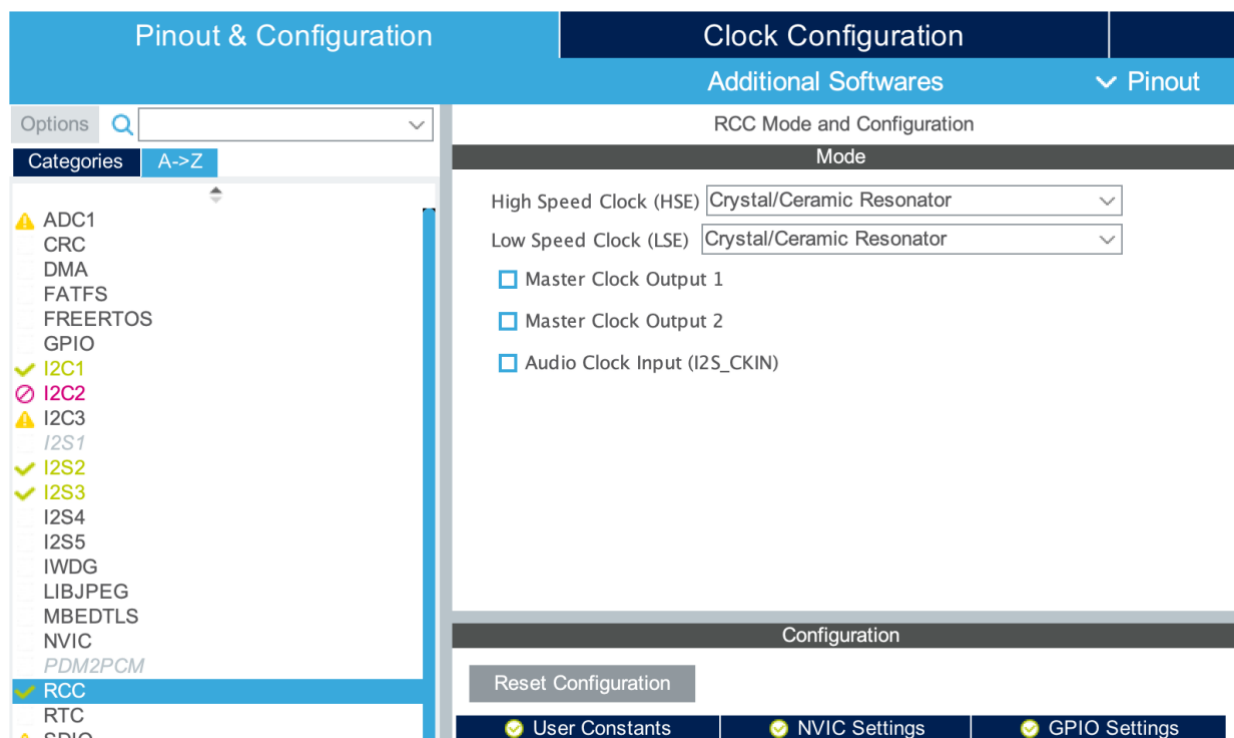
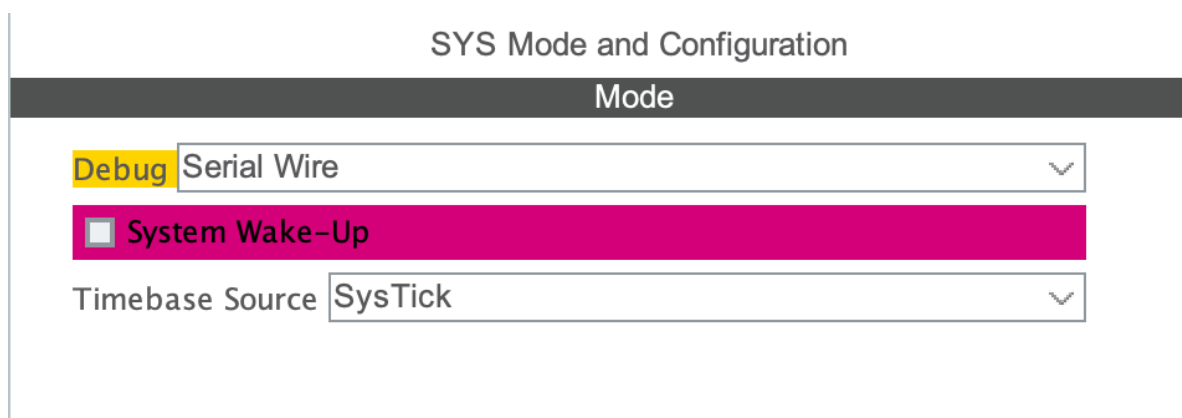


Створення Проекту за допомогою CUBE MX

- 1) Запустити CUBE MX та обрати потрібний мікроконтролер.
- 2) У лівому меню обрати розділ RCC та встановити для High Speed clock та для Low Speed Clock – Crystal/Ceramic Resonator



- 3) У розділі SYS встановити “Serial Wire”.



4) Потрібно налаштувати таймер та DMA. Отже, почне з таймера, було обрано TIM2 для налаштування. Відкриваємо конфігурацію TIM2:

CLOCK SOURCE – INTERNAL CLOCK

CHANNEL 2 – PWM GENERATION CH2

TIM2 Mode and Configuration	
Mode	
Slave Mode	<div>Disable</div>
Trigger Source	<div>Disable</div>
Clock Source	<div>Internal Clock</div>
Channel1	<div>Disable</div>
Channel2	<div>PWM Generation CH2</div>
Channel3	<div>Disable</div>
Channel4	<div>Disable</div>
Combined Channels	<div>Disable</div>
<input checked="" type="checkbox"/> Use ETR as Clearing Source	
<input type="checkbox"/> XOR activation	
<input checked="" type="checkbox"/> One Pulse Mode	

5) Наступним кроком буде налаштування таймера. Відносно початкових налаштувань потрібно лише змінити параметр COUNTER

PERIOD - 89

TIM2 Mode and Configuration

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ GPIO Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

◀▶

i

▼ Counter Settings

Prescaler (PSC – 16 bits value)

0

Counter Mode

Up

Counter Period (AutoReload R...

89

Internal Clock Division (CKD)

No Division

▼ Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)

Disable (Trigger input effect not delayed)

Trigger Event Selection

Reset (UG bit from TIMx_EGR)

▼ PWM Generation Channel 2

Mode

PWM mode 1

Pulse (32 bits value)

0

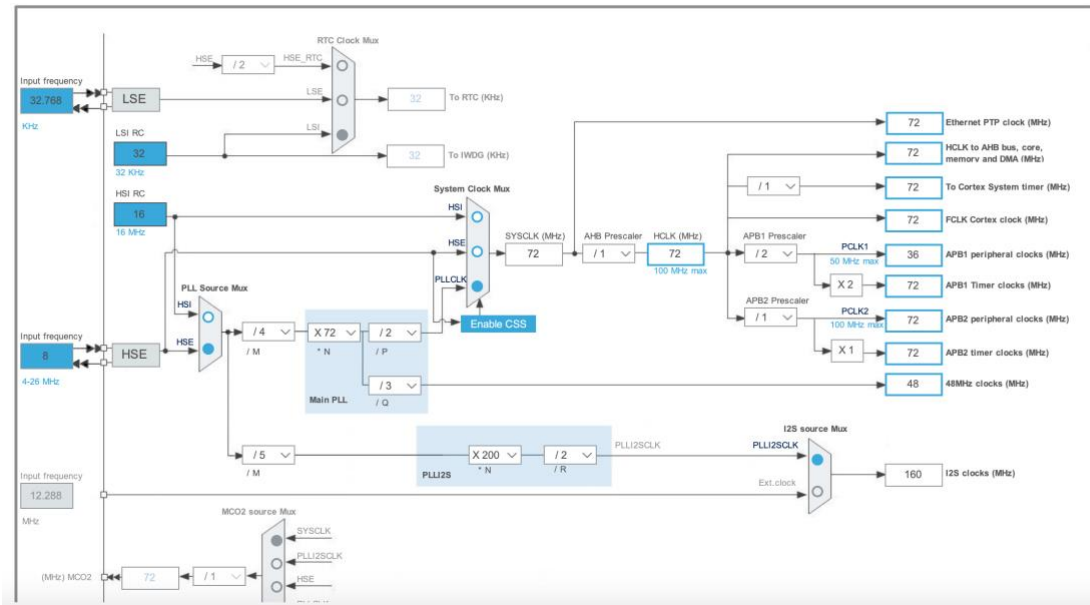
Fast Mode

Disable

CH Polarity

High

6) Відкриваємо Clock configuration та встановлюємо частоту 72 мГц – оптимальна частота для нашого проекту.



7) Повернемо до налаштування DMA. Знову відкриваємо TIM2 та обираємо меню «DMA». Створюємо новий потік для DMA та робимо наступні маніпуляції:

- INCREMENT ADDRESS – MEMORY
- DMA REQUEST – TIM2_CH2/CH4
- STREAM – DMA1 STREAM
- DIRECTION – MEMORY TO PERIPHERAL
-
- DATA WIDTH – HALF WORD

DMA Request	Stream	Direction
TIM2_CH2/CH4	DMA1 Stream 6	Memory To Peripheral

	Peripheral	Memory
Increment Address	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Data Width	Half Word <input type="button" value="v"/>	Half Word <input type="button" value="v"/>
Burst Size	<input type="button" value="v"/>	<input type="button" value="v"/>

8) Генеруємо код.

Створюємо проект, де буде знаходитись реалізація. Виконуємо наступні налаштування, які знаходяться на зображеннях нижче.

Project	<div>Project Settings</div> <div>Project Name</div> <div>ws2812b</div> <div>Project Location</div> <div>/Users/anastasiatraverse/Desktop/POC/diod/</div> <div>Browse</div> <div>Application Structure</div> <div>Basic <input type="button" value="v"/></div> <div><input type="checkbox"/> Do not generate the main()</div> <div>Toolchain Folder Location</div> <div>/Users/anastasiatraverse/Desktop/POC/diod/ws2812b/</div> <div>Toolchain / IDE</div> <div>SW4STM32 <input type="button" value="v"/></div> <div><input checked="" type="checkbox"/> Generate Under Root</div>
Code Generator	
Advanced Settings	<div>Linker Settings</div> <div>Minimum Heap Size</div> <div>0x200</div> <div>Minimum Stack Size</div> <div>0x400</div>
	<div>Mcu and Firmware Package</div> <div>Mcu Reference</div> <div>STM32F411VETx</div> <div>Firmware Package Name and Version</div> <div>STM32Cube FW_F4 V1.23.0 <input type="button" value="v"/></div> <div><input checked="" type="checkbox"/> Use latest available version</div> <div><input checked="" type="checkbox"/> Use Default Firmware Location</div> <div>atraverse/STM32Cube/Repository/STM32Cube_FW_F4_V1.23.0</div> <div>Browse</div>

Project
Code Generator
Advanced Settings

STM32Cube Firmware Library Package

- ☐ Copy all used libraries into the project folder
- ☒ Copy only the necessary library files
- ☐ Add necessary library files as reference in the toolchain project configuratio...

Generated files

- ☒ Generate peripheral initialization as a pair of '.c/.h' files per peripheral
- ☒ Backup previously generated files when re-generating
- ☒ Keep User Code when re-generating
- ☒ Delete previously generated files when not re-generated

HAL Settings

- ☒ Set all free pins as analog (to optimize the power consumption)
- ☒ Enable Full Assert

Template Settings

Select a template to generate customized code
Settings...

9) Коли код згенерувався варто перевірити налаштування пінів, який ви обрали для передачі інформації світлодіодам.

У нашому випадку PB8. Основне, що потрібно перевірити швидкість на піні.

Тобто, потрібно перевірити рядок `GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;`

Потрібно швидкість – `GPIO_SPEED_FREQ_VERY_HIGH;`

```

/*Configure GPIO pin : PB8 */
GPIO_InitStruct.Pin = GPIO_PIN_8;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

```