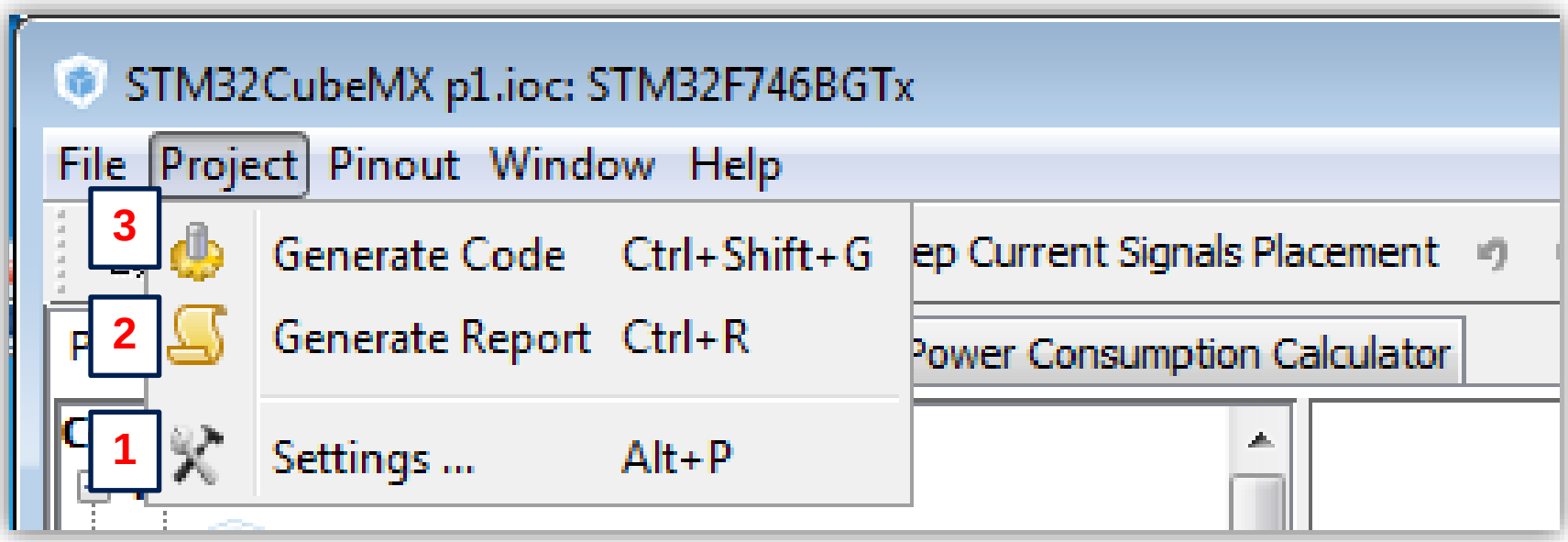
CUBE - Start new project generate the code for some GUI 1/3



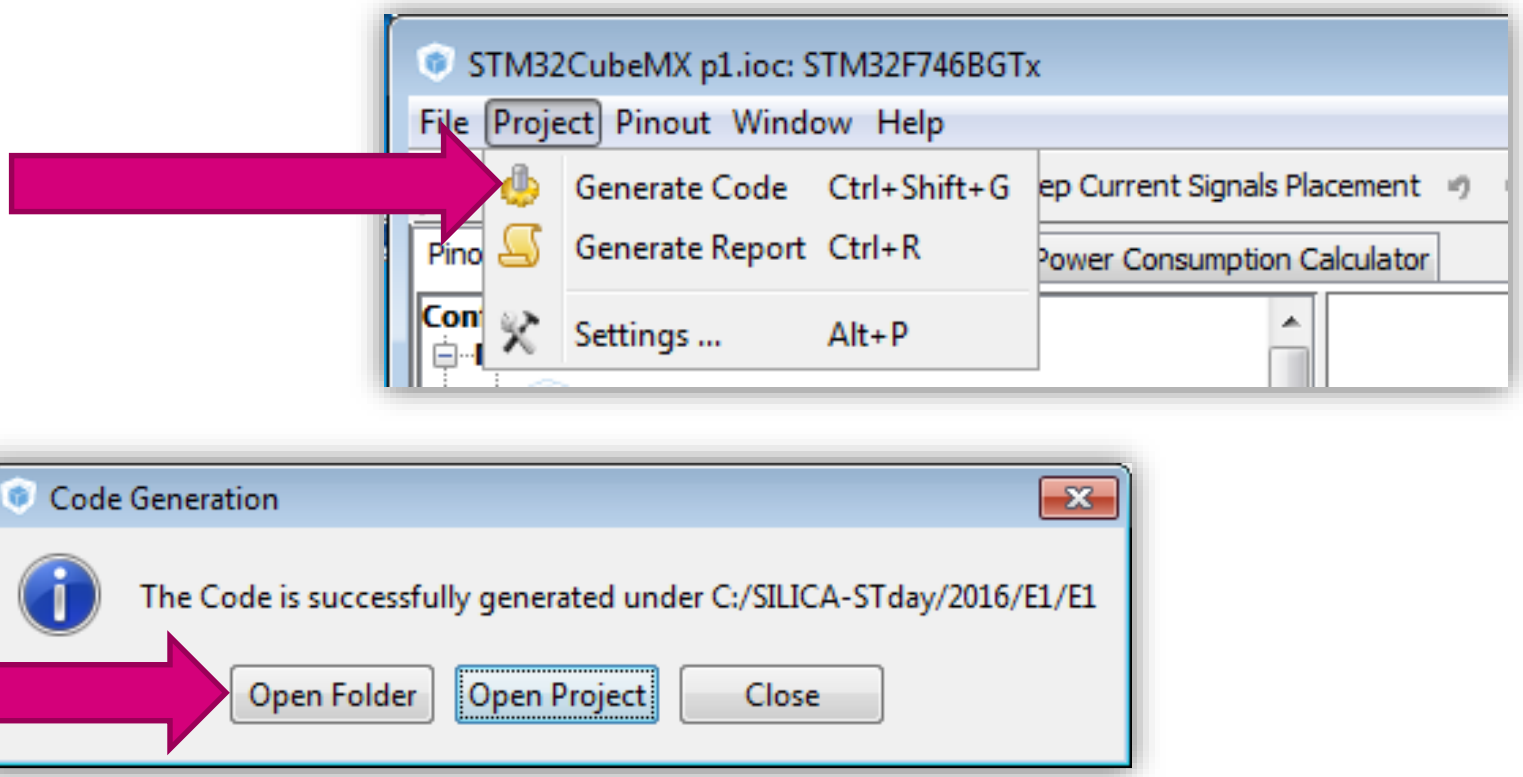
CUBE - Start new project generate the code for some GUI 2/3



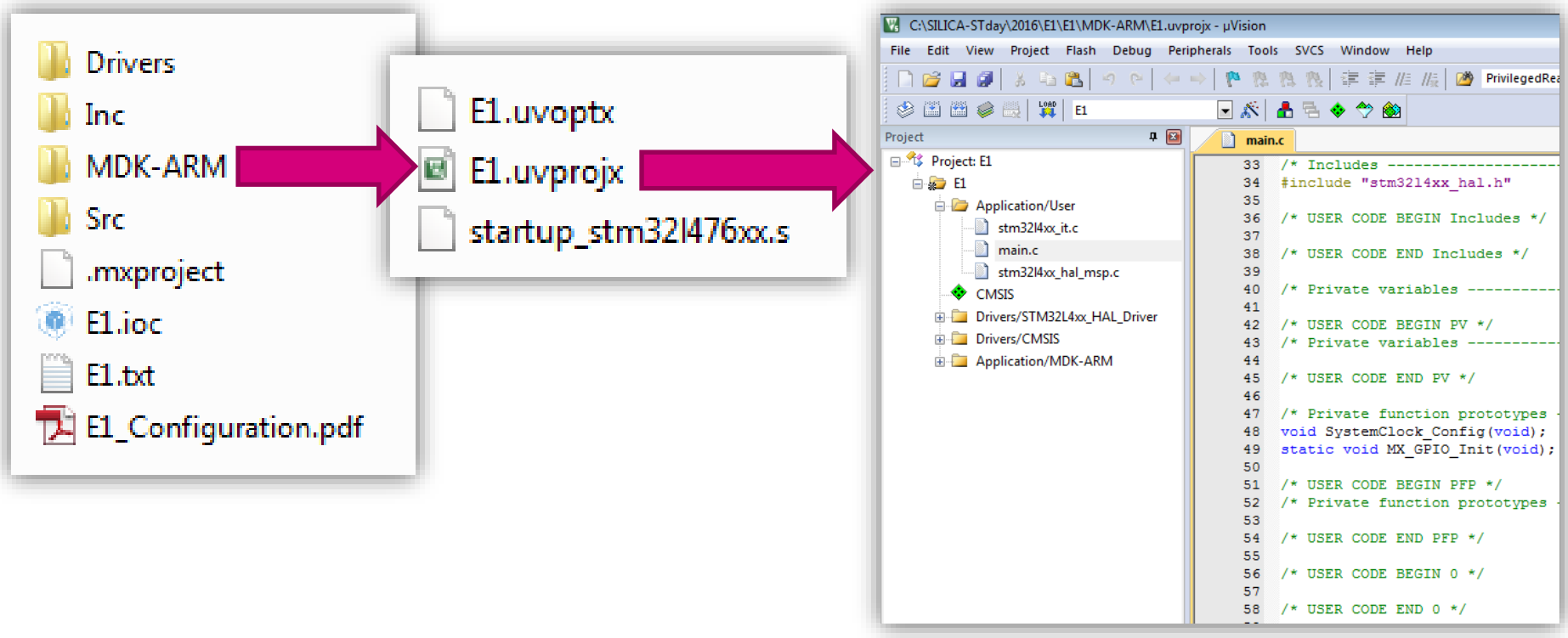
CUBE - Start new project generate the code for some GUI 1/3



CUBE - Start new project generate the code 2/3



CUBE - Start new project generate the code 3/3



CUBE - Start new project add code for flashing LEDs

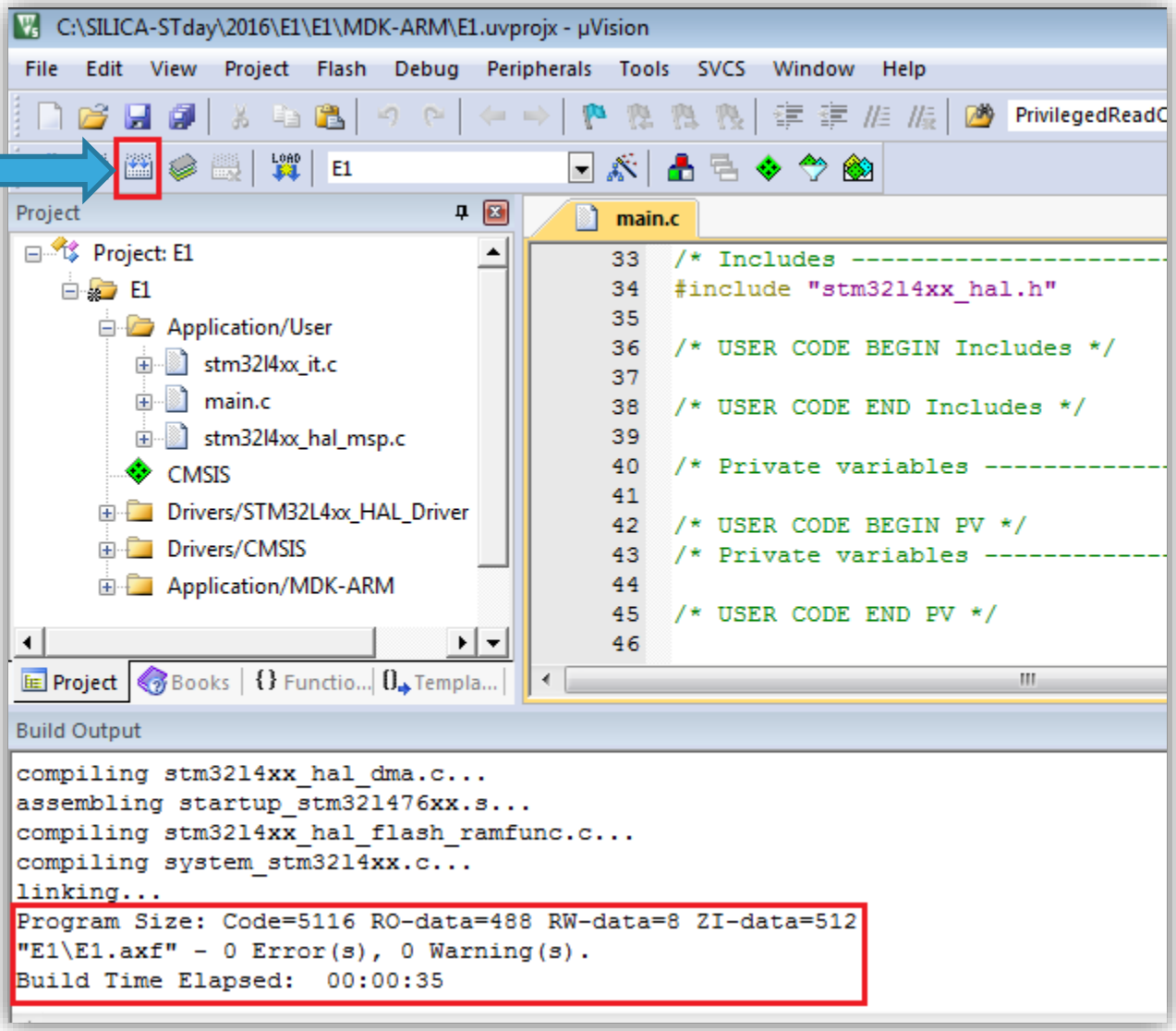
main.c

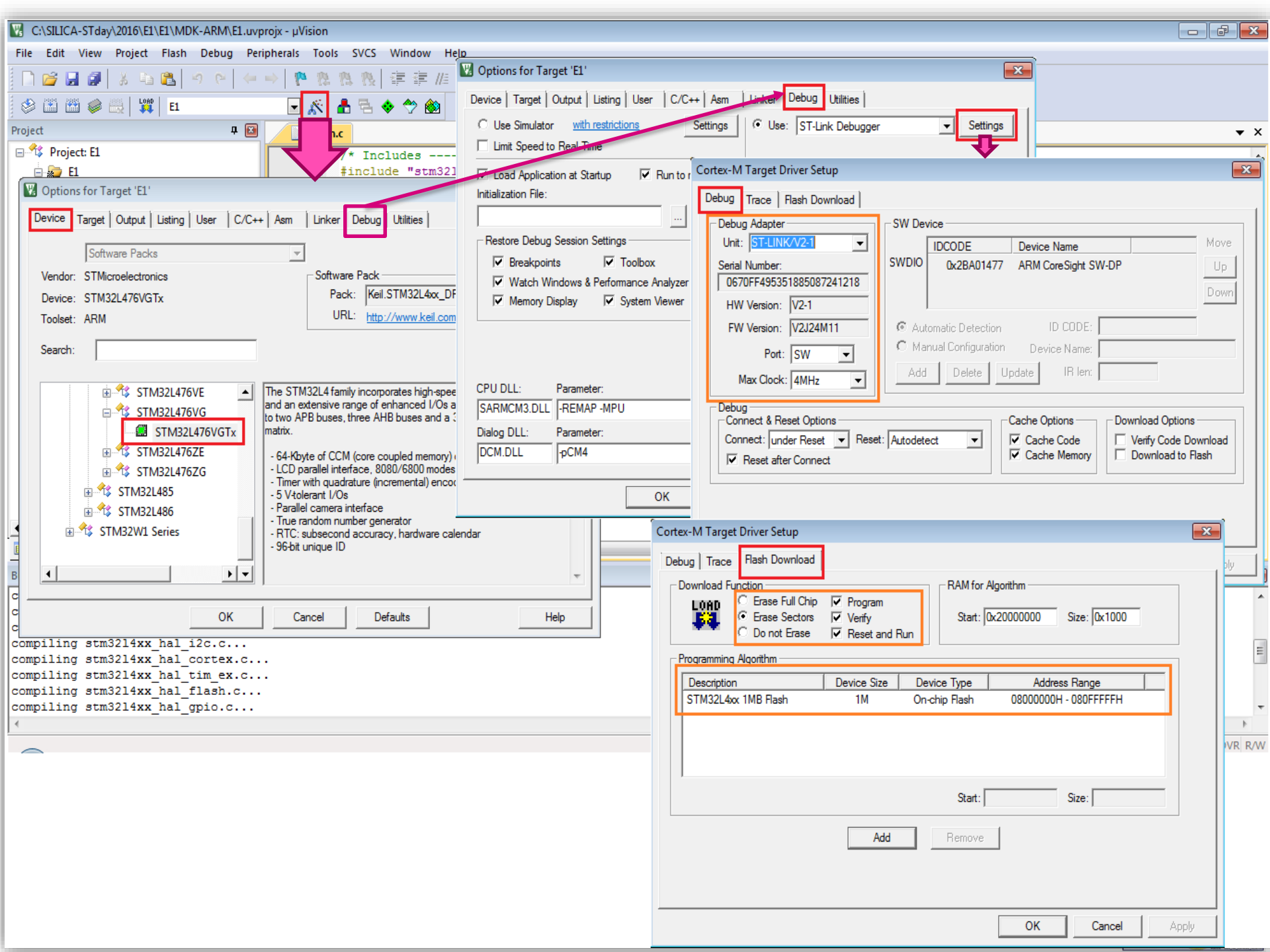
```
74
75    /* Initialize all configured peripherals */
76    MX_GPIO_Init();
77
78    /* USER CODE BEGIN 2 */
79
80    /* USER CODE END 2 */
81
82    /* Infinite loop */
83    /* USER CODE BEGIN WHILE */
84    while (1)
85    {
86    /* USER CODE END WHILE */
87
88    /* USER CODE BEGIN 3 */
89      HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_2);
90      HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_8);
91      /* Delay 200 ms */
92      HAL_Delay(200);
93    }
94    /* USER CODE END 3 */
95
96  }
97
```

See the:
UM1884 - Description of STM32L4 HAL and LL drivers

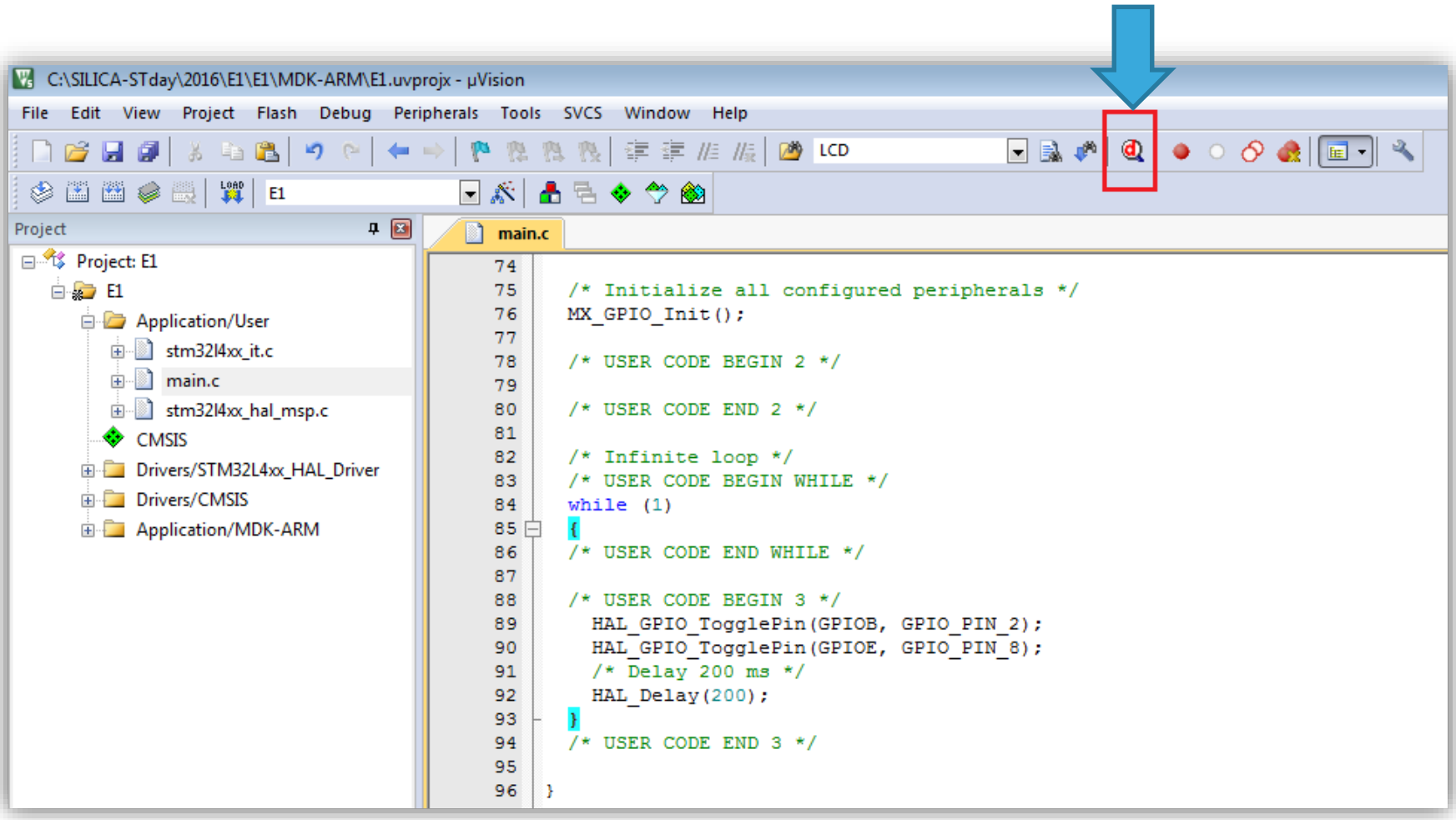
CUBE - Start new project compile and debug

COMPILE

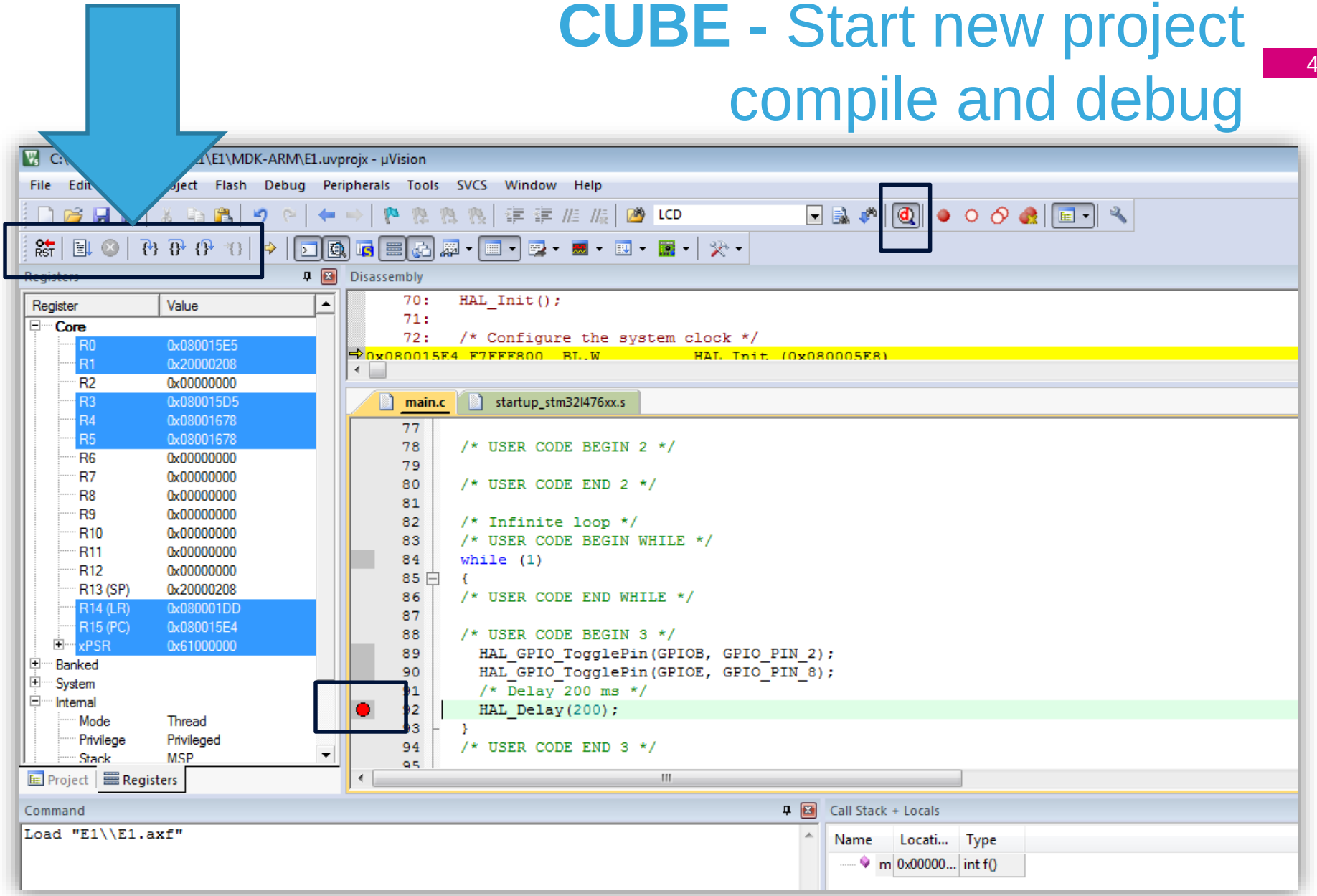



C:\SILICA-STday\2016\E1\E1\MDK-ARM\E1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project: E1
E1
Options for Target 'E1'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Software Packs
Vendor: STMicroelectronics
Device: STM32L476VGTx
Toolset: ARM
Software Pack
Pack: Keil.STM32L4xx_DF
URL: http://www.keil.com
Search:
STM32L476VE
STM32L476VG
STM32L476VGTx
STM32L476ZE
STM32L476ZG
STM32L485
STM32L486
STM32W1 Series
The STM32L4 family incorporates high-spee
and an extensive range of enhanced I/Os a
to two APB buses, three AHB buses and a
matrix.
- 64-Kbyte of CCM (core coupled memory)
- LCD parallel interface, 8080/6800 modes
- Timer with quadrature (incremental) encoc
- 5 V-tolerant I/Os
- Parallel camera interface
- True random number generator
- RTC: subsecond accuracy, hardware calendar
- 96-bit unique ID
OK Cancel Defaults Help
Options for Target 'E1'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Use Simulator with restrictions Settings
Limit Speed to Real-Time
Use: ST-Link Debugger Settings
Load Application at Startup Run to
Initialization File:
Restore Debug Session Settings
Breakpoints Toolbox
Watch Windows & Performance Analyzer
Memory Display System Viewer
CPU DLL: Parameter:
SARMCM3.DLL -REMAP -MPU
Dialog DLL: Parameter:
DCM.DLL -pCM4
OK
Cortex-M Target Driver Setup
Debug Trace Flash Download
Debug Adapter
Unit: ST-LINK/V2-1
Serial Number:
0670FF495351885087241218
HW Version: V2-1
FW Version: V2J24M11
Port: SW
Max Clock: 4MHz
SW Device
IDCODE Device Name
SWDIO 0x2BA01477 ARM CoreSight SW-DP
Move Up Down
Automatic Detection ID CODE:
Manual Configuration Device Name:
Add Delete Update IR len:
Debug
Connect & Reset Options
Connect: under Reset Reset: Autodetect
Reset after Connect
Cache Options
Cache Code
Cache Memory
Download Options
Verify Code Download
Download to Flash
Cortex-M Target Driver Setup
Debug Trace Flash Download
Download Function
Erase Full Chip Program
Erase Sectors Verify
Do not Erase Reset and Run
RAM for Algorithm
Start: 0x20000000 Size: 0x1000
Programming Algorithm
Description Device Size Device Type Address Range
STM32L4xx 1MB Flash 1M On-chip Flash 08000000H - 080FFFFFH
Start: Size:
Add Remove
OK Cancel Apply
compiling stm32l4xx_hal_i2c.c...
compiling stm32l4xx_hal_cortex.c...
compiling stm32l4xx_hal_tim_ex.c...
compiling stm32l4xx_hal_flash.c...
compiling stm32l4xx_hal_gpio.c...

CUBE - Start new project compile and debug



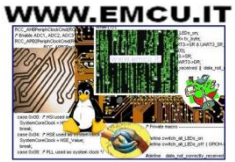
CUBE - Start new project compile and debug



TrueSTUDIO Lite

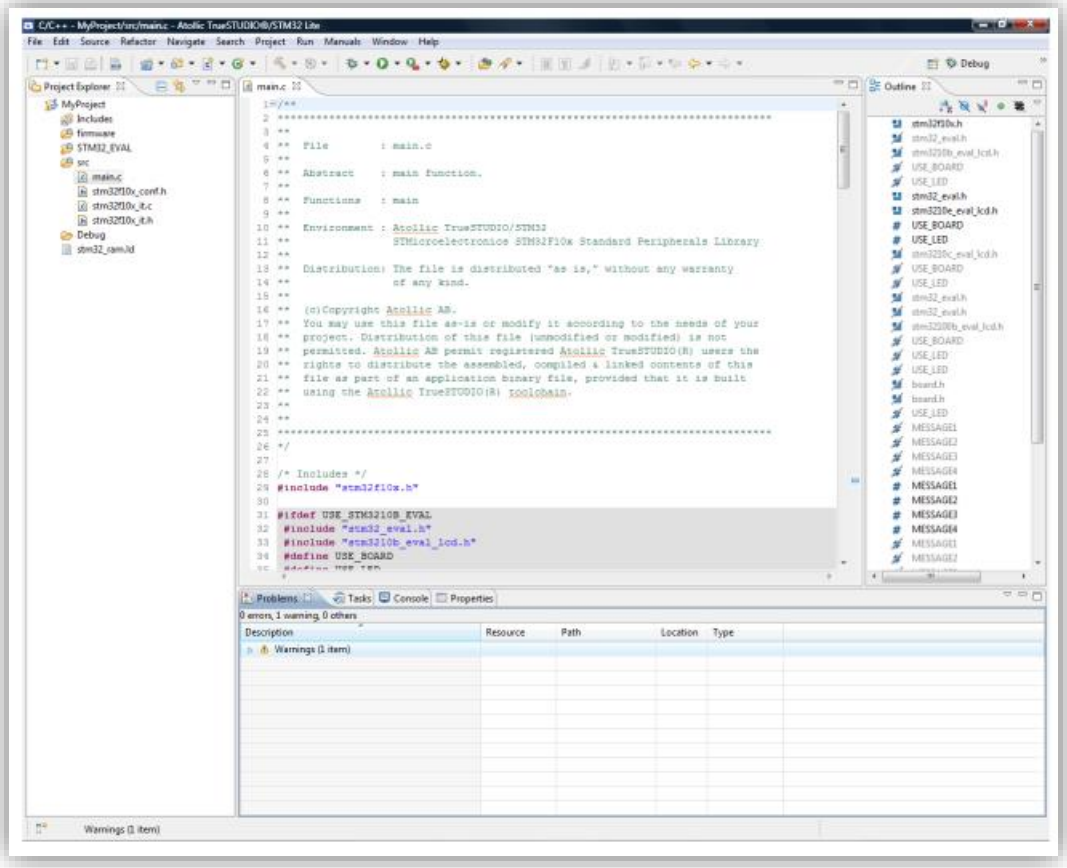
free tools for ARM

Free to download. Free to use. Free to share.



ATOLLIC

High-quality embedded systems software.
This includes compiling and assembling C/C++ source code, and linking object files and runtime libraries.
Core, multi-core and multi-processor systems.
Tight integration with popular debug probes including J-Link, ST-Link, and Multi-link.



	TrueSTUDIO Lite	TrueSTUDIO Pro
	FREE	49EUR/59USD per month
Features:	The best free ARM development tools on the planet!	Superior code analysis and advanced debug tools.
Project management and example projects	✓	✓
Edit/Compile/Debug	✓	✓
Tools for team collaboration	✓	✓
Unlimited code-size	✓	✓
Static code analysis tools		✓
Advanced debugging capabilities		✓
Technical support		✓

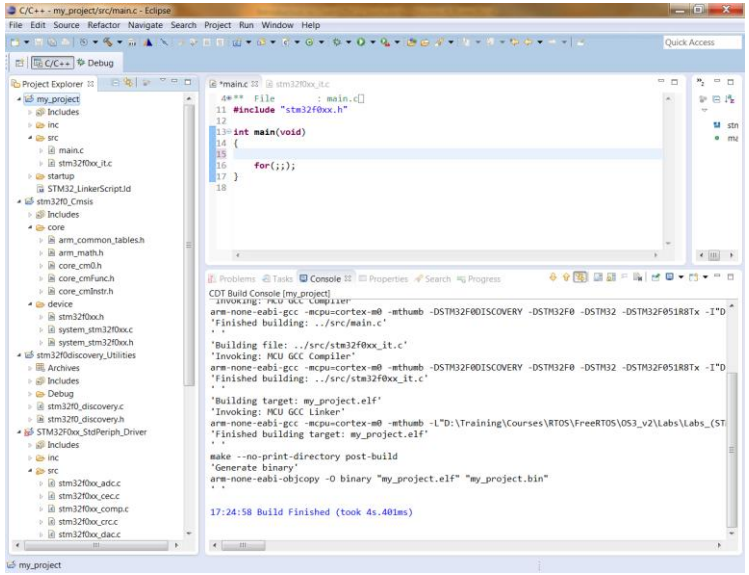
AC6 – SW4STM32

AC6 – SW4STM32

- Toolset developed by **AC6** partner
- Current name : « **System Workbench for STM32** » i.e **SW4STM32**
- A complete expandable free toolset :
  - A user-friendly graphical environment based on **Eclipse**
  - **GCC-based** full-fledged toolchains (without code size limitation)
  - **Multiple OS support** (Windows, Linux, OSX)
- A community website www.openSTM32.org :
  - Direct download area for toolset and add-ons
  - Wiki for toolset documentation
  - Support forums
  - Blogs

AC6 – SW4STM32

- Define your target platform from a predefined set of elements :
  - Choose any STM32 mcu configuration, any STM32 reference boards (Nucleo, Discovery, Eval) or define your own board.
- **Configure**, **compile** and install your application :
  - Use the automatically linked **libraries** providing **CMSIS**, Peripherals and Board related drivers.
- **Flash** and **Debug** your application on the target with built-in or external hardware debug probe :
  - Run, break, single step through your whole program.
  - Display MCU-specific registers in a convenient way.



Free CooCox CoIDE

Free/open ARM Cortex-M Development Tool-chain

Free CooCox CoIDE

- CooCox is a **free integrated development evironment** focusing on ARM **Cortex Mx** based MCU.

- It uses **GCC** tool chain and is based on **Eclipse**.

- CooCox is devoted to providing developers with a free/open and easy-to-use ARM Cortex Mx tool-chain, including:
  - **CoLinkEx**, a low-cost and hardware open debug adapter (compatible with ST-LINK-v2)
  - **CoIDE**, an integrated development environment for developing based on code components like stacking up building blocks
  - **CoSmart**, a graphical code generation tool.

- Meanwhile, CooCox encourages users to share open source code components in CooCox community. By providing a series of open source software such as CoX, a unified standard peripheral library for ARM Cortex Mx based microcontrollers.

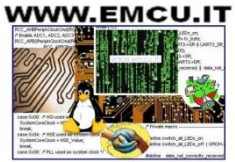
- **CoOS**, a lightweight **real-time operating system** and a number of drivers based on CoX, CooCox is committed to building an open-source and sharing community.

Free MDK-ARM

life.augmented

AVNET Memec

SILICA An Avnet Company



Free MDK-ARM

- Multi-year agreement with ARM on MDK-ARM :
- Free MDK-ARM licenses for STM32 based on **Cortex-M0/M0+** cores :
  - Applicable immediately to all **STM32F0** and **STM32L0** mcus.
  - PC-locked renewable 3-Year licenses.
  - **No code size limit**.
  - Windows support only.
  - Multiple languages supported : English, Japanese, Korean.
  - **Technical support included**.
- Direct download from Keil website :
  - **No limit of number of downloads by customer.**
  - Direct access to configuration files for all STM32 based on Cortex-M0/M0+ cores.
  - **Free access to MDK-ARM periodic updates**.

Thank you !