

## Лабораторна робота №8

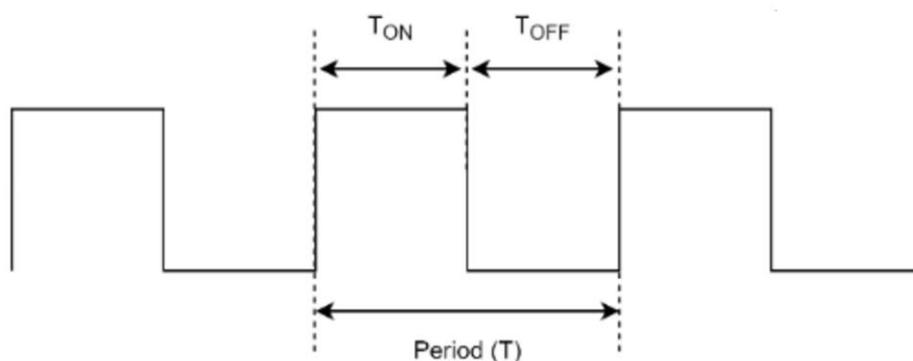
### ШИРОТНО-ІМПУЛЬСНА МОДУЛЯЦІЯ ТА ЇЇ ВИКОРИСТАННЯ

Мета: отримати базові навички управління широтно-імпульсною модуляцією. Програмна взаємодія та виконання прикладних задач в мікроконтролерах STM32.

#### Теоретичні відомості

**Широтно - імпульсна модуляція ( ШІМ )** – це метод генерації безперервного змінного цифрового сигналу ВИСОКОГО/НИЗЬКОГО рівня та програмного керування його шириною та частотою імпульсу. Деякі навантаження, такі як (світлодіоди, двигуни тощо), реагують на **середню напругу** сигналу, яка зростає зі збільшенням ширини імпульсу ШІМ-сигналу. Цей метод широко використовується у вбудованих системах для керування яскравістю світлодіодів, швидкістю двигуна та іншими застосуваннями.

ШІМ-сигнал має кілька характеристик. Перша з них – це частота, яка, по суті, є мірою того, як швидко ШІМ-сигнал чергується між ВИСОКИМ та НИЗЬКИМ рівнем. Частота вимірюється в Гц і є оберненою величиною повного періоду часу. Ось як це виглядає графічно та як вона представлена в математичній формулі.



$$Frequency[PWM] = \frac{1}{Period(T)} Hz$$

ШВД (коефіцієнт заповненості) ШІМ – найважливіша характеристика, яка нас завжди цікавить. Це міра того, як довго сигнал ШІМ залишається увімкненим відносно повного періоду циклу ШІМ. Рівняння коефіцієнта заповненості ШІМ виглядає наступним чином:

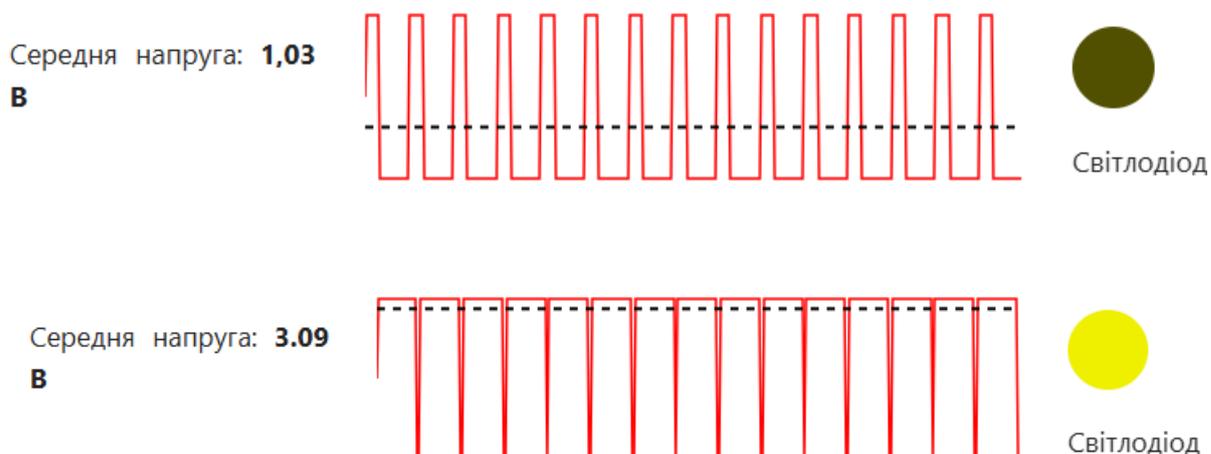
$$DutyCycle[PWM] = \frac{T_{ON}}{T_{ON}+T_{OFF}} \times 100 = \frac{T_{ON}}{Period(T)} \times 100[\%]$$

Штучний цикл зазвичай виражається у відсотках ( % ), оскільки це співвідношення між двома часовими величинами. І він безпосередньо впливає на загальну (середню) напругу ШІМ, на яку реагує більшість пристроїв..

Роздільна здатність ШІМ виражається в (бітах). Це кількість бітів, які використовуються для представлення значення шпаруватості. Вона може бути 8, 10, 12 або навіть 16 біт. Роздільну здатність ШІМ можна визначити як кількість дискретних рівнів шпаруватості між 0% і 100%. Чим вища роздільна здатність ШІМ, тим більша кількість дискретних рівнів у всьому діапазоні шпаруватості ШІМ.

Роздільна здатність ШІМ лише **3 біти** означає, що існує лише **8 дискретних рівнів** для робочого циклу в усьому діапазоні (від 0% до 100%). З іншого боку, ШІМ з роздільною

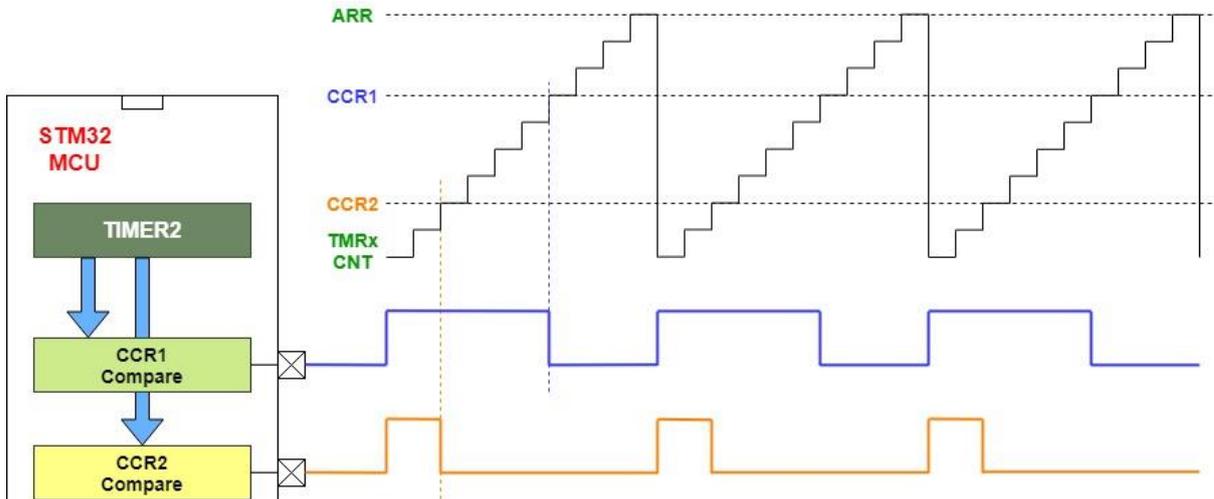
здатністю **8 бітів** матиме **256 дискретних рівнів** для робочого циклу в усьому діапазоні (від 0% до 100%). Ви можете скористатися інтерактивним інструментом нижче, щоб перевірити це самостійно.



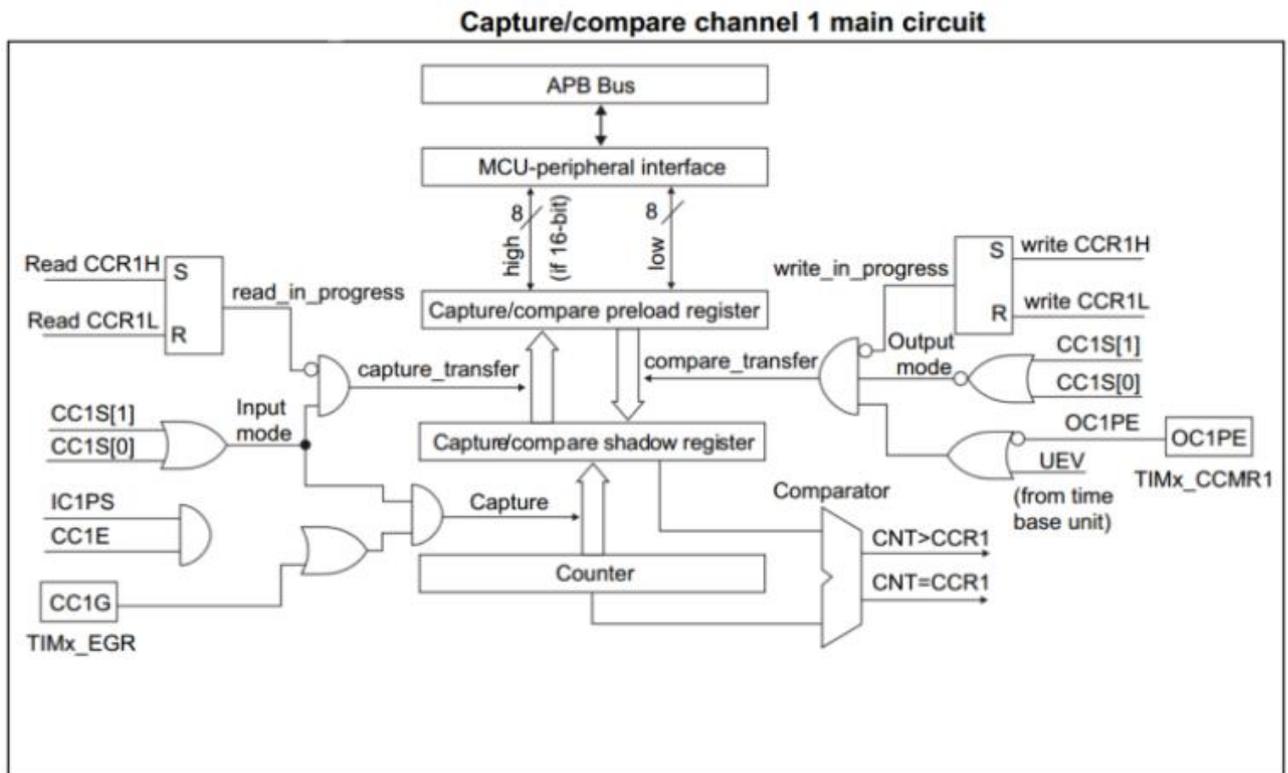
Коли таймер синхронізується з внутрішнього джерела та рахує до значення регістра автоматичного перезавантаження, вихідний контакт каналу переводиться у **ВИСОКИЙ** рівень. І залишається таким, доки лічильники таймера не досягнуть значення регістра  $CCRx$ , подія збігу призводить до того, що вихідний контакт каналу переводиться у **НИЗЬКИЙ** рівень. І залишається таким, доки таймер не рахує до значення регістра автоматичного перезавантаження, і так далі.

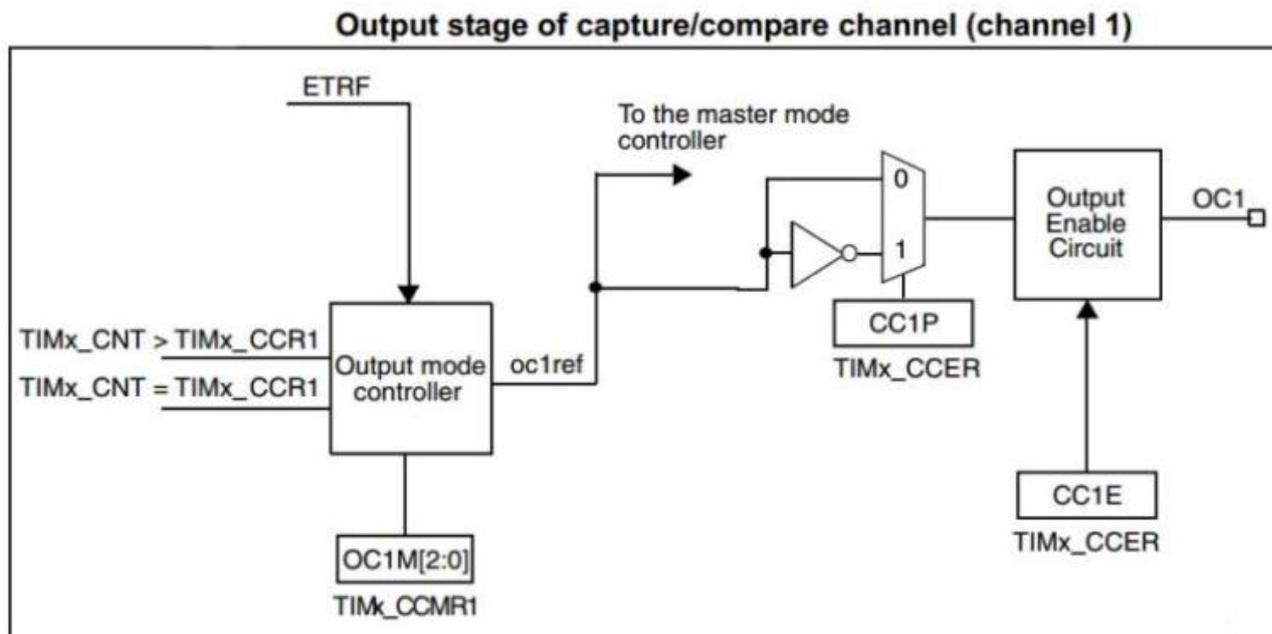
Отримана форма хвилі називається ШІМ (імпульсно-широотно-модульованим) сигналом. Його частота визначається внутрішнім тактовим генератором, передділювачем та регістром  $ARRx$ . А його шпаруватість визначається значенням регістра  $CCRx$  каналу. ШІМ не завжди повинен дотримуватися точно такої ж процедури генерації ШІМ, проте це найпростіший і найлегший для розуміння режим. Він називається режимом ШІМ з підвищеним рахунком. Ми обговоримо подальші вдосконалені методи генерації ШІМ у цій серії навчальних посібників.

На наступній діаграмі показано, як значення  $ARR$  впливає на період (частоту) ШІМ-сигналу. А також як значення  $CCRx$  впливає на шпаруватість відповідного ШІМ-сигналу. Також проілюстровано весь процес генерації ШІМ-сигналу в режимі нормального рахування з висхідним відліком.



Кожен канал захоплення/порівняння побудований навколо регістра захоплення/порівняння (включаючи тінювий регістр), вхідного каскаду для захоплення (з цифровим фільтром, мультиплексуванням та попереднім дільником) та вихідного каскаду (з компаратором та керуванням виходом). Вихідний каскад генерує проміжну форму сигналу, яка потім використовується для опорної: OCxRef (активний високий рівень). Полярність діє на кінці ланцюга.





Один таймер STM32 зазвичай має кілька каналів (4, 6 або будь-яку іншу кількість, зазначену в технічному описі). Тому, використовуючи один таймер, ви можете незалежно генерувати кілька ШІМ-сигналів з різною шпаруватістю, звичайно, але вони матимуть однаковий час (однакову частоту), і всі вони будуть синхронізовані. Ми зробимо це в другій лабораторній роботі цього посібника після того, як налаштуємо один ШІМ-канал і запустимо все в роботу.

Ось знімок діаграми загального призначення Timer2, який підкреслює наявність кількох вихідних каналів порівняння та вихідних драйверів.

Режим широтно-імпульсної модуляції дозволяє генерувати сигнал з частотою, що визначається значенням регістра  $TIMx\_ARR$ , та шпаруватістю, що визначається значенням регістра  $TIMx\_CCRx$ . Режим ШІМ можна вибрати незалежно на кожному каналі (один ШІМ на вихід  $OCx$ ), записавши 110 (режим ШІМ 1) або '111 (режим ШІМ 2) у біти  $OCxM$  регістра  $TIMx\_CCMRx$ .

Користувач повинен увімкнути відповідний регістр попереднього завантаження, встановивши біт  $OCxPE$  в регістрі  $TIMx\_CCMRx$ , а також регістр попереднього завантаження автоматичного перезавантаження, встановивши біт  $ARPE$  в регістрі  $TIMx\_CR1$ .

Полярність  $OCx$  програмується програмно за допомогою біта  $CCxP$  у регістрі  $TIMx\_CCER$ . Її можна запрограмувати як активний високий або активний низький рівень. Цей варіант підійде для застосувань, де потрібно генерувати комплементарні ШІМ-сигнали.

У режимі ШІМ (1 або 2),  $TIMx\_CNT$  та  $TIMx\_CCRx$  завжди порівнюються, щоб визначити, чи  $TIMx\_CCRx \leq TIMx\_CNT$  або  $TIMx\_CNT \leq TIMx\_CCRx$  (залежно від напрямку лічильника). Таймер здатний генерувати ШІМ у режимі вирівнювання по фронту або по центру залежно від бітів  $CMS$  у регістрі  $TIMx\_CR1$ .

У різних застосуваннях вам знадобиться генерувати ШІМ-сигнал із певною частотою. У керуванні серводвигунами, драйверах світлодіодів, драйверах двигунів та багатьох інших ситуаціях вам знадобиться встановити потрібну частоту для вихідного ШІМ-сигналу.

Період ШІМ ( $1/F_{\text{ШІМ}}$ ) визначається такими параметрами: значенням ARR, значенням попереднього дільника та самим внутрішнім тактовим генератором, який керує модулем таймера FCLK. Наведена нижче формула використовується для розрахунку  $F_{\text{ШІМ}}$  для виходу. Ви можете встановити тактовий генератор, який ви використовуєте, попередній дільник, та знайти значення ARR, щоб керувати  $F_{\text{ШІМ}}$  та отримати потрібне значення.

$$F_{\text{PWM}} = \frac{F_{\text{CLK}}}{(ARR + 1) \times (PSC + 1)}$$

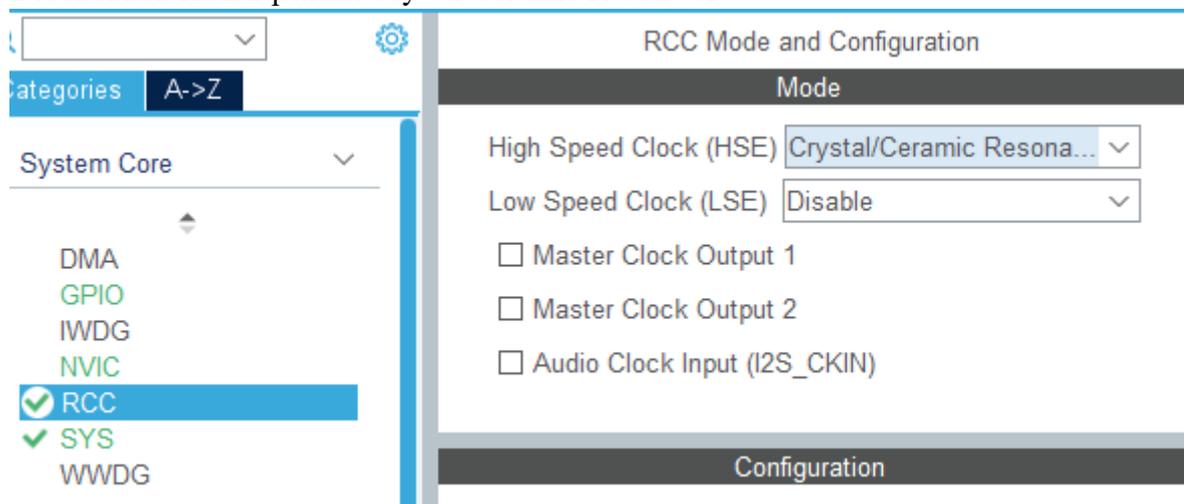
У звичайних налаштуваннях, припускаючи, що ви використовуєте модуль таймера в режимі ШІМ та генеруєте ШІМ-сигнал у режимі вирівнювання по фронту з конфігурацією рахунку вперед. Відсоток шпаруватості контролюється зміною значення регістра CCRx. А шпаруватість дорівнює  $(CCRx/ARR)$  [%].

$$\text{DutyCycle}_{\text{PWM}}[\%] = \frac{CCRx}{ARRx}[\%]$$

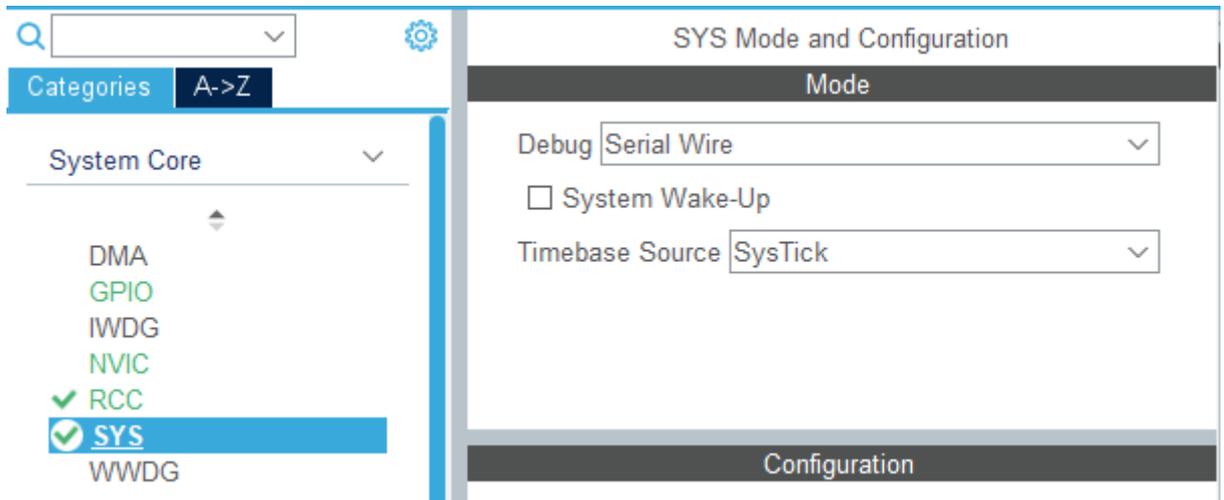
Однією з найважливіших властивостей ШІМ-сигналу є роздільна здатність. Це кількість дискретних рівнів шпаруватості, які можна встановити. Це число визначає, скільки кроків може пройти шпаруватість, доки не досягне максимального значення. Отже, розмір кроку або кількість кроків шпаруватості може показати, наскільки точно можна змінювати шпаруватість, щоб досягти певного відсотка. Це може бути надзвичайно важливим у деяких аудіозастосуваннях, системах керування двигунами або навіть системах керування освітленням.

Для включення ШІМ каналу в мікроконтролерах STM32 потрібно:

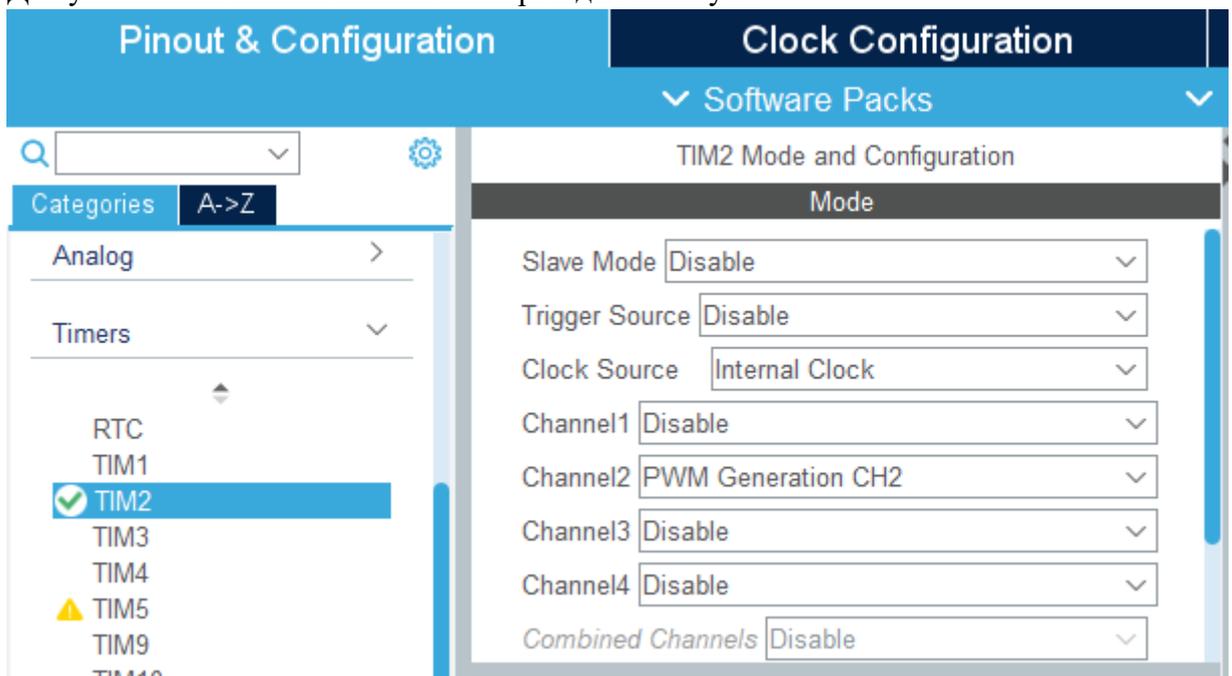
1. Встановити HSE в режим Crystal/Ceramic Resonator:



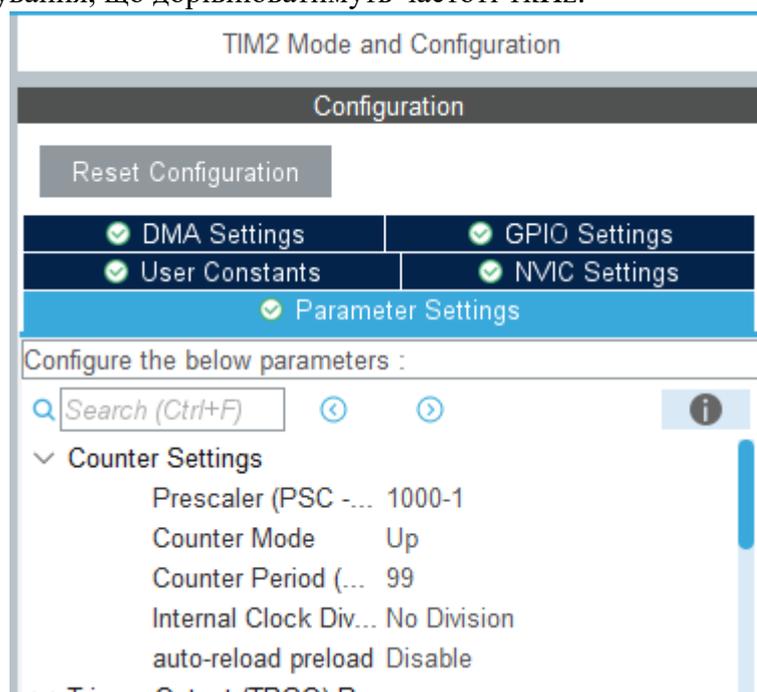
2. Для дебагу увімкнути Serial Wire



3. Для увімкнення самого ШІМ в таймері 2 для каналу 2:



4. Задати налаштування, що дорівнюватимуть частоті 1kHz:



5. В основному файлі main.c додати код включення ШІМ сигналу для таймера 2 канал 2:

```
/* USER CODE BEGIN 2 */
  HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);
/* USER CODE END 2 */
```

6. Підключити світлодіод до плати PA1 та реалізувати код плавного миготіння:

```
while (1)
{
  /* USER CODE END WHILE */
  for(int i=0; i<100; i++){
    __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_2, i);
    HAL_Delay(3);
  }
  for(int j=100; j>0; j--){
    __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_2, j);
    HAL_Delay(3);
  }
  /* USER CODE BEGIN 3 */
}
```

### Індивідуальне завдання

1. Підключити енкодер до плати та реалізувати регулювання інтенсивністю світлодіоду.
2. Підключити 2 світлодіоди та реалізувати різні режими асинхронного миготіння. Зміна режиму відбувається за допомогою натискання кнопки користувача.

### Зміст звіту

1. Тема та мета лабораторної роботи.
2. Скріншот конфігурації мікропроцесора (файл .ioc) для кожного пункту індивідуального плану;
3. Код файлу main.c функції main для кожного пункту індивідуального плану;
4. Висновки.

### Контрольні запитання

1. Що таке ШІМ-сигнал і для чого він використовується?
2. Як визначається робочий цикл (duty cycle) ШІМ-сигналу?
3. Як впливає зміна робочого циклу на яскравість світлодіода?
4. Чим відрізняється зміна частоти ШІМ від зміни робочого циклу?
5. Які переваги ШІМ перед аналоговим керуванням струмом?