STM32CubeIDE basics

EXTI lab: Handling external interrupts using HAL libraries

life.augmented

STM32 CubeIDE

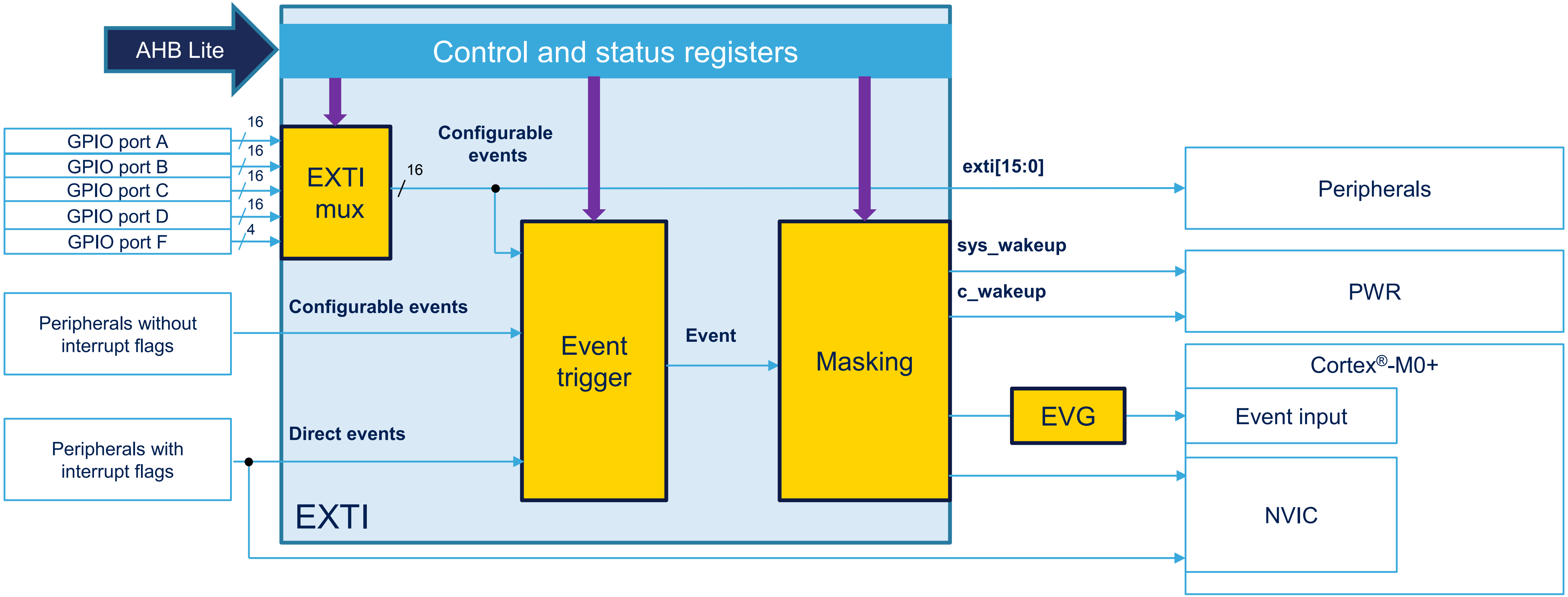
Lab: NVIC + External Interrupts

Objective:

- In this project we are going to configure the GPIO that is connected to the user button as External Interrupt (EXTI) with falling edge trigger.

- We will also configure the Interrupt Controller: the NVIC.

EXTI - block diagram

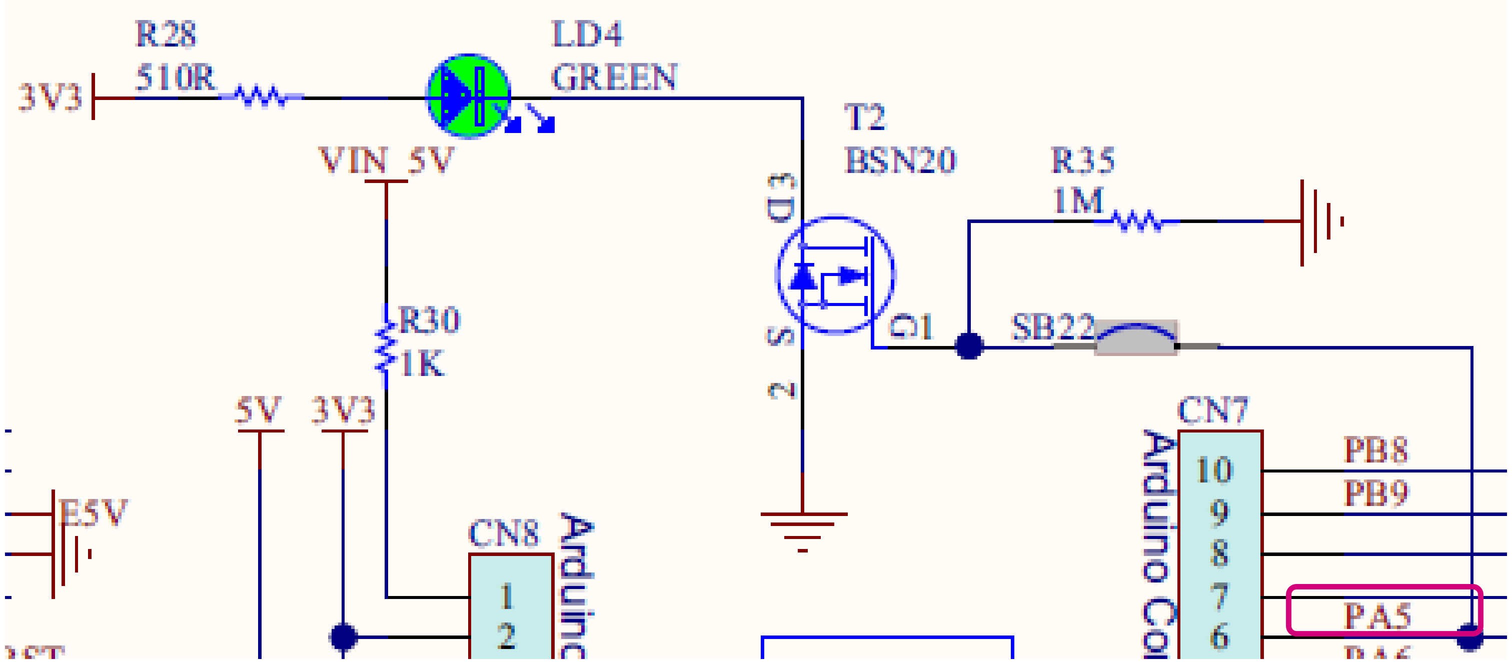


EVG: EVent Generator

Pin Configuration

Green LED

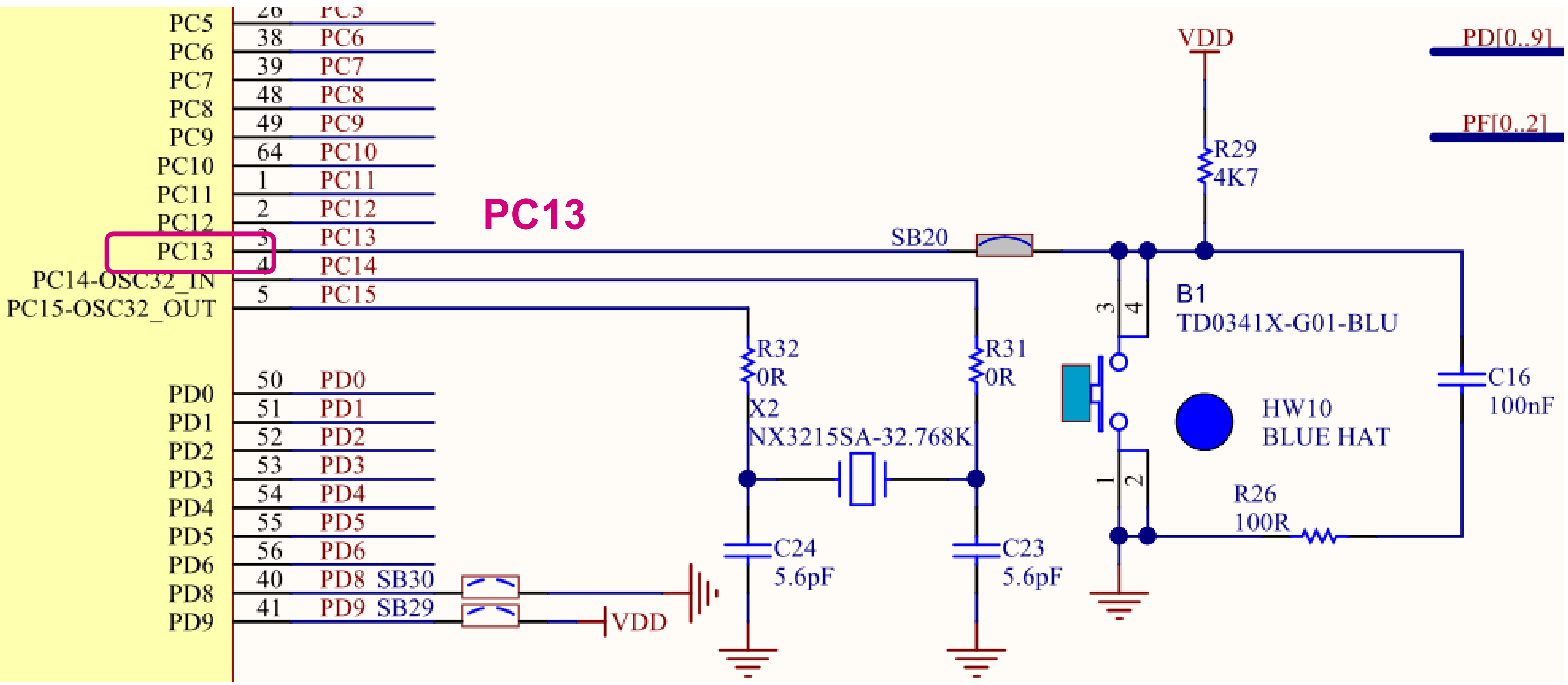
- In this example we are going to use one of the LEDs present on the STM32G0 Nucleo board (connected to PA5 as seen in the schematic below)



Pins Configuration

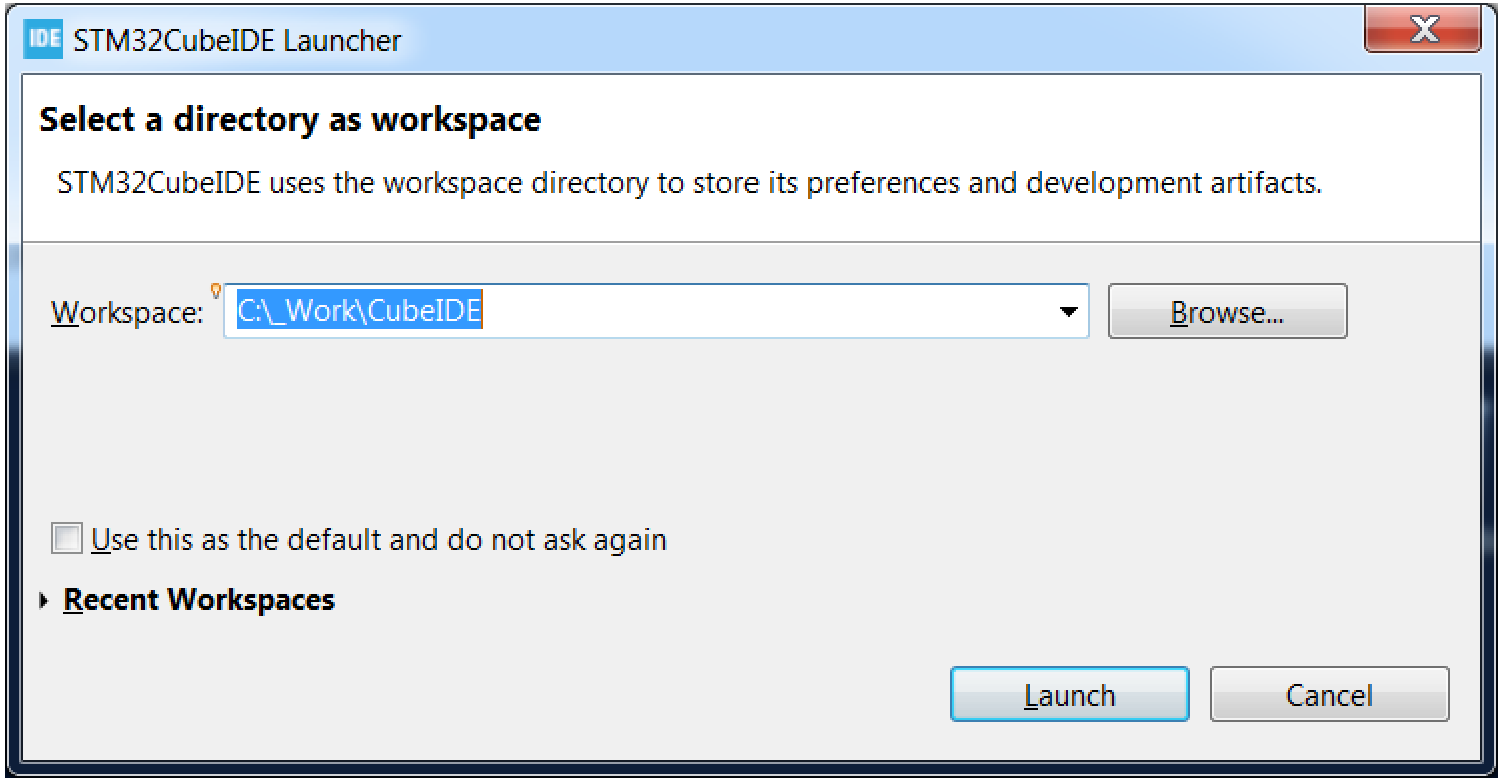
external interrupt button

- In this example we are going to use one button - blue one (connected to PA5 as seen in the schematic below)



Start a new workspace

- Run STM32CubeIDE
- Select a folder to store a workspace



STM32CubeIDE Launcher

Select a directory as workspace

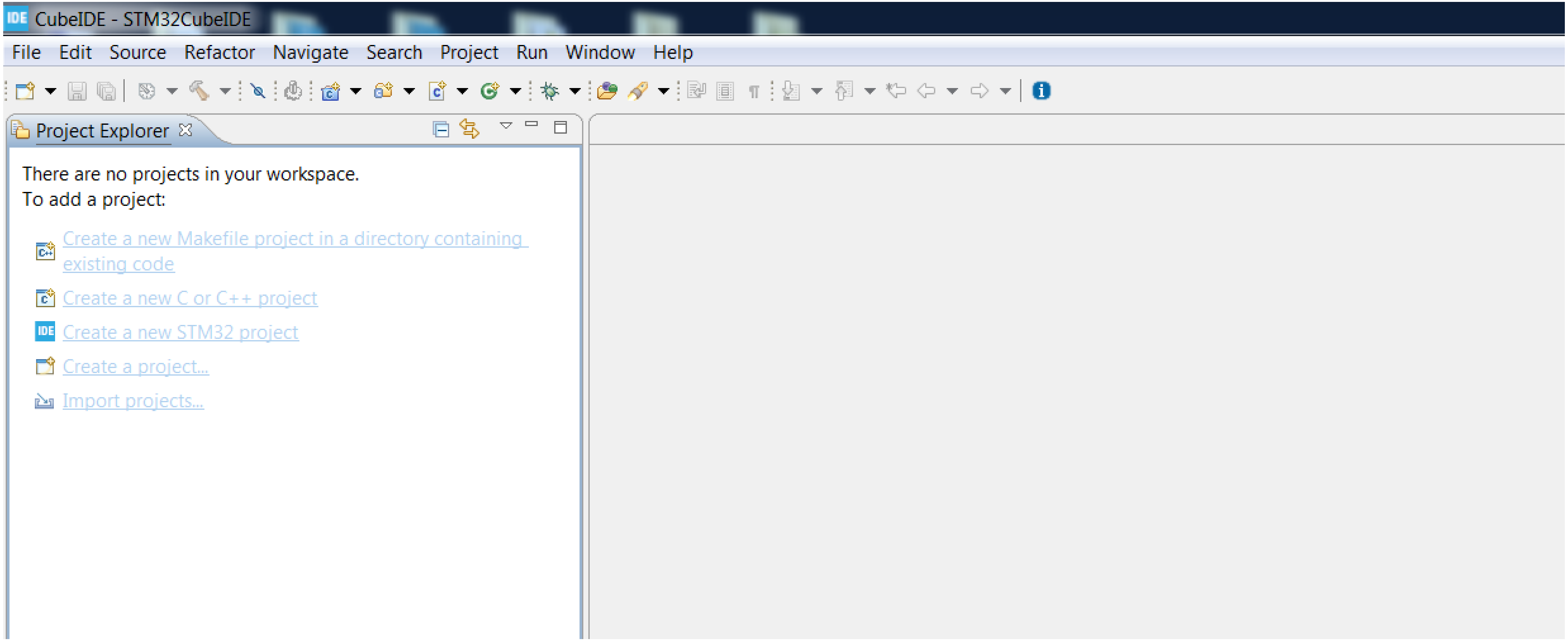
STM32CubeIDE uses the workspace directory to store its preferences and development artifacts.

Workspace: C:_Work\CubeIDE

☐ Use this as the default and do not ask again

Recent Workspaces

View on an empty workspace

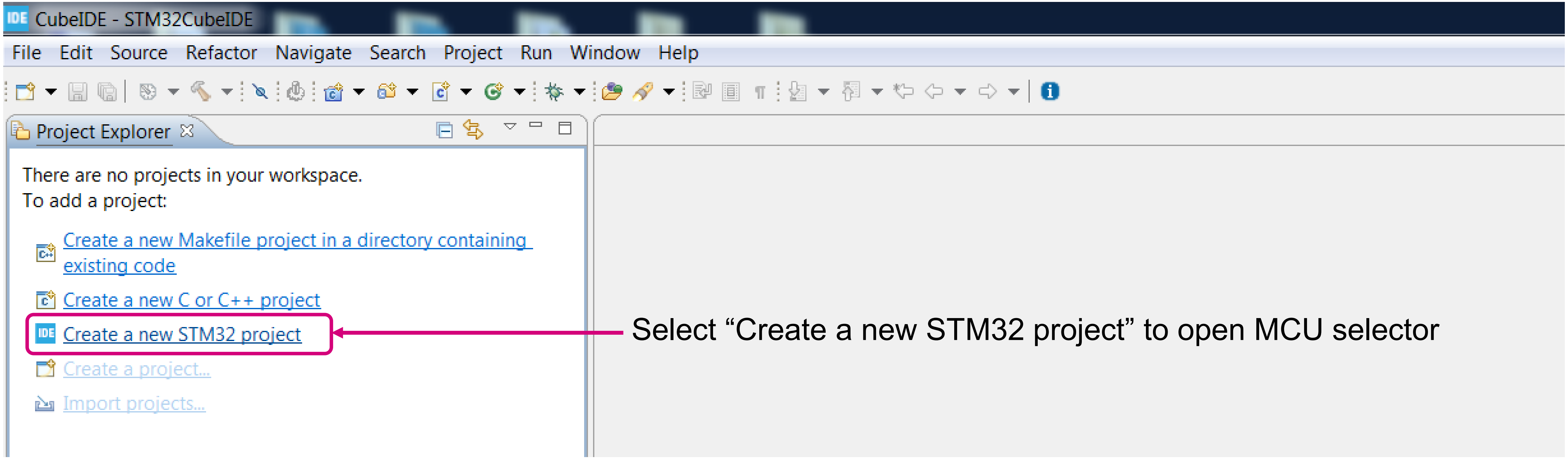

CubeIDE - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer

There are no projects in your workspace.
To add a project:

- Create a new Makefile project in a directory containing existing code
- Create a new C or C++ project
- Create a new STM32 project
- Create a project...
- Import projects...

Create a new project

- Click on “Create a new STM32 project”



CubeIDE - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer

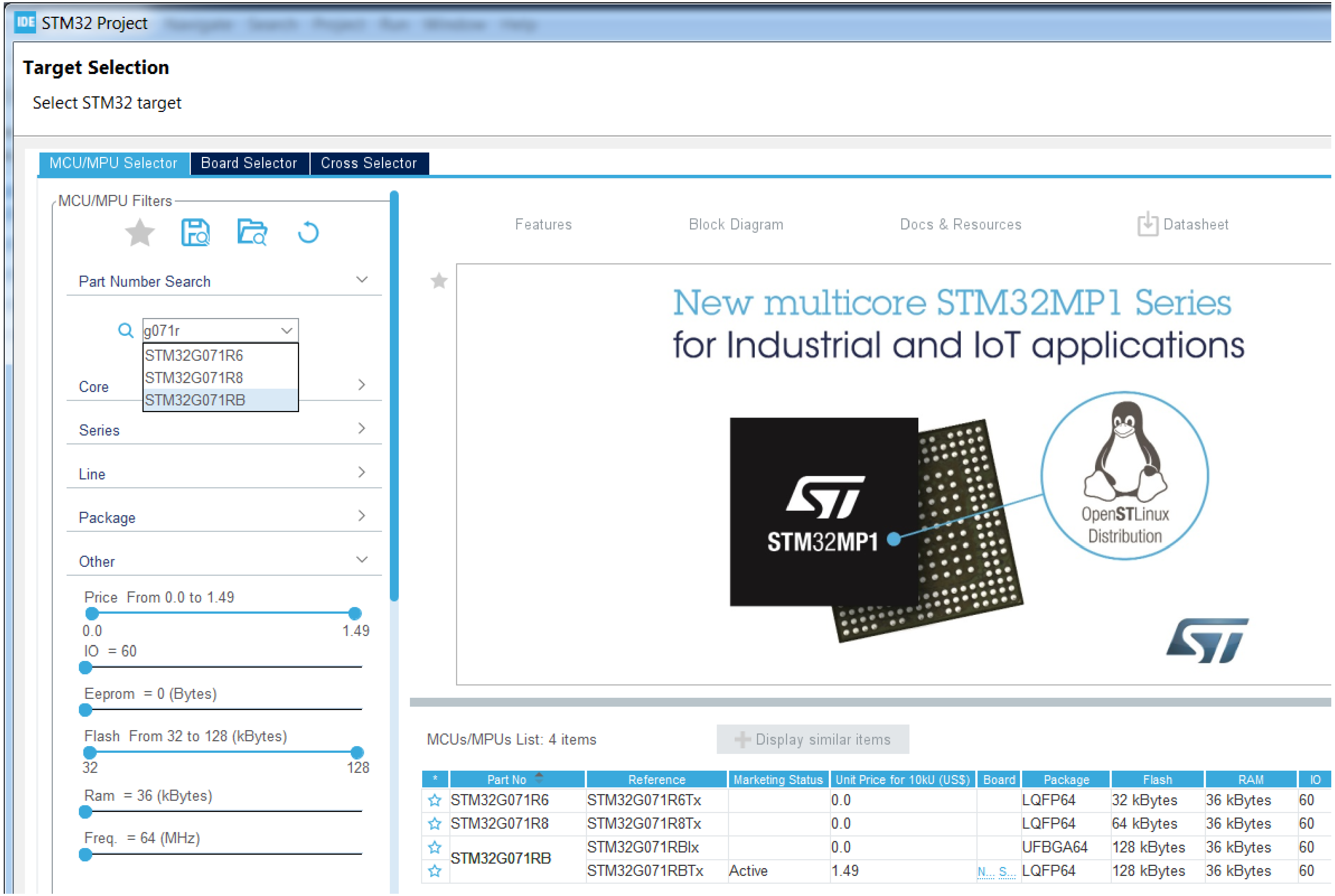
There are no projects in your workspace.
To add a project:

- Create a new Makefile project in a directory containing existing code
- Create a new C or C++ project
- Create a new STM32 project
- Create a project...
- Import projects...

Select “Create a new STM32 project” to open MCU selector

Select target MCU: STM32G071RBTx

- We will use STM32G071RBTx MCU



STM32 Project

Target Selection

Select STM32 target

g071r

STM32G071R6
STM32G071R8
STM32G071RB

Price From 0.0 to 1.49
0.0 1.49
IO = 60
Eeprom = 0 (Bytes)
Flash From 32 to 128 (kBytes)
32 128
Ram = 36 (kBytes)
Freq. = 64 (MHz)

MCUs/MPUs List: 4 items

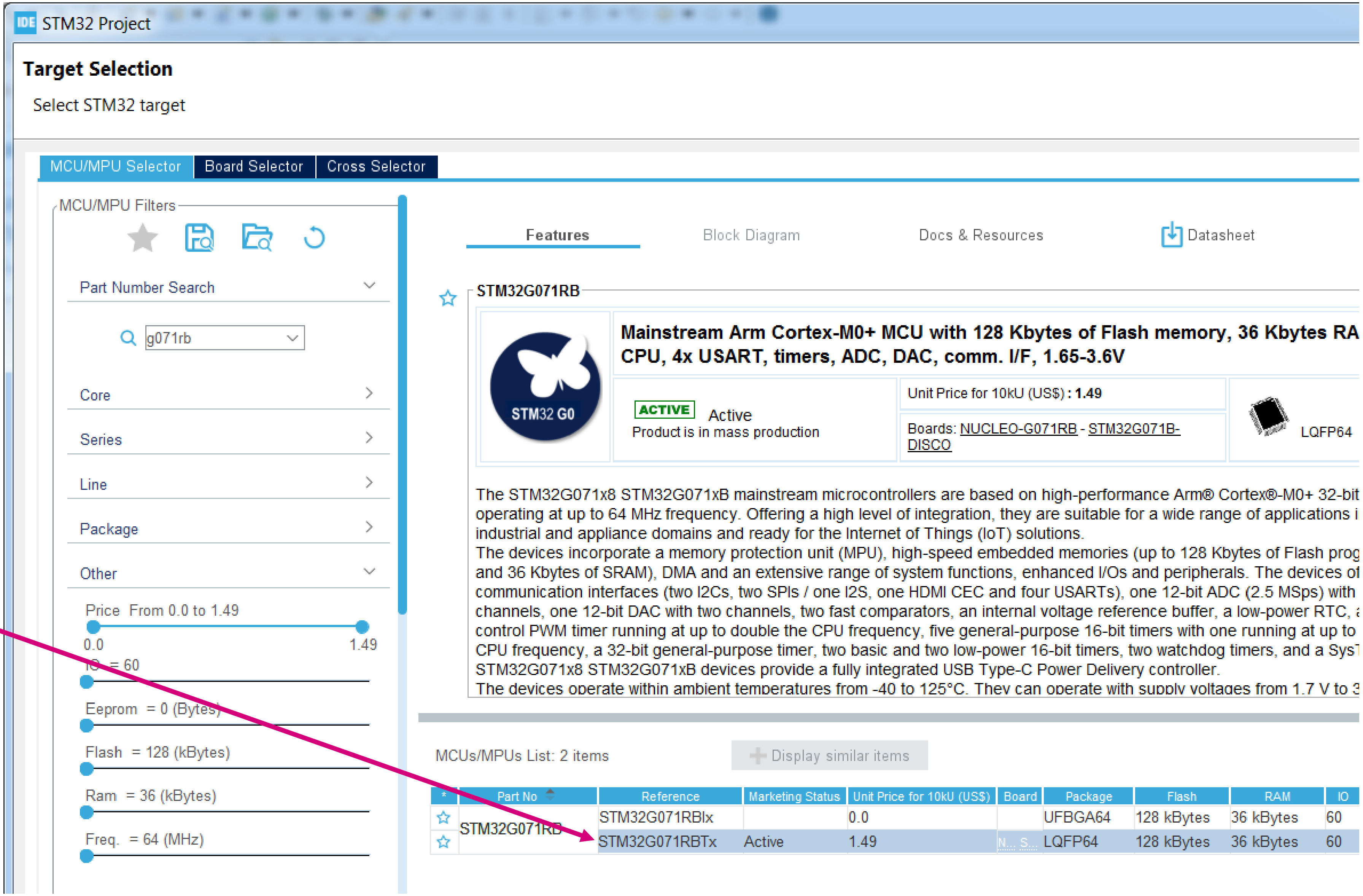
Display similar items

*	Part No	Reference	Marketing Status	Unit Price for 10kU (US$)	Board	Package	Flash	RAM	IO
☆	STM32G071R6	STM32G071R6Tx		0.0		LQFP64	32 kBytes	36 kBytes	60
☆	STM32G071R8	STM32G071R8Tx		0.0		LQFP64	64 kBytes	36 kBytes	60
☆	STM32G071RB	STM32G071RBIx		0.0		UFBGA64	128 kBytes	36 kBytes	60
☆		STM32G071RBTx	Active	1.49	N... S...	LQFP64	128 kBytes	36 kBytes	60

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



STM32 Project

Target Selection

Select STM32 target

MCU/MPU Filters

Part Number Search: g071rb

Core

Series

Line

Package

Other

Price From 0.0 to 1.49

0.0 1.49

IO = 60

Eeprom = 0 (Bytes)

Flash = 128 (kBytes)

Ram = 36 (kBytes)

Freq. = 64 (MHz)

STM32G071RB

Mainstream Arm Cortex-M0+ MCU with 128 Kbytes of Flash memory, 36 Kbytes RA CPU, 4x USART, timers, ADC, DAC, comm. I/F, 1.65-3.6V

ACTIVE Active
Product is in mass production

Unit Price for 10kU (US$) : 1.49

Boards: NUCLEO-G071RB - STM32G071B-DISCO

LQFP64

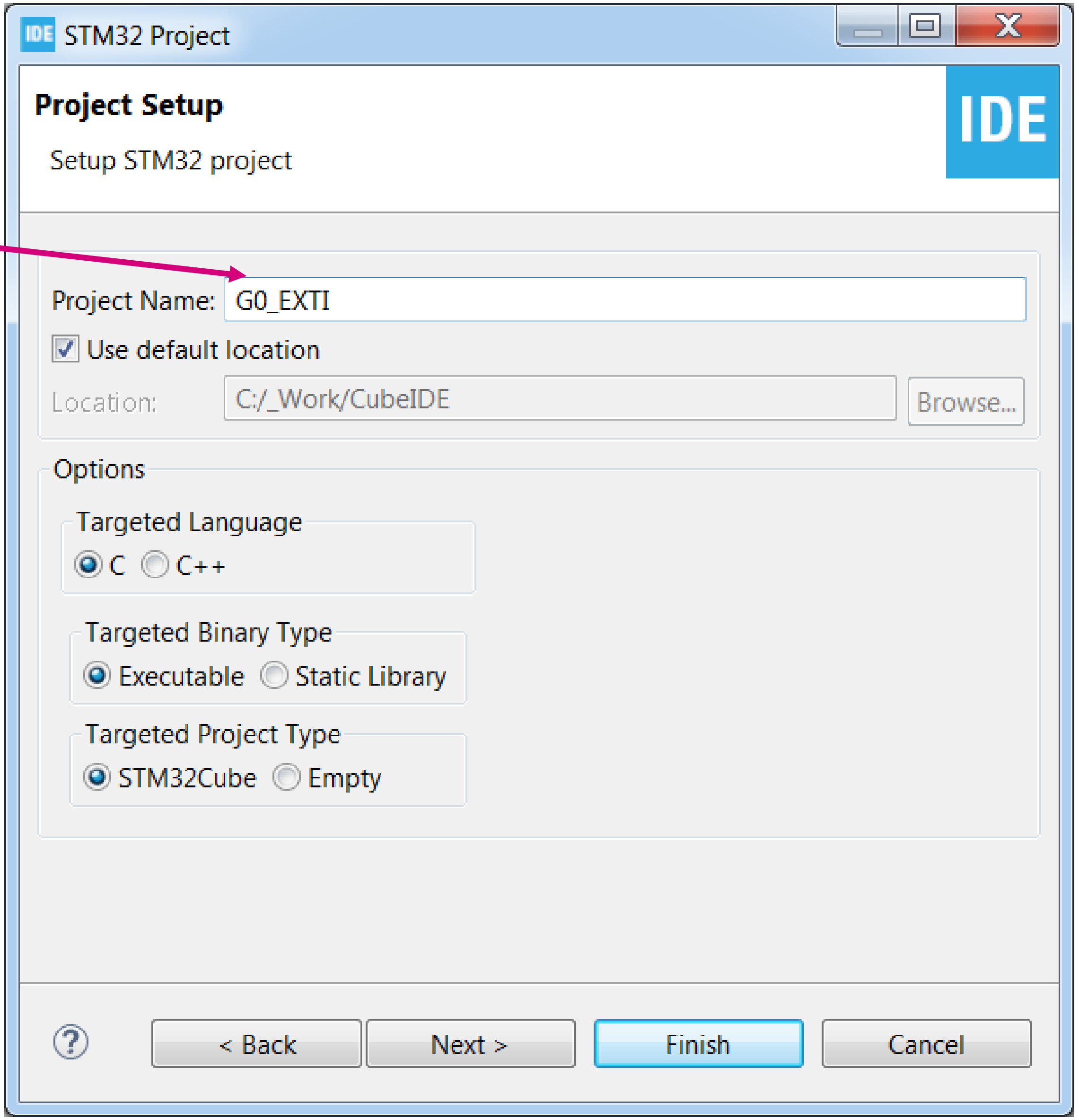
The STM32G071x8 STM32G071xB mainstream microcontrollers are based on high-performance Arm® Cortex®-M0+ 32-bit operating at up to 64 MHz frequency. Offering a high level of integration, they are suitable for a wide range of applications i industrial and appliance domains and ready for the Internet of Things (IoT) solutions.
The devices incorporate a memory protection unit (MPU), high-speed embedded memories (up to 128 Kbytes of Flash prog and 36 Kbytes of SRAM), DMA and an extensive range of system functions, enhanced I/Os and peripherals. The devices of communication interfaces (two I2Cs, two SPIs / one I2S, one HDMI CEC and four USARTs), one 12-bit ADC (2.5 MSps) with channels, one 12-bit DAC with two channels, two fast comparators, an internal voltage reference buffer, a low-power RTC, a control PWM timer running at up to double the CPU frequency, five general-purpose 16-bit timers with one running at up to CPU frequency, a 32-bit general-purpose timer, two basic and two low-power 16-bit timers, two watchdog timers, and a SysT STM32G071x8 STM32G071xB devices provide a fully integrated USB Type-C Power Delivery controller.
The devices operate within ambient temperatures from -40 to 125°C. They can operate with supply voltages from 1.7 V to 3

MCUs/MPUs List: 2 items

*	Part No	Reference	Marketing Status	Unit Price for 10kU (US$)	Board	Package	Flash	RAM	IO
☆	STM32G071RB	STM32G071RBIx		0.0		UFBGA64	128 kBytes	36 kBytes	60
☆		STM32G071RBTx	Active	1.49	N... S...	LQFP64	128 kBytes	36 kBytes	60

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)



STM32 Project

Project Setup

Setup STM32 project

Project Name: G0_EXTI

☑ Use default location

Location: C:/_Work/CubeIDE

Options

Targeted Language

● C ○ C++

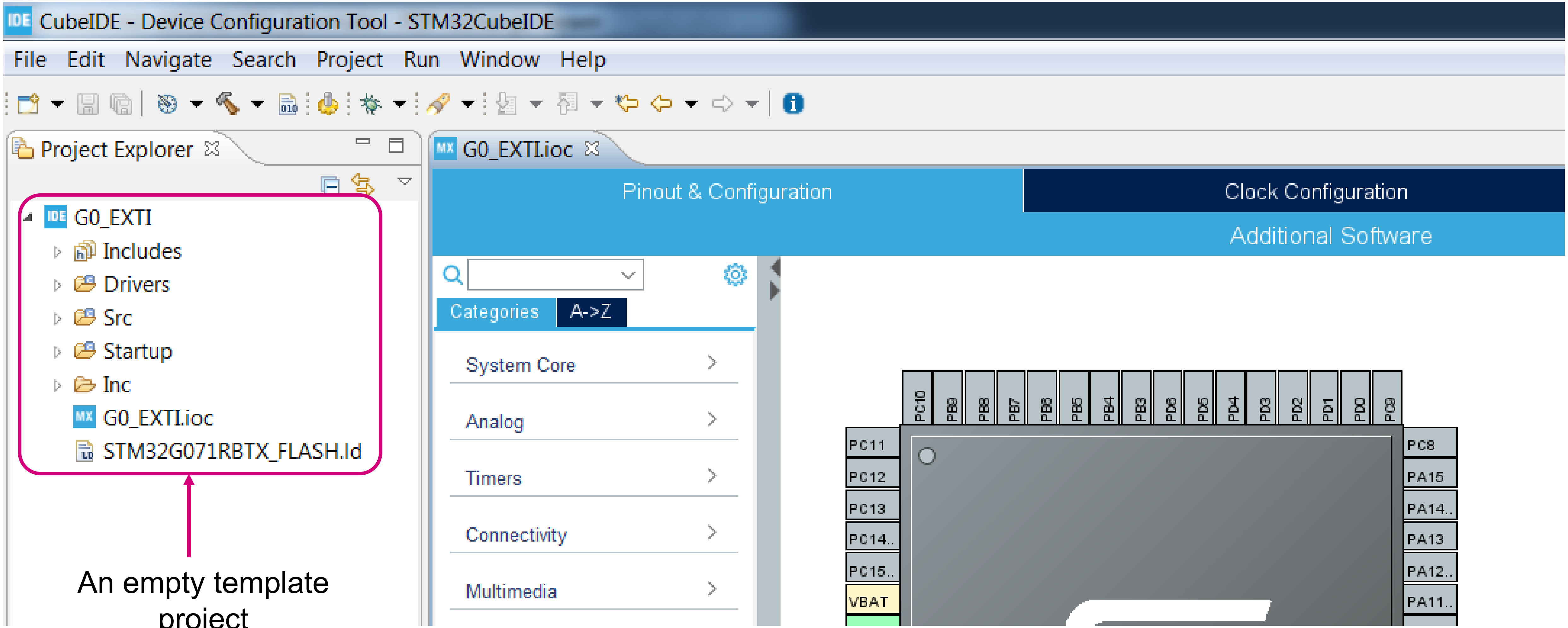
Targeted Binary Type

● Executable ○ Static Library

Targeted Project Type

● STM32Cube ○ Empty

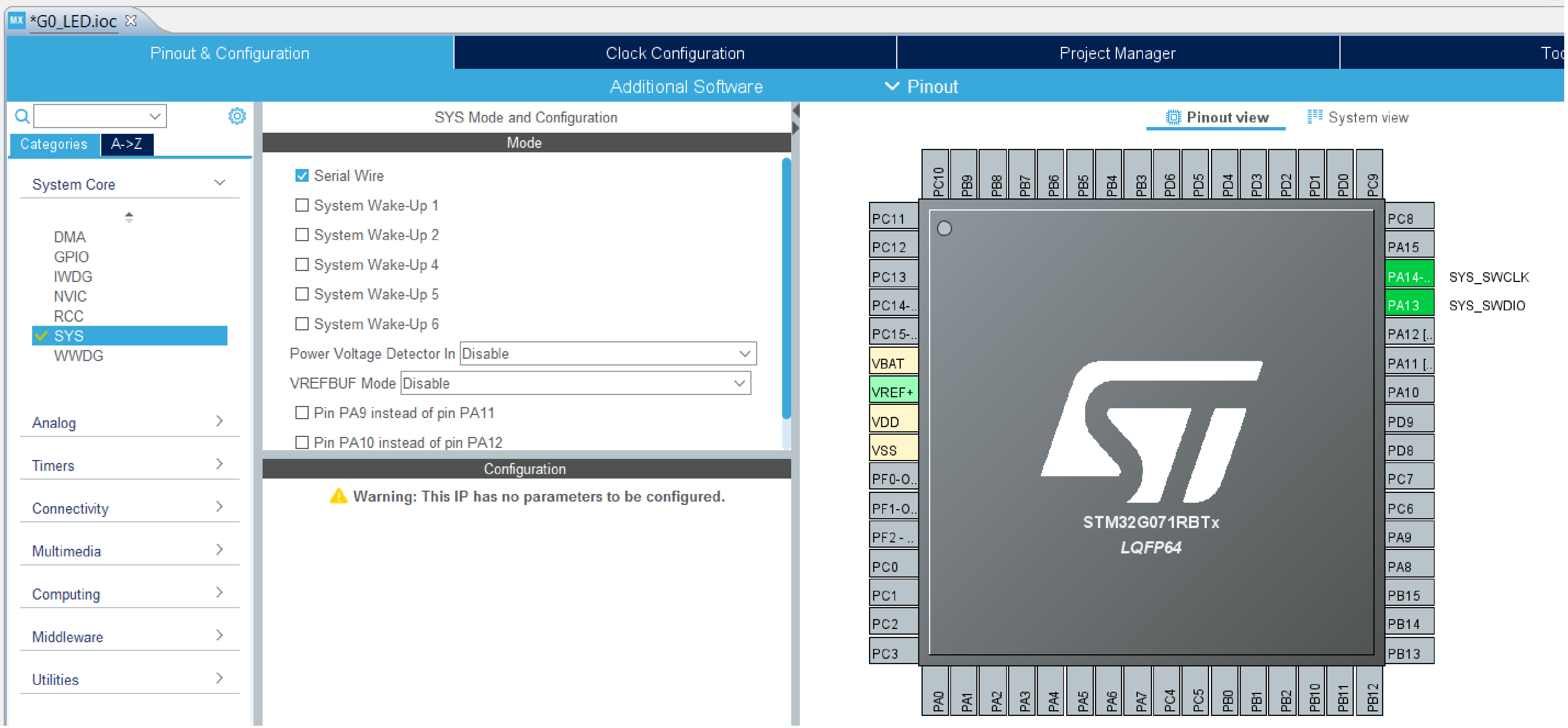
New project in STM32CubeMX perspective



An empty template project

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



*G0_LED.ioc

Pinout & Configuration | Clock Configuration | Project Manager | Too

Additional Software | Pinout

Categories | A->Z

System Core: DMA, GPIO, IWDG, NVIC, RCC, ✓ SYS, WWDG

Analog, Timers, Connectivity, Multimedia, Computing, Middleware, Utilities

SYS Mode and Configuration

Mode

☑ Serial Wire

□ System Wake-Up 1

□ System Wake-Up 2

□ System Wake-Up 4

□ System Wake-Up 5

□ System Wake-Up 6

Power Voltage Detector In: Disable

VREFBUF Mode: Disable

□ Pin PA9 instead of pin PA11

□ Pin PA10 instead of pin PA12

Configuration

⚠ Warning: This IP has no parameters to be configured.

Pinout view | System view

STM32G071RBTx LQFP64

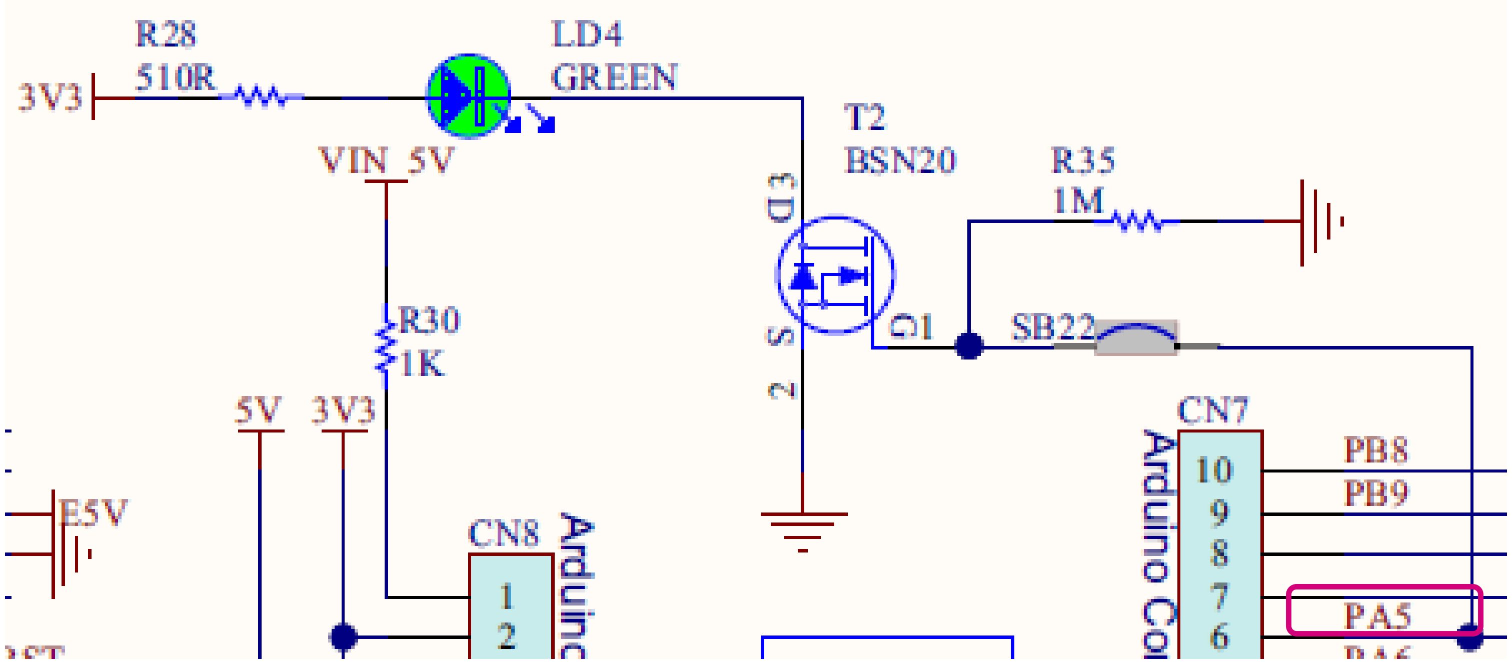
PA14-.. SYS_SWCLK

PA13 SYS_SWDIO

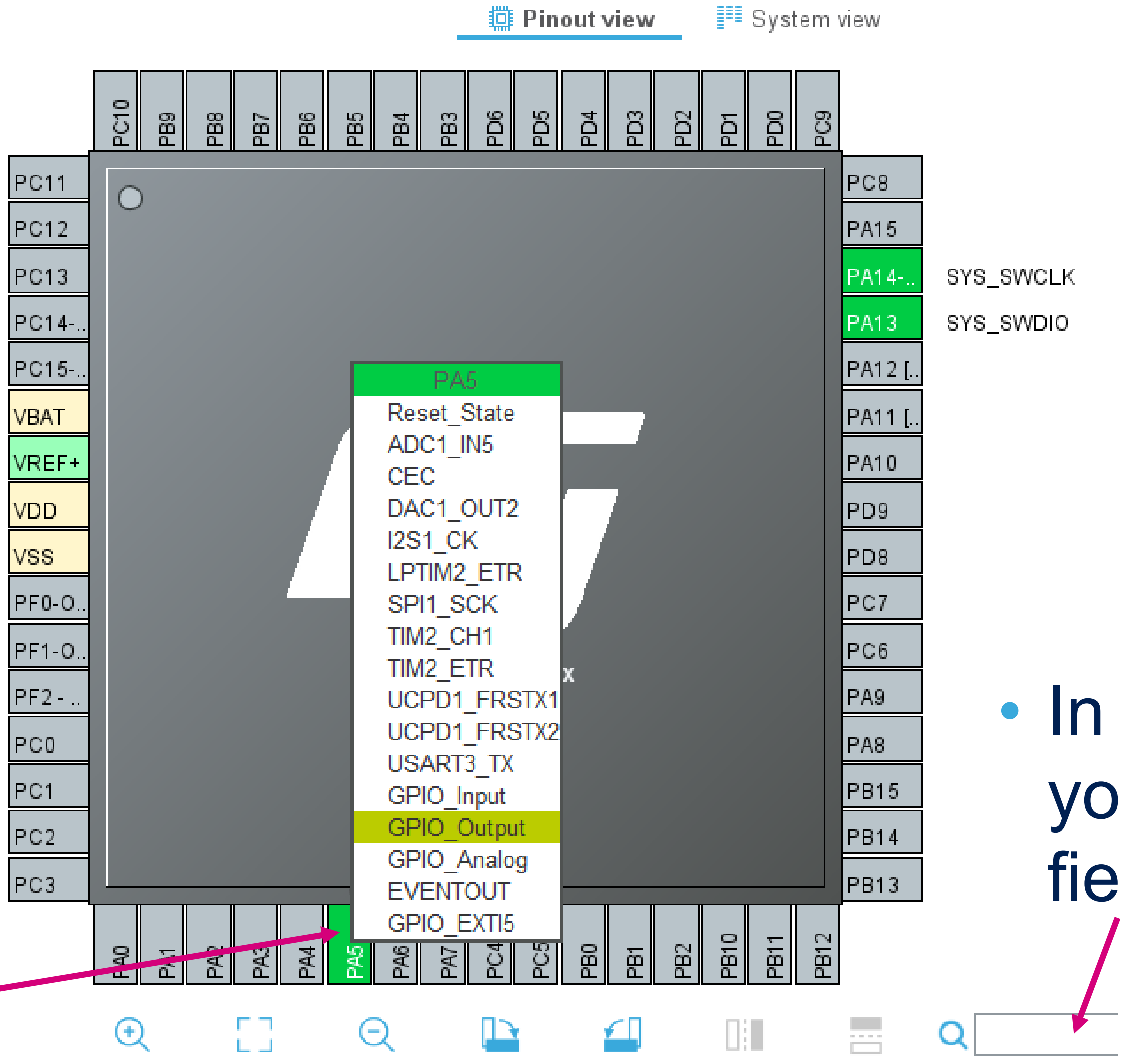
Pin Configuration

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)



Configuring PA5 as Output

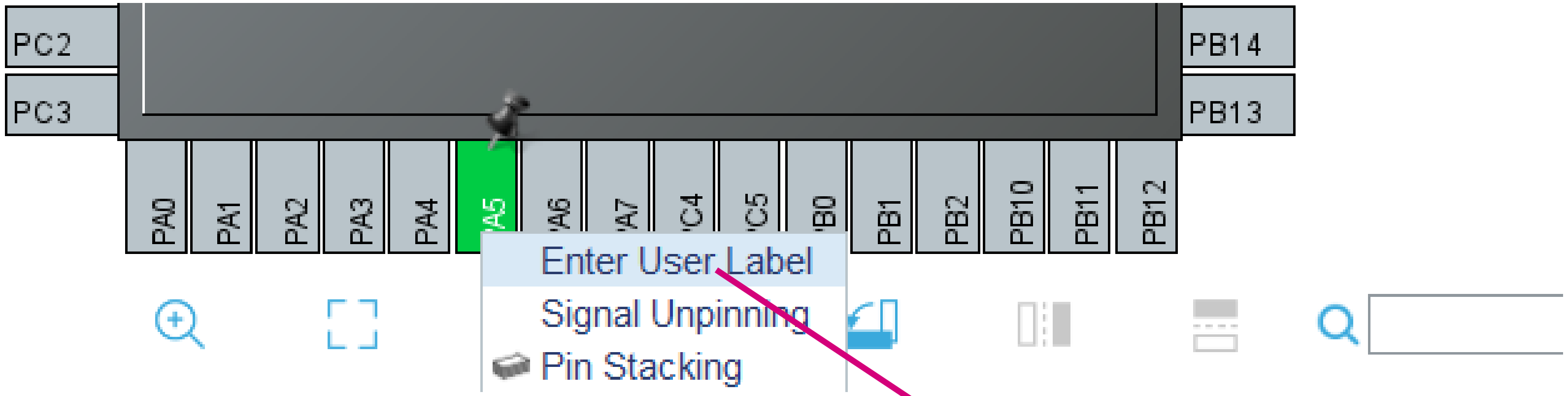


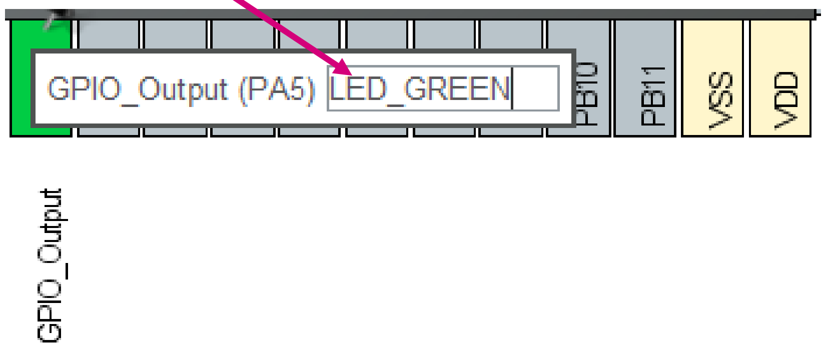
- In bigger packages you can use search field to locate the pin

- Using configure PA5 as GPIO_Output

Assign label to PA5

- Using 🖱 select Enter User Label and insert LED_GREEN label





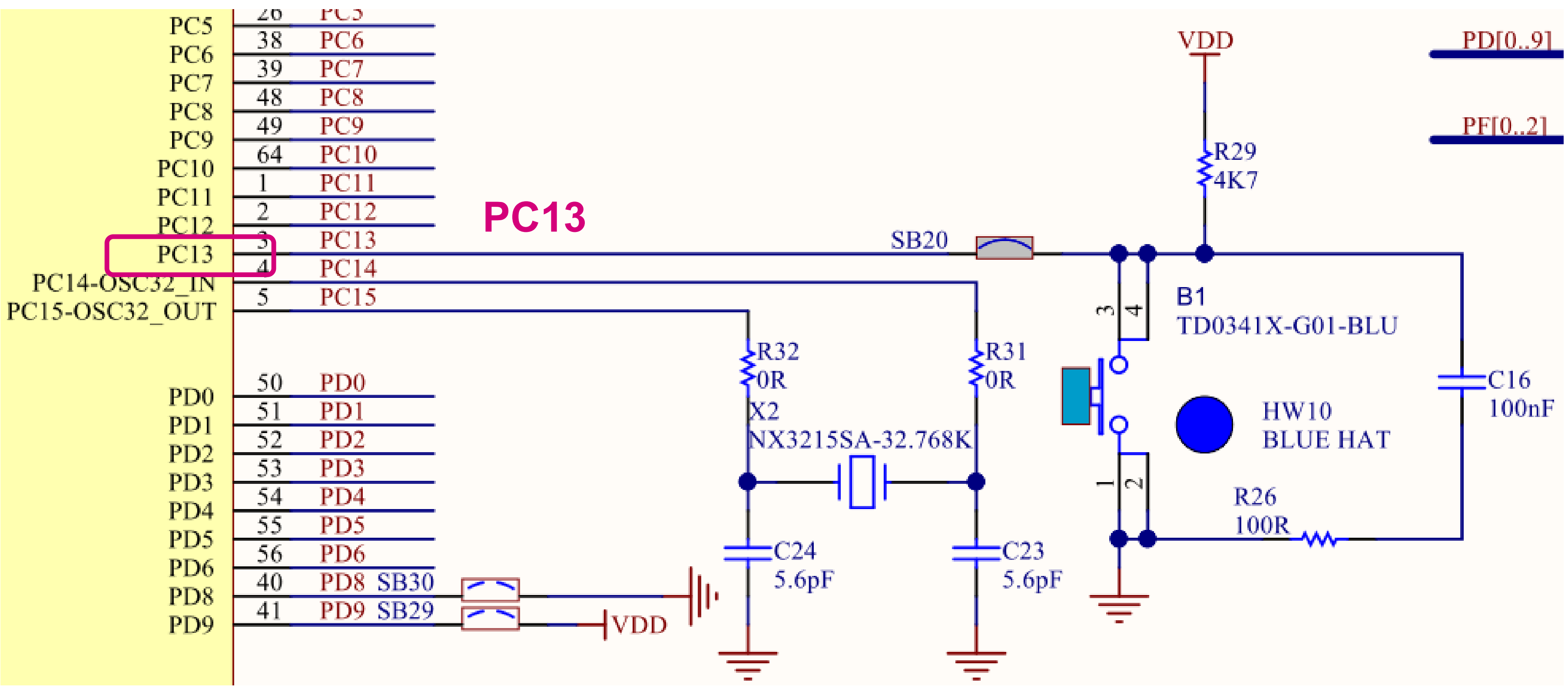
GPIO_Output (PA5) LED_GREEN

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

Pins Configuration

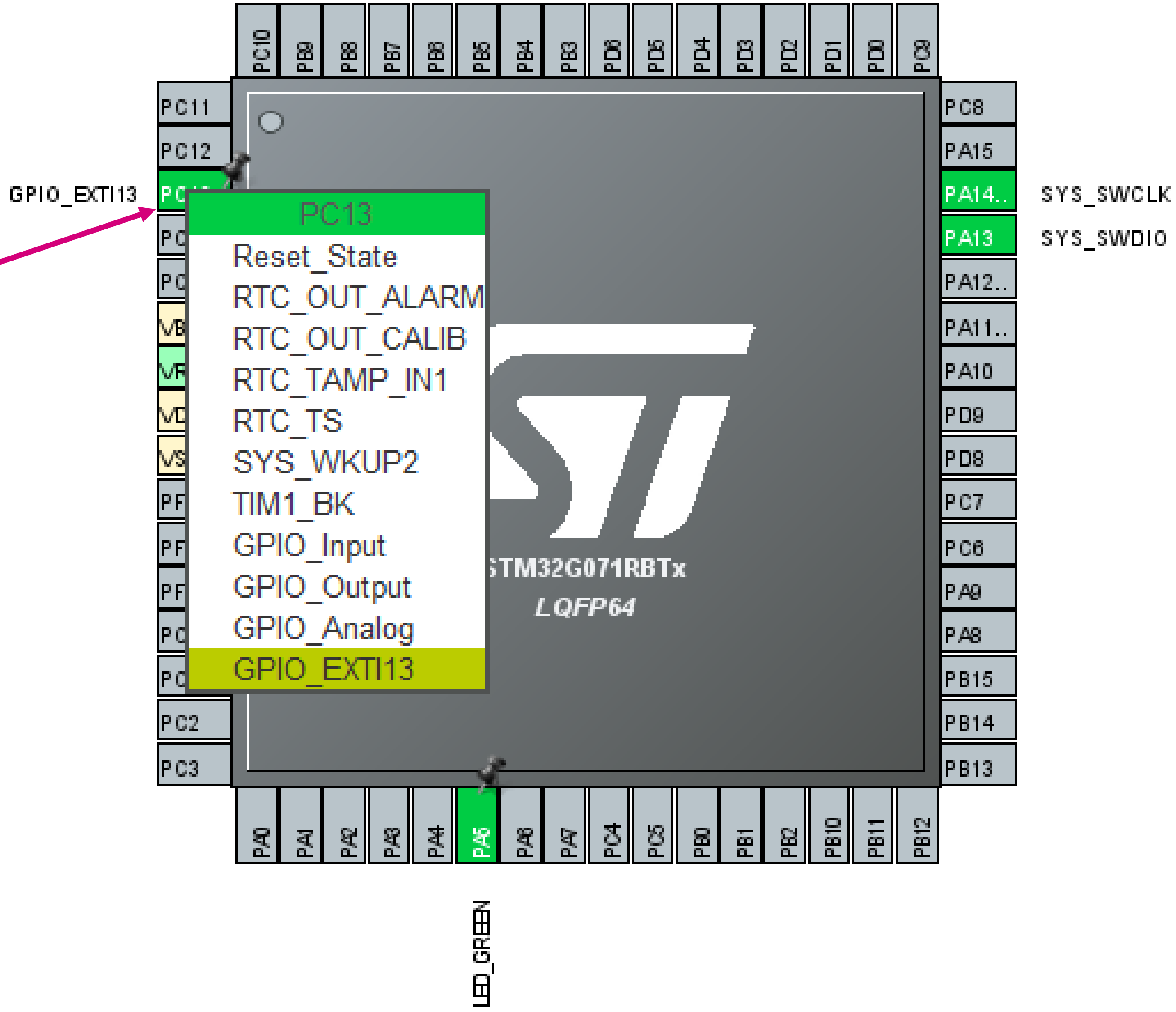
external interrupt button

- In this example we are going to use one button - blue one (connected to PA5 as seen in the schematic below)



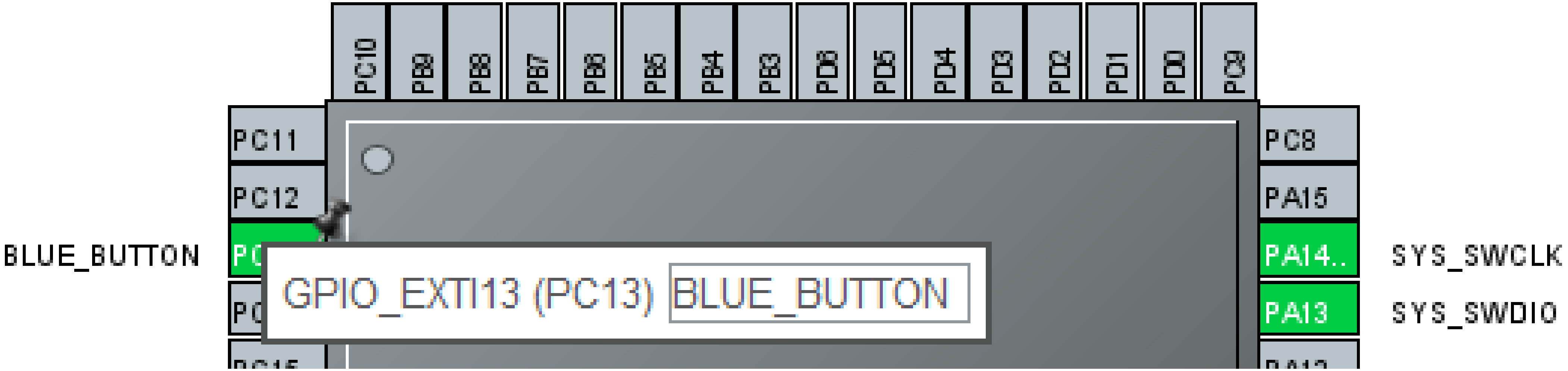
Configuring PC13 as GPIO_EXTI13

- Using configure PC13 as GPIO_EXTI13



Assign label to PC13

- Using 🖱 select Enter User Label and insert BLUE_BUTTON label



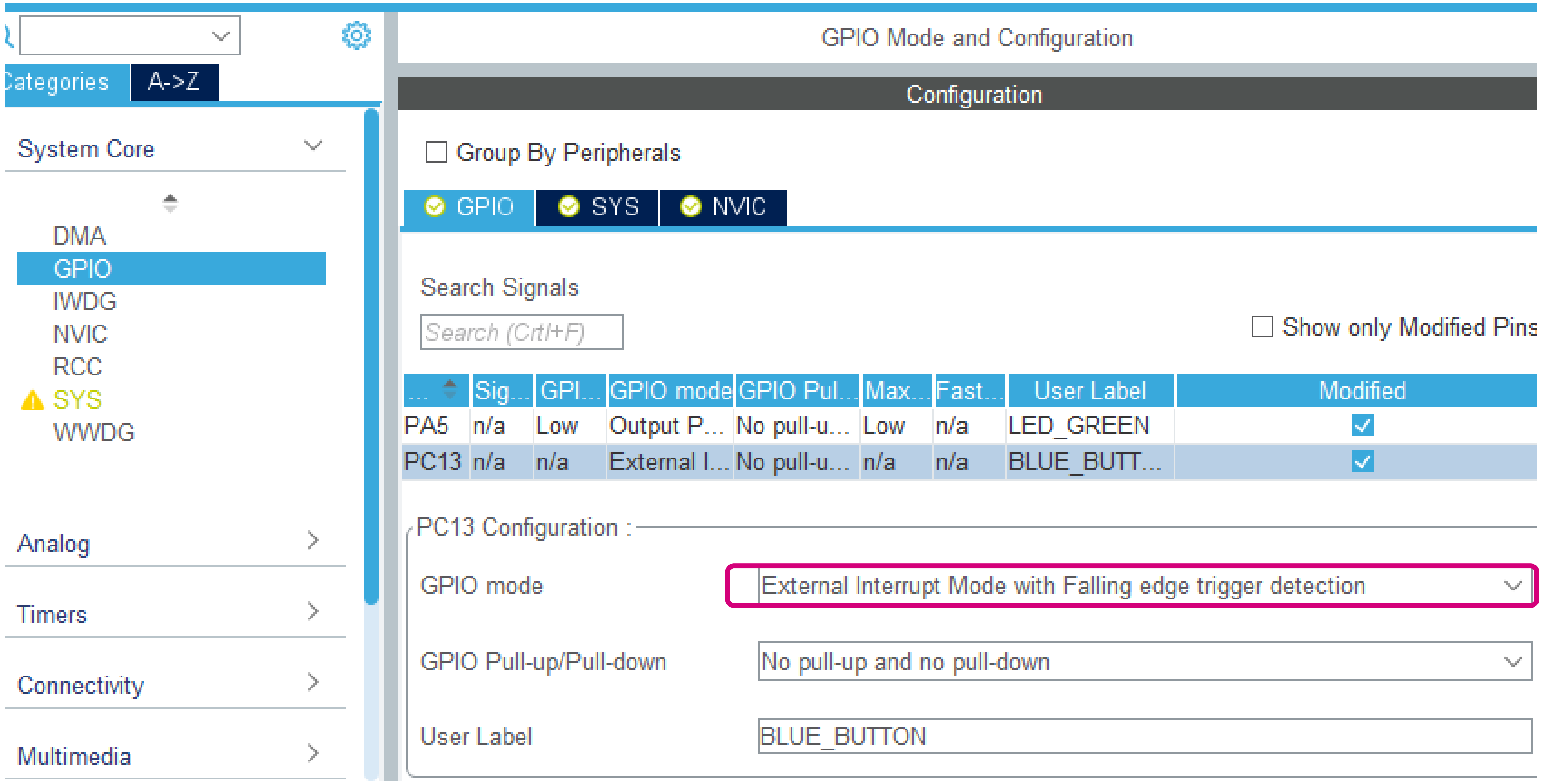
GPIO_EXTI13 (PC13) BLUE_BUTTON

Hint:

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

Configure PC13

- Go to System Core -> GPIO, GPIO tab
- Select PC13 from the list
- Change GPIO mode to External Interrupt Mode with Falling edge trigger detection



☐ Group By Peripherals

☐ Show only Modified Pins

...	Sig...	GPI...	GPIO mode	GPIO Pul...	Max...	Fast...	User Label	Modified
PA5	n/a	Low	Output P...	No pull-u...	Low	n/a	LED_GREEN	☑
PC13	n/a	n/a	External I...	No pull-u...	n/a	n/a	BLUE_BUTT...	☑

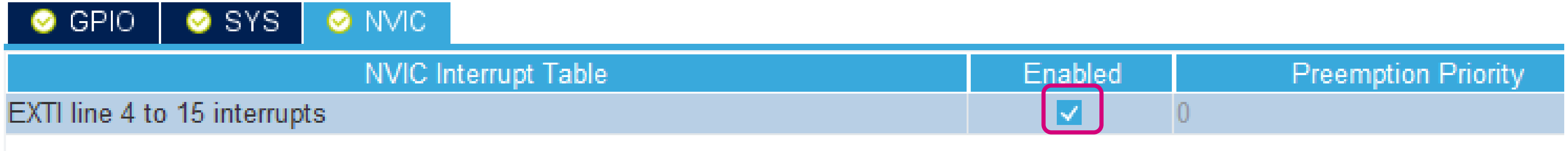
PC13 Configuration :

GPIO mode: External Interrupt Mode with Falling edge trigger detection

GPIO Pull-up/Pull-down: No pull-up and no pull-down

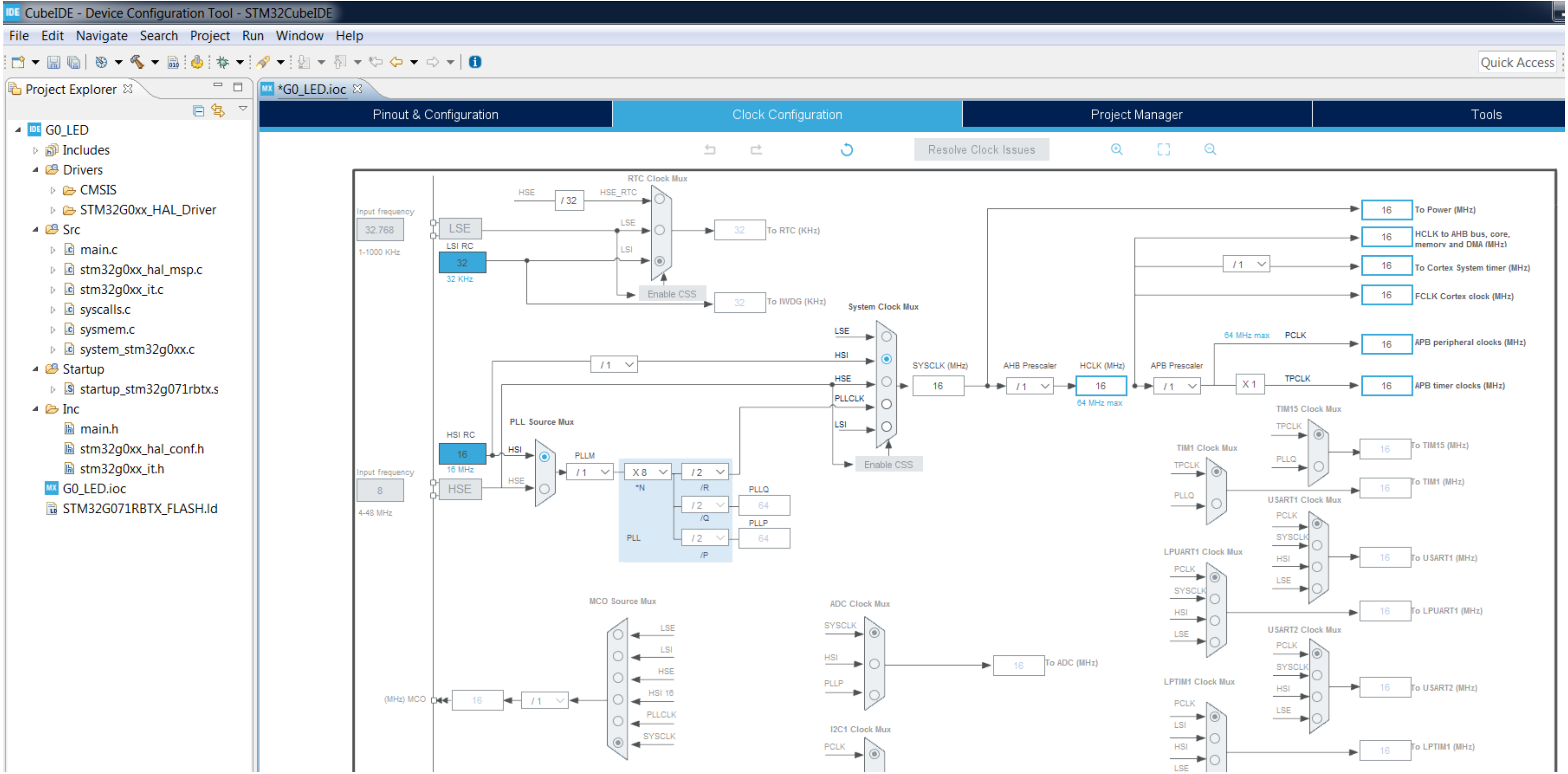
User Label: BLUE_BUTTON

- Switch to NVIC tab within GPIO configuration and enable EXTI Line 4 to 15 interrupt



NVIC Interrupt Table	Enabled	Preemption Priority
EXTI line 4 to 15 interrupts	☑	0

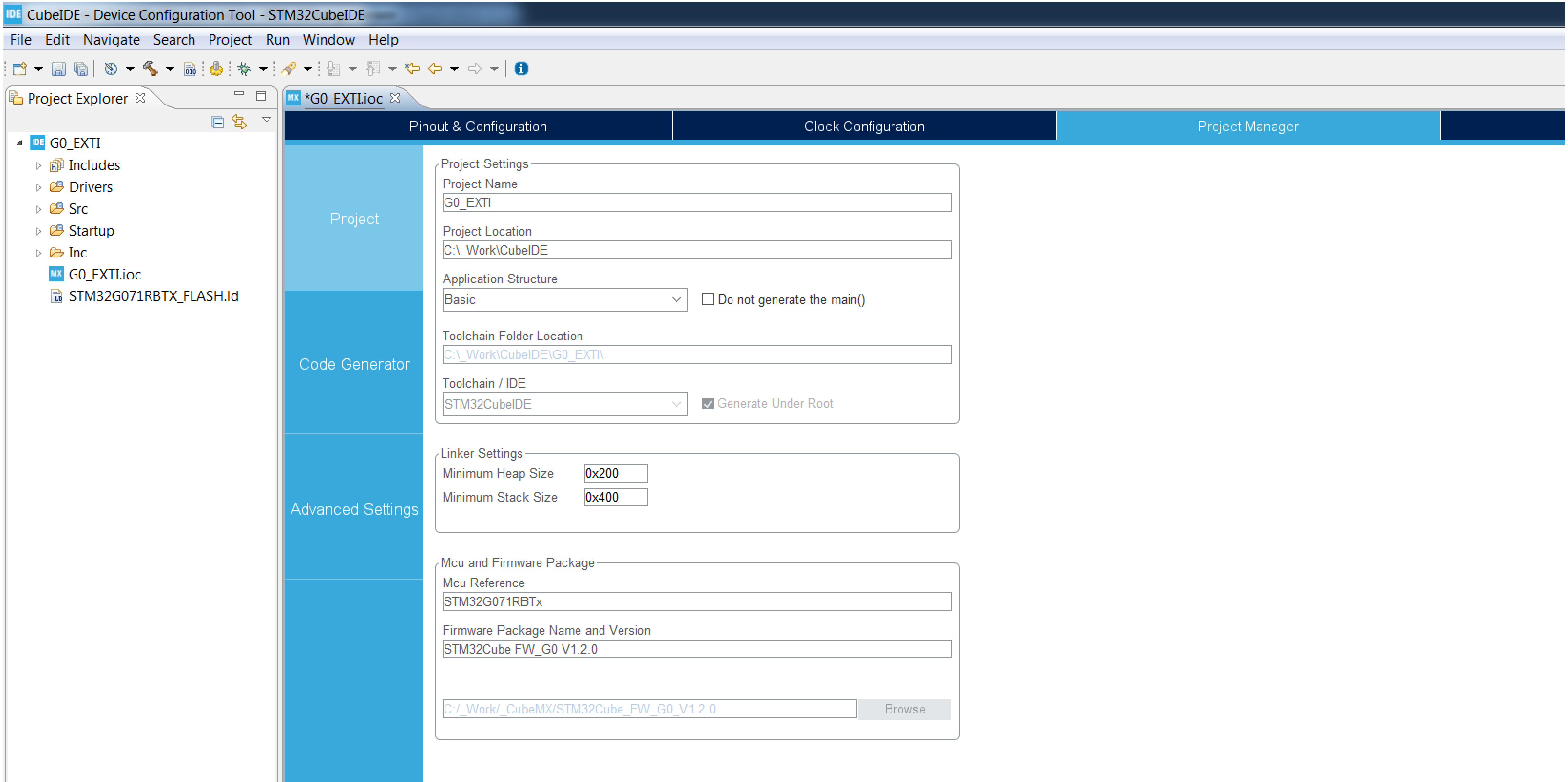
Default clock configuration – no change



- 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

LSE Input frequency: 32.768
LSI RC: 32
HSI RC: 16
HSE Input frequency: 8
To RTC (KHz): 32
To IWDG (KHz): 32
HSI divider: / 1
PLLM: / 1
*N: X 8
/R: / 2
/Q: / 2
/P: / 2
PLLQ: 64
PLLP: 64
System Clock Mux: HSI
SYSCLK (MHz): 16
AHB Prescaler: / 1
HCLK (MHz): 16
Cortex System timer divider: / 1
APB Prescaler: / 1
To Power (MHz): 16
HCLK to AHB bus, core, memory and DMA (MHz): 16
To Cortex System timer (MHz): 16
FCLK Cortex clock (MHz): 16
APB peripheral clocks (MHz): 16
APB timer clocks (MHz): 16
To TIM15 (MHz): 16
To TIM1 (MHz): 16
To USART1 (MHz): 16
To LPUART1 (MHz): 16
To USART2 (MHz): 16
To LPTIM1 (MHz): 16
To ADC (MHz): 16
MCO divider: / 1
(MHz) MCO: 16

Basic project settings – no change



Project Settings

Project Name: G0_EXTI

Project Location: C:_Work\CubeIDE

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

Toolchain Folder Location: C:_Work\CubeIDE\G0_EXTI\

Toolchain / IDE: STM32CubeIDE ☑ Generate Under Root

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Mcu and Firmware Package

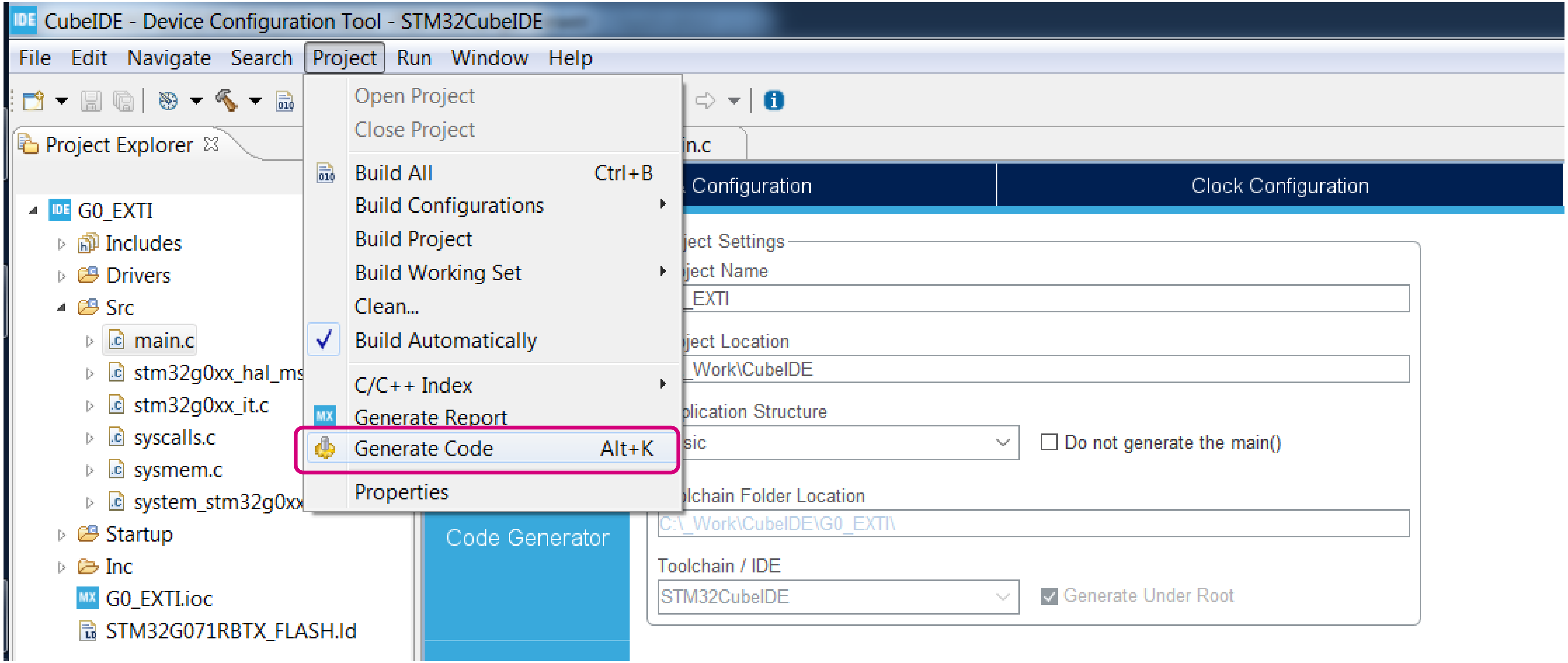
Mcu Reference: STM32G071RBTx

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 our project we have just prepared



Adding code

main.c file

```
/* USER CODE BEGIN PV */
uint8_t flag=0;
```

Variable declaration

…

```
if(1==flag)
{
HAL_GPIO_TogglePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin);
flag=0;
}
/* USER CODE END WHILE */
```

Green LED pin toggling in case flag=1 and clear flag variable afterwards

…

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Falling_Callback(uint16_t GPIO_Pin)
{
flag=1;
}
```

Set flag to 1 in case of blue button press

… 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 on each blue button press

Thank you

/STM32 @ST_World community.st.com

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