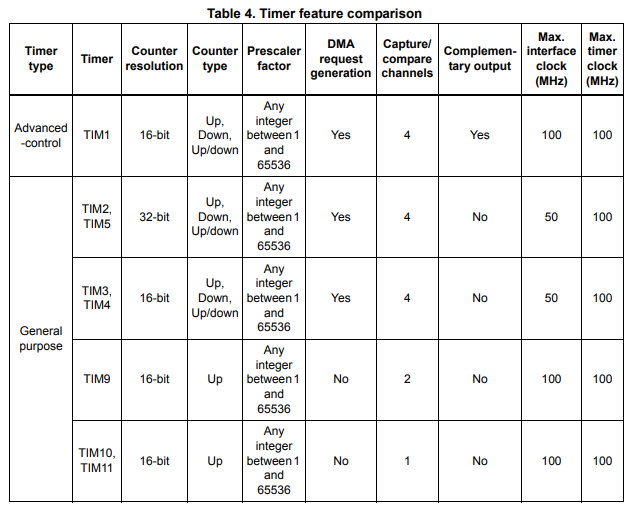
**Sử dụng Timer Input Capture**

**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 …

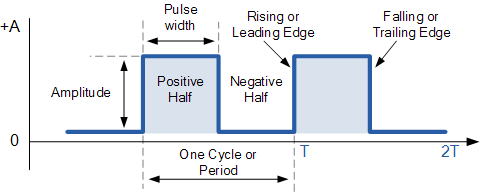
STM32F411 có 8 bộ Timer, trong đó có 1 bộ Advanced – control timer (TIM1) thường được các bộ thư viện sử dụng để tạo bộ đếm thời gian chuẩn của hệ thống (như ngắt System Tick, các hàm tạo Delay, TimeOut…), và 7 bộ General–Purpose Timer (TIM2 đến TIM5 và TIM9 đến TIM11).

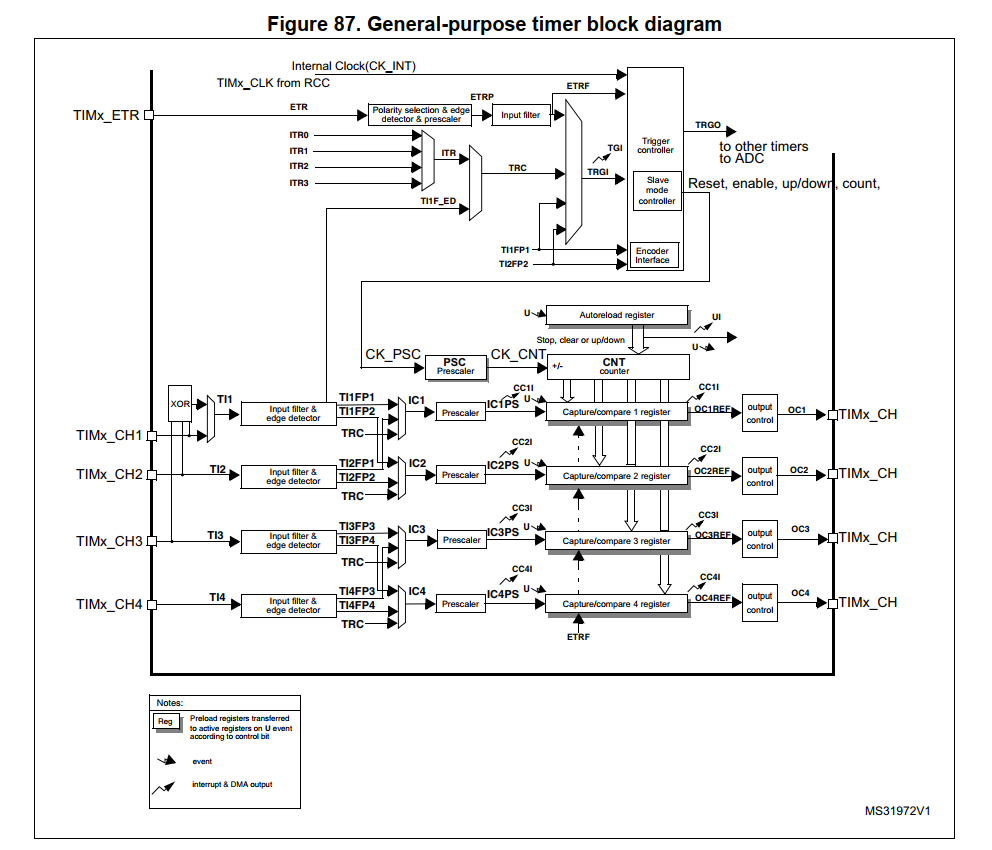


**2.Các kênh Capture**

Tương tự như Compare thì mode Capture cũng có 4 kênh.

*Input Capture : Ở chế độ Input capture, thanh ghi* *TIMx\_CCRx* *của kênh đầu vào tương ứng sẽ được sử dụng để lưu giá trị của CNT khi phát hiện sự thay đổi mức logic (sườn lên/ sườn xuống) như được cấu hình trước đó. Từ đó có thể biết được khoảng thời gian giữa 2 lần có sườn lên hoặc sườn xuống.*



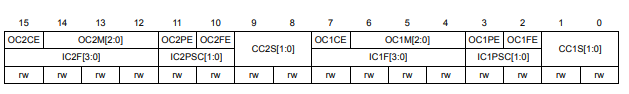


**Một số thanh ghi quan trọng của các General Timer (TIM2 ,…,TIM5)**

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

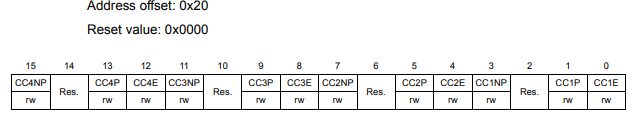
Address offset: 0x18

Reset value: 0x0000

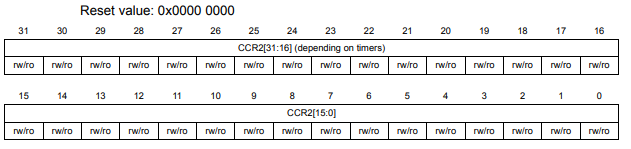


Thanh ghi cấu hình chức năng hoạt động của Timer mode Capture/Compare

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



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



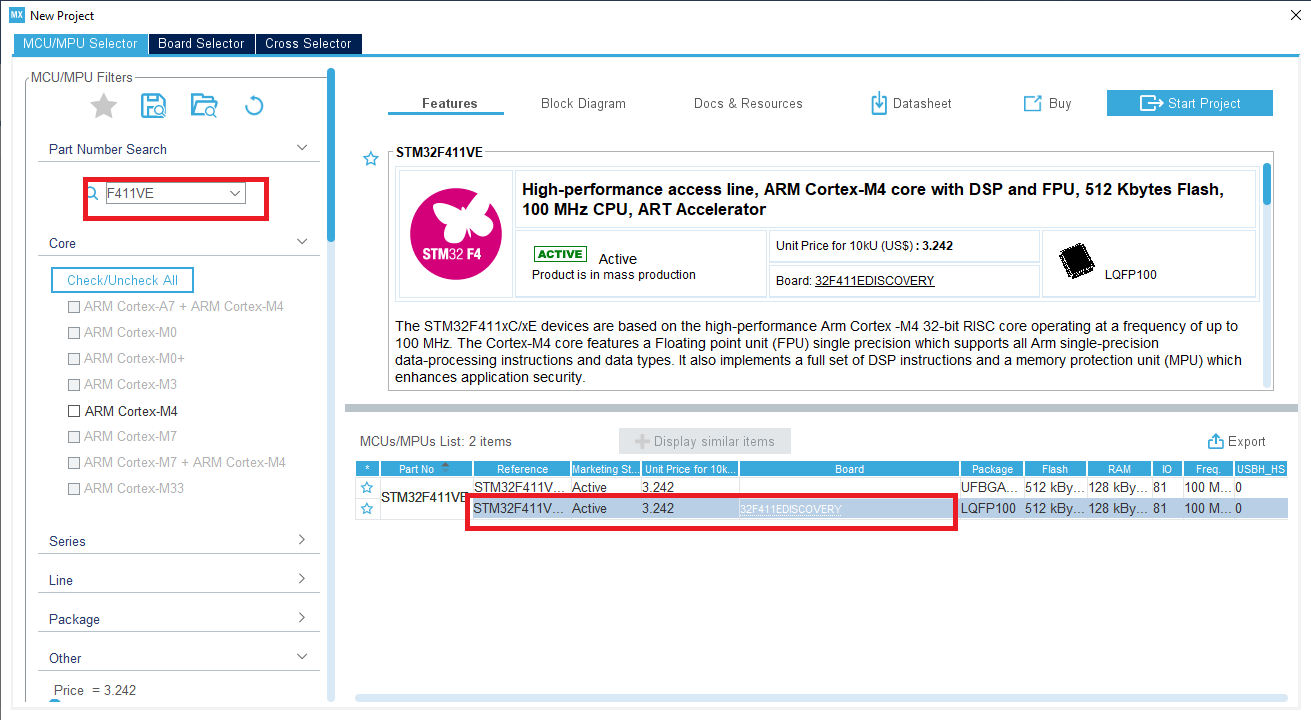
Thanh ghi ứng với các Channel sẽ lấy giá trị CNT khi tín hiệu đầu vào thay đổi theo mức logic 1 xuống 0 hay từ 0 lên 1.

**Ý tưởng demo: Mình sẽ cấu hình cho Timer đếm từ 0→999999 và cấu hình Input Capture bắt xung sườn lên. Cứ mỗi lần mình bấm nút PA0 trên KIT, sẽ có sự thay đổi mức logic từ 0→1 và bộ Input Capture ghi giá trị thanh ghi** *CNT* **tới thanh ghi** *TIMx\_CCRx.* **Cuối cùng mình sẽ đọc giá trị này ra để sử dụng cho các mục đích khác.**

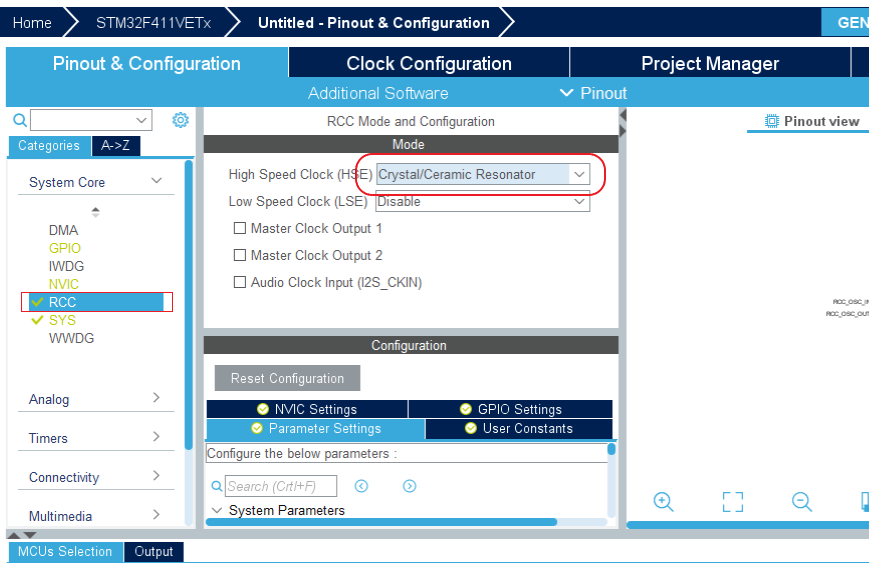
**II, Lập trình**

Mở phần mềm STM CubeMX, 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 hỗ trợ và cần chú ý phần này để cấu hình tân số hoạt động cho Chip.



Sau đó cấu hình Chip sử dụng thạch anh ngoài hàn sẵn trên board mạch.



Cấu hình Chip Debug bằng mode SWD.

A screenshot of a social media post

Description automatically generated

Cấu hình button trên board là TIM2\_CH1 ứng với chân PA0 và Chọn nguồn clock là xung clock nội và mode là Input\_Capture cho CH1.

A screenshot of a computer

Description automatically generated

Tiếp theo ta sang cấu hình thông số cho Timer. Ở đây mình chọn Prescaler = 9999, Period = 999999. Chế độ Counter Up.

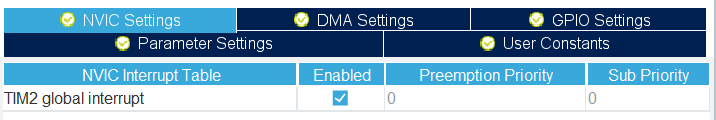
Đối với cấu hình như trên = 10,000 (Hz). Tức cứ sau CNT lại tăng lên 1 giá trị. Và CNT sẽ tăng tới giá trị 999999 (đếm từ 0 → 999999) sẽ được reset về 0.

A screenshot of a cell phone

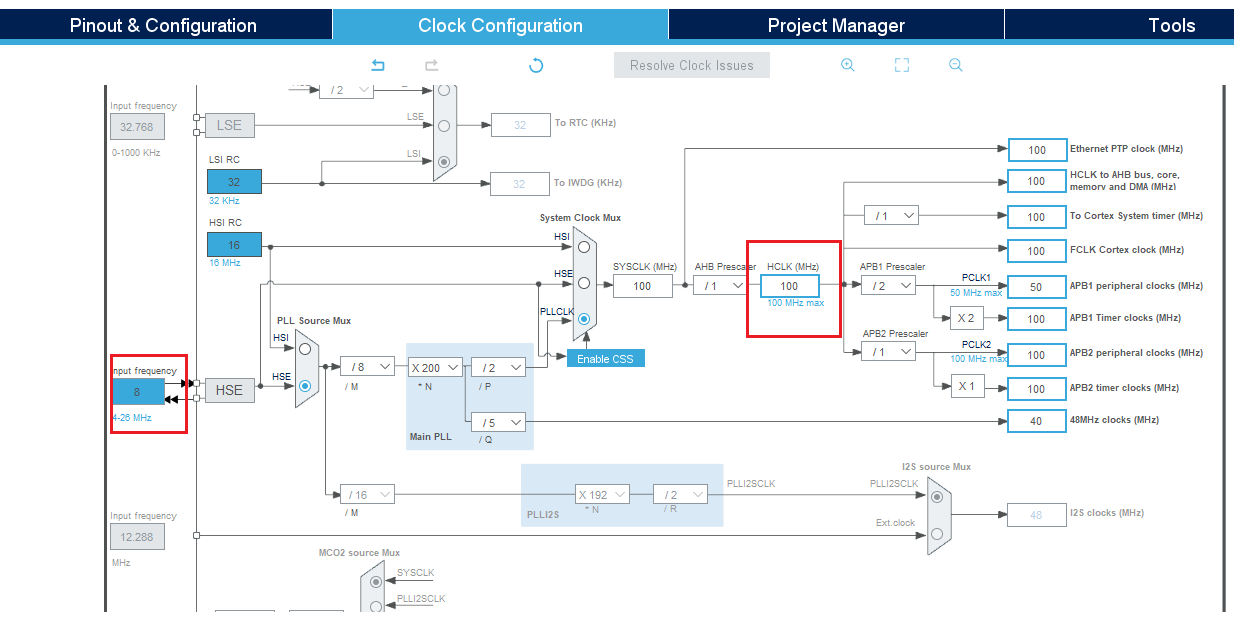
Description automatically generated

Polarity Selection : Rising Edge. Tức là khi chân PA0 nhận được 1 xung sườn lên (từ 0→1) nó sẽ ghi giá trị CNT vào thanh ghi TIMx\_CCRx.

Tiếp theo ta sẽ Enable ngắt Timer lên



Giờ thì chuyển sang Tab Clock Configuration. Chọn đầu vào Clock thạch anh 8MHz. Chọn HCLK là 100MHz, tần số tối đa mà Chip hỗ trợ đạt được.



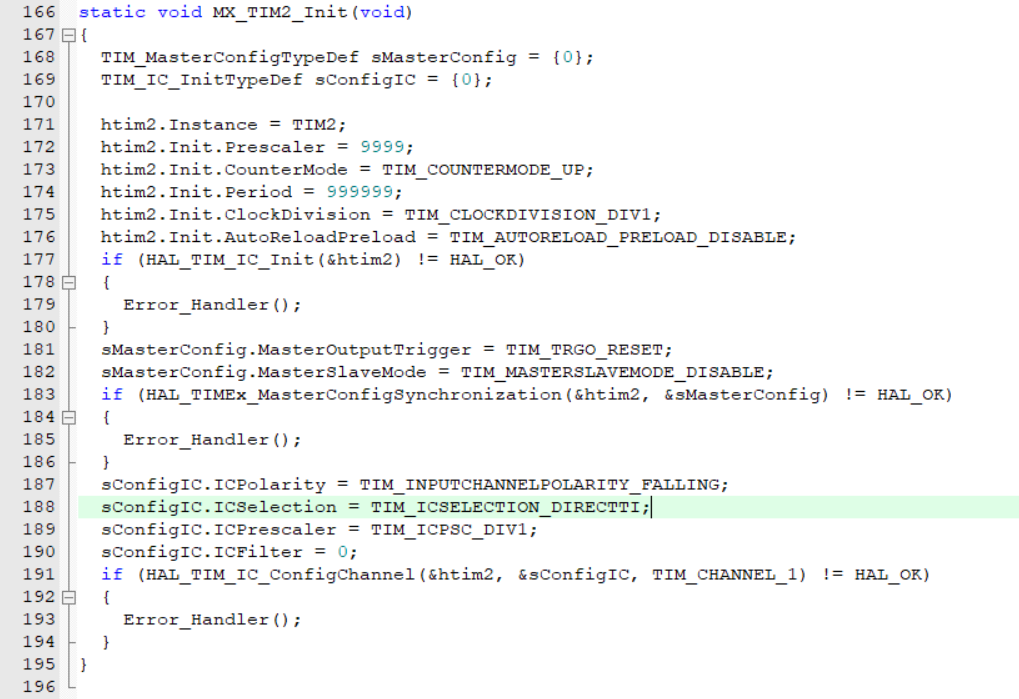
Cuối cùng chọn file và sinh code cho Project

A screenshot of a social media post

Description automatically generated

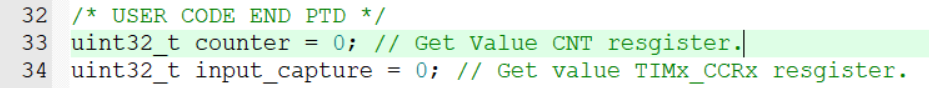
Chọn những thư viện cần thiết để sinh code nhanh hơn và giảm dung lượng Project nhé.

Ở KeilC ta sẽ có các hàm khởi tạo cho Timer 2 và Channel input Capture tương tự như các bài trước.

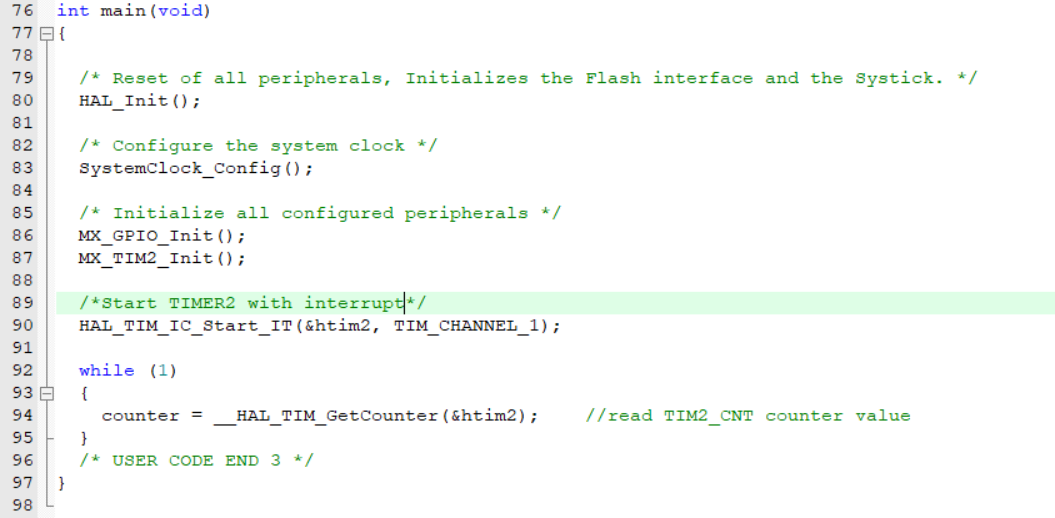


Cube đã tự sinh ra hàm khởi tạo cho chúng ta và bạn có thể sửa lại các thuộc tính ngay tại đây.

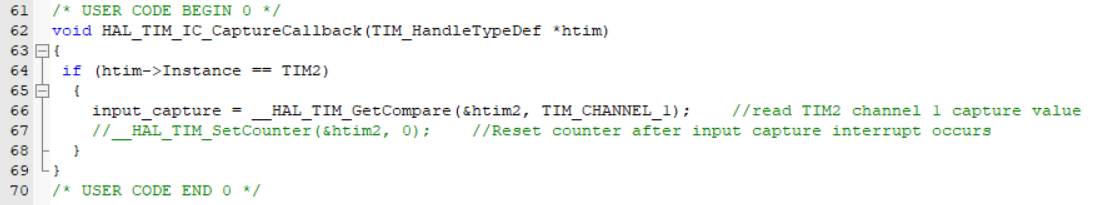
Trong main.c khai báo cáo biến cần thiết để lấy được thời gian khi nhấn nút.



Trong hàm main mình sẽ khởi tạo các Module cần thiết. Và liên tục đọc về giá trị CNT để hiển thị lên Watch.



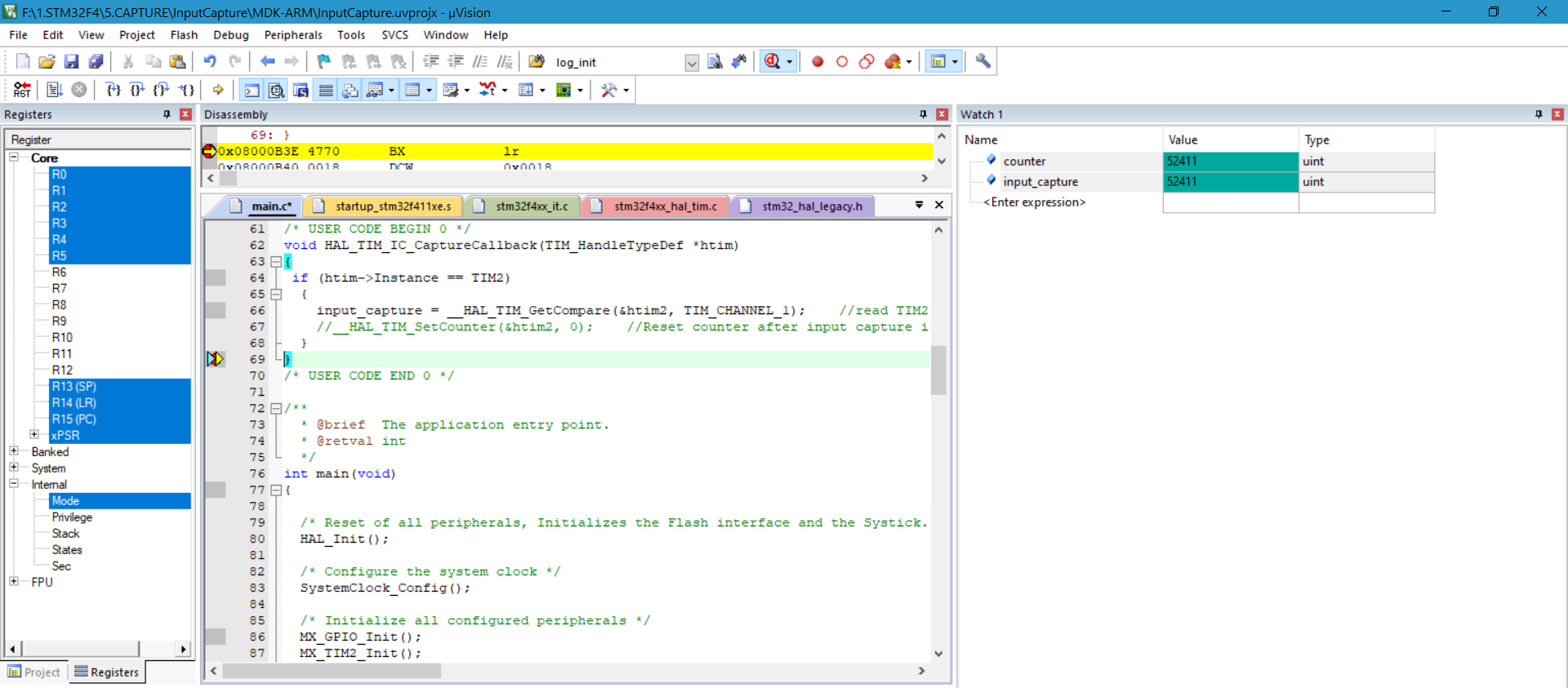
Tiếp theo mình sẽ khai báo 1 hàm xử lý ngắt khi có Input Capture xảy ra.



Như vậy mỗi khi mình bấm nút, mình sẽ đọc lại giá trị của thanh ghi TIM2\_CCR1 và in lên Watch.

Tiến hành Build Code và Nạp chương trình xuống KIT.

**Kết quả Demo**



Như các bạn thấy. Khi mình bắt đầu nạp chương trình, thanh ghi CNT bắt đầu đếm từ 0 sau đó tăng dần, khi mình bấm nút. Một ngắt Input Capture xảy ra và nó ngay lập tức bắt lấy giá trị của thanh ghi CNT và ghi vào TIM2\_CCR1. Kết quả như trên hình.

Các bạn có thể ứng dụng chức năng này để tạo ra các mạch đếm độ rộng xung.

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

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