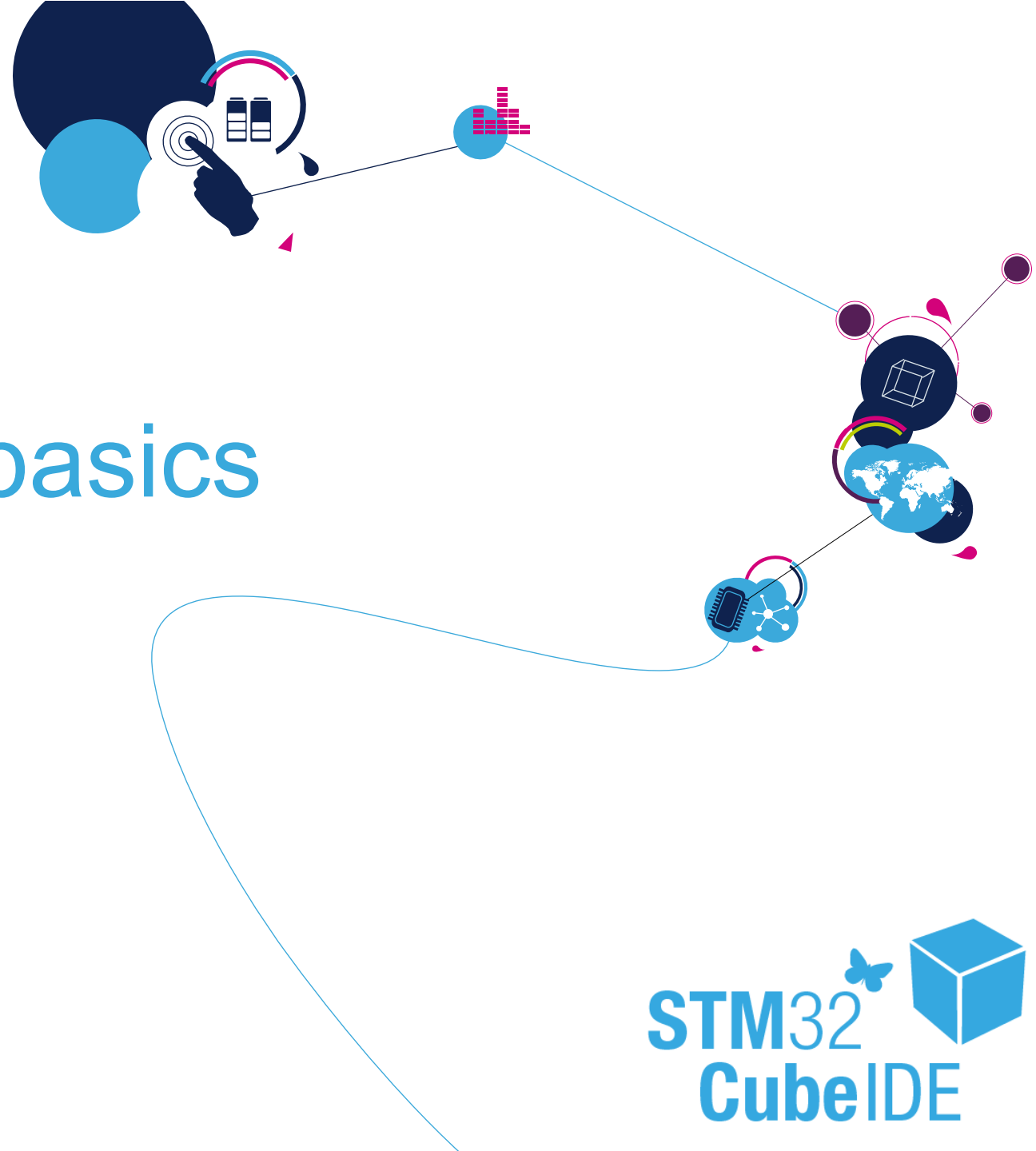
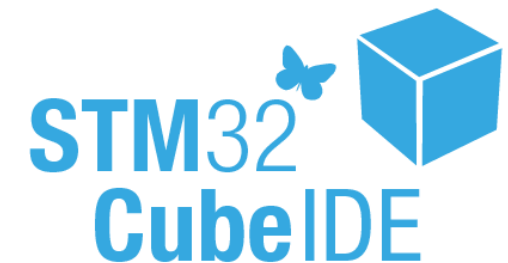


# STM32CubeIDE basics

Timer lab: PWM generation using HAL library



## Objective:

- Now let's use a more advanced peripheral like the Timer.
- In this lab we are going to configure a Timer in a PWM mode to blink the LED that we previously controlled with a GPIO.
- PA5 has an alternate Timer channel alternate function which is Timer 2 Channel 1: TIM2\_CH1 that we will be using.

# Timer Parameters Calculation

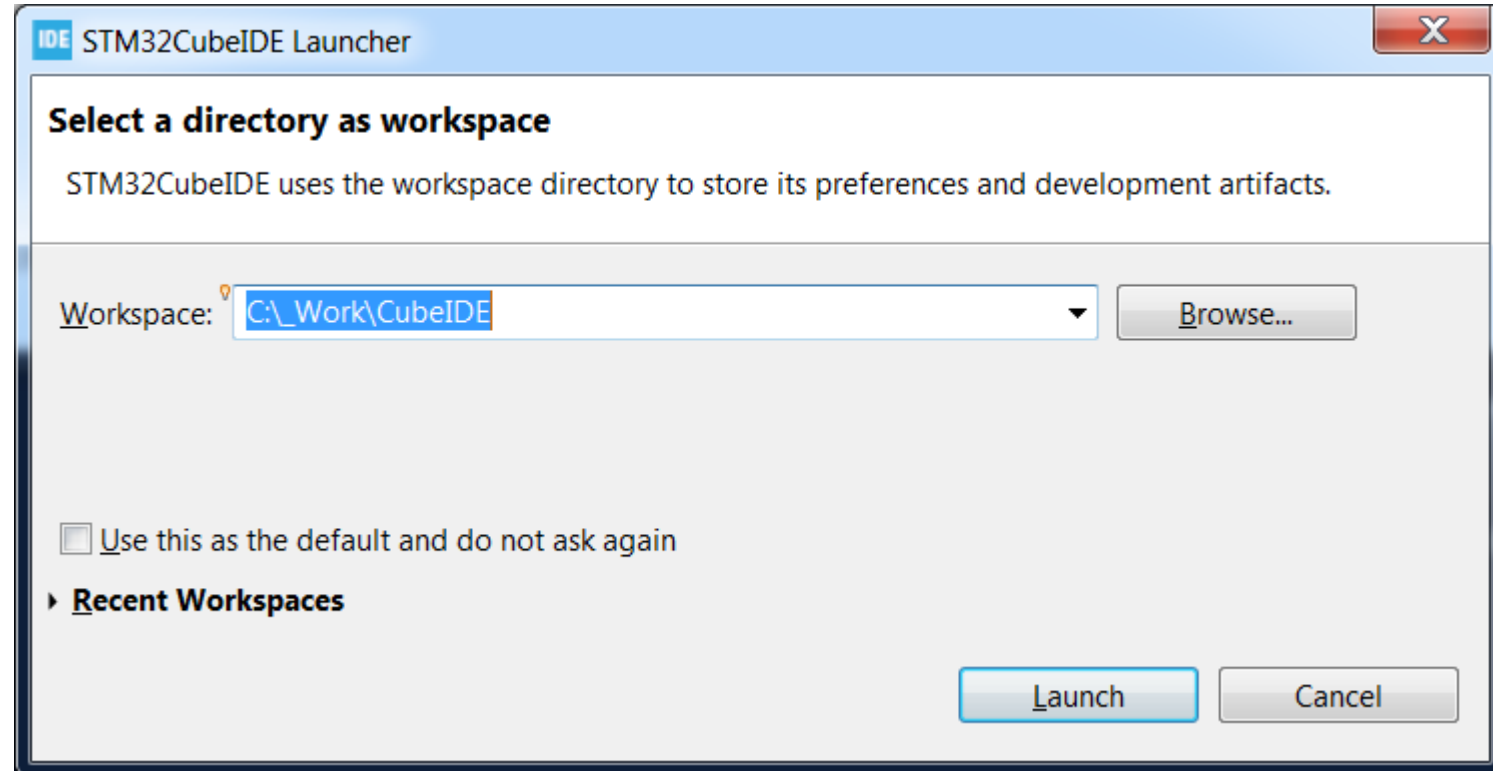
- We can calculate Timer parameters for 1 Hz period @ 50% duty cycle
- If we choose Timer input clock to be 8 MHz
- Let us say the prescaler is div-by-128;  $8\text{MHz}/128 = 62500 \text{ Hz}$ 
  - So, prescaler register should be  $128-1 = 127$  (actual divide is  $\text{PSC}+1$ )
  - If we set Period = 62499 (as we calculate from 0); Pulse = 31250, we get 1 Hz Period @ 50% duty cycle

# ... time for hands-on

- In case you would like to create a new workspace and project
  - [go here](#)
- In case you would like to create a new project within current workspace
  - [go here](#)

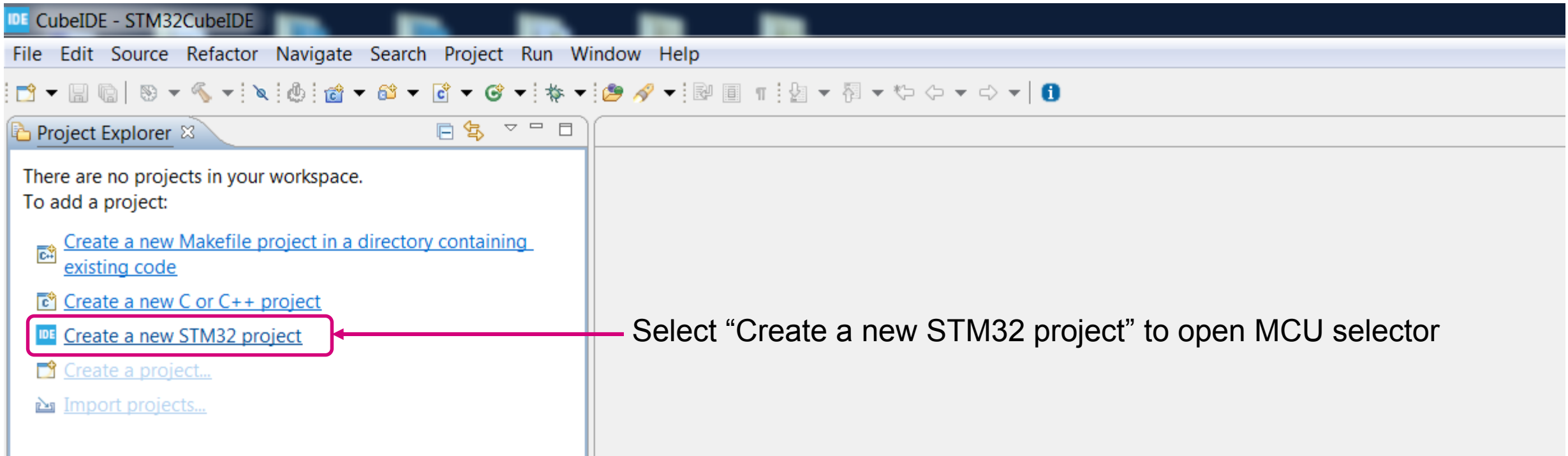
# Start a new workspace

- Run STM32CubeIDE
- Select a folder to store a workspace



# Create a new project

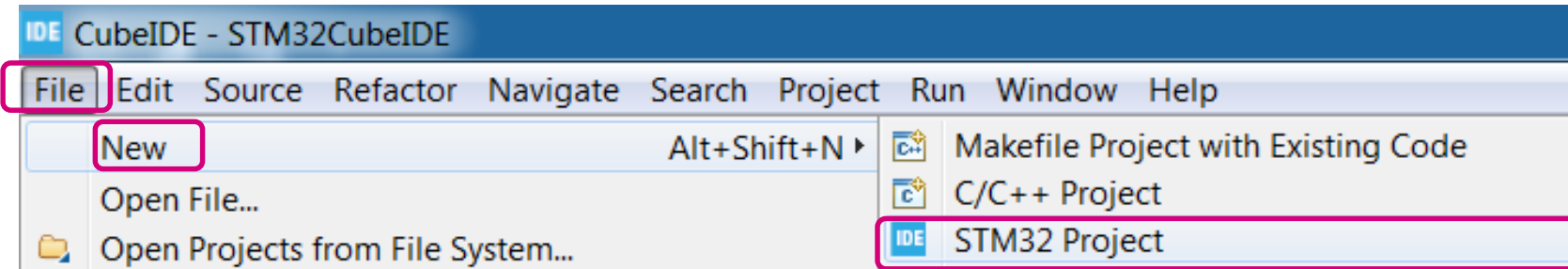
- Click on “Create a new STM32 project”



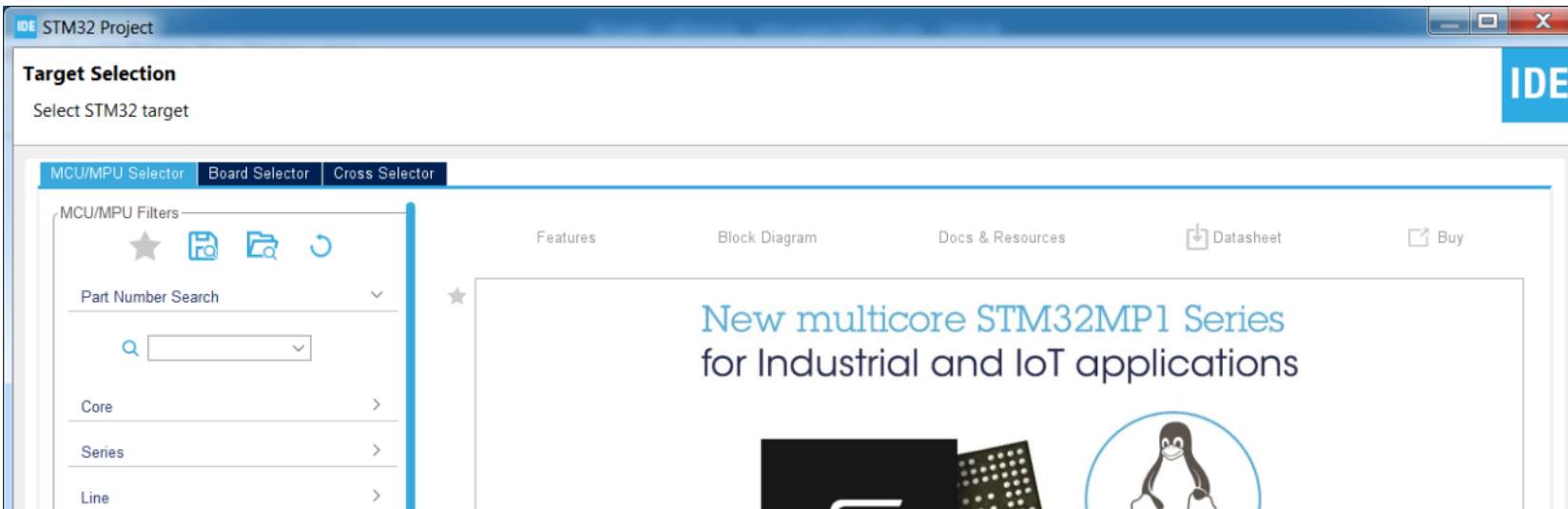
Select “Create a new STM32 project” to open MCU selector

# Start a new project within workspace

- Go to File -> New -> STM32 Project

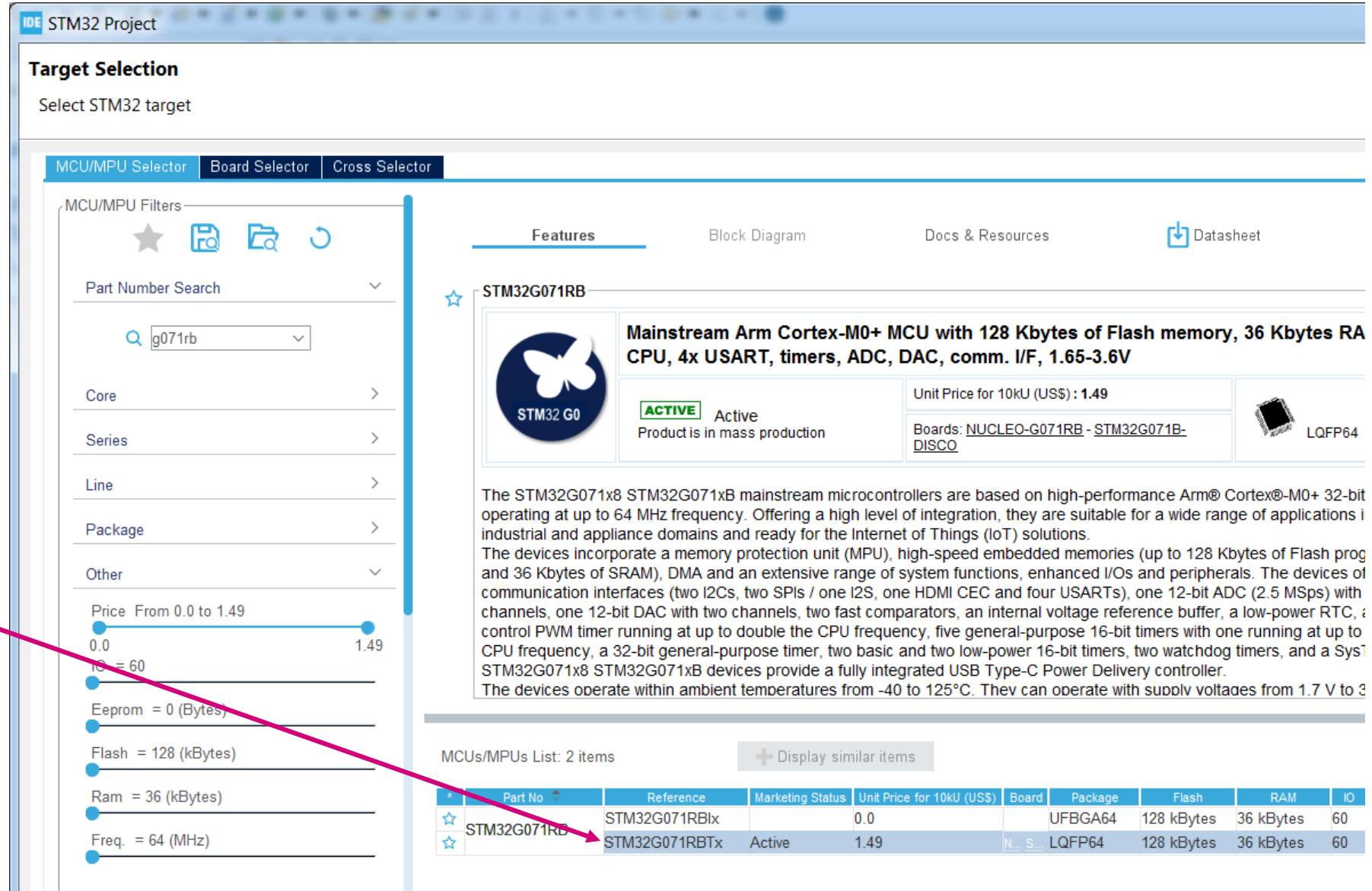


- As a result “target selection” window will be displayed



# 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



The screenshot shows the STM32 Project Target Selection window. The 'MCU/MPU Selector' tab is active, displaying a search bar with 'g071rb' and various filters. The 'Features' section for the selected MCU, STM32G071RB, is shown, including its main features, unit price, and a list of boards. A pink arrow points from the text 'double click on the part number' to the 'Part No.' column in the 'MCUs/MPUs List' table, specifically highlighting the 'STM32G071RBTx' entry.

**Target Selection**  
Select STM32 target

MCU/MPU Selector | Board Selector | Cross Selector

MCU/MPU Filters

Part Number Search

g071rb

Core

Series

Line

Package

Other

Price From 0.0 to 1.49

0.0 1.49

IC = 60

Eeprom = 0 (Bytes)

Flash = 128 (kBytes)

Ram = 36 (kBytes)

Freq. = 64 (MHz)

**Features** | Block Diagram | Docs & Resources | Datasheet

★ 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 in 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

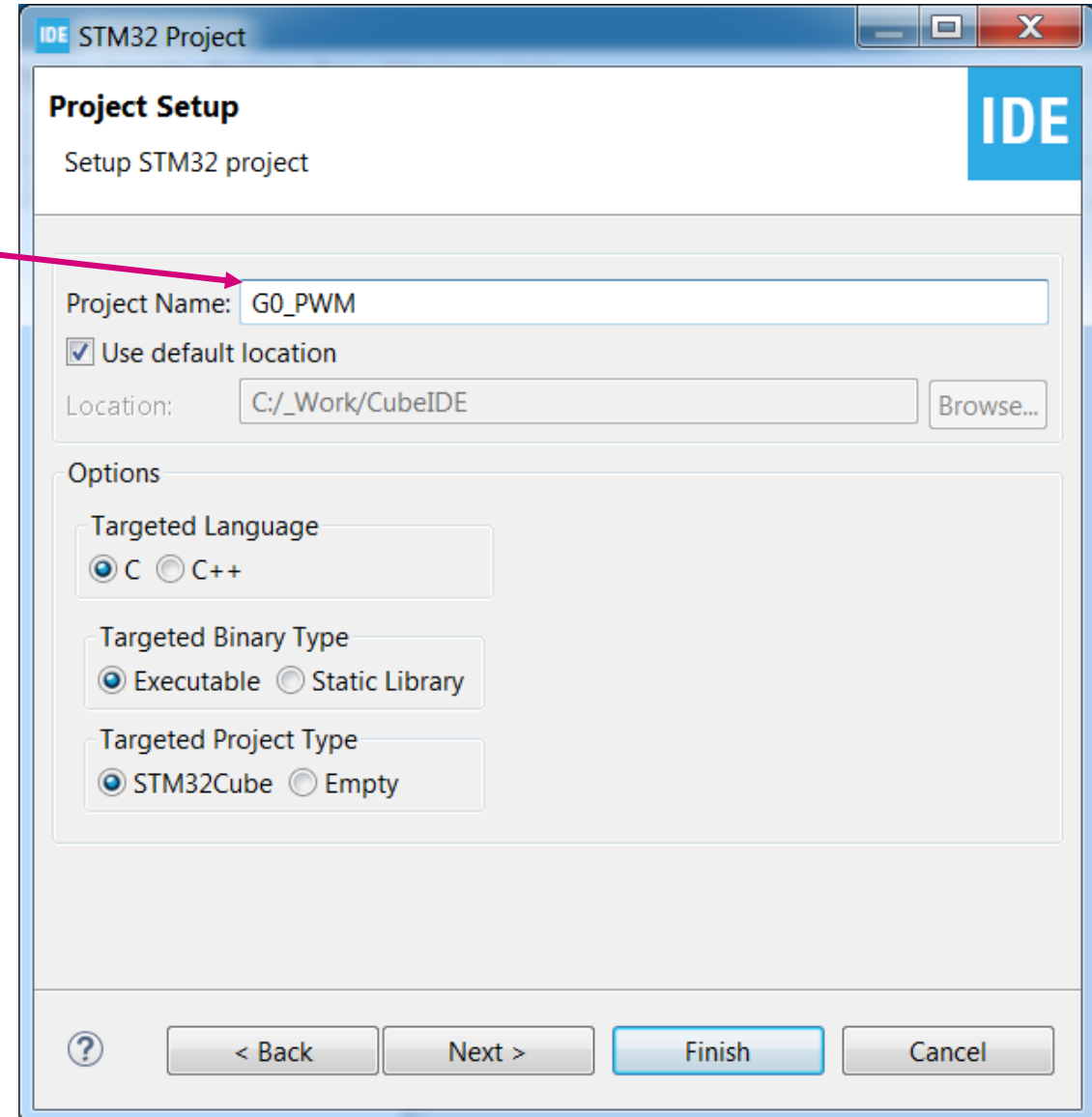
Display similar items

★	Part No	Reference	Marketing Status	Unit Price for 10kU (US\$)	Board	Package	Flash	RAM	IO
★	STM32G071RB	STM32G071RB1x	0.0			UFPGA64	128 kBytes	36 kBytes	60
★	STM32G071RBTx	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)



**Project Setup**  
Setup STM32 project

Project Name: G0\_PWM

☒ Use default location

Location: C:/\_Work/CubeIDE Browse...

**Options**

Targeted Language  
☒ C ☐ C++

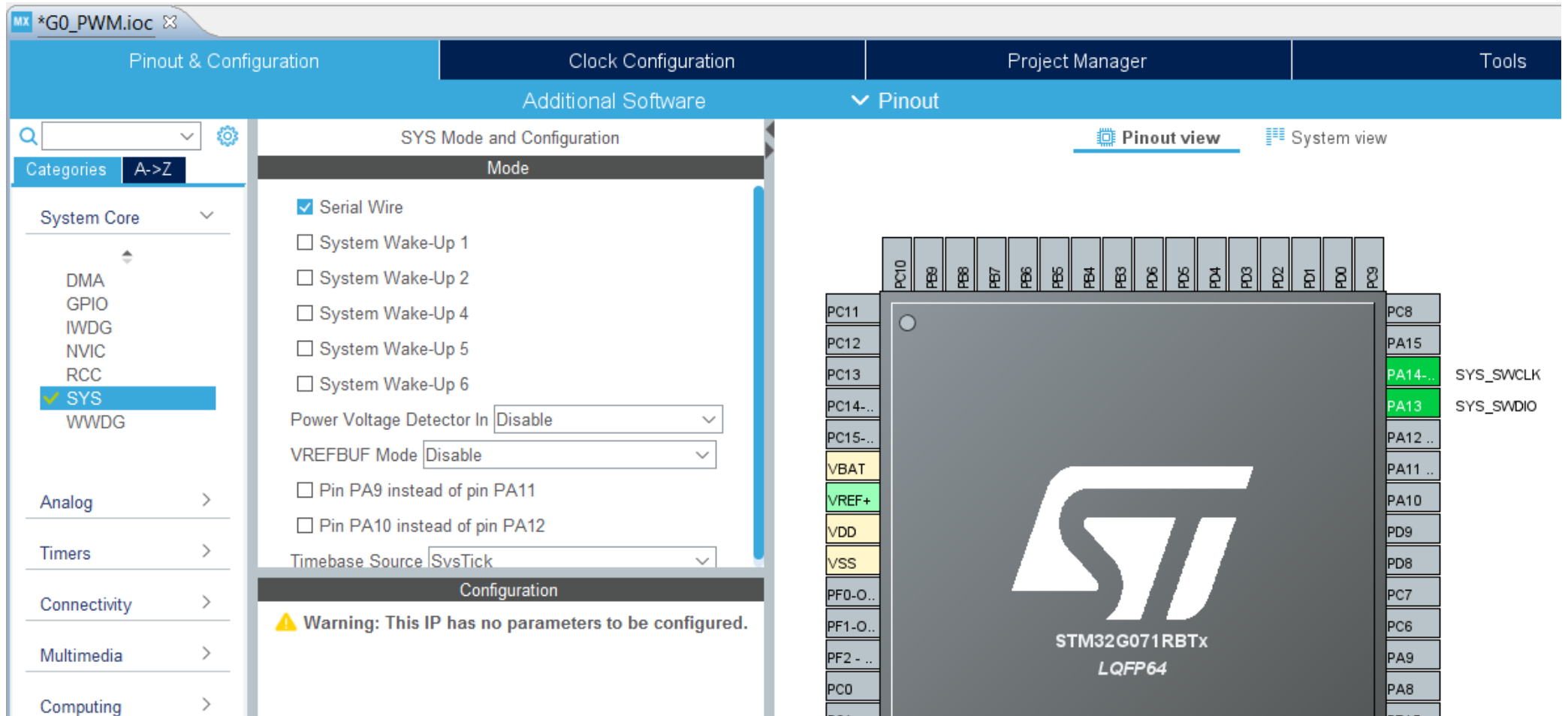
Targeted Binary Type  
☒ Executable ☐ Static Library

Targeted Project Type  
☒ STM32Cube ☐ Empty

? < Back Next > Finish Cancel

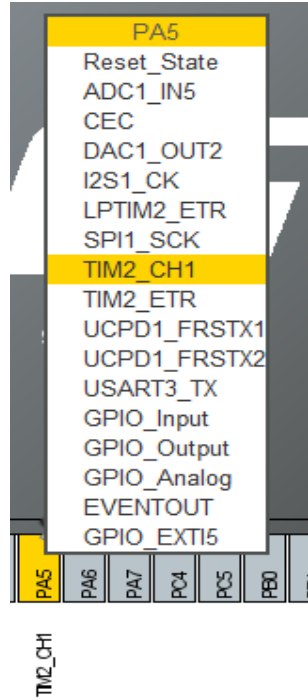
# Enabling Serial Wire debug interface


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

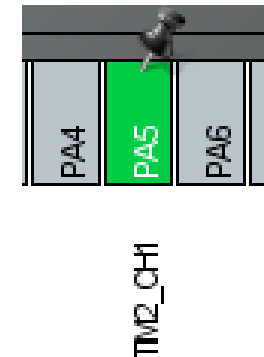
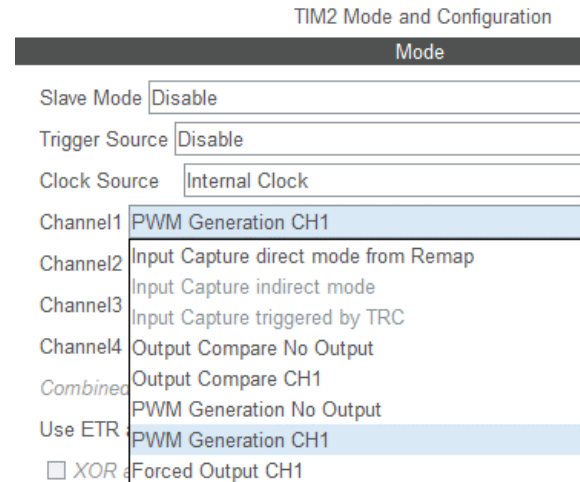


# Configuring PA5 as TIM2\_CH1 PWM

method 1



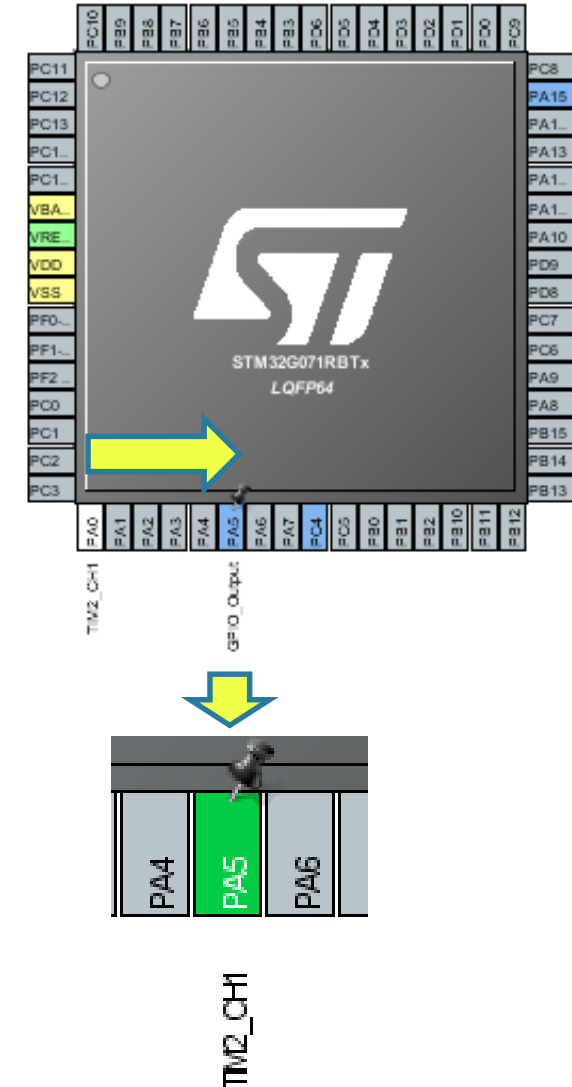
- Click  over PA5 pin and select TIM2\_CH1 function
  - PA5 pin become yellow with pin (booked but not configured pin) and TIM2\_CH1 label
- In Timers->TIM2->Mode window select:
  - Clock Source: Internal Clock
  - Channel1: PWM Generation CH1
- After this PA5 become green with TIM2\_CH1 label



# Configuring PA5 as TIM2\_CH1 PWM

method 2

- By default the tool will configure Timer 2 CH1 to PA0
- We want to remap it to PA5
  - NOTE: PA5 is connected to LD4
- Hold “Ctrl” button and left mouse click on PA0
- Then drag the mouse pointer to PA5 and then release



# Timer Parameters Calculation

- We can calculate Timer parameters for 1 Hz period @ 50% duty cycle
- If we choose Timer input clock to be 8 MHz
- Let us say the prescaler is div-by-128;  $8\text{MHz}/128 = 62500 \text{ Hz}$ 
  - So, prescaler register should be  $128-1 = 127$  (actual divide is  $\text{PSC}+1$ )
  - If we set Period = 62499 (as we calculate from 0); Pulse = 31250, we get 1 Hz Period @ 50% duty cycle

# Clock Tree Configuration

MX STM32CubeMX pwm.ioc: STM32G071RBTx



File

Window

Help



Home

/ STM32G071RBTx

/ pwm.ioc

- Clock Configuration

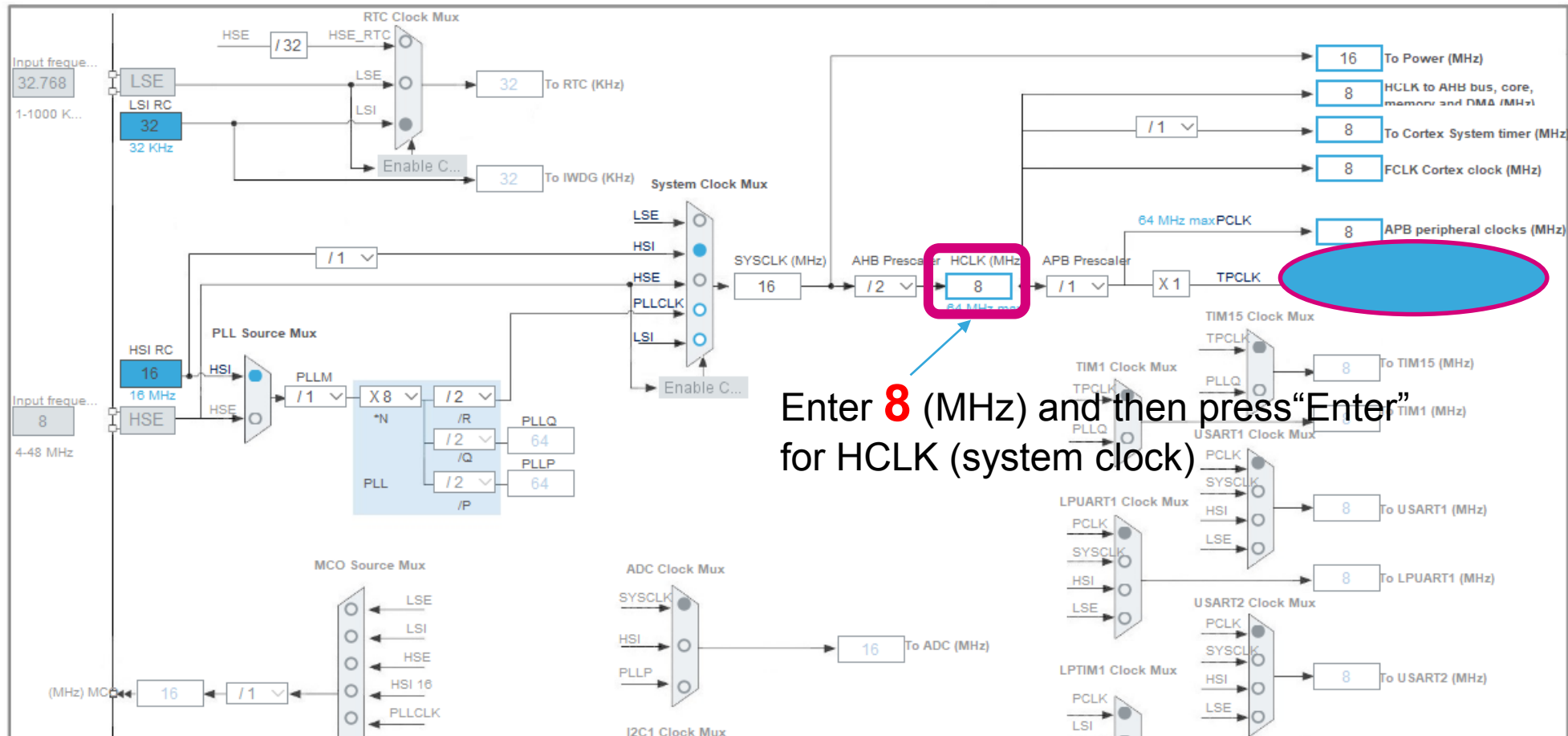
GENERATE CODE

Pinout & Configuration

Clock Configuration

Project Manager

Tools



Enter 8 (MHz) and then press "Enter" for HCLK (system clock)

# Configuring TIM2\_CH1 (1Hz, 50% dc)

- Select the Pinout & Configuration
- In Parameters Settings of the TIM2
- Configure 1 Hz timer
  - Prescaler = 127 **1**
  - Period = 62499 **2**
- Set CH1 PWM
  - Pulse = 31250 **3**

TIM2 Mode and Configuration

Mode

Slave Mode

Trigger Source

Clock Source

Channel1

Configuration

Reset Configuration

☒ NVIC Settings ☒ DMA Settings ☒ Parameter Settings

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC - 16 bits value)  **1**

Counter Mode

Counter Period (AutoReload Regi...  **2**

Internal Clock Division (CKD)

auto-reload preload

> Trigger Output (TRGO) Parameters

> Clear Input

PWM Generation Channel 1

Mode

Pulse (32 bits value)  **3**

Fast Mode

CH Polarity


- Generate project using Project->Generate Code
- Open the main.c, Add the following code before the while(1) loop in order to start the PWM Timer:

```
/* USER CODE BEGIN 2 */
if (HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1) != HAL_OK)
    Error_Handler();
```



# ... 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 (this time controlled by Timer2)

