STM32CubeIDE basics

GPIO lab: LED toggling using HAL libraries

life.augmented

STM32 CubeIDE

Lab: LED toggling using HAL library

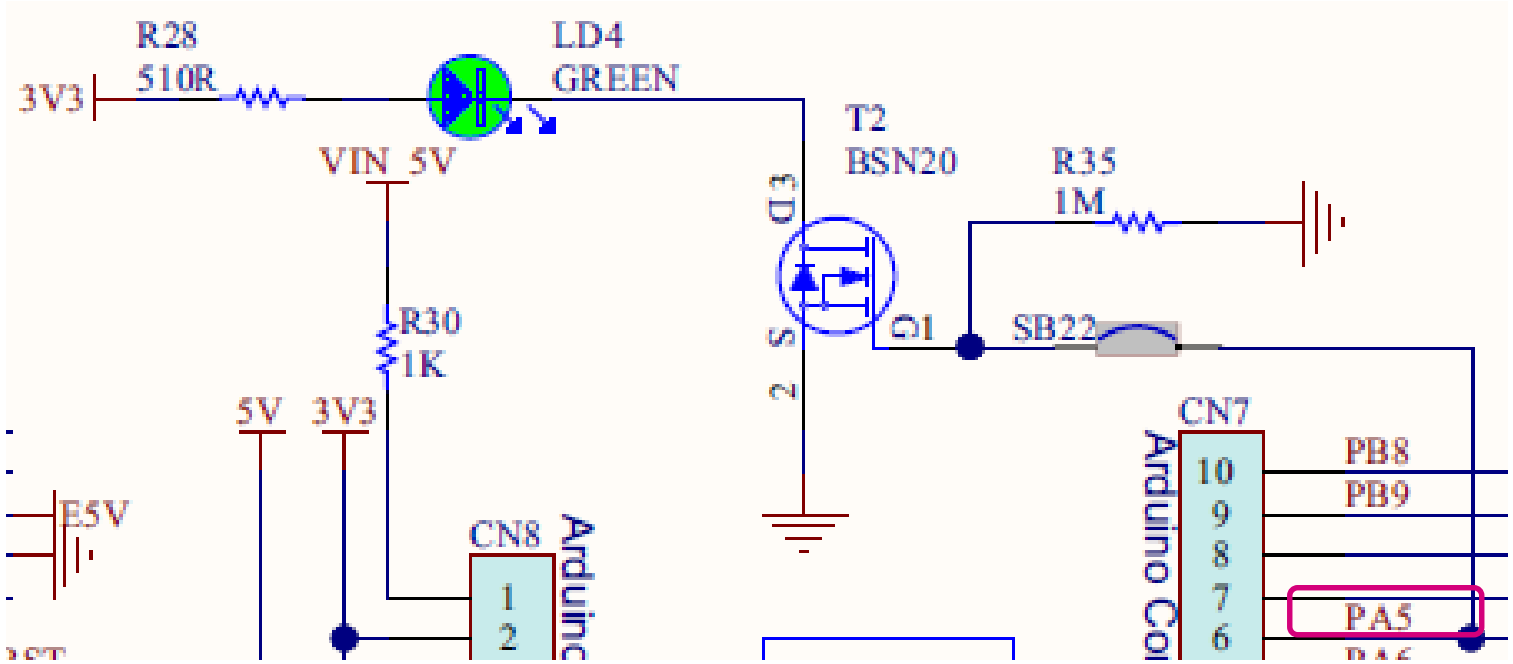
Objective:

- In this project we are going to toggle GREEN LED connected to PA5 pin on NUCLEO-G071RB board using Hardware Abstraction Layer (HAL) library

Pin Configuration

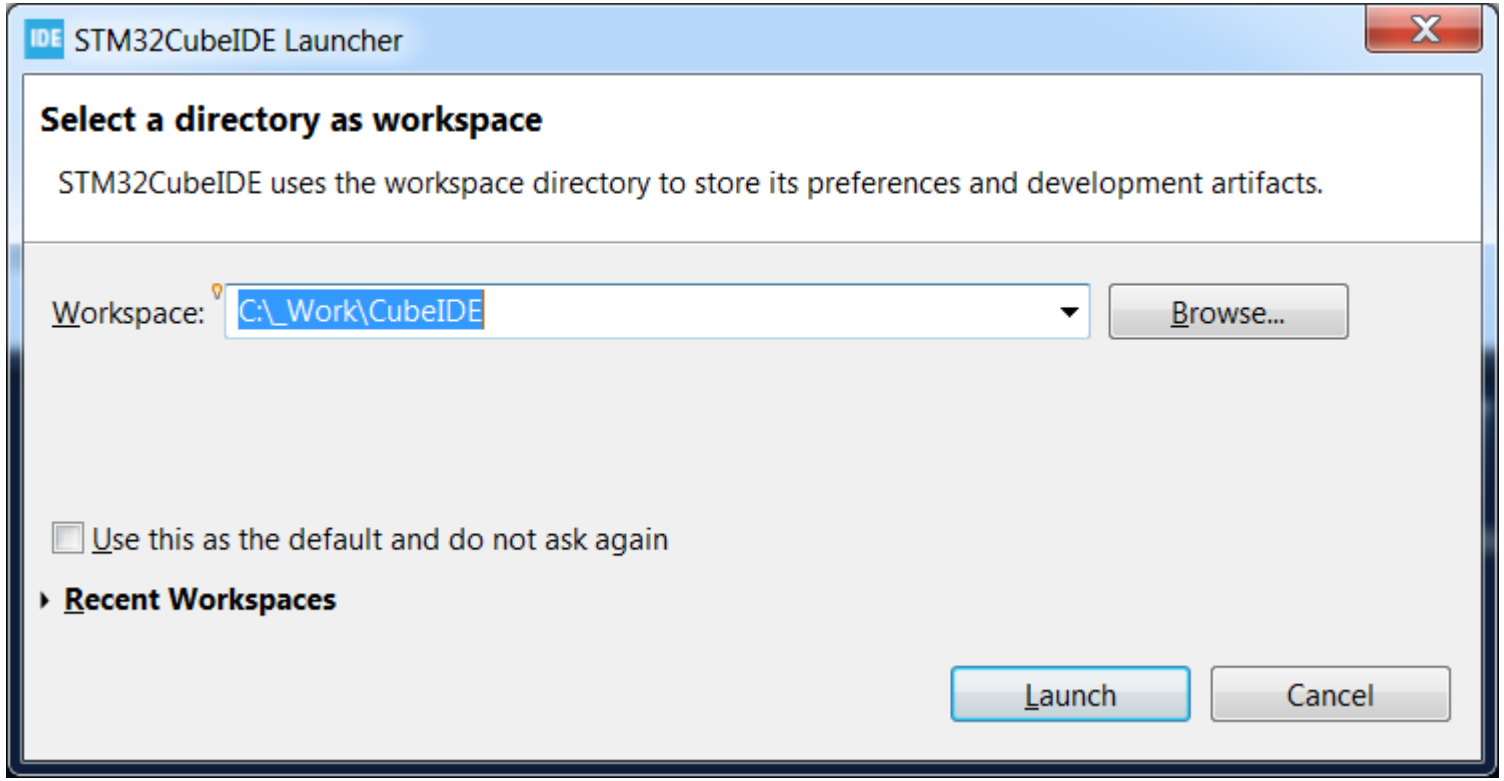
Green LED

- In this example we are going to use one of the LEDs present on the STM32G0 Nucleo board (connected to PA5 as seen in the schematic below)

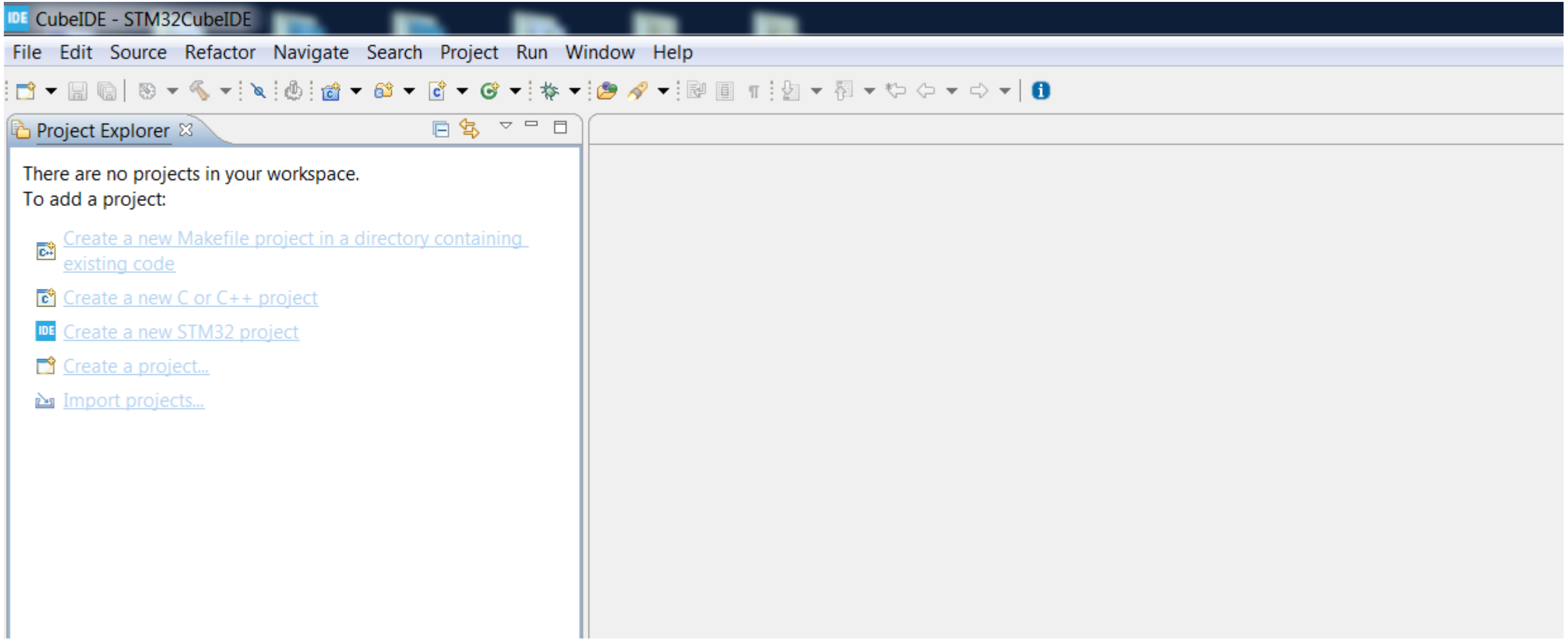


Start a new workspace

- Run STM32CubeIDE
- Select a folder to store a workspace

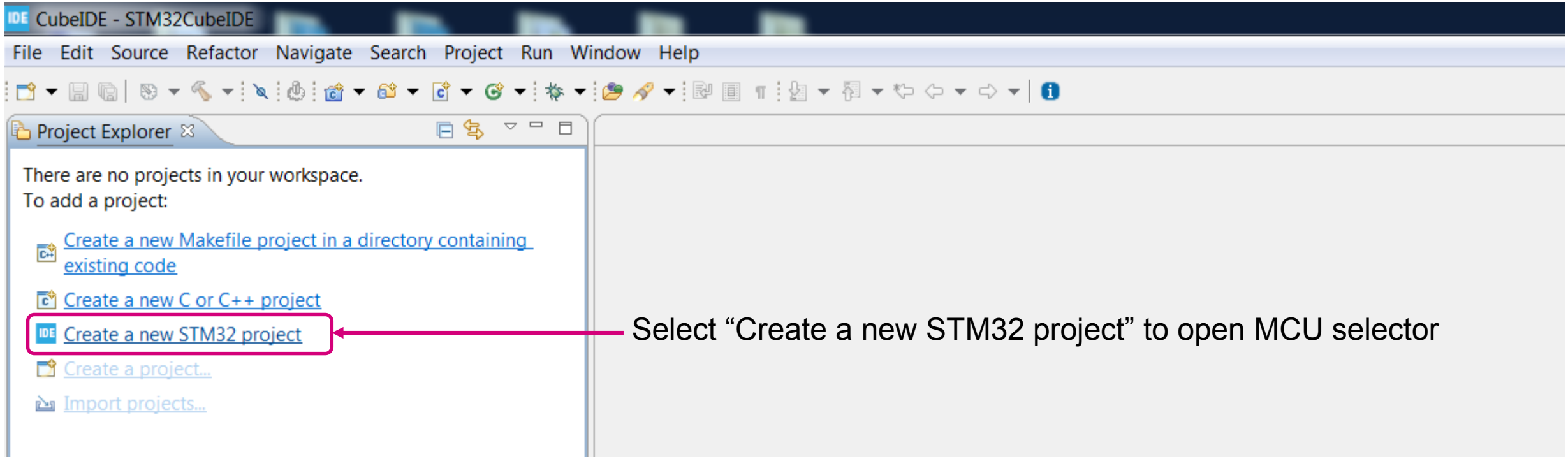


View on an empty workspace



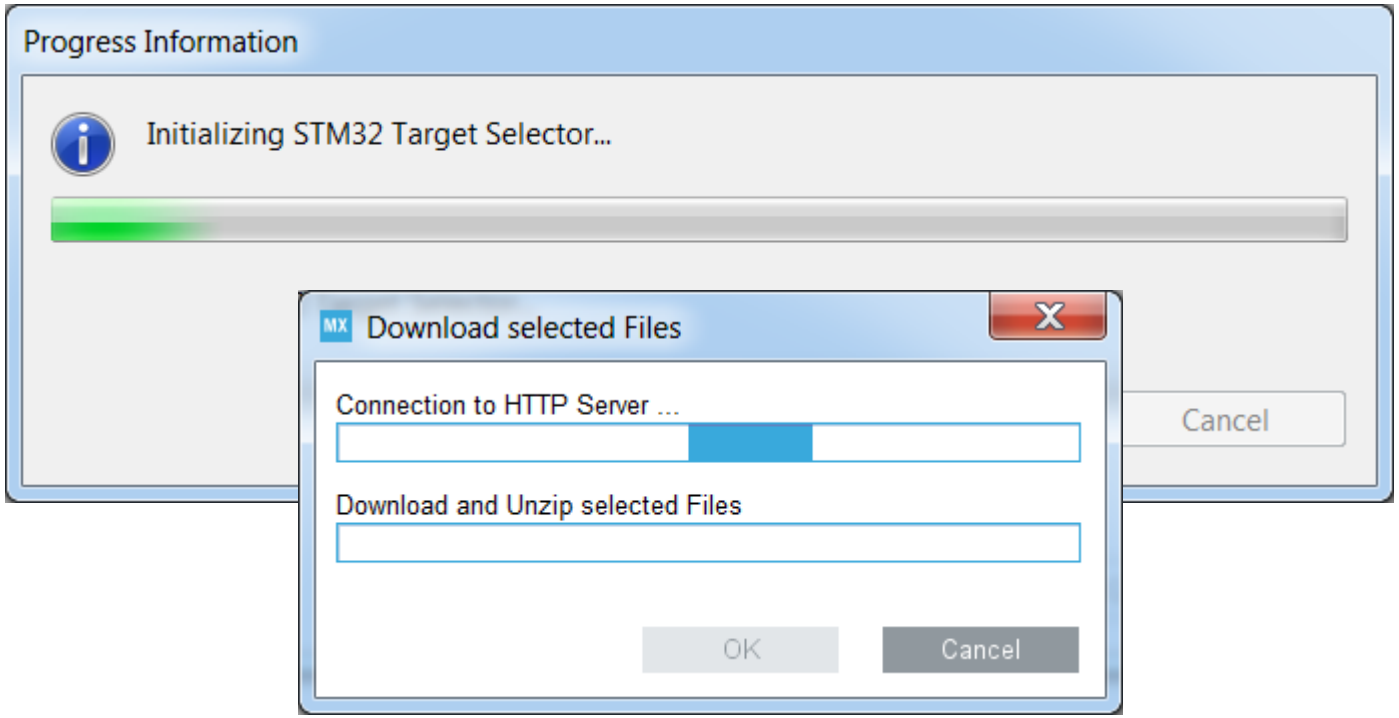
Create a new project

- Click on “Create a new STM32 project”



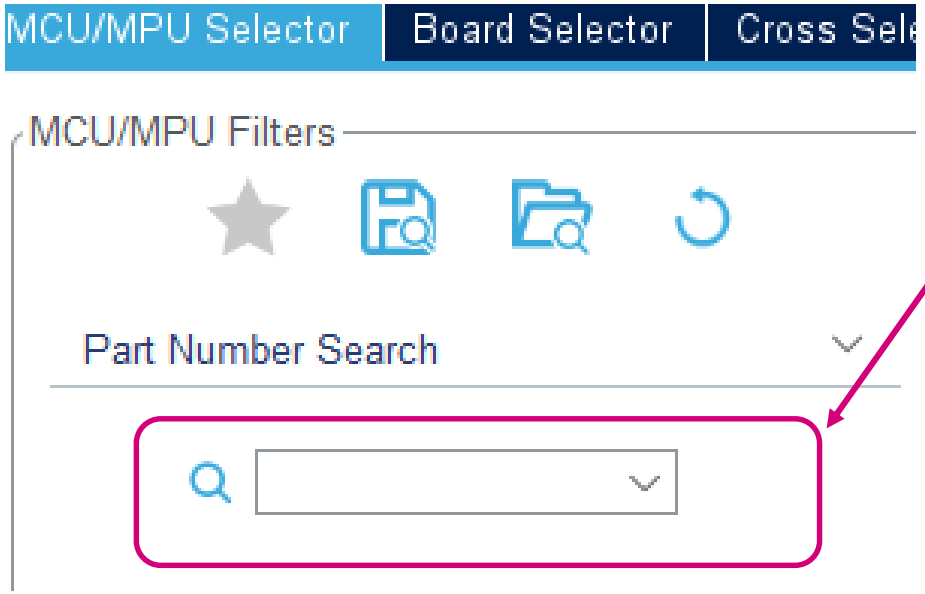
Checking for libraries update

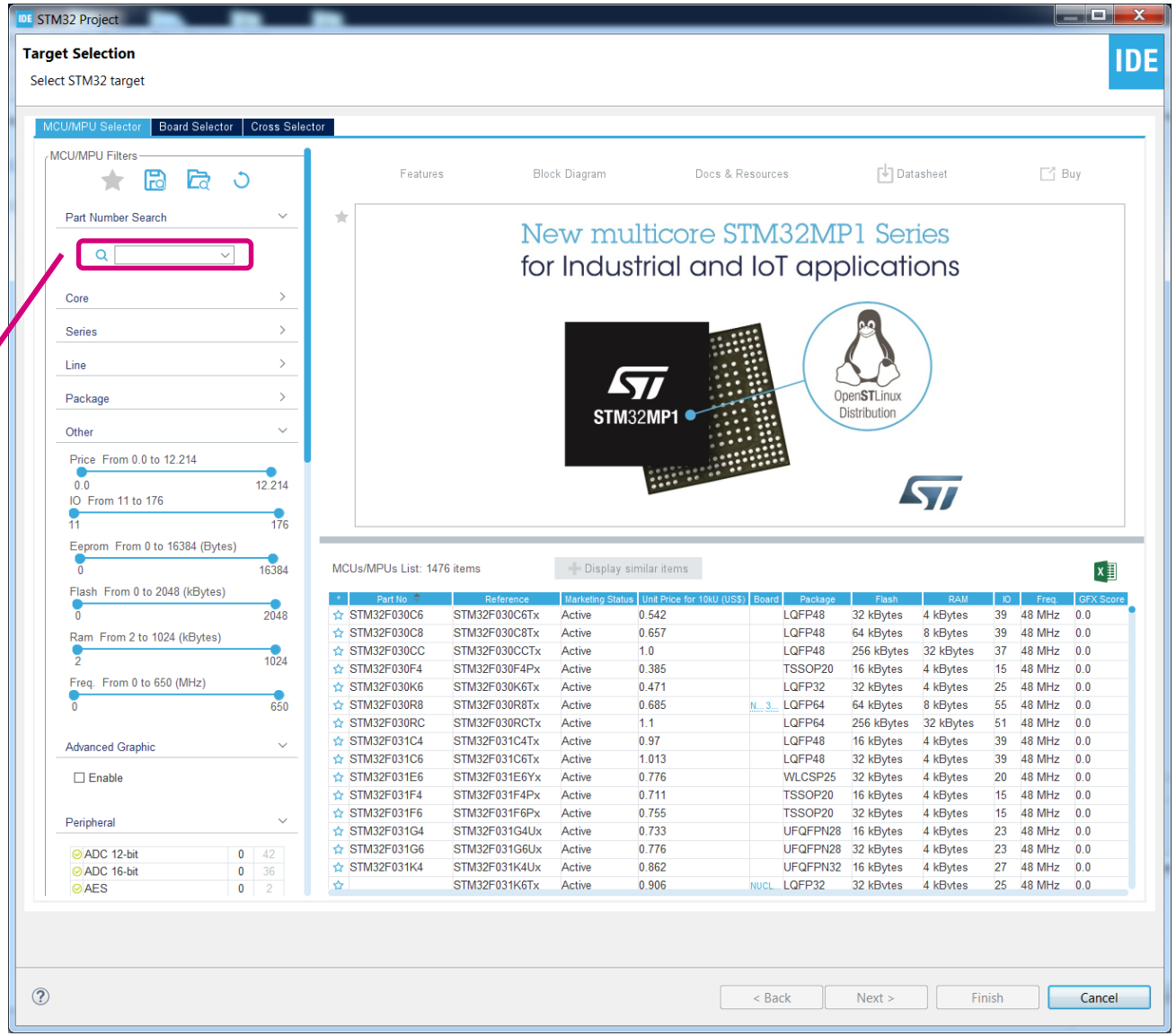
- Application will look for updates on the server
- It may take a while …



MCU/MPU selector

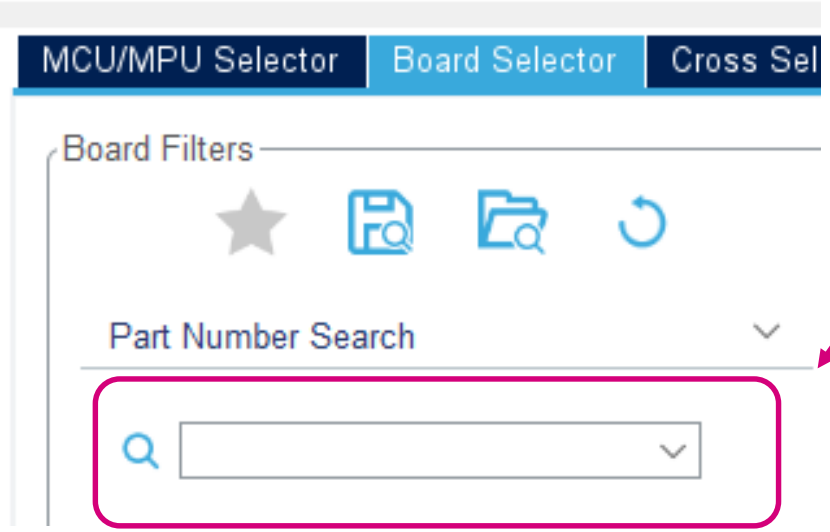
- As a next step we should select target MCU

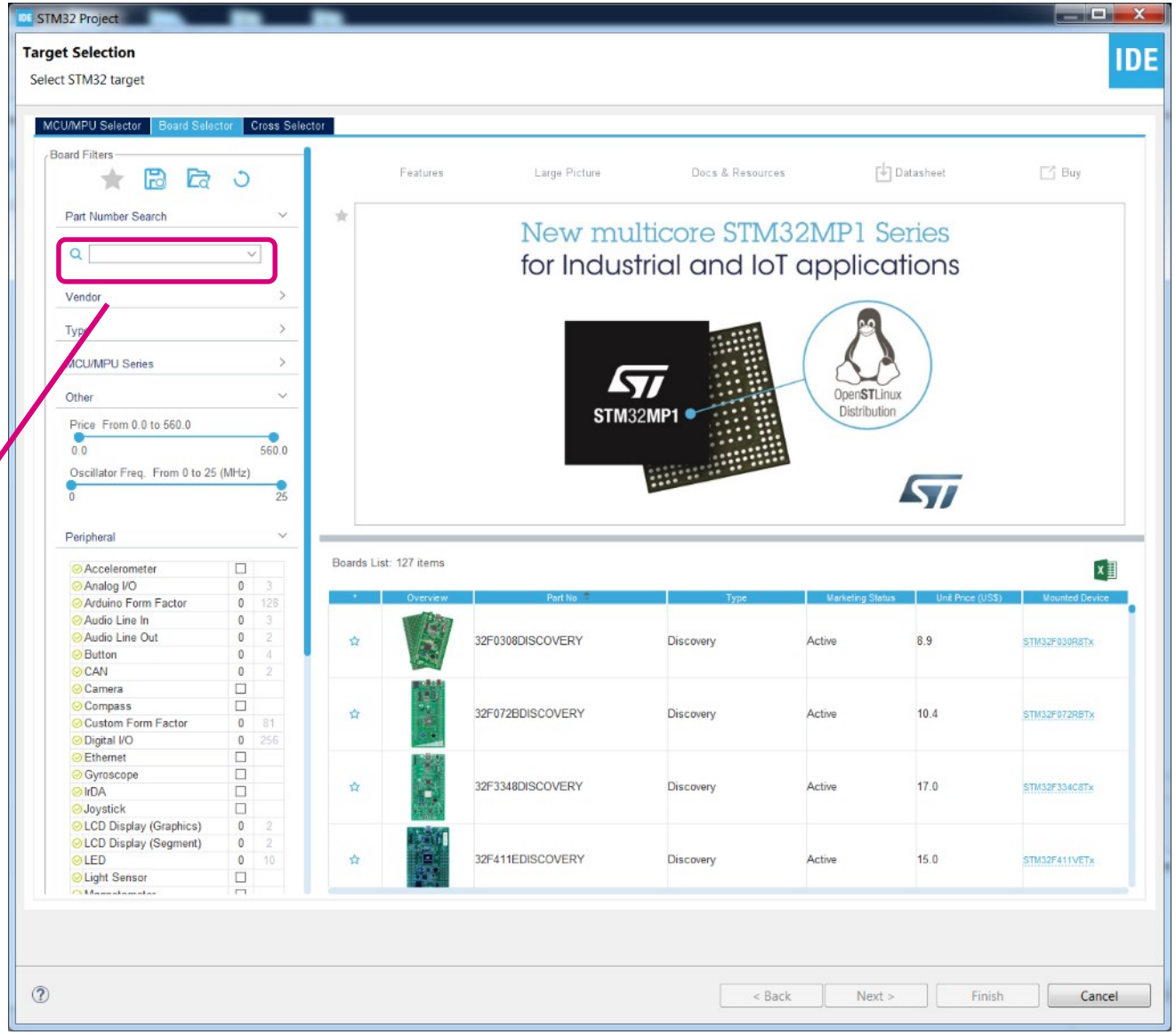




ST MCU board selector

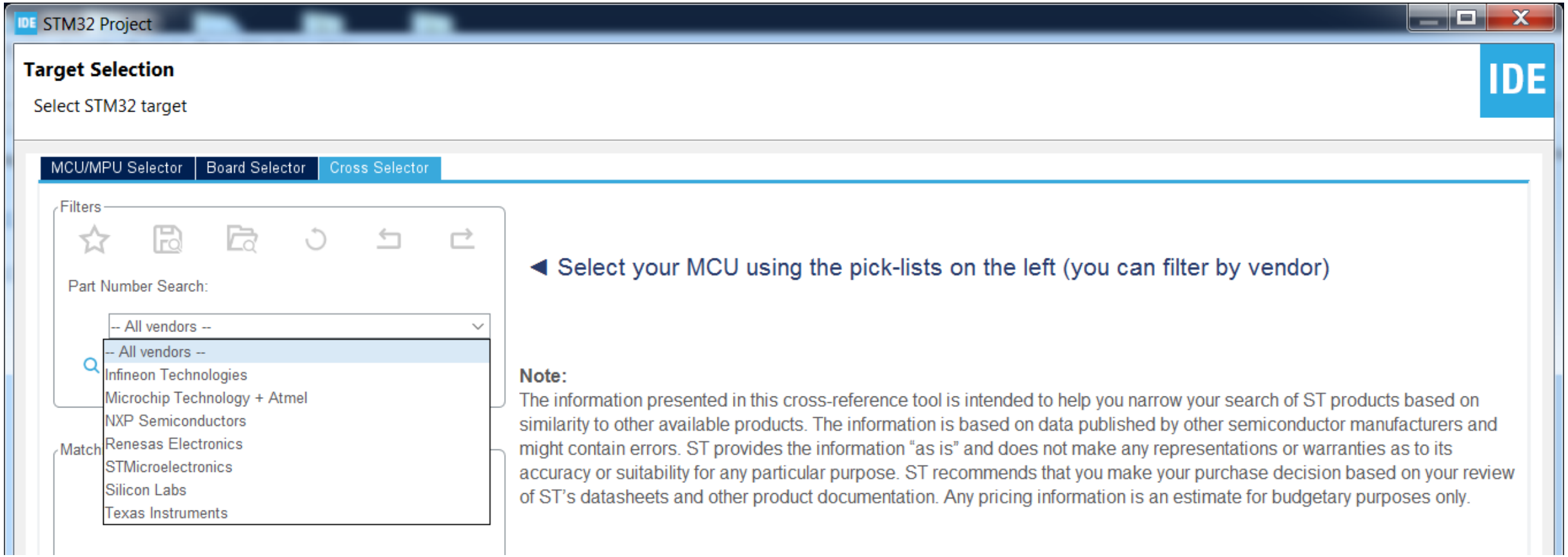
- We can select an ST board instead





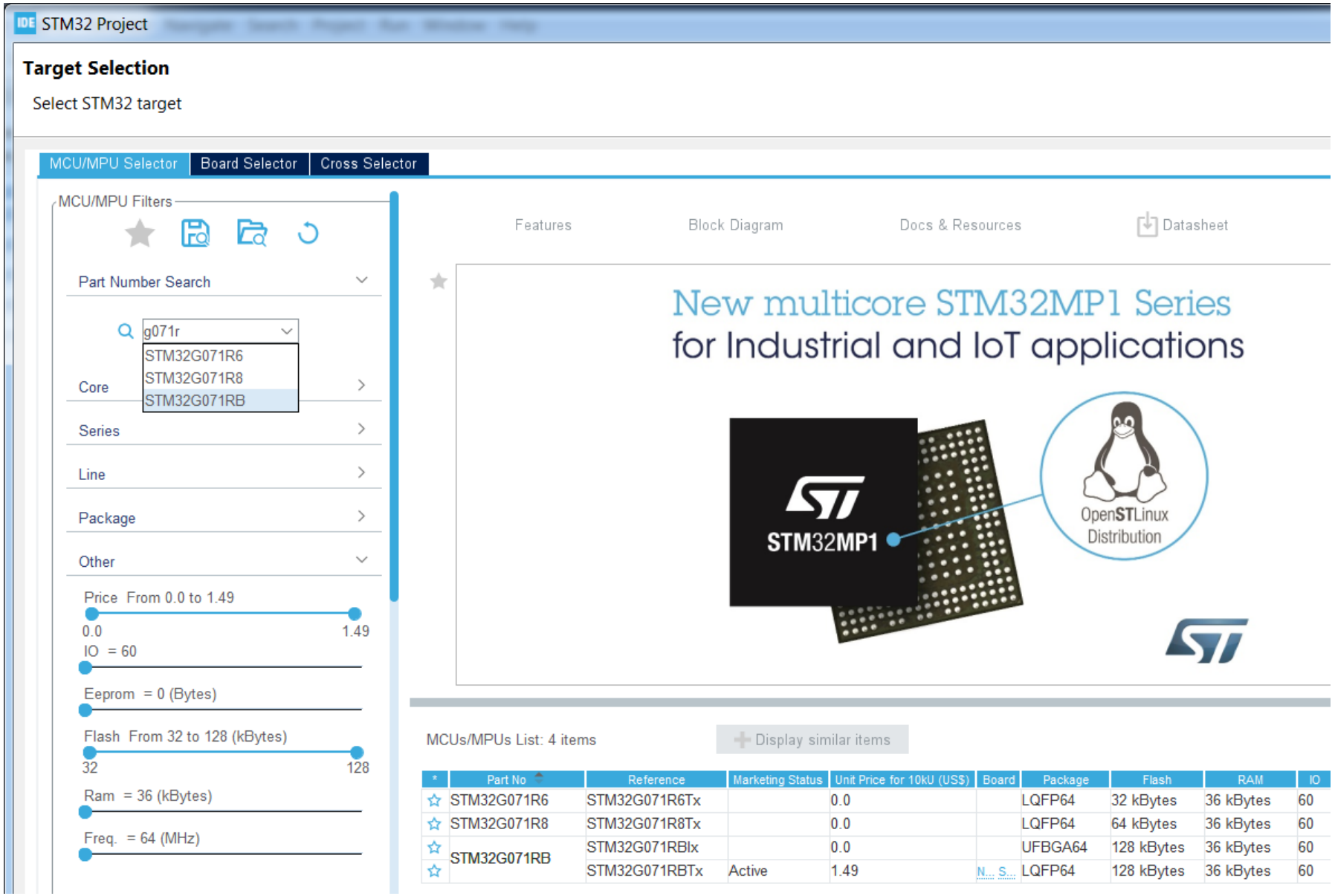
Cross reference selector

- It is possible to use a cross reference to select an STM32 device



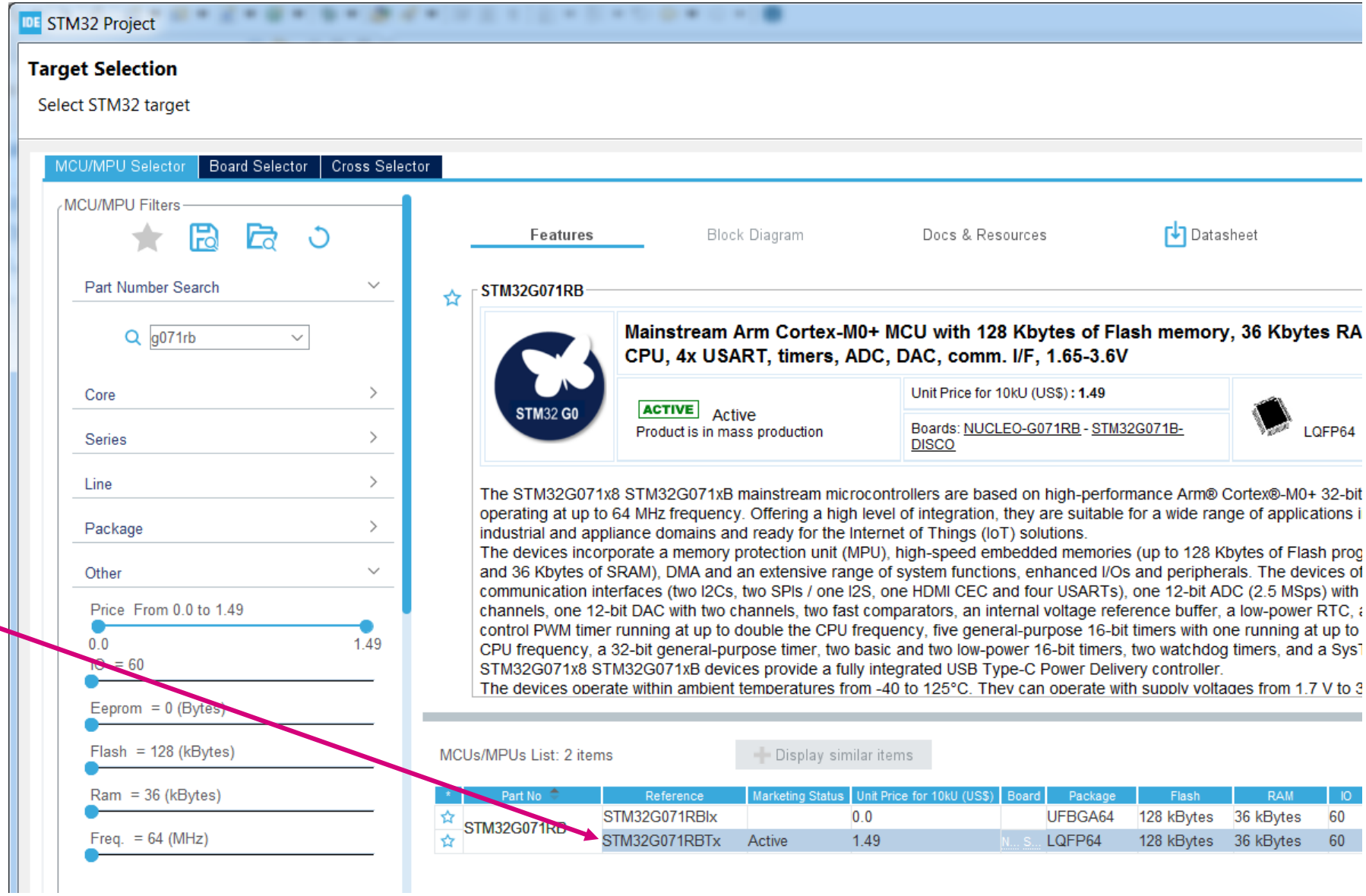
Select target MCU: STM32G071RBTx

- We will use STM32G071RBTx MCU



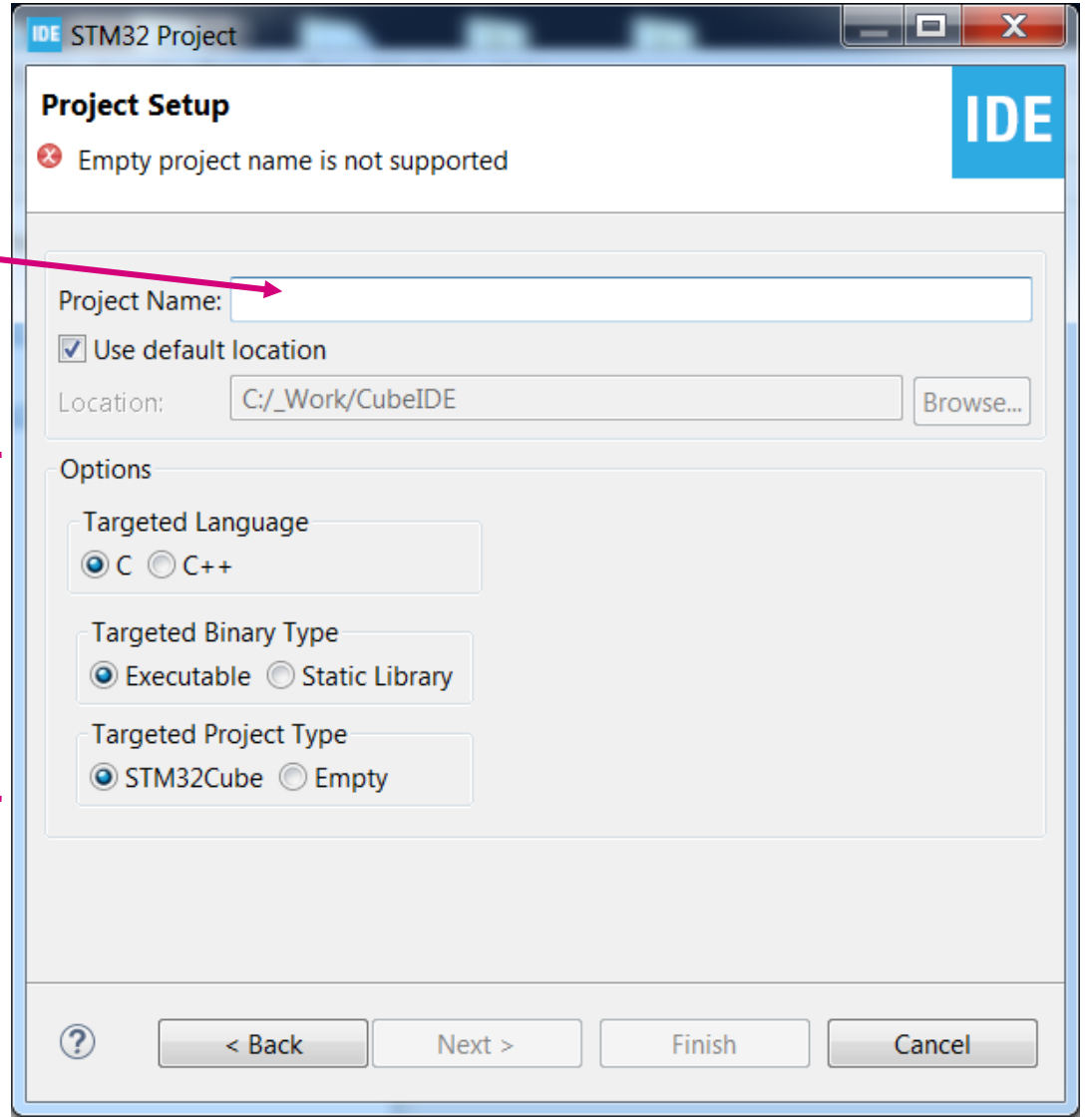
Select target MCU: STM32G071RBTx

- It is possible to view on main MCU features, download its documentation
- To start a new project we need to double click on the part number



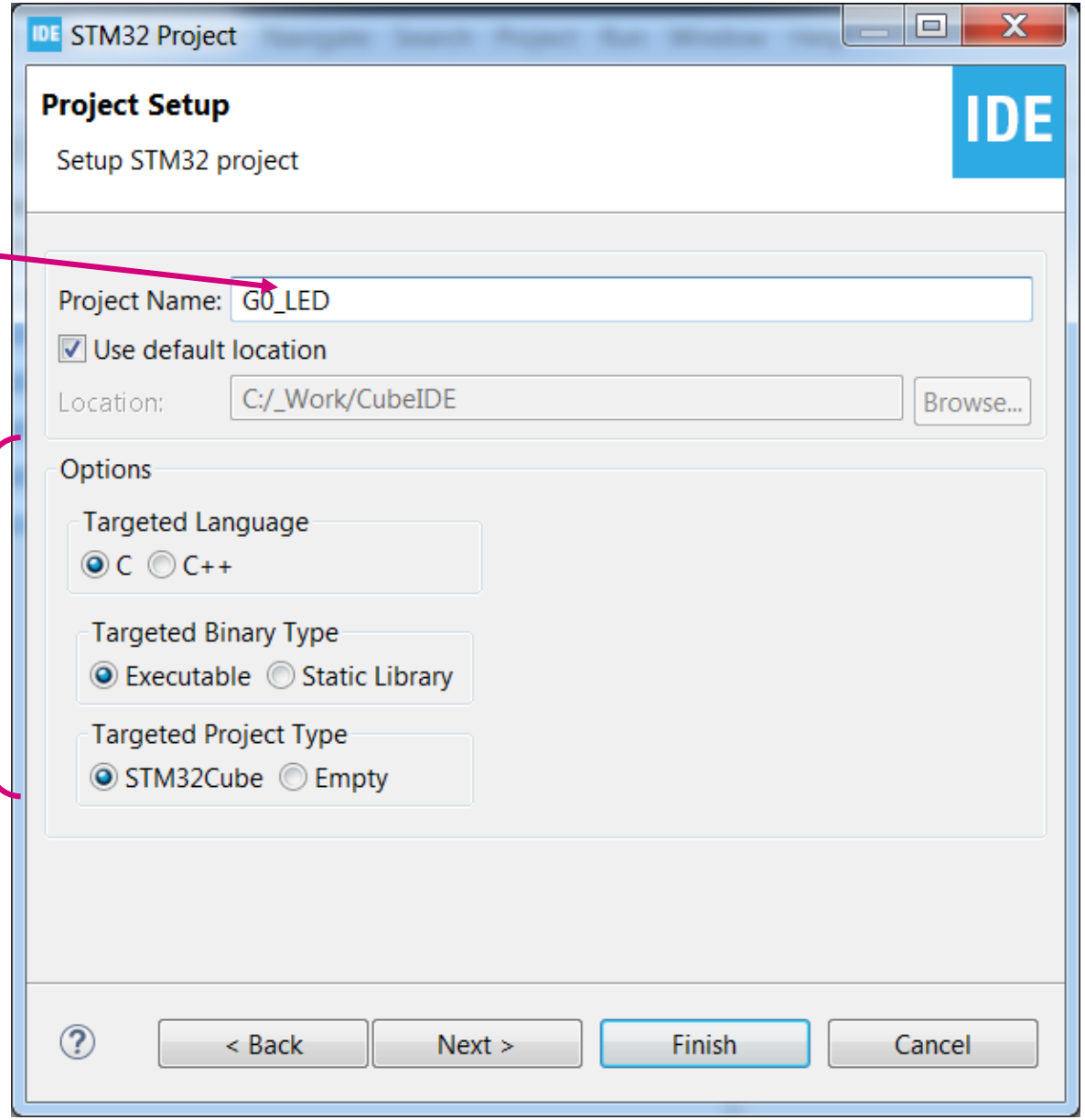
Enter project name

- Specify project name, optionally its location (if different from workspace one)
- Additionally we can specify target language (C or C++), binary type (executable or static library) and project type (generated by STM32CubeMX or an empty one)



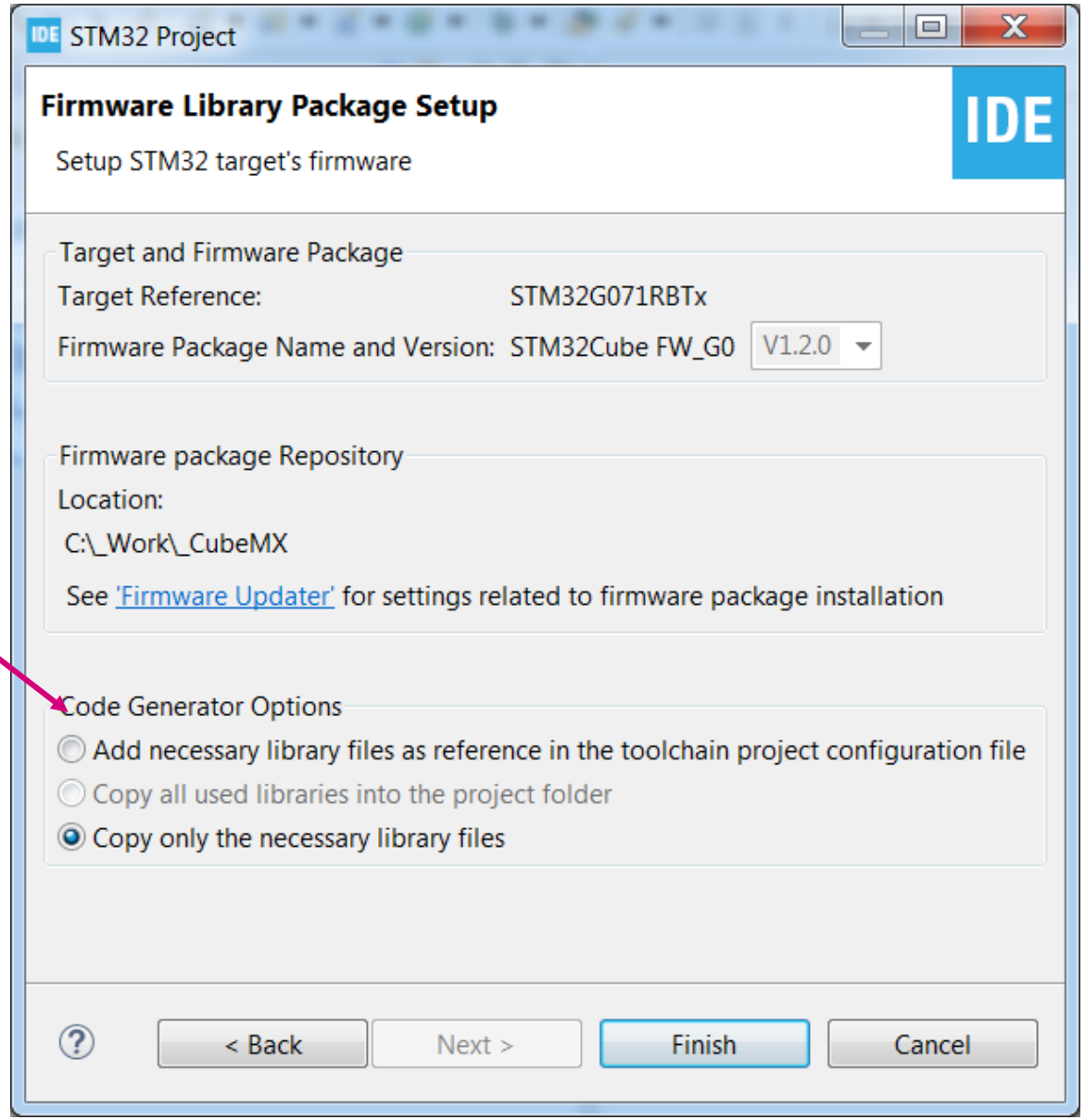
Enter project name

- Specify project name, optionally its location (if different from workspace one)
- Additionally we can specify target language (C or C++), binary type (executable or static library) and project type (generated by STM32CubeMX or an empty one)



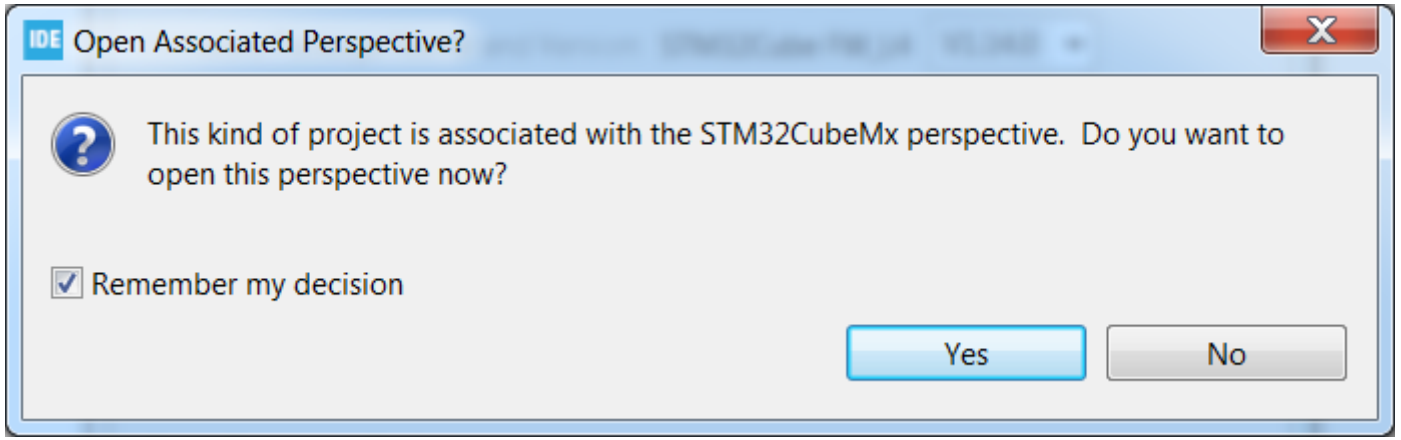
Select code generation options

- It is possible to specify code generator options (which library files would be added to the project)



Confirm change of perspective

- At the end of project generation there is an information window about change of perspective to STM32CubeMX one



New project in STM32CubeMX perspective

CubeIDE - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- G0_LED
  - Includes
  - Drivers
  - Src
  - Startup
  - Inc
  - G0_LED.ioc
  - STM32G071RBTX_FLASH.ld

An empty template project

G0_LED.ioc

Pinout & Configuration | Clock Configuration

Additional Software | Pinout

Categories | A->Z

System Core
Analog
Timers
Connectivity
Multimedia
Computing
Middleware

Pinout view

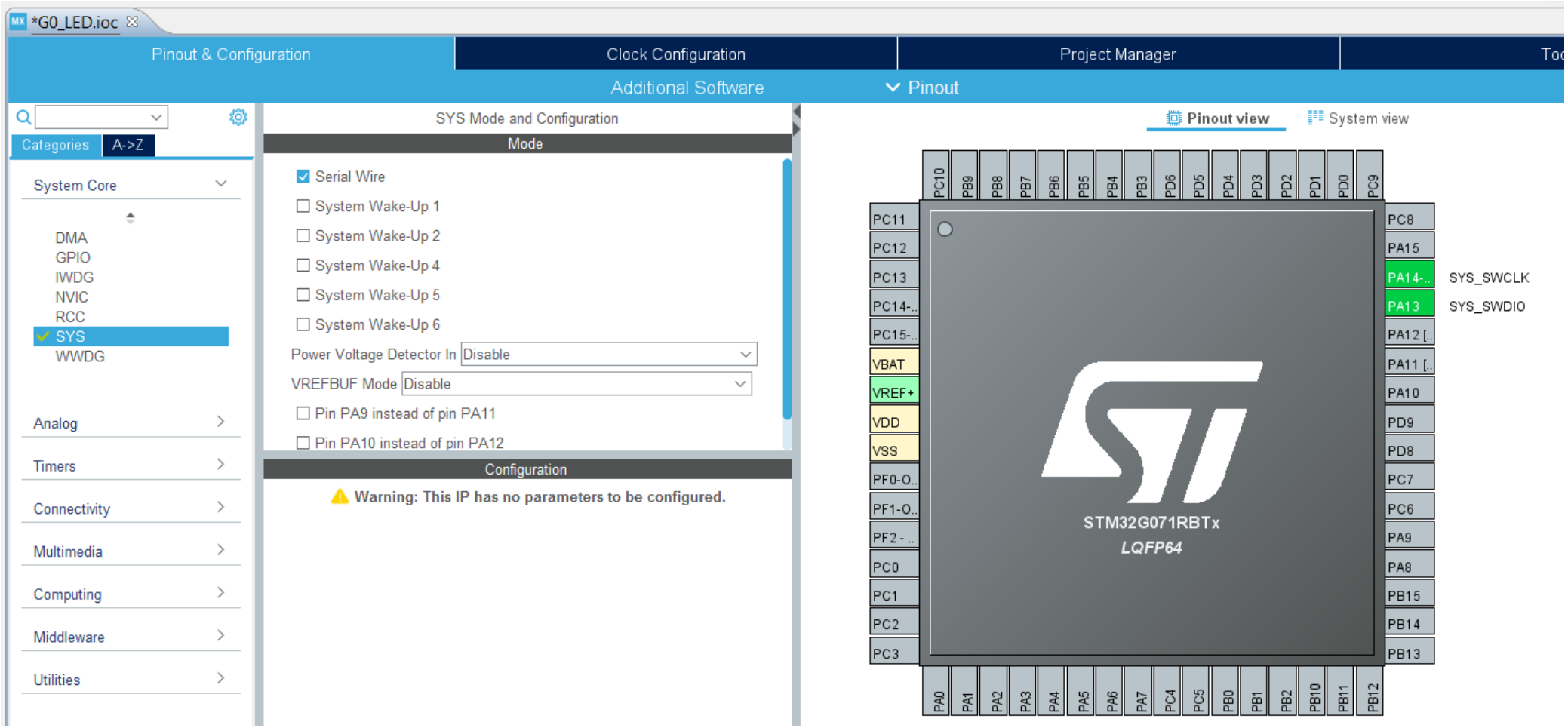
PC10 PB9 PB8 PB7 PB6 PB5 PB4 PB3 PD6 PD5 PD4 PD3 PD2 PD1 PD0 PC9

PC11 PC12 PC13 PC14-.. PC15-.. VBAT VREF+ VDD VSS

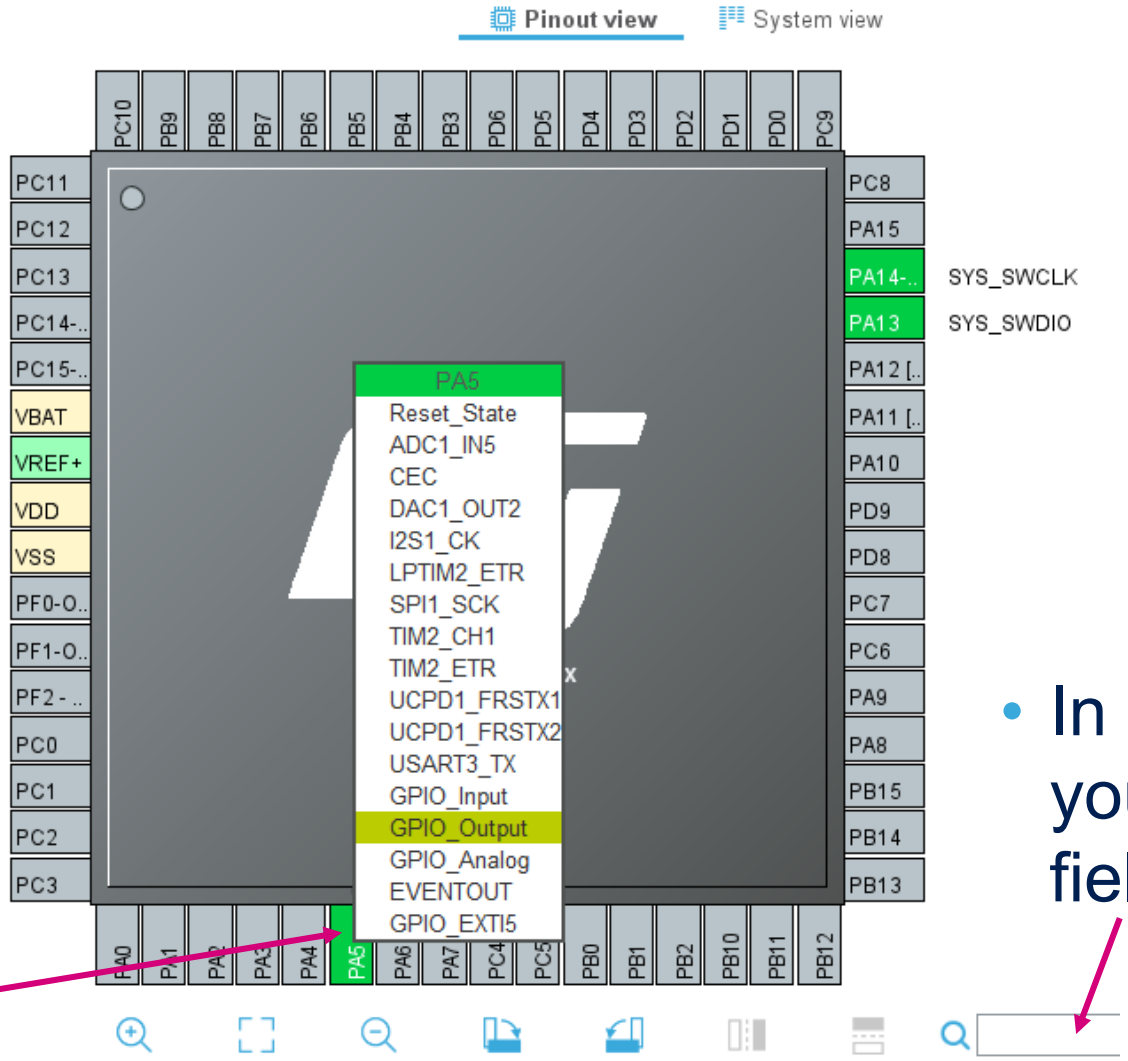
PC8 PA15 PA14-.. PA13 PA12 [.. PA11 [.. PA10 PD9 PD8

Enabling Serial Wire debug interface

- Select “Serial Wire” from System Core -> SYS peripheral group
- As a result PA13 and PA14 will be assigned to SWD interface



Configuring PA5 as Output

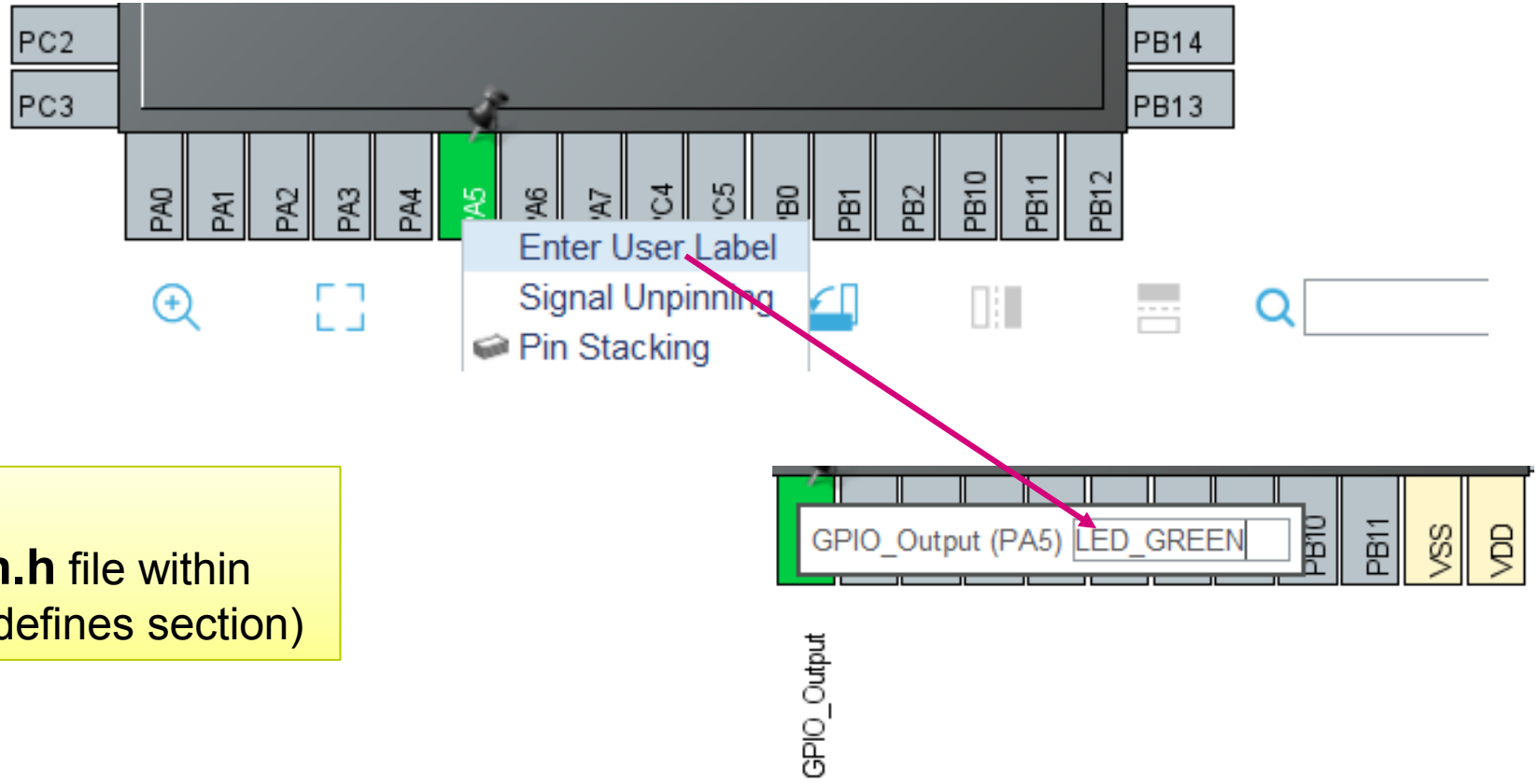


- Using configure PA5 as GPIO_Output

- In bigger packages you can use search field to locate the pin

Assign label to PA5

- Using select Enter User Label and insert LED_GREEN label



Hint:
Labels are defined in **main.h** file within generated project (private defines section)

Default clock configuration – no change

CubeIDE - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Quick Access

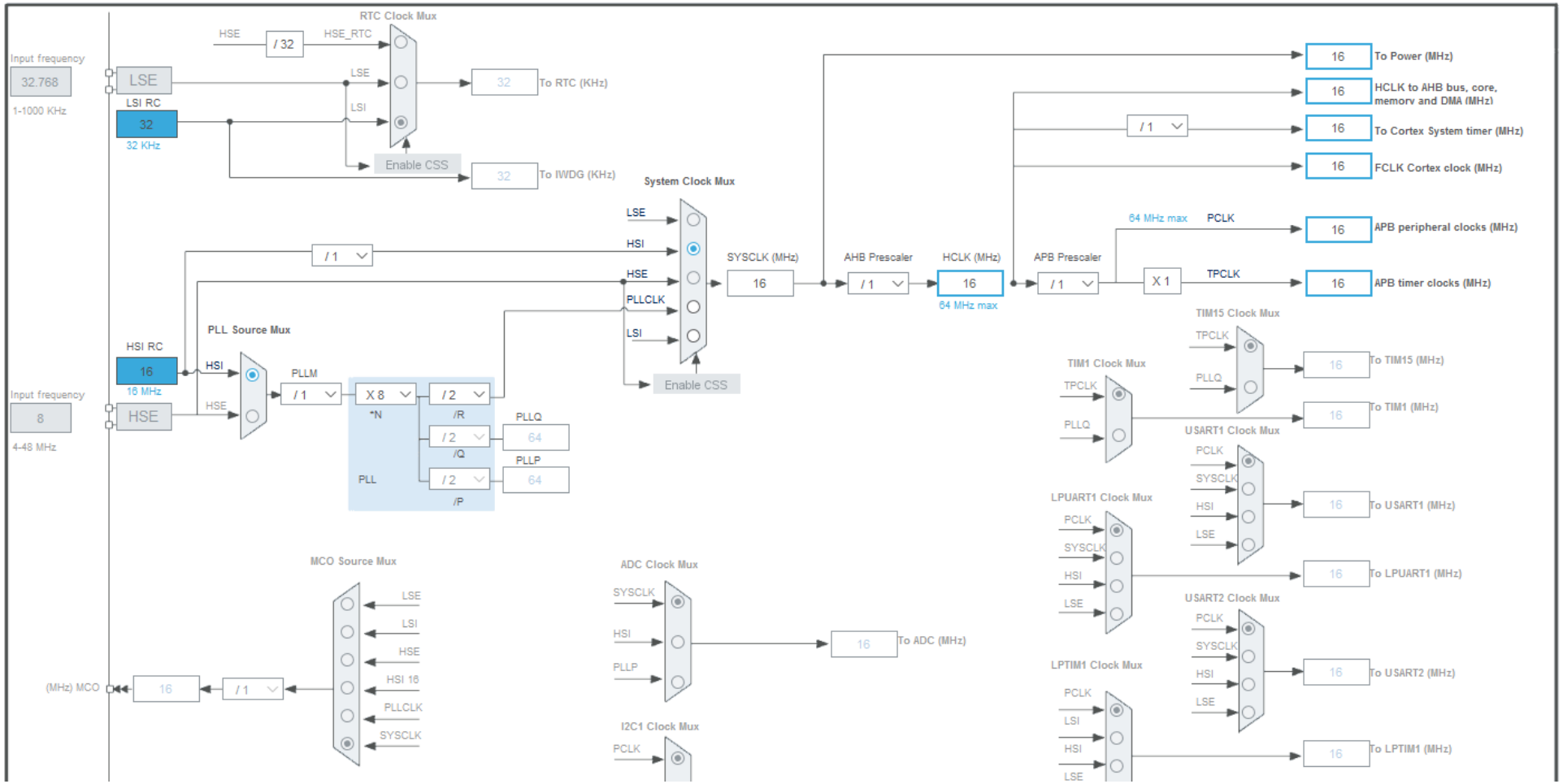
Project Explorer

- G0_LED
  - Includes
  - Drivers
    - CMSIS
    - STM32G0xx_HAL_Driver
  - Src
    - main.c
    - stm32g0xx_hal_msp.c
    - stm32g0xx_it.c
    - syscalls.c
    - sysmem.c
    - system_stm32g0xx.c
  - Startup
    - startup_stm32g071rbtx.s
  - Inc
    - main.h
    - stm32g0xx_hal_conf.h
    - stm32g0xx_it.h
  - G0_LED.ioc
  - STM32G071RBTX_FLASH.ld

*G0_LED.ioc

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Resolve Clock Issues



Basic project settings – no change

CubeIDE - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- G0_LED
  - Includes
  - Drivers
    - CMSIS
    - STM32G0xx_HAL_Driver
  - Src
    - main.c
    - stm32g0xx_hal_msp.c
    - stm32g0xx_it.c
    - syscalls.c
    - sysmem.c
    - system_stm32g0xx.c
  - Startup
    - startup_stm32g071rbtx.s
  - Inc
    - main.h
    - stm32g0xx_hal_conf.h
    - stm32g0xx_it.h
  - G0_LED.ioc
  - STM32G071RBTX_FLASH.ld

*G0_LED.ioc

Pinout & Configuration | Clock Configuration | Project Manager

Project | Code Generator | Advanced Settings

Project Settings

Project Name: G0_LED

Project Location: C:_Work\CubeIDE

Application Structure: Basic — ☐ Do not generate the main()

Toolchain Folder Location: C:_Work\CubeIDE\G0_LED\

Toolchain / IDE: STM32CubeIDE — ☑ Generate Under Root

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Mcu and Firmware Package

Mcu Reference: STM32G071RBTx

Firmware Package Name and Version: STM32Cube FW_G0 V1.2.0

C:/_Work/_CubeMX/STM32Cube_FW_G0_V1.2.0 — Browse

Code generation

- It is necessary to add to an empty template project our configuration done

CubeIDE - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Project Explorer

G0_LED
Includes
Drivers
CMSIS
STM32G0xx_HAL_
Src
main.c
stm32g0xx_hal_m
stm32g0xx_it.c
syscalls.c
sysmem.c
system_stm32g0xx.c
Startup
startup_stm32g071rbtx.s
Inc
main.h

Open Project
Close Project
Build All Ctrl+B
Build Configurations
Build Project
Build Working Set
Clean...
Build Automatically
C/C++ Index
Generate Report
Generate Code Alt+K
Properties

Configuration
Clock Configuration

Code Generator

ject Settings
ject Name
_LED
ject Location
_Work\CubeIDE
plication Structure
sic
Do not generate the main()
olchain Folder Location
C:_Work\CubeIDE\G0_LED\
Toolchain / IDE
STM32CubeIDE
Generate Under Root

Linker Settings
Minimum Heap Size 0x200

Adding code

- We can use Ctrl+Space help to find proper function or argument
- Please use “USER CODE” sections only

```
94
95    /* Infinite loop */
96    /* USER CODE BEGIN WHILE */
97    while (1)
98    {
99      HAL_GPIO_TogglePin(LED_, GPIO_Pin)
100     /* USER CODE END WHILE
101
102     /* USER CODE BEGIN 3 */
103   }
104   /* USER CODE END 3 */
105 }
106
107 /**
108   * @brief System Clock Con
109   * @retval None
110   */
```

void HAL_GPIO_TogglePin(**GPIO_TypeDef * GPIOx**, uint16_t GPIO_Pin)

- # LED_GREEN_GPIO_Port
- # LED_GREEN_Pin
- LL_RCC_PLL_EnableDomain_ADC(void) : void
- LL_RCC_PLL_EnableDomain_I2S1(void) : void
- LL_RCC_PLL_EnableDomain_SYS(void) : void
- LL_RCC_PLL_EnableDomain_TIM1(void) : void
- LL_RCC_PLL_EnableDomain_TIM15(void) : void
- # LL_RCC_LSEDRIVE_HIGH
- # LL_RCC_LSEDRIVE_LOW
- # LL_RCC_LSEDRIVE_MEDIUMHIGH
- # LL_RCC_LSEDRIVE_MEDIUMLOW
- # LPTIM_CFGR_PRELOAD_Msk

Press 'Ctrl+Space' to show Template Proposals

Console

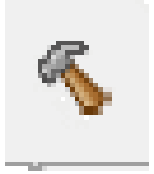
No consoles to display at this time.

Final code (main.c file)

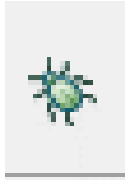
- We need to toggle LED_GREEN pin with 500ms delay in between

```
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    HAL_GPIO_TogglePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin);
    HAL_Delay(500);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```

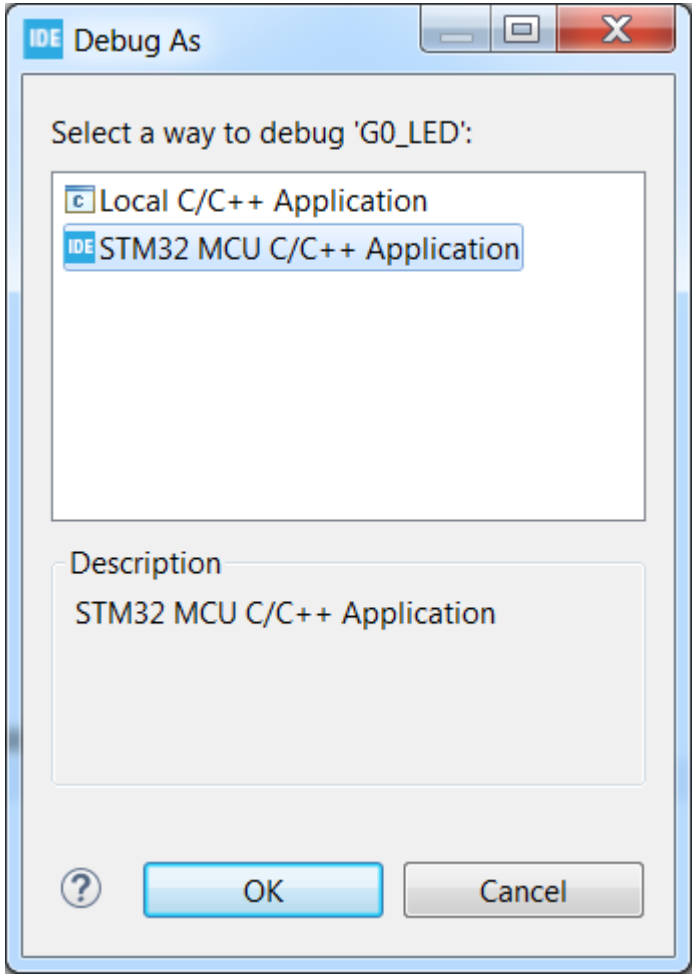
Compile and run debug session



- We can compile the code with “hammer” or build with automatic start of debug session using “bug” button

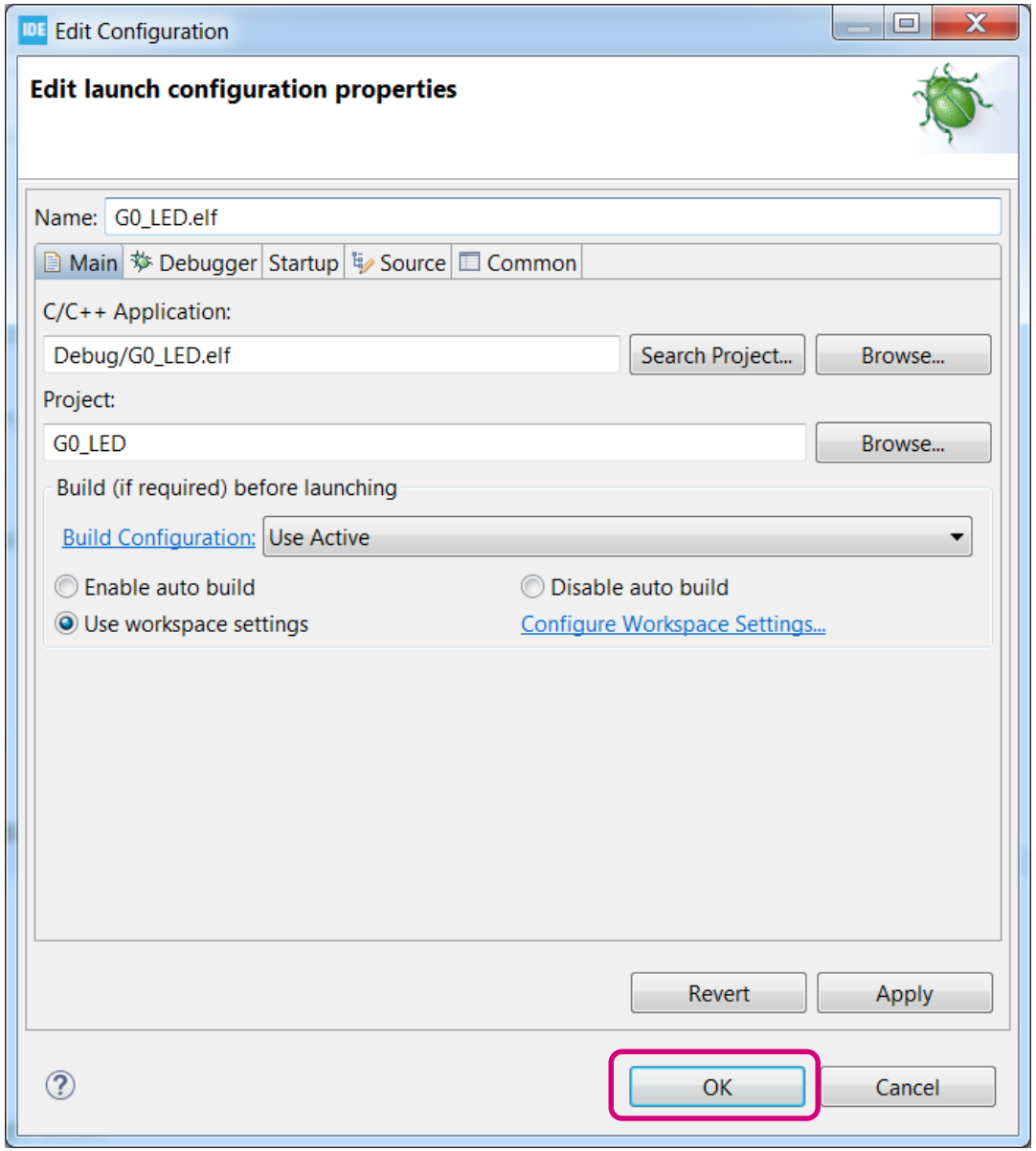


- Starting debug session we will be asked how we would like to debug the code



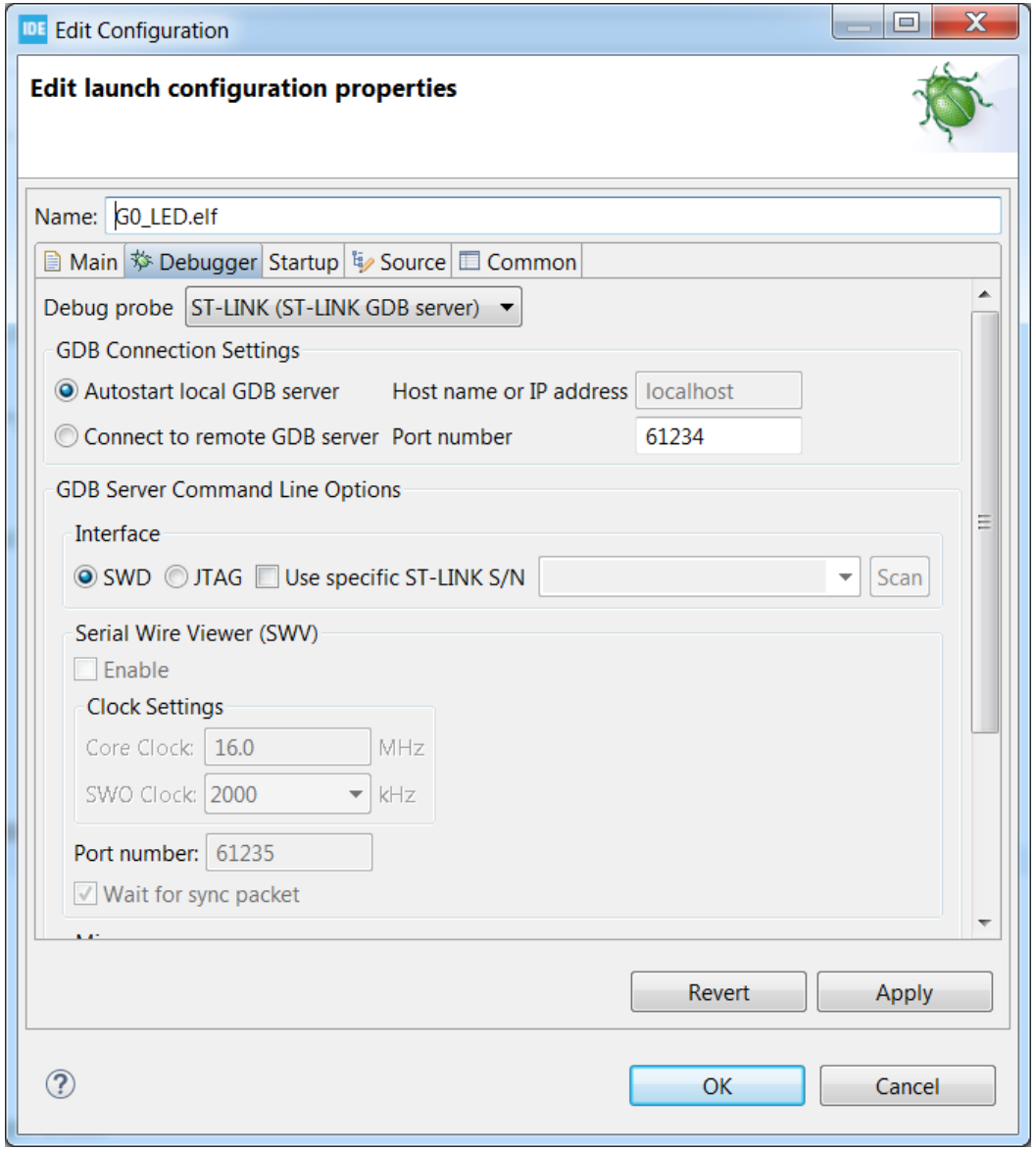
Debug settings 1/3

- When running debug (after successful compilation) there is no need to configure anything for debug
- Just connect target board to STLink and press ok



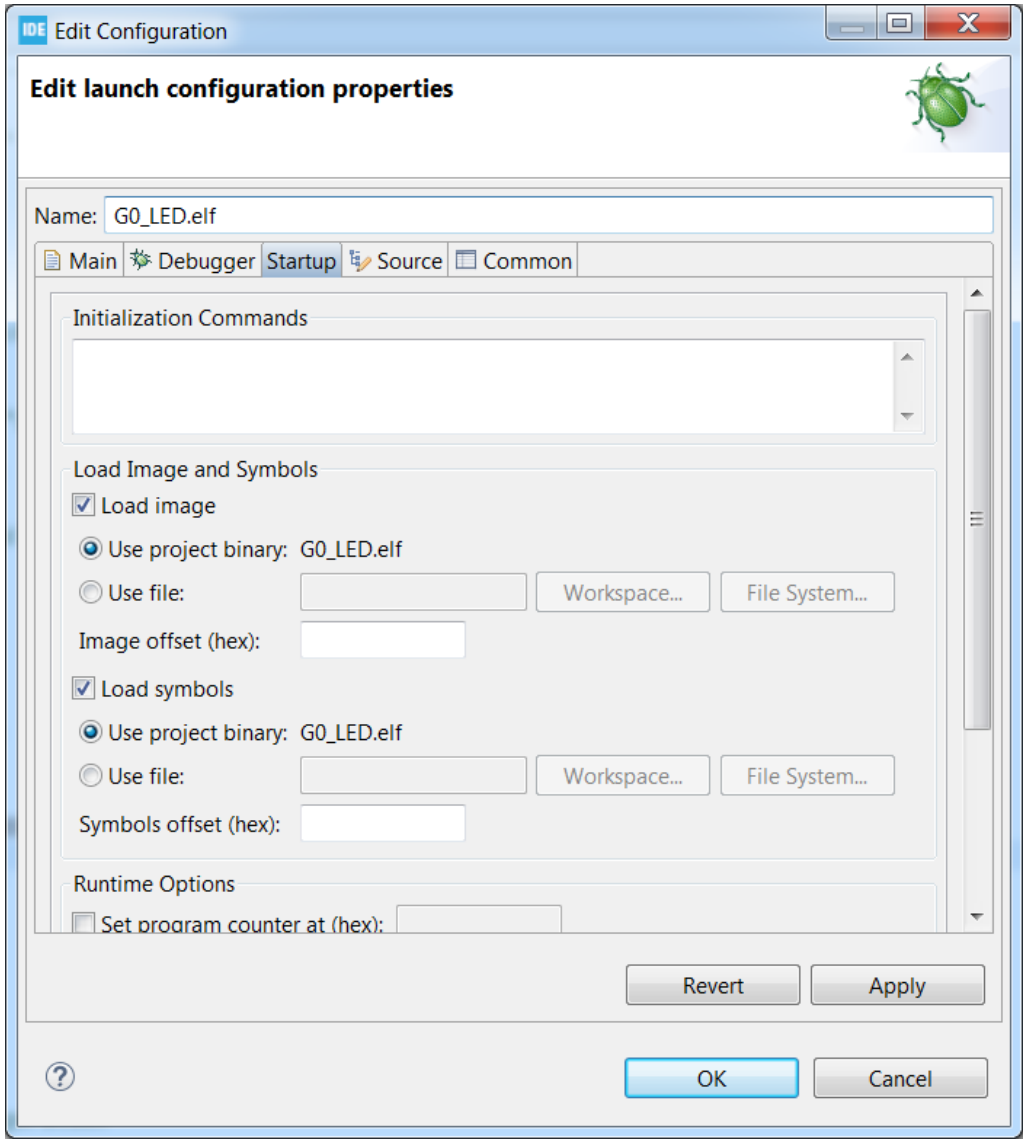
Debug settings 2/3

- It is possible to select debug probe and its interface



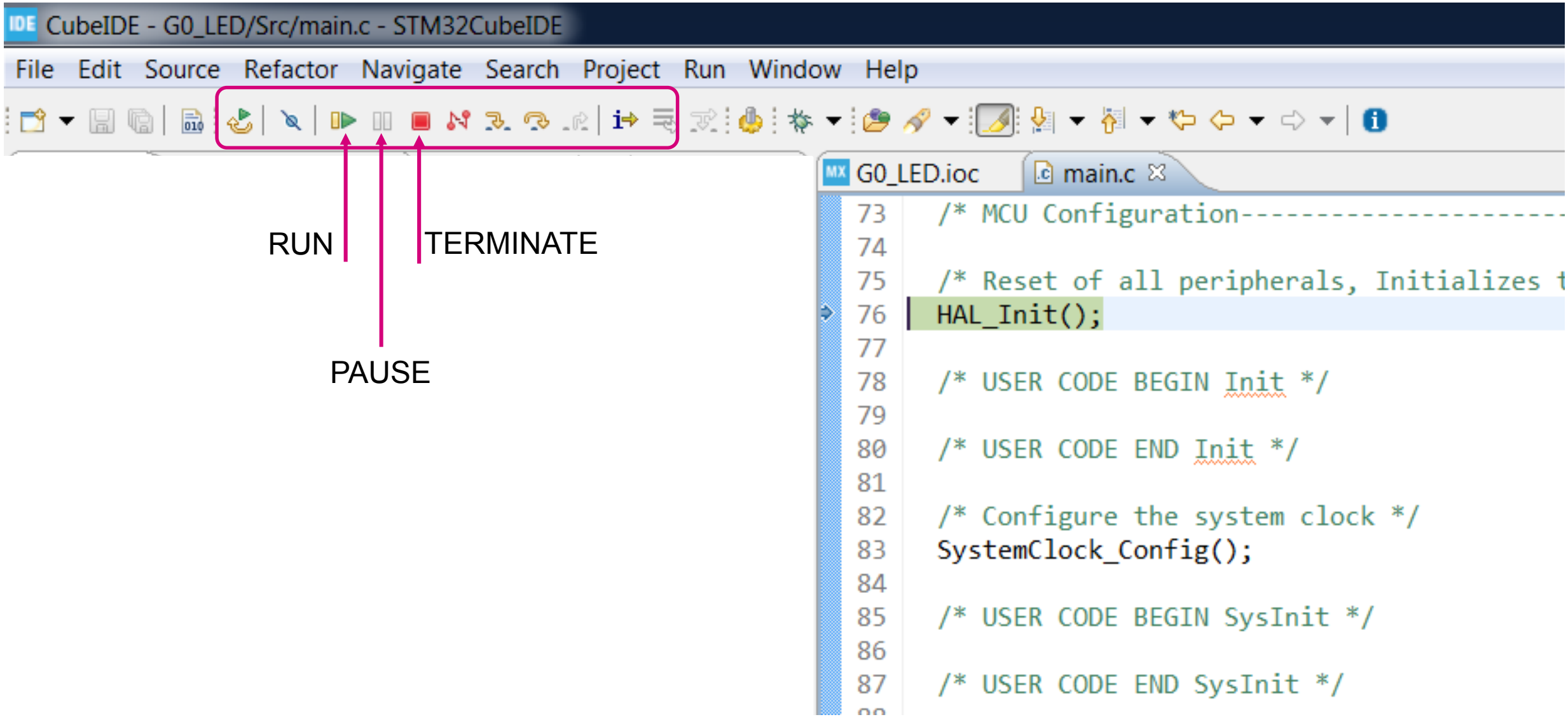
Debug settings 3/3

- Some more advanced commands are possible as well



Debug perspective (toolbar)

- Once we enter into debug session, a new debug toolbar pop up



Debug perspective – registers view

- Within debug perspective it is possible to monitor code flow, variables and registers content (on application pause, breakpoint stop or live)

CubeIDE - G0_LED/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug | Project Explorer

- G0_LED.elf [STM32 MCU Debugging]
  - G0_LED.elf [cores: 0]
    - Thread #1 [main] 1 [core: 0] (Running)
  - C:/ST/STM32CubeIDE_1.0.0/STM32CubeIDE/plugins/com.st.str
  - ST-LINK (ST-LINK GDB server)

G0_LED.ioc | main.c

```
73  /* MCU Configuration----------------------------------------
74
75  /* Reset of all peripherals, Initializes the Flash interface and
76  HAL_Init();
77
78  /* USER CODE BEGIN Init */
79
80  /* USER CODE END Init */
81
82  /* Configure the system clock */
83  SystemClock_Config();
84
85  /* USER CODE BEGIN SysInit */
86
87  /* USER CODE END SysInit */
88
89  /* Initialize all configured peripherals */
90  MX_GPIO_Init();
91  /* USER CODE BEGIN 2 */
92
93  /* USER CODE END 2 */
94
```

Variables | Breakpoints | Expressions | Modules | Live Expressions | SFRs

type filter text

Register	Address	Value
IWDG		
WWDG		
FLASH		
DBG		
RCC		
PWR		
DMA		
DMAMUX		
GPIOA		
MODER	0x50000000	0xebfff7ff
OTYPER	0x50000004	0x0
OSPEEDR	0x50000008	0xc000000
PUPDR	0x5000000c	0x24000000
IDR	0x50000010	0x4020
ODR	0x50000014	0x20
BSRR	0x50000018	

Project / debug explorer

Source code

Variables, breakpoints, registers view

… Let’s check it

- After all code processing we can build the project, start debug session and run the application
- As an effect Green LED should toggle each 1second

STM32 CubeIDE

Thank you

/STM32

@ST_World

community.st.com

life.augmented

www.st.com/stm32g0, www.st.com/stm32cubeide