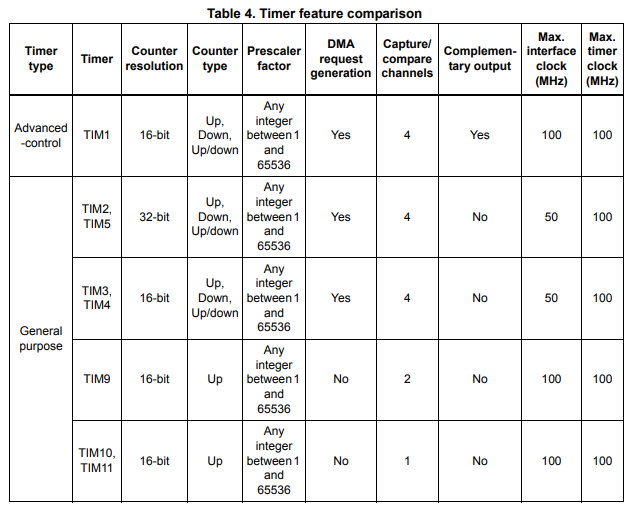
**Sử dụng Timer mode Output Compare**

**I, Kiến thức cần chuẩn bị**

**1.Timer**

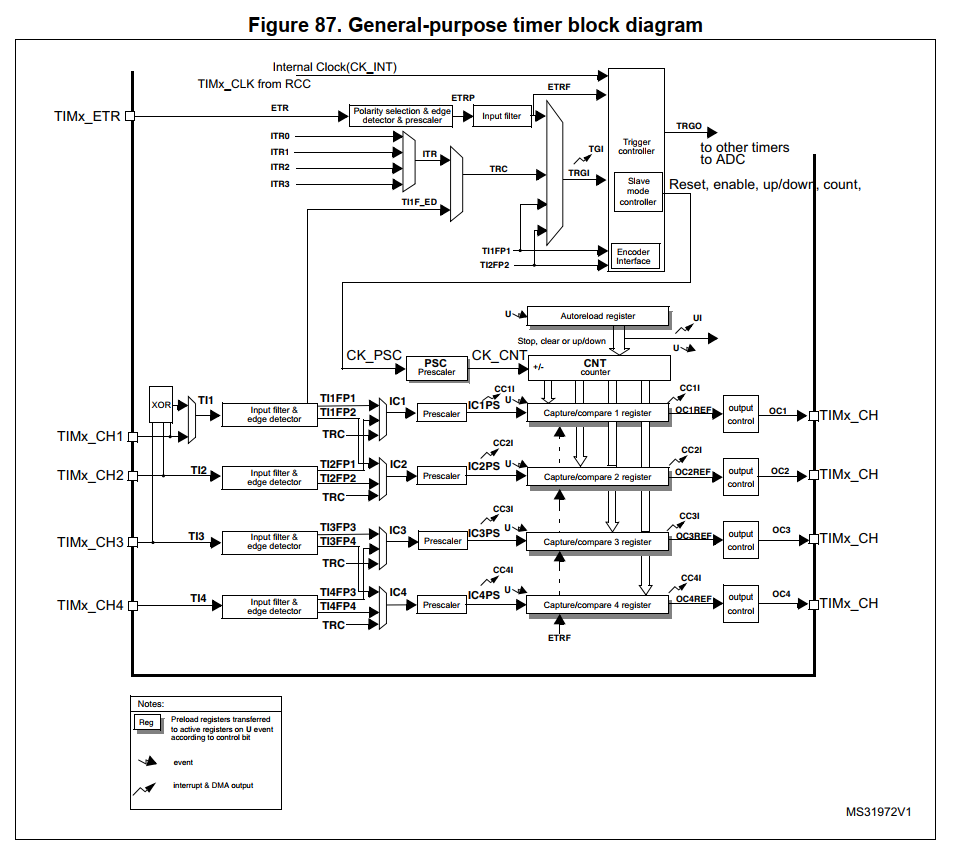
Timer là một loại ngoại vi được tích hợp ở hầu hết các vi điều khiển, cung cấp cho người dùng nhiều ứng dụng như xác định chính xác một khoảng thời gian, đo – đếm xung đầu vào, điều khiển dạng sóng đầu ra, băm xung PWM.

STM32F411 có 8 bộ Timer, trong đó có 1 bộ Advanced – control timer (TIM1) thường được các bộ thư viện sử dụng để tạo bộ đếm thời gian chuẩn của hệ thống (như ngắt System Tick, hàm tạo Delay, TimeOut…), và 7 bộ General – purpose timer (TIM2 đến TIM5 và TIM9 đến TIM11).

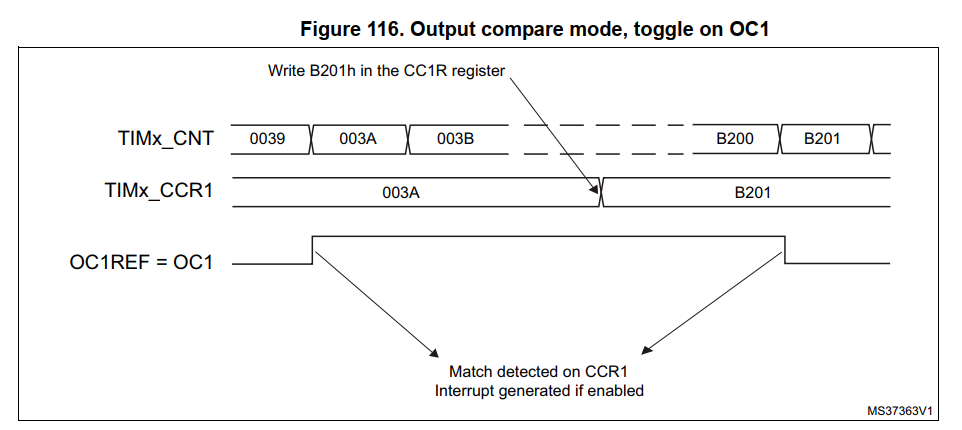


**2.Các kênh Compare**

**Output compare:**Đây là một chức năng của bộ Timer với rất nhiều chế độ hoạt động được hỗ trợ. Cơ chế hoạt động chung là sẽ kiểm tra giá trị TIMxCNT đếm tới giá trị bằng với giá trị được nạp sẵn vào thanh ghi TIMx\_CCRy thì một ngắt sẽ được tạo ra đồng thời có thể xuất xung đầu ra tại OCy output.



Lấy ví dụ ta chọn chế độ khi TIMx\_CNT = TIMx\_CCRy có ngắt xảy ra, chân OCy sẽ Toggle mức logic đầu ra.



Còn lại chúng ta có tới 8 chế độ để sử dụng với bộ Output Compare của Timer.

A screenshot of a cell phone

Description automatically generated

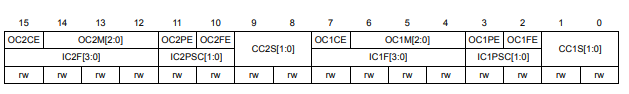
**Vậy Output Compare và PWM khác nhau ở chỗ nào ?**

**Một số thanh ghi quan trọng cho các General Timer (TIM2 ,..,TIM5):**

**TIMx\_CCMR1 (TIMx capture/compare mode register 1)**

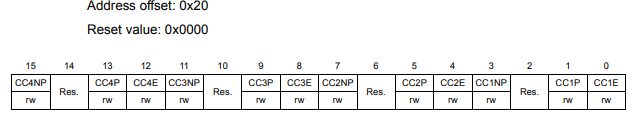
Address offset: 0x18

Reset value: 0x0000



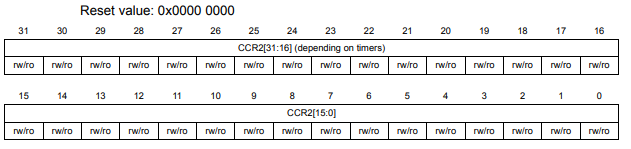
Thanh ghi cấu hình chức năng hoạt động của Timer Mode Capture/Compare

**TIMx\_CCER (TIMx capture/compare enable register )**



Thanh ghi enable chức năng Output Compare và cấu hình xung đầu ra của chân OCy.

**TIMx\_CCRy (TIMx capture/compare register , channel y** với y từ 1 đến 4**)**



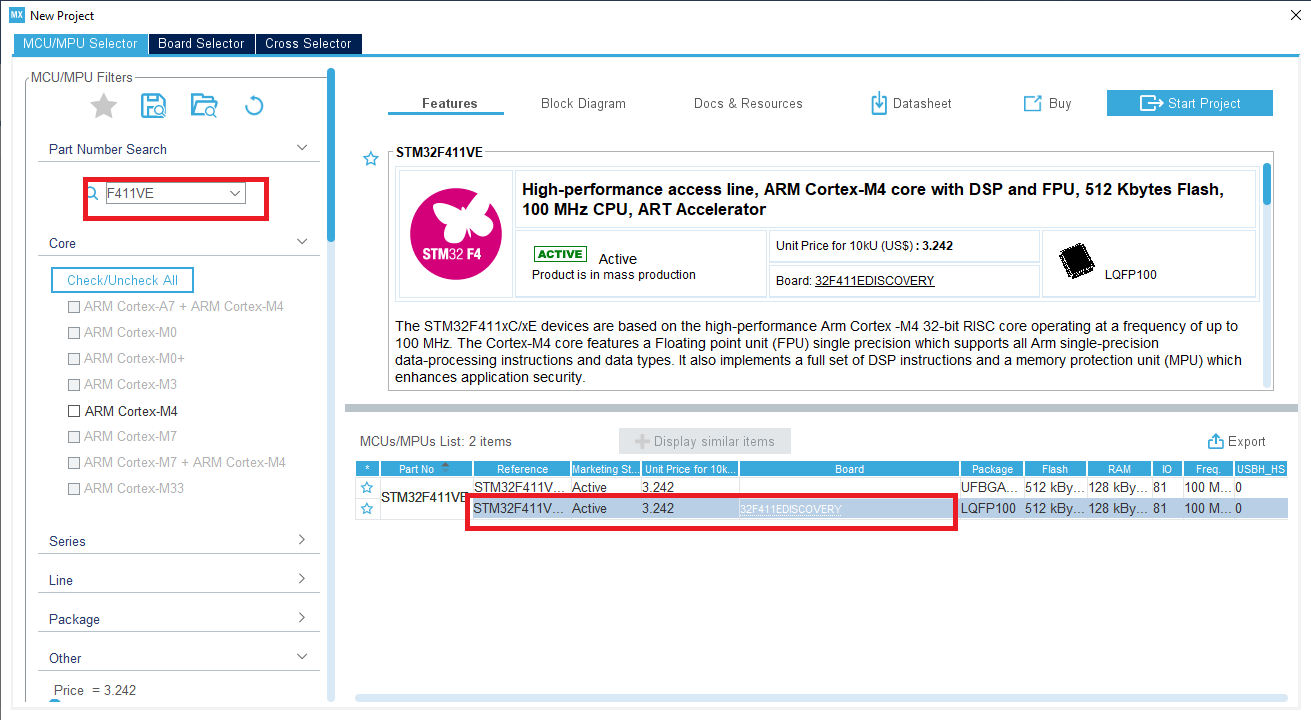
Thanh ghi ứng với các Channel chứa giá trị để so sánh với CNT.

**Ý tưởng demo: Mình sẽ cấu hình Timer đến từ 0 → 9999, trong đó sẽ nạp sẵn giá trị 999 vào thanh ghi TIMx\_CCRy và đảo mức logic ở chân OCy. Sau đó mình sẽ reset TIMx\_CNT về 0 và bắt đầu lại chu trình. Các bạn sẽ nhìn thấy sự thay đổi xung nhờ việc nhấp nháy LED tại chân OCy nhé.**

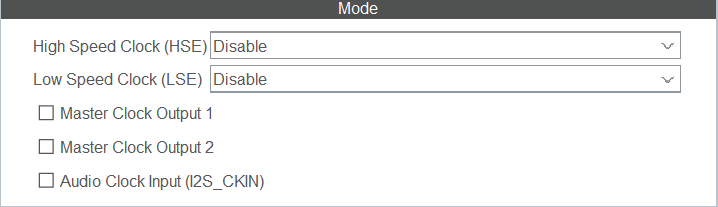
**II, Lập trình**

Mở phần mềm STMCubeMX, chọn dòng chip bạn sử dụng. Ở đây mình chọn chip STM32F411VE.

Đối với các dòng chip STM32 đời 4, tất cả mọi câu lệnh khi sử dụng thư viện HAL đều giống nhau. Chỉ khác nhau phần cấu hình Clock phụ thuộc riêng vào mỗi Chip.



Bài này mình sẽ sử dụng nguồn dao đông RC nội để cấp Clock cho Chip nhé.



Cấu hình Chip Debug sử dụng mode SWD.

A screenshot of a social media post

Description automatically generated

Ở đây mình sử dụng TIM4\_CH1 để trùng với 1 chân LED trên DEV KIT, như vậy xung xuất ra trên chân OC1 sẽ làm cho LED sáng tối theo ý mình điều chỉnh.

Cấu hình chân là Output Compare lấy nguồn clock từ Clock nội (HSI) được chip hỗ trợ sẵn.

Cấu hình khi có ngắt xảy ra (TIM4\_CNT = TIM4\_CCR1) chân OC1 sẽ đảo mức logic (Toggle on match). Mình muốn nhấp nháy LED mà.

Chọn mức Polarity là mức High (cao) nhé.

A screenshot of a social media post

Description automatically generated

Ở đây mình chọn Prescaler = 9999 và Period = 9999 → .

Ở đây mình cài đặt giá trị TIM4\_CCR1 là 999. Nghĩa là khi TIM4\_CNT đếm từ 0 → 999 sẽ xảy ra ngắt Output Compare.

Tiếp theo mình sẽ cấu hình tần số Clock cho Chip như mọi bài mình chọn sử dụng nguồn Clock nội (HSI) được chip hỗ trợ sẵn, như vậy mình sẽ không cần hàn thêm thạch anh ngoại mà chương trình vẫn chạy được. Clock đi qua bộ nhân tần PLLCLK để đạt được tần số hoạt động tối đa mà chip hỗ trợ HCLK = 100MHz. Việc còn lại Cube MX sẽ tự cấu hình cho các bạn.

A screenshot of a social media post

Description automatically generated

Cuối cùng chọn File và sinh code cho Project.

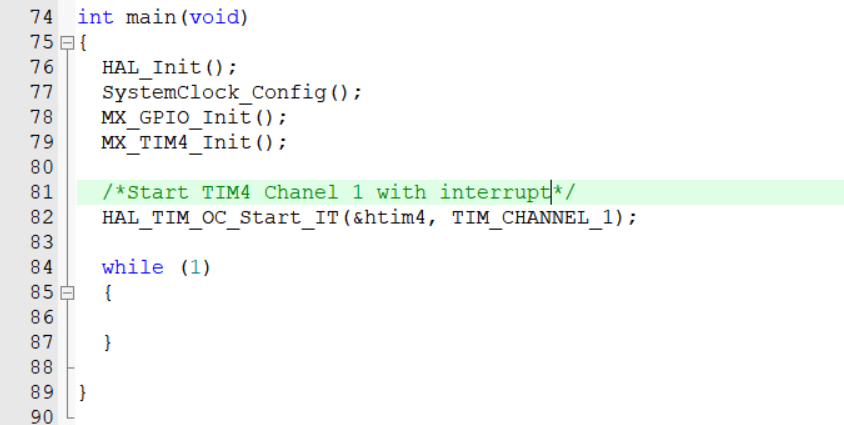
A screenshot of a social media post

Description automatically generated

Chọn những thư viện cần thiết để sinh code nhanh hơn và giảm dung lượng Project nhé.

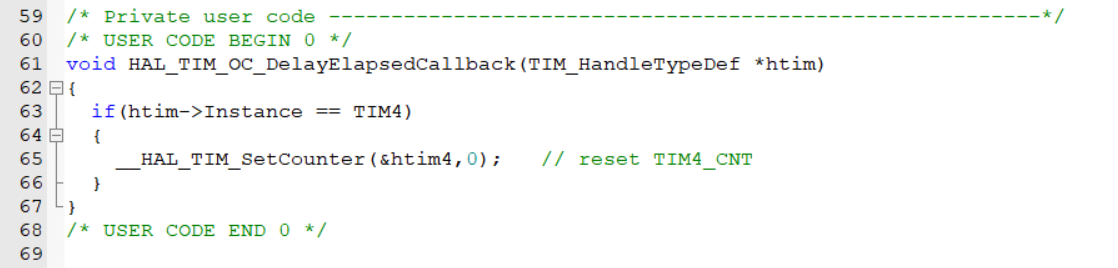
Nào bắt đầu đi vào sửa code nhé !!!

Ở hàm main, mình khai báo các hàm quan trọng.



Khi Output Compare xảy ra ngắt, mình sẽ reset biến đếm CNT về 0 và chân OC1 sẽ đảo mức logic làm thay đổi trạng thái LED.

**Các bạn thử tính xem chu kì nhấp nháy LED sẽ là bao nhiêu nhé ?**



**Kết quả Demo trên chân xuất Output của TIM4**

A screenshot of a computer

Description automatically generated

Đây chỉ là 1 trong 8 chế độ hoạt động của bộ Output Compare thôi nhé các bạn. Chúc các bạn thành công !!!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DEVIOT - CÙNG NHAU HỌC LẬP TRÌNH IOT

📌 Website: deviot.vn

📌 FanPage: Deviot - Thời sự kỹ thuật & IoT

📌 Group: Deviot - Cùng nhau học lập trình IOT

📌 Hotline: 0969.666.522

📌 Address: Số 101C, Xã Đàn 2

📌 Đào tạo thật, học thật, làm thật