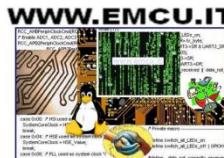
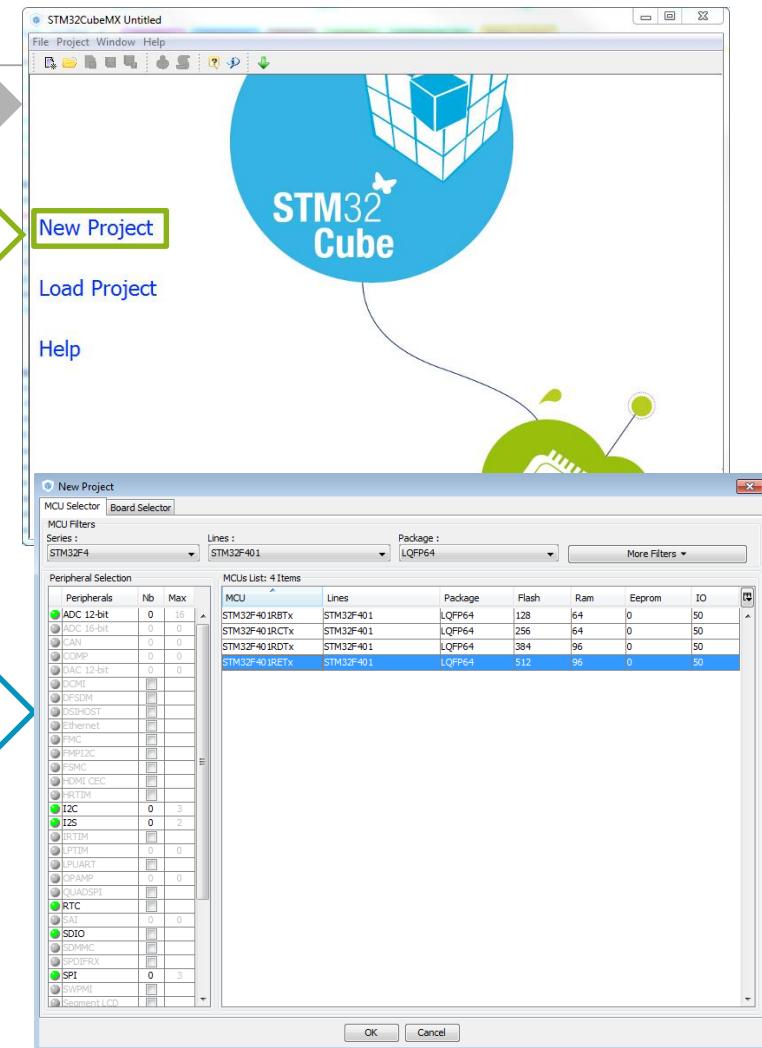
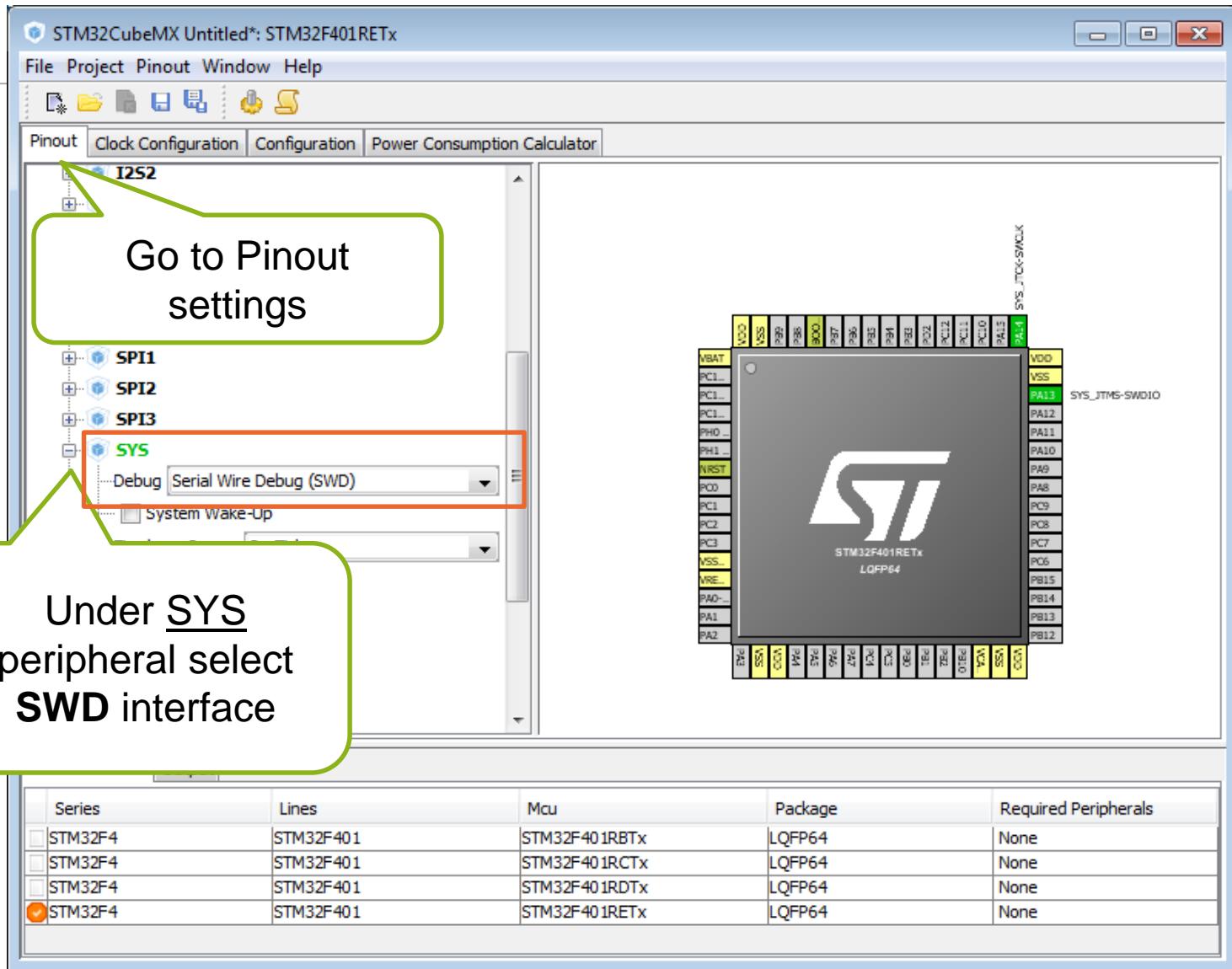


Create new project using CubeMX

- Run CubeMX tool
- Start **new project**
 - Click “New Project” desktop shortcut, or
 - Go to “Menu->File->New Project”
- Filter:
 - Series: STM32F4
 - Line: STM32F401
 - Package: LQFP64
- Select: **STM32F401RE**



Configure debug interface



STM32CubeMX Untitled*: STM32F401RETx

File Project Pinout Window Help

Pinout Clock Configuration Configuration Power Consumption Calculator

I2S2

SPI1

SPI2

SPI3

SYS

Debug Serial Wire Debug (SWD)

System Wake-Up

Go to Pinout settings

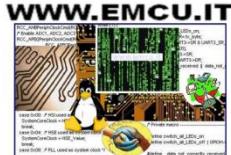
Under SYS peripheral select SWD interface

STM32F401RETx LQFP64

Series Lines Mcu Package Required Peripherals

STM32F4	STM32F401	STM32F401RBTx	LQFP64	None
STM32F4	STM32F401	STM32F401RCTx	LQFP64	None
STM32F4	STM32F401	STM32F401RDTx	LQFP64	None
<input checked="" type="checkbox"/> STM32F4	STM32F401	STM32F401RETx	LQFP64	None

Configure LSE resonator (32,768 KHz)



The screenshot shows the STM32CubeMX software interface for the STM32F401RETx. The top menu bar includes File, Project, Pinout, Window, and Help. The Pinout tab is selected, displaying a tree view of peripherals and their configuration options. The RCC peripheral is expanded, showing the following settings:

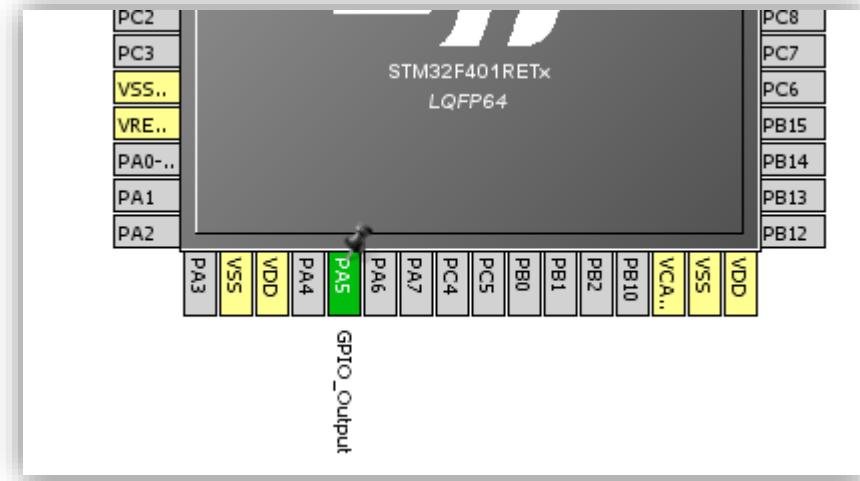
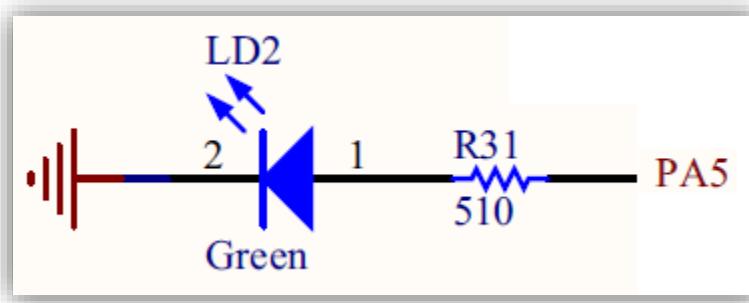
- High Speed Clock (HSE): Disable
- Low Speed Clock (LSE): Crystal/Ceramic Res... (highlighted with a red box)
- Master Clock Output 1
- Master Clock Output 2
- Audio Clock Input (I2S_CKIN)

The pinout diagram for the STM32F401RETx LQFP64 package is shown on the right, with pins labeled from PA0 to PA15 and PB0 to PB15. The VDD and VSS pins are also indicated.

MCUs Selection	Output			
Series	Lines	Mcu	Package	Required Peripherals
STM32F4	STM32F401	STM32F401RBTx	LQFP64	None
STM32F4	STM32F401	STM32F401RCTx	LQFP64	None
STM32F4	STM32F401	STM32F401RDTx	LQFP64	None
<input checked="" type="checkbox"/> STM32F4	STM32F401	STM32F401RETx	LQFP64	None

Configure GPIO for LED toggling

Configure LED pin as GPIO_Output



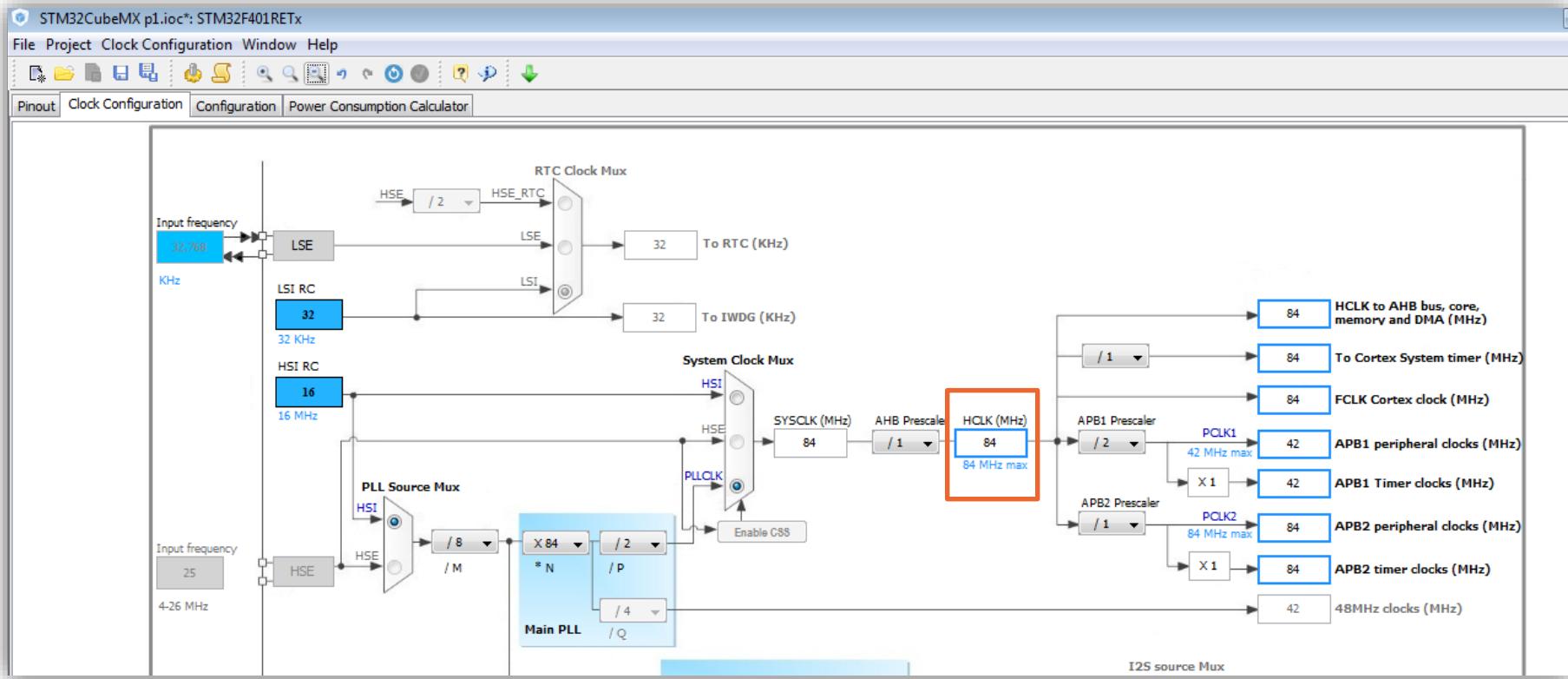
31



11 June 2016



Clock configuration



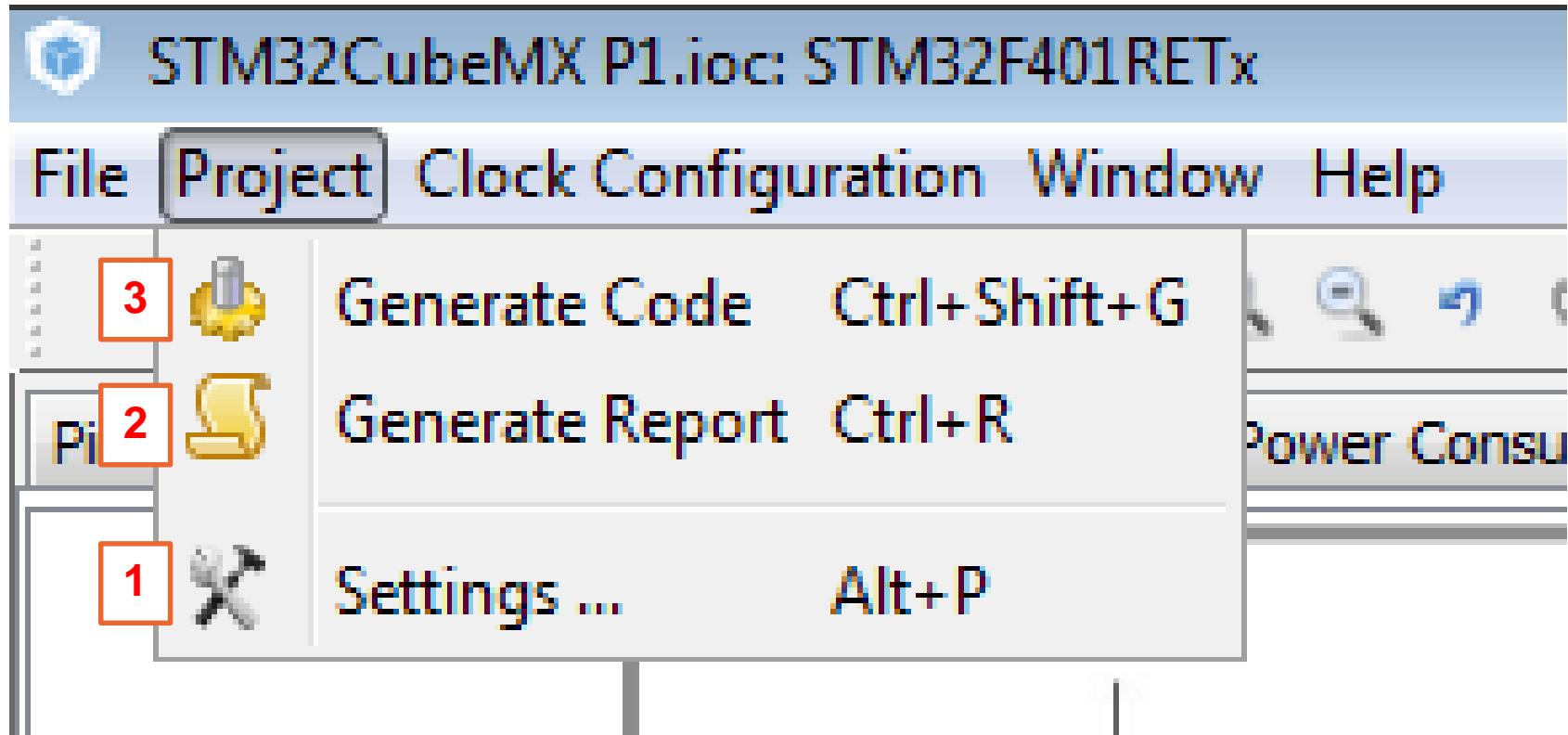
32



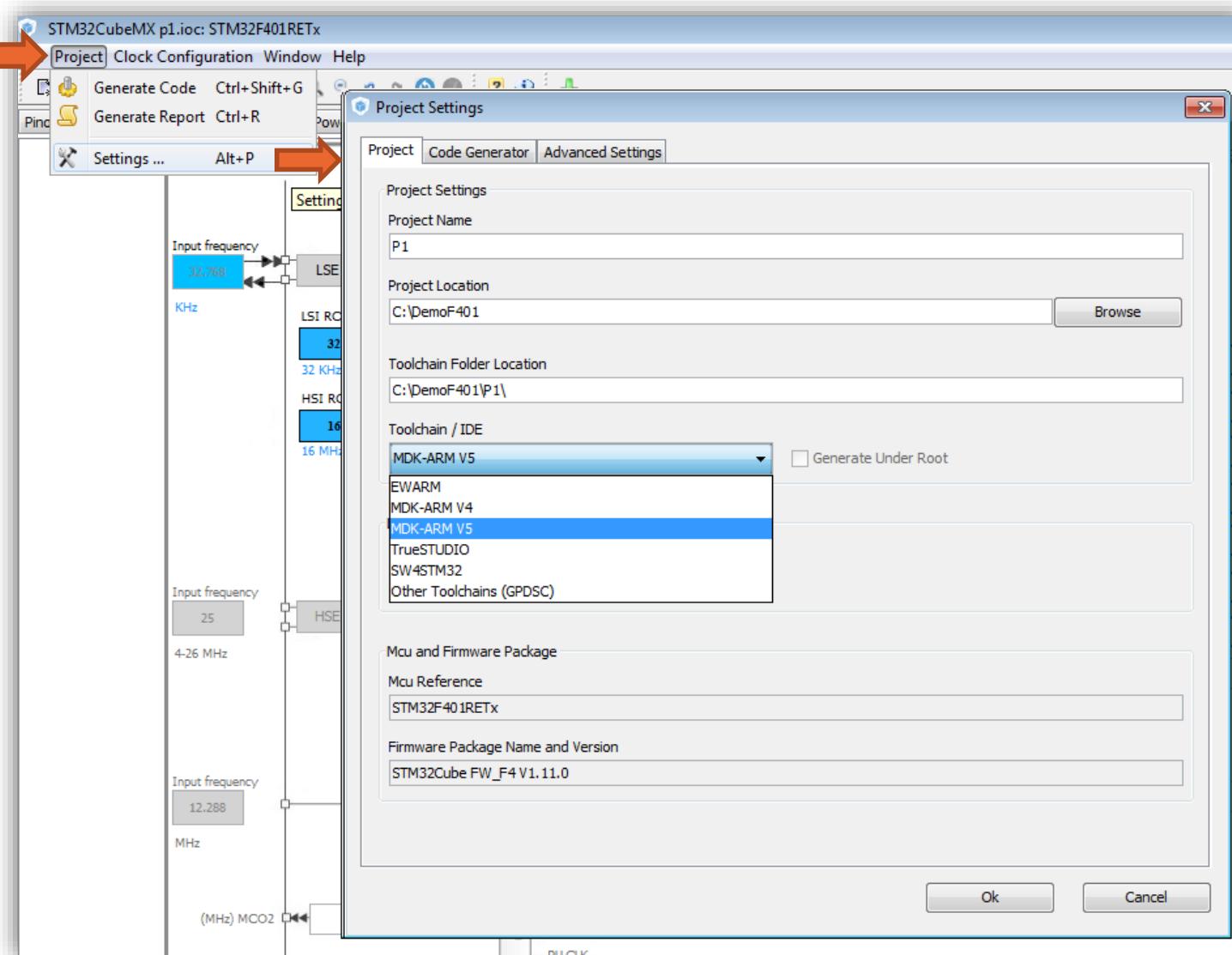
11 June 2016



CubeMX generate the code for some GUI 1/3

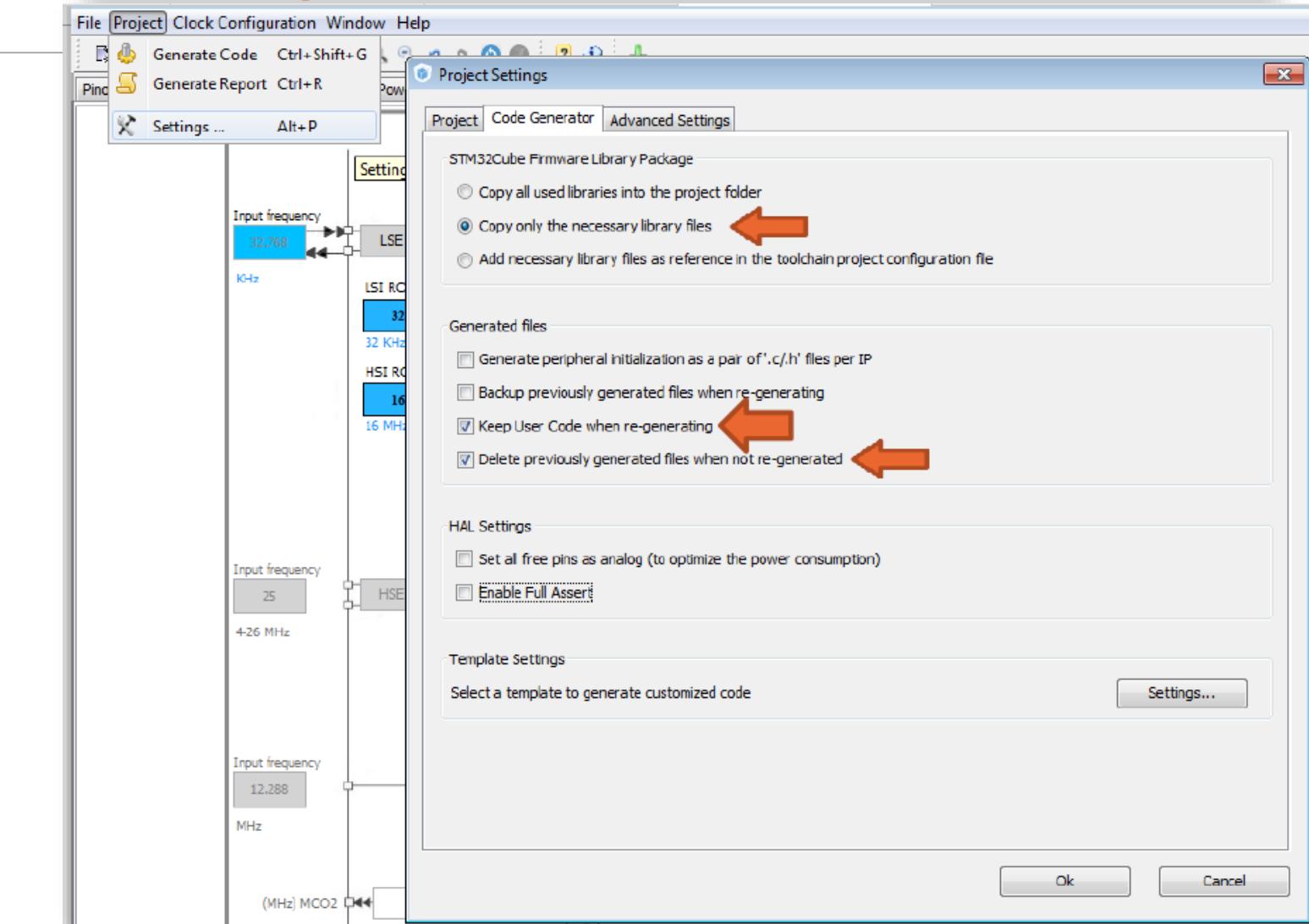


CubeMX generate the code for some GUI 2/3

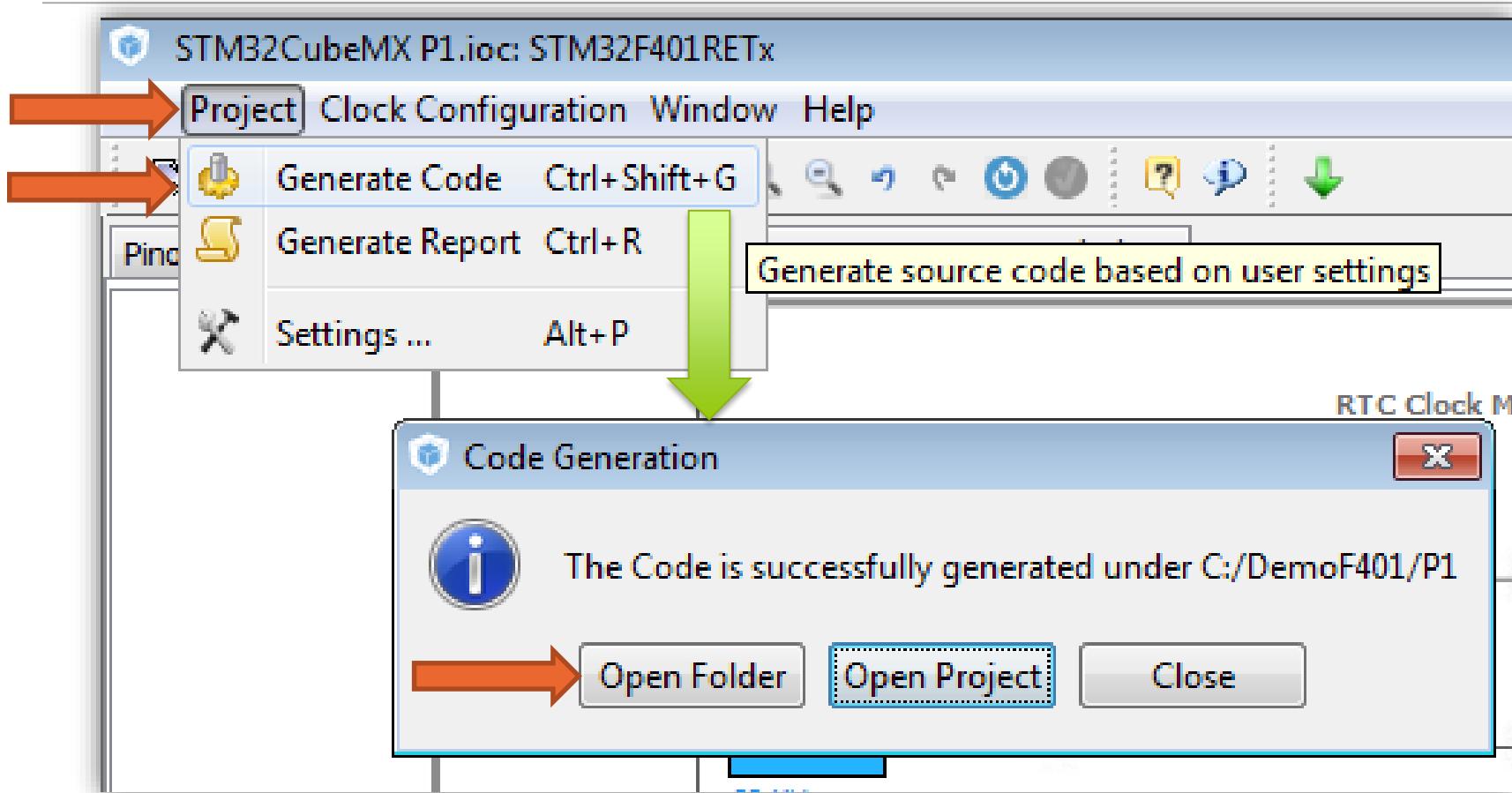




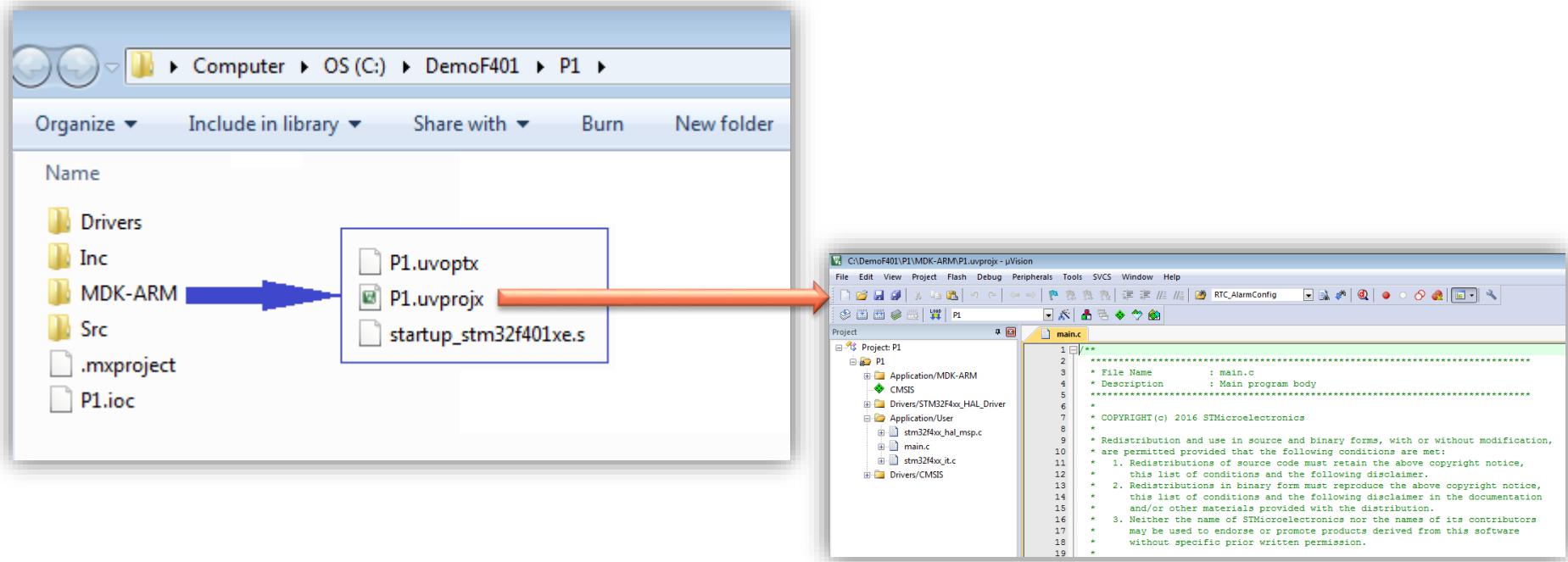
CubeMX generate the code for some GUI 3/3



CubeMX generate the code 1/3



CubeMX generate the code 2/3



37



11 June 2016



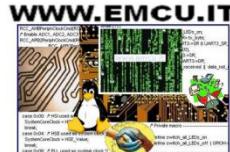
CubeMX add code for flashing LEDs

```
68 /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
69 HAL_Init();
70
71 /* Configure the system clock */
72 SystemClock_Config();
73
74 /* Initialize all configured peripherals */
75 MX_GPIO_Init();
76
77 /* USER CODE BEGIN 2 */
78
79 /* USER CODE END 2 */
80
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(GPIOA, GPIO_PIN_5);
90 HAL_Delay(200);
91 }
92 /* USER CODE END 3 */
93
94 }
```

The C code generated by CubeMX provides user sections as illustrated here. They allow user C code to be inserted and preserved at next C code generation. So add your code in these dedicated sections so as to be kept by CubeMX upon regeneration.



See the:
UM1725 - Description
of STM32F4 HAL
drivers



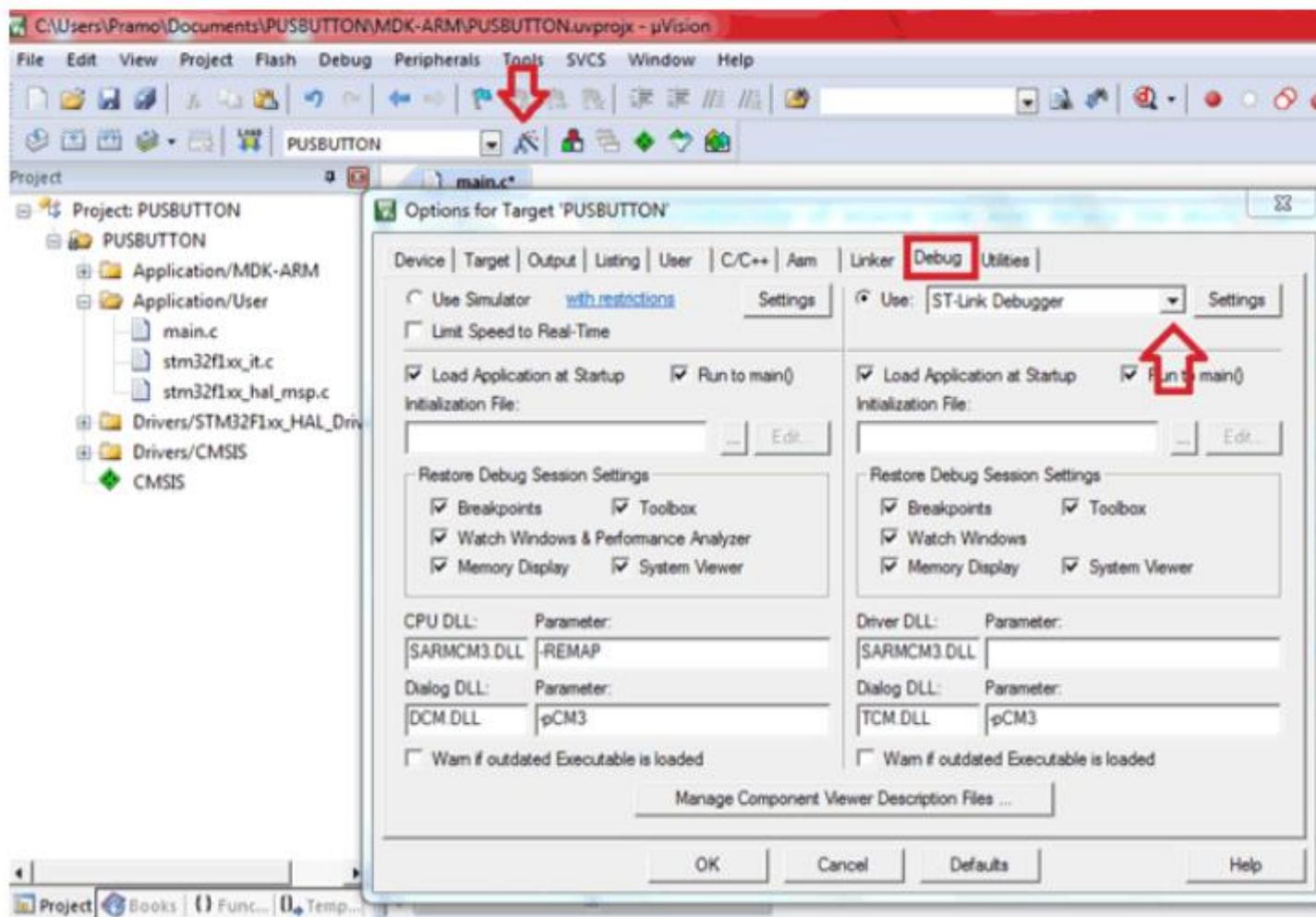
38



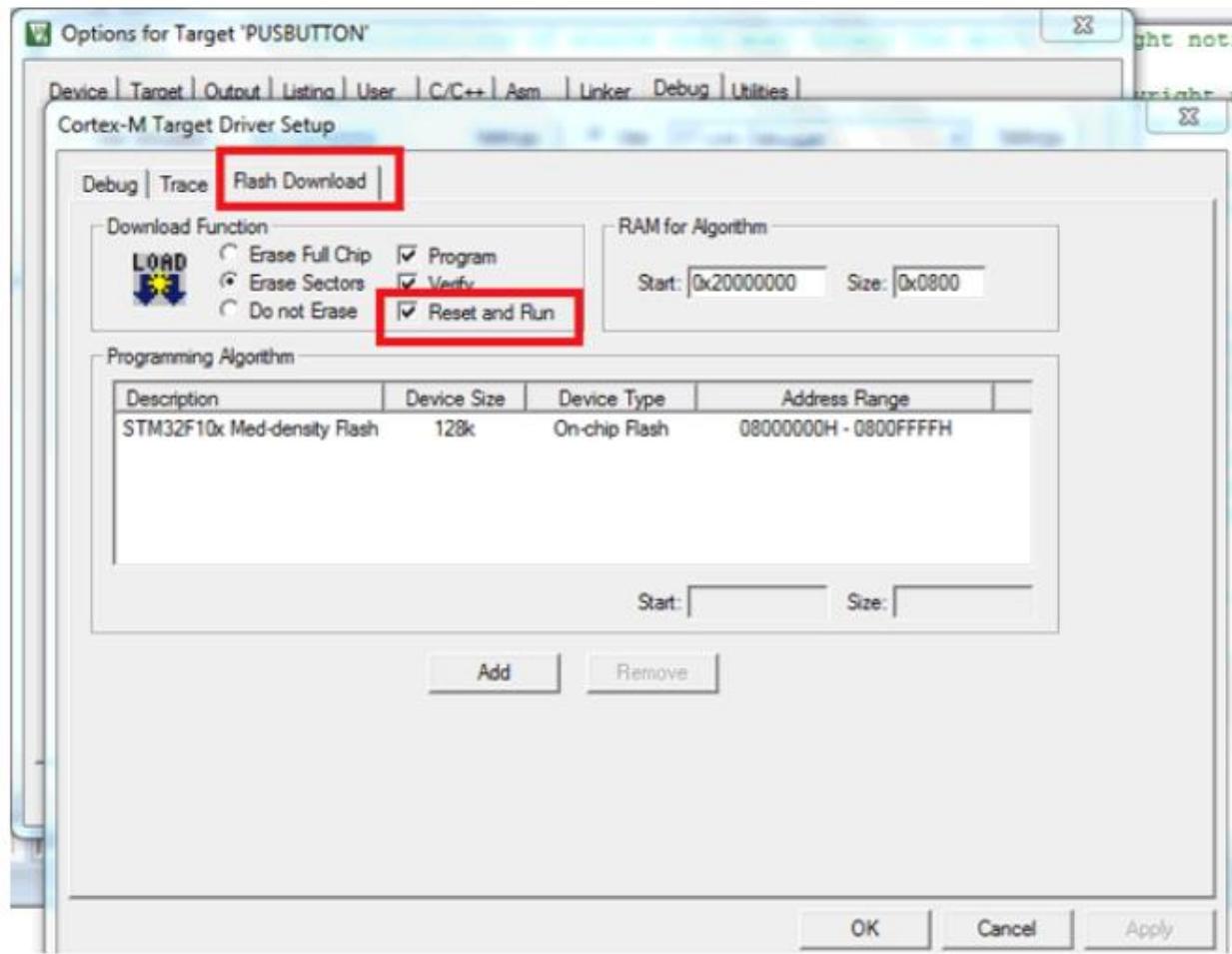
11 June 2016



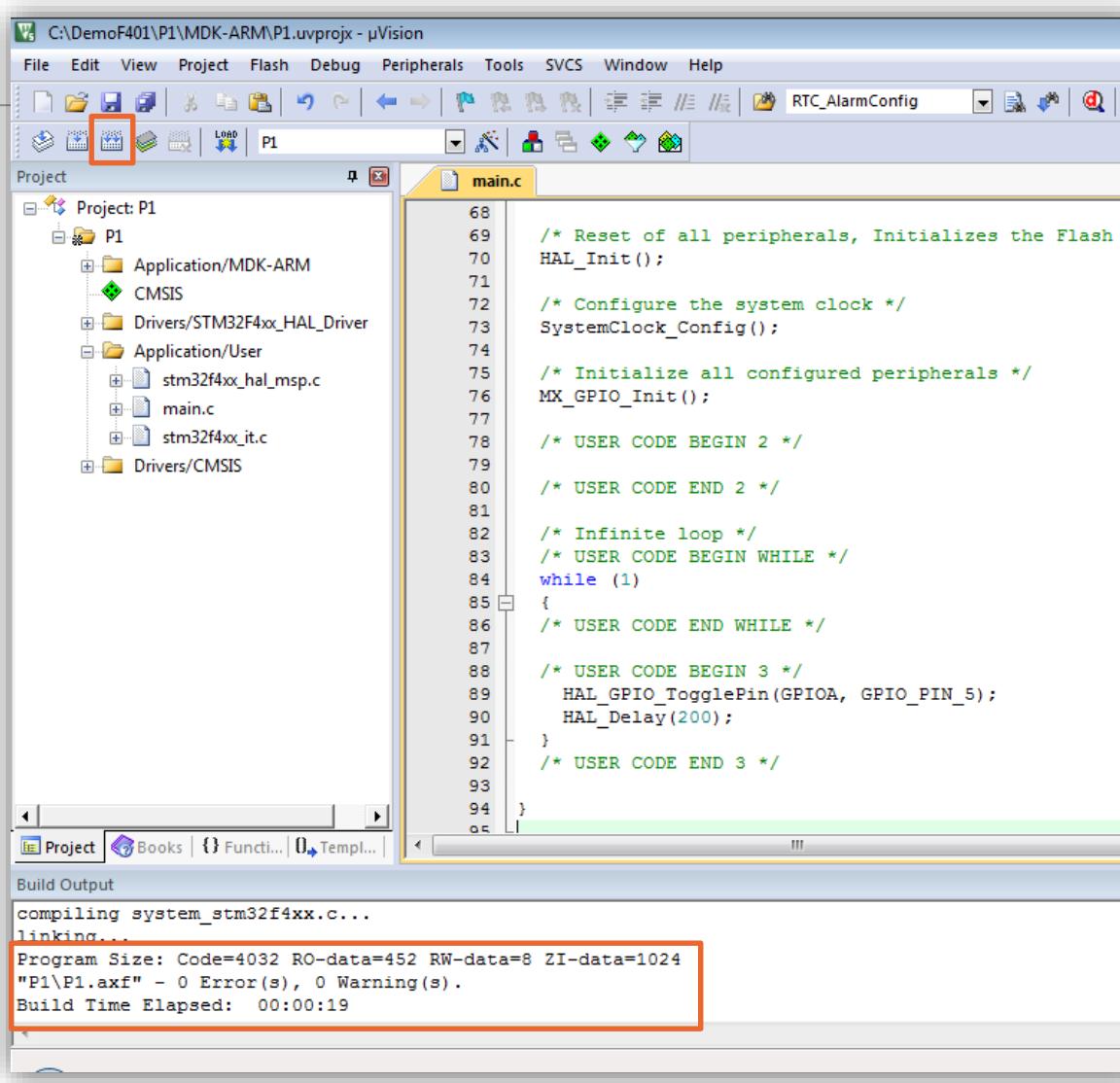
After finish editing the code, click the **Options for Target** icon under the **debug tab** select ST-LINK Debugger



Also, click on **Settings** button and then under **Flash Download** tab tick the **Reset and Run** check box and click 'ok'.



CubeMX compile and debug – 1/3



The screenshot shows the ST-Connect IDE interface with the following details:

- Project:** C:\DemoF401\P1\MDK-ARM\P1.uvprojx - µVision
- File Menu:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Build, Run, and others.
- Project Explorer:** Shows the project structure under "Project: P1".
 - P1 folder contains:
 - Application/MDK-ARM
 - CMSIS
 - Drivers/STMB2F4xx_HAL_Driver
 - Application/User
 - stm32f4xx_hal_msp.c
 - main.c (highlighted)
 - stm32f4xx_it.c
 - Drivers/CMSIS
- Code Editor:** Displays the main.c file content, which includes initialization code for peripherals, system clock configuration, and an infinite loop with GPIO toggling.
- Build Output:** Shows the compilation and linking process:

```
compiling system_stm32f4xx.c...
linking...
Program Size: Code=4032 RO-data=452 RW-data=8 ZI-data=1024
"P1\P1.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:19
```



39



11 June 2016



CubeMX compile and debug – 2/3

The screenshot shows the ST MDK-ARM IDE interface. The title bar reads "C:\DemoF401\P1\MDK-ARM\P1.uvprojx - µVision". The menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar has various icons for project management and debugging. A search icon is highlighted with a red box. The left pane shows the project structure under "Project: P1": P1 (Application/MDK-ARM, CMSIS, Drivers/STMB2F4xx_HAL_Driver, Application/User (stm32f4xx_hal_msp.c, main.c, stm32f4xx_it.c), Drivers/CMSIS). The right pane displays the code editor for "main.c" with the following content:

```
68  /* Reset of all peripherals, Initializes the Flash interface of the
69  * HAL_Init();
70
71  /* Configure the system clock */
72  SystemClock_Config();
73
74  /* Initialize all configured peripherals */
75  MX_GPIO_Init();
76
77  /* USER CODE BEGIN 2 */
78
79  /* USER CODE END 2 */
80
81  /* Infinite loop */
82  /* USER CODE BEGIN WHILE */
83  while (1)
84  {
85      /* USER CODE END WHILE */
86
87
88  /* USER CODE BEGIN 3 */
89  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
90  HAL_Delay(200);
91  }
92  /* USER CODE END 3 */
93
94 }
```

The bottom pane shows the "Build Output" window with the following log:

```
compiling system_stm32f4xx.c...
linking...
Program Size: Code=4032 RO-data=452 RW-data=8 ZI-data=1024
"P1\P1.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:19
```



40



11 June 2016



CubeMX compile and debug – 3/3

