TIMER\_PWM(stm32f103c8t6)

1, giới thiệu chung

- Timer là một trong những ngoại vi của stm32. Timer có thể chia ra thành các loại Base TImer(timer chỉ có chức năng đếm (up counting)) Tim2 ,Genera Timer(các Timer có chức năng mục đích chung) Advangte Timer

- Các chức năng chung là :

+ Đếm (Timer Base)

+ Đếm/Tính toán tần số/ chu kỳ tín hiệu vào (Input Capture)

+ tạo ra các loại sóng vuông khác nhau (Output Compare)

+ …

- phần mềm lập trình : stm32cubemx & keilc v5/ stm32cubeide

1 Timer Base

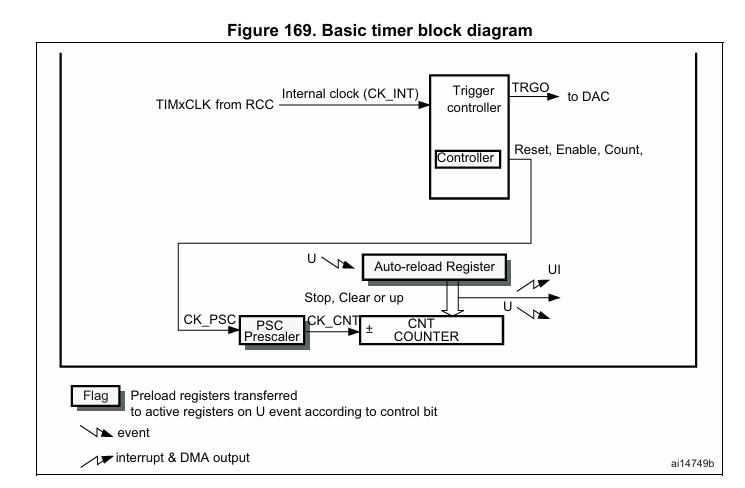
- Các khối chung của basic timer

+ TIM\_CLK (Timer Clock) : bộ tạo dao động cho timer

+ TIMx\_CNT (counter register) : thanh ghi đếm

+ TIMx\_ARR (auto-reload register) : thanh ghi tự động tải

+ TIMx\_PSC (prescaler Register) : thanh ghi chia tần số

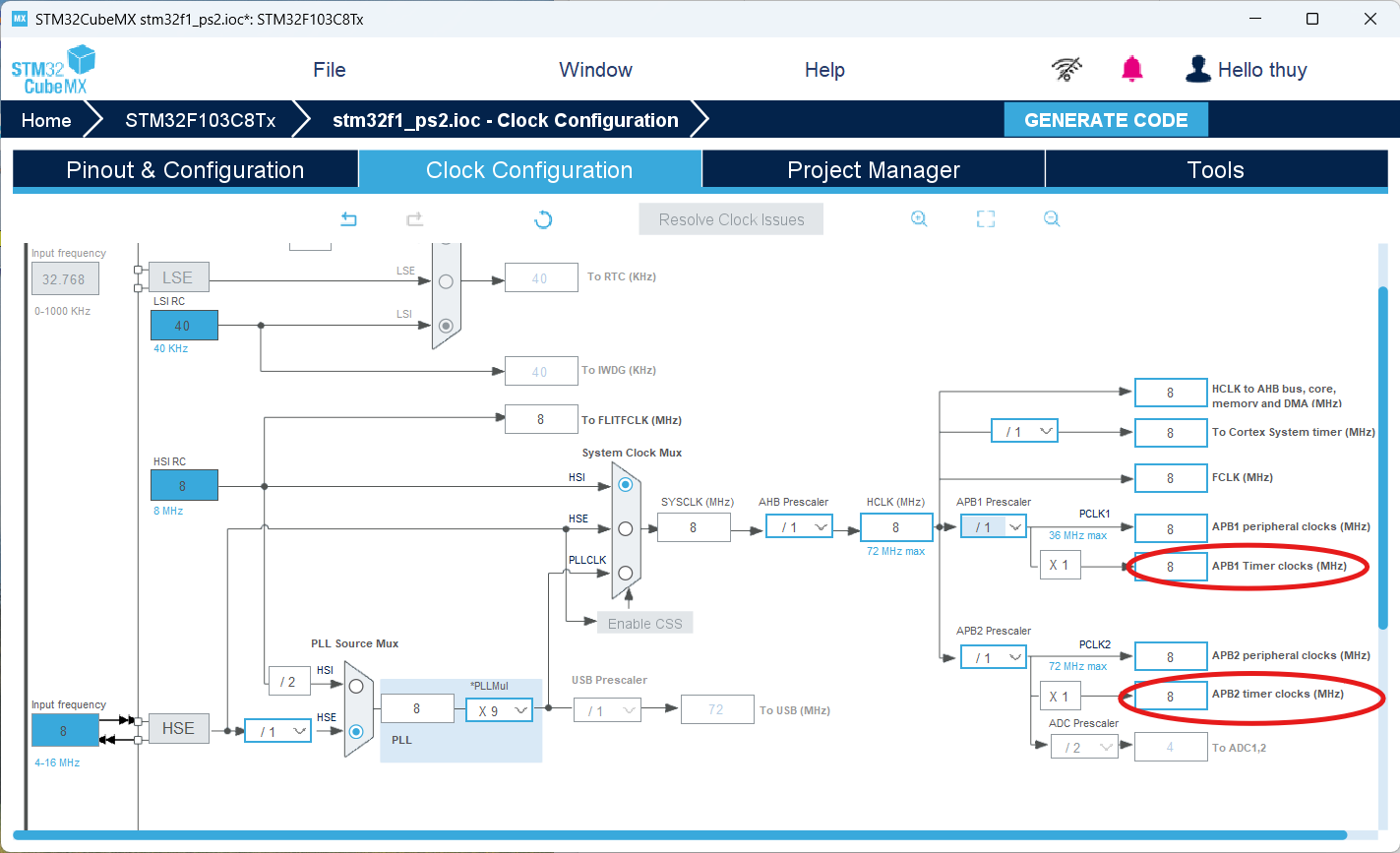


bộ tạo dao động cho timer:

+ tùy thuộc vào sự liên kết bus (APB1 / APB2) với các Timer chia ra làm 2 loại xung clock cho timer

APB1: TIM2 TIM3 TIM4

APB2: TIM1

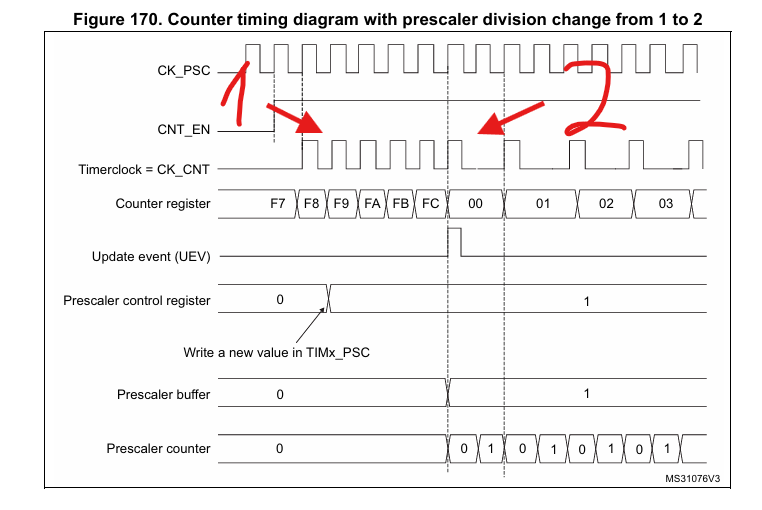


Những yếu tố ảnh hướng đến tần số đếm(F CNT\_CLK)

- (F CNT\_CLK) : tần số đếm : số lượng xung được xuất ra trọng 1 s

+ Giá trị ghi vào thanh ghi TIMx\_PSC (prescaler Register- 16bit) : thanh ghi chia tần số Giá trị được ghi vào thanh ghi PSC ảnh hưởng đến tần số đếm (CK\_CNT)

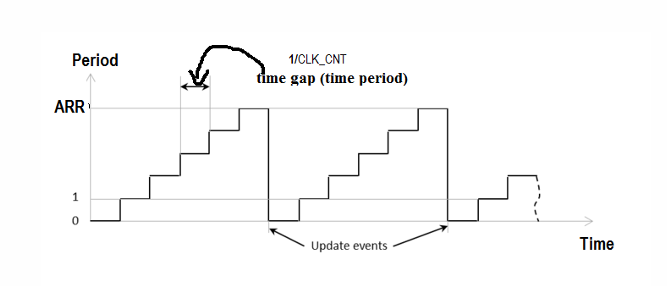
(F CK\_CNT) = (F TIM\_CLK) / (PSC + 1)



1. : với 1 xung TIM\_CLK ta thu được 1 xung CNT\_CLK
2. : với 2 xung TIM\_CLK ta thu được 1 xung CNT\_CLK

=> tương tự với các mức thay đổi với giá trị nạp vào PSC register 3,4,5….n

Chu kỳ đếm (ARR value)



Update\_event = TIM\_CLK/ ((PSC + 1) \* (ARR + 1))

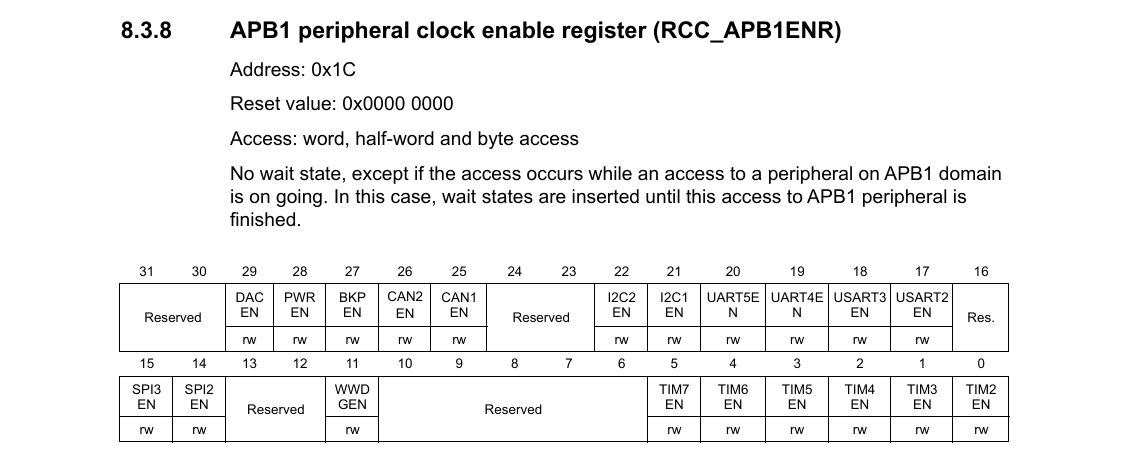
Cách hoạt động Timer Base (đối với trường hợp đếm lên)

Khi Timer được kích hoạt (CEN = 1 <TIMx\_CR1>) giá trị trong thanh ghi CNT được set = 0 và đếm lên theo tần số đếm CNT\_CLK cho đến khi đạt đến giá trị trong ghi ARR -> CNT set về 0 , (Ngắt update events có thể sảy ra ngắt nếu UIE = 1<TIMx\_DIER>)

Cấu hình thanh ghi Timer Base (TIM2) với chế độ đếm lên

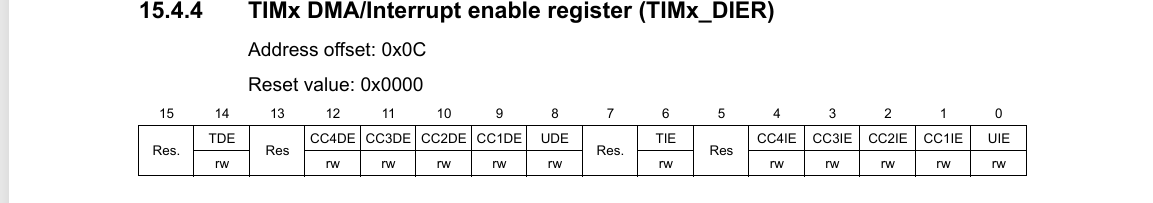
\* Đối với khối RCC

+ cấp clk cho TIM2 : TIM2EN = 1

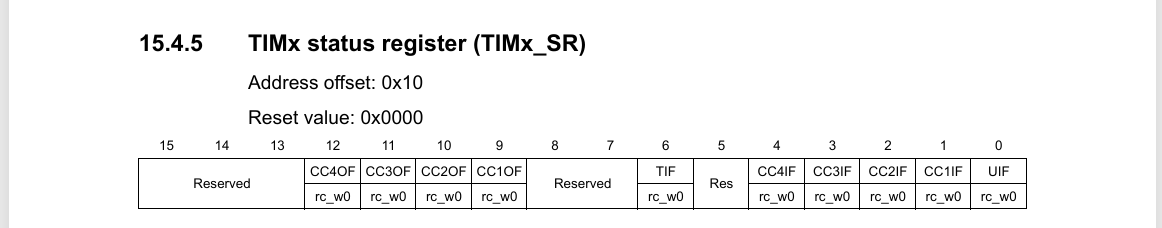


\* Đối với khối TIM

+ Set bit UIE : nếu muốn tạo ra ngắt



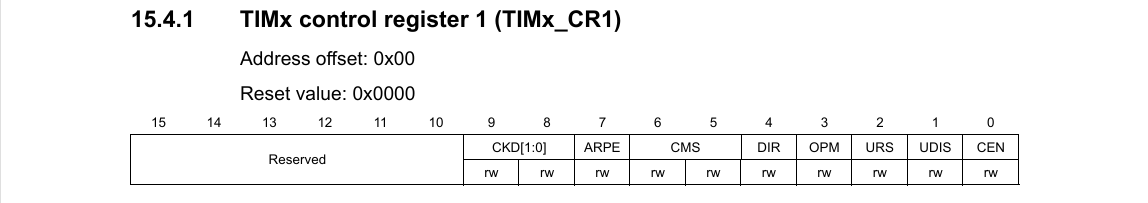
+ Khi sảy ra ngắt : trong hàm ngắt Clear UIF



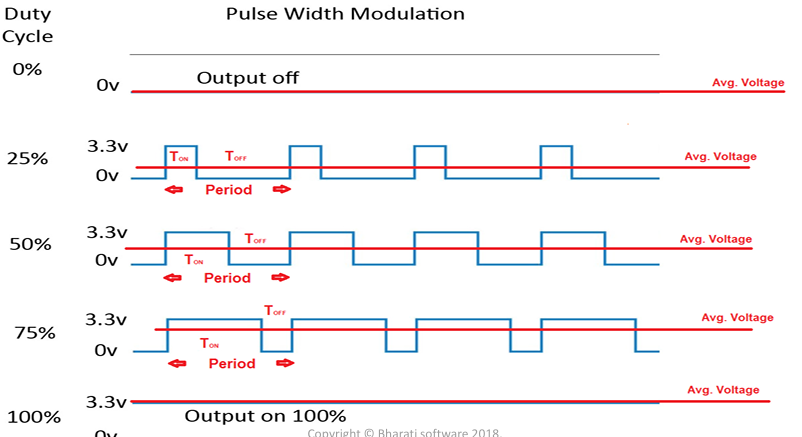
+ Nạp vào giá trị cho thanh ghi PSC và ARR

+ DIR = 0

+ set CEN (sau cùng)



PWM (pulse width modulation) : điều chế độ rộng xung



- Chu kỳ của 1 xung(Ton + Toff) phụ thuộc vào tần số đếm của Timer và giá trị ghi được ghi vào thanh ghi ARR

Tpwm = (PSC + 1) \* (ARR+1)/f(Timer)

- Duty cycles : là giá trị biểu diễn bằng tỉ số % Ton/Toff

- Ảnh hưởng của giá trị ghi vào TIMx\_CCRx(16bit) và giá trị bit CCxP(TIMx\_CCER) tới xung PWM

+ Với CCxP = 0: OC active high

Khi giá trị đếm trong thanh ghi CNT < giá trị nạp vào CCRx thì TIMx\_Channel\_y xuất ra mức cao

Khi giá trị đếm trong thanh ghi CNT > giá trị nạp vào CCRx thì TIMx\_Channel\_y xuất ra mức thấp

Ton = (PSC + 1) \* (CCRy+1)/f(Timer)

Toff = Tpwm - Ton

+ với CCxP = 1: OC active low

Hoàn toàn ngược lại với trường hợp trên

Toff = (PSC + 1) \* (CCRy+1)/f(Timer)

Ton = Tpwm - Ton

Cấu hình thanh ghi với TIM2 Channel 1

\* Cấu hình TIMER base chế độ đếm lên và không set bit CEN(bên trên)

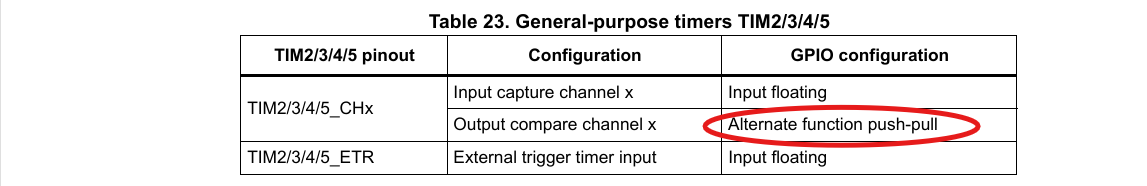
\* Cấu hình TIMER Channel

\* đối với khối RCC

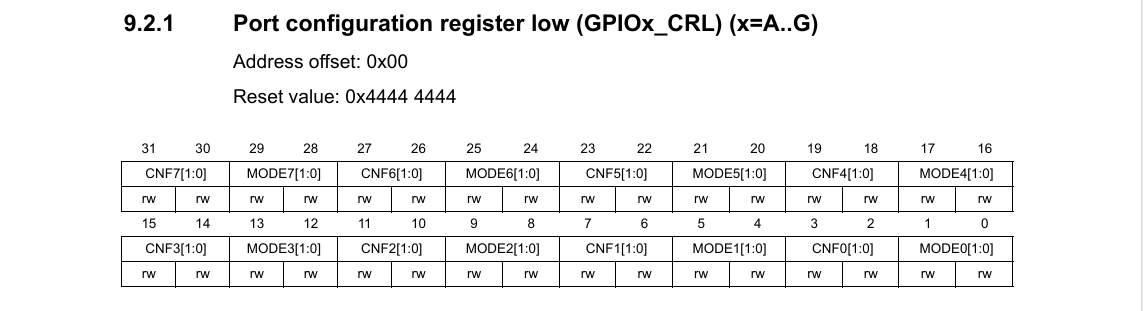
Set GPIOAEN = 1

GPIOx

\* đối với khối GPIO



Cấu hình PA0 chế độ alternate function push - pull

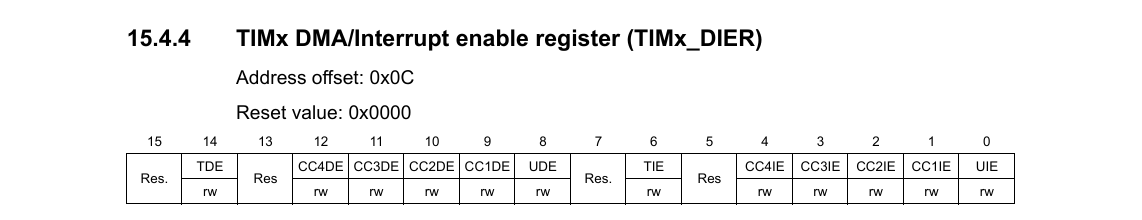


CNF0 = 0x2 (alternative function push pull)

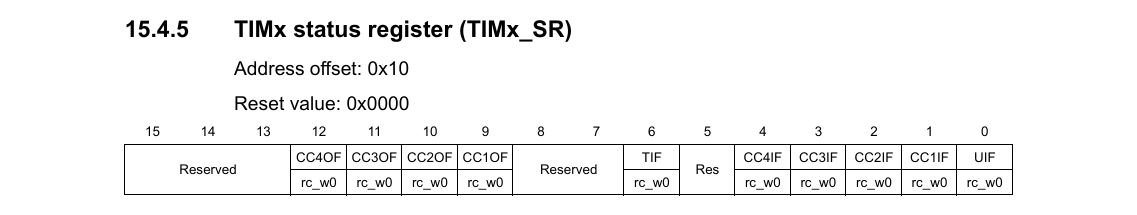
MODE0 = 0x3 (output mode 50MHz)

\*Cấu hình TIM\_OC

Cấu hình ngắt nếu cần CC1IE = 1

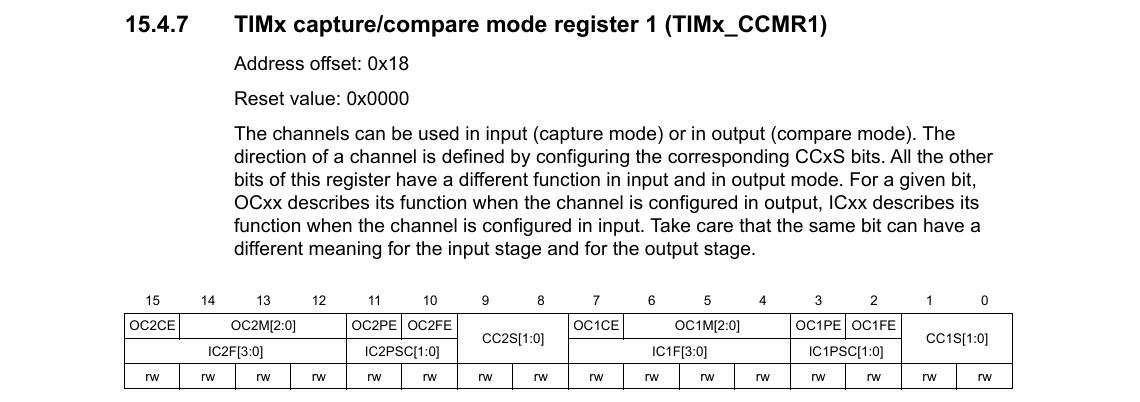


Nếu sử dụng ngắt CC1IF



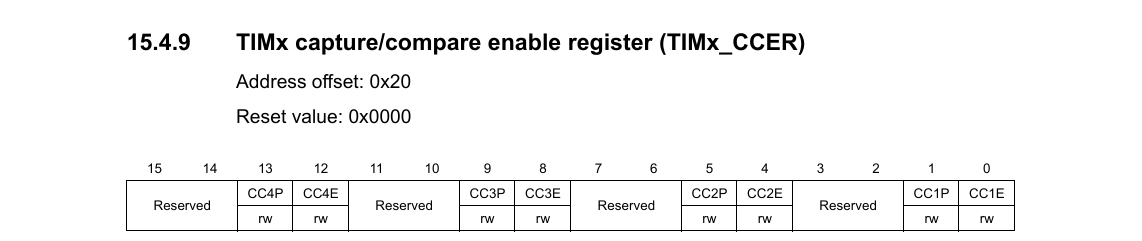
Cc1s = 0x0 -> CH1 output

OC1M = 0x6 -> TIMx\_PWM mode 1



CCP1P = 0 -> OC1 active high.

CC1E = 1 -> OC1 active

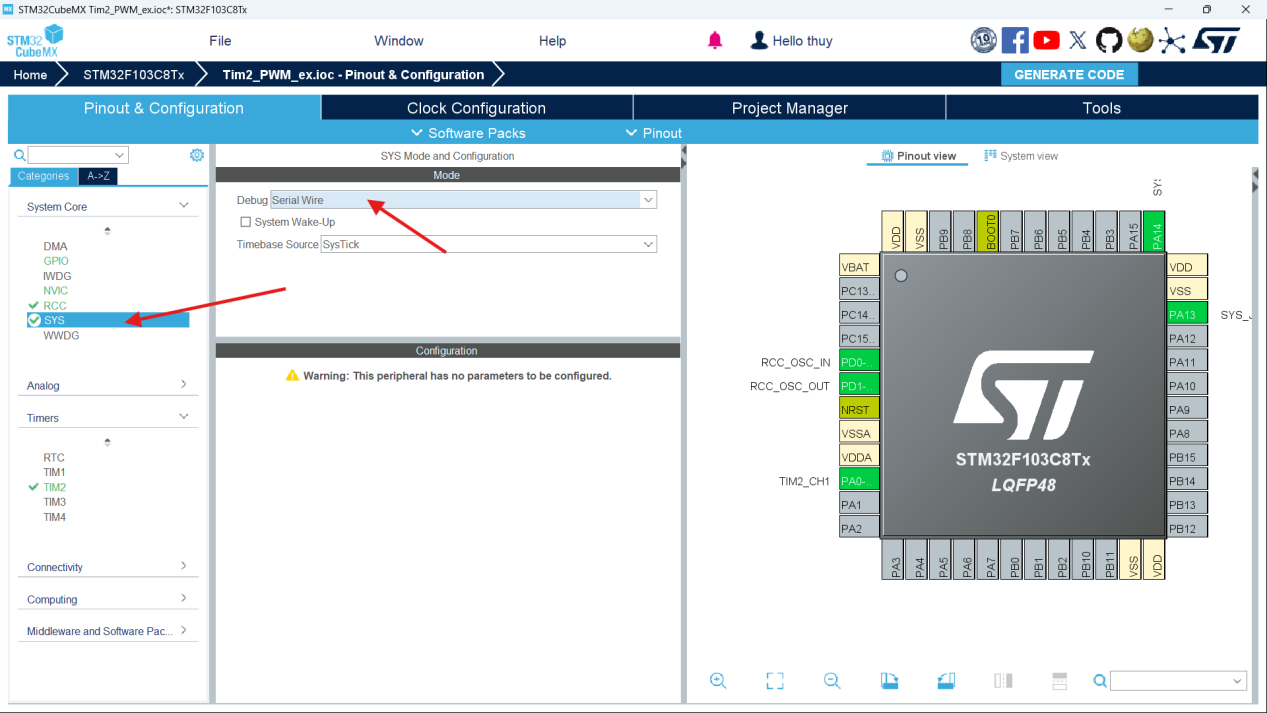


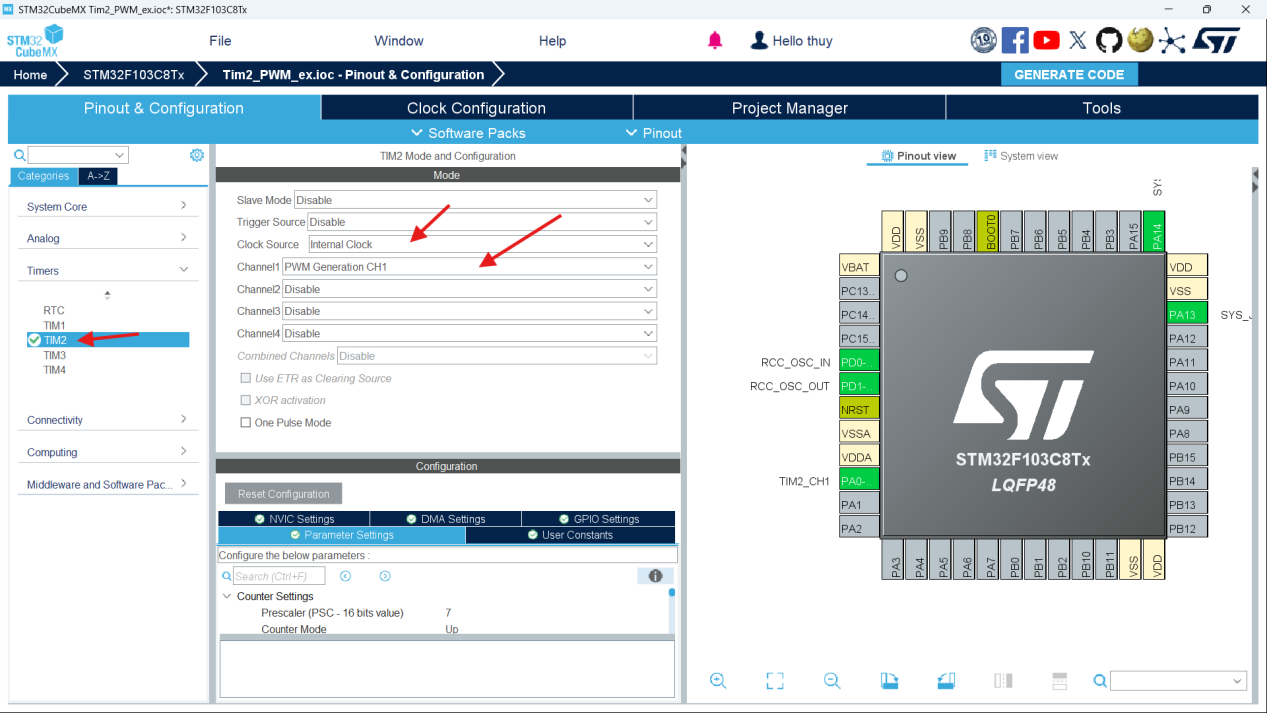
Set giá trị vào thanh ghi TIMx\_CCR1

TIMx\_CR1 : CEN = 1

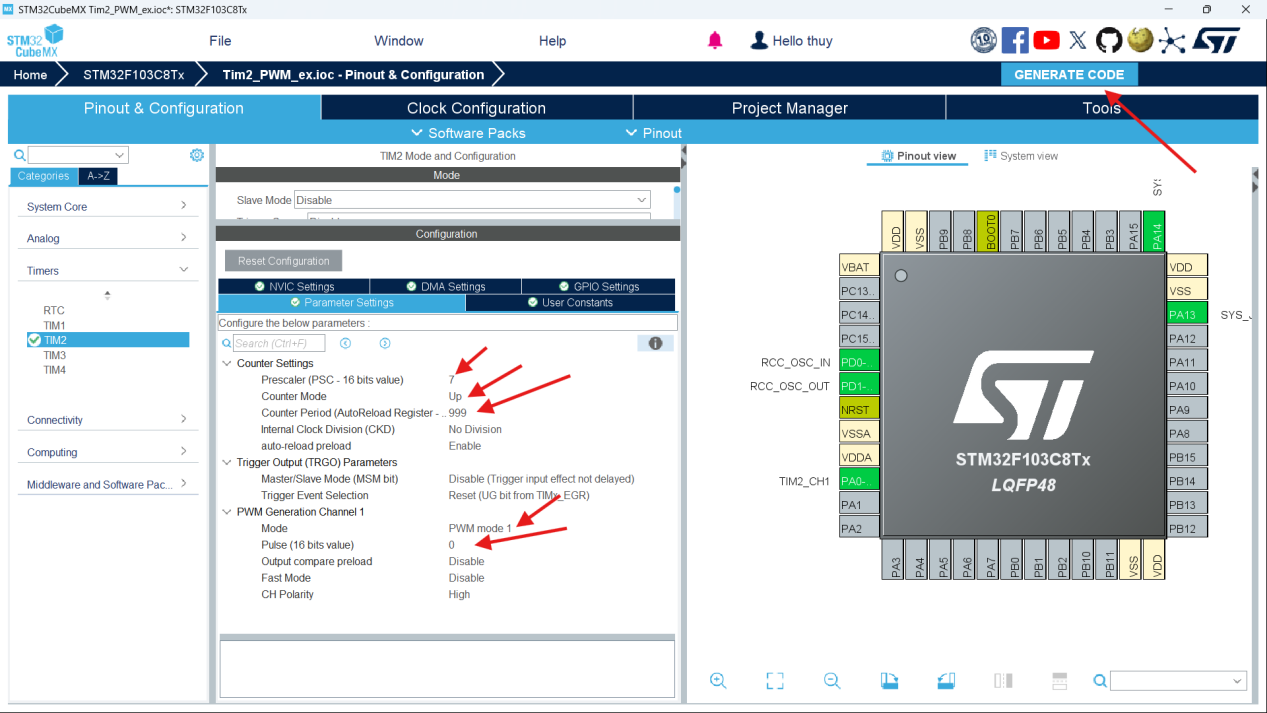
Cấu hình bằng phần mềm : Cấu hình xung PWM 1khz cho TIM2 Channel1

+ cấu hình trên mx

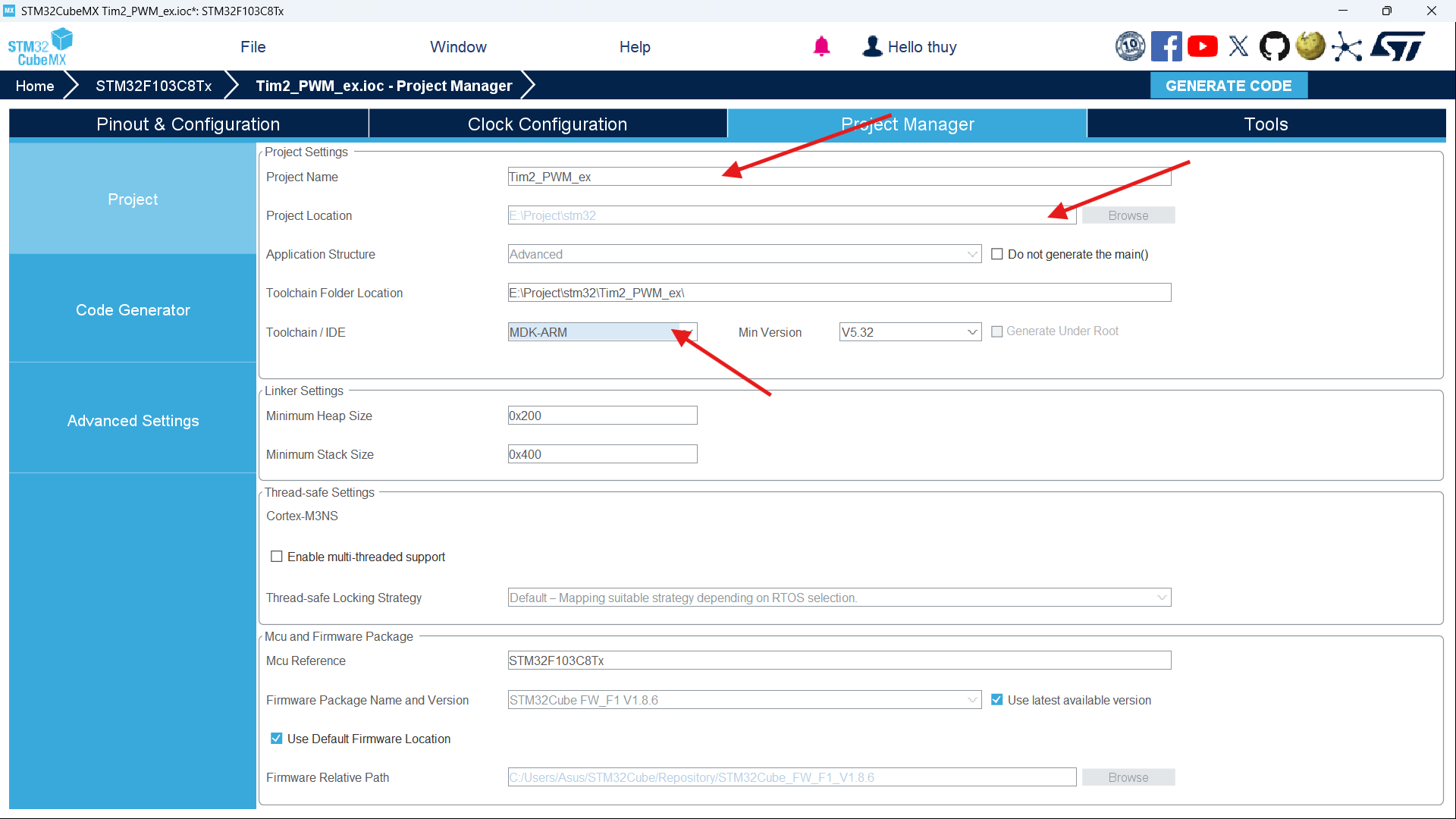




Cấu hình xong ấn generate code



Đặt tên file vs chọn path lưu file toolchain ide chọn ide phù hợp (keic -> mdk-arm)

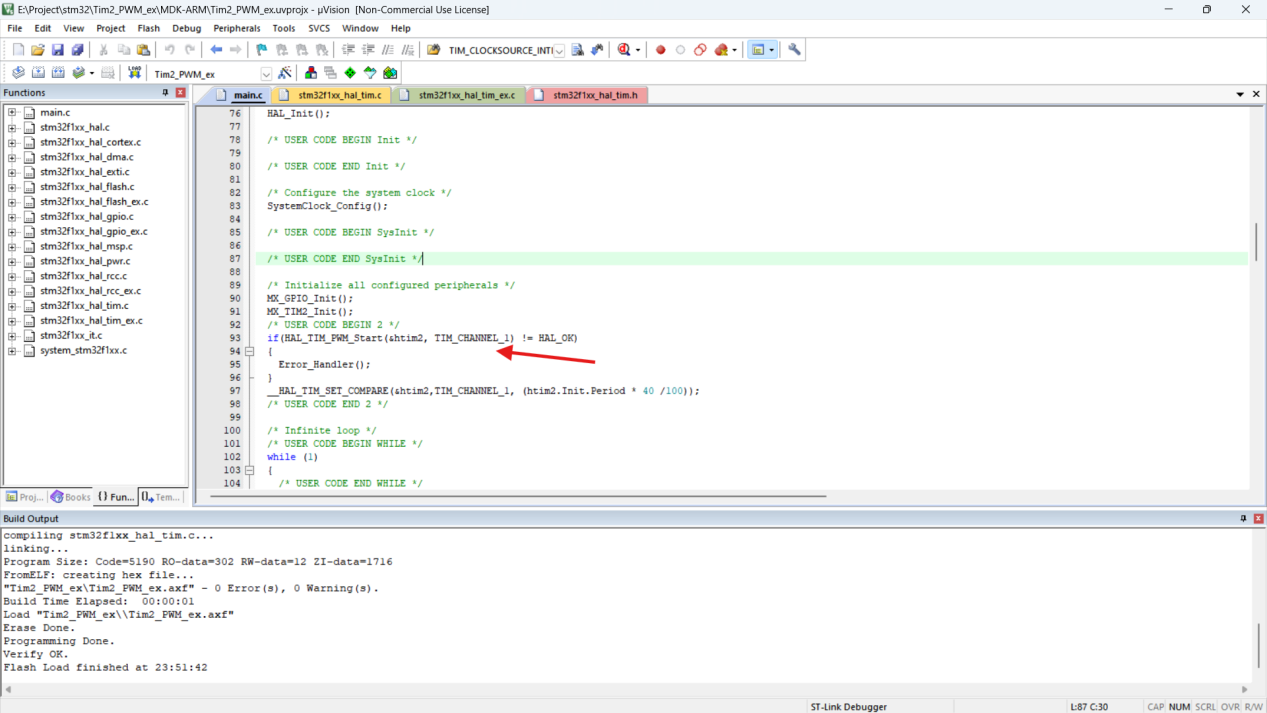


Cấu hình trong keilc

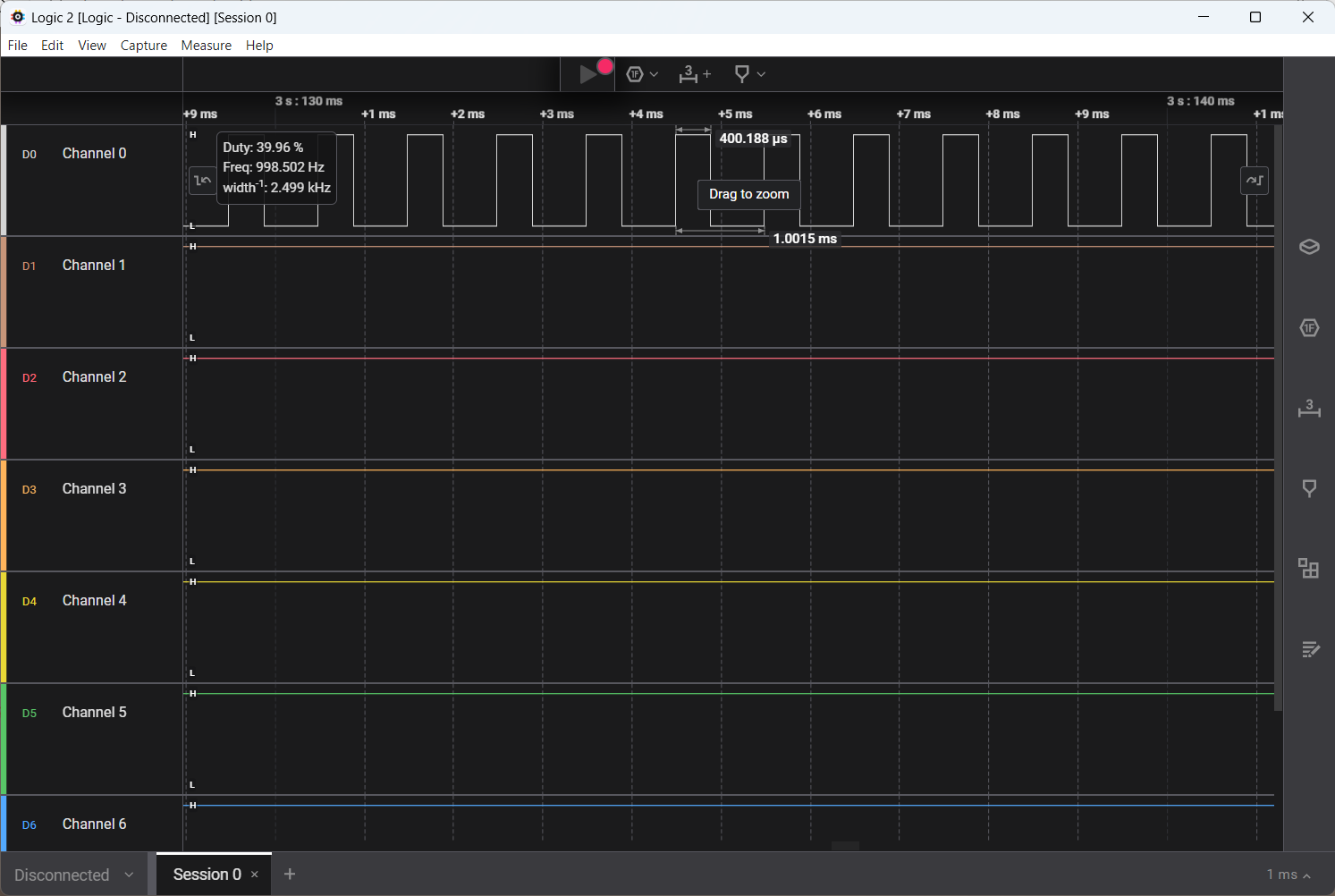
Trong hàm mai gọi hàm HAL\_TIM\_PWM\_Start()

- có thể thay đổi tần số chu kỳ/tần số của xung pwm bằng hàm \_\_HAL\_TIM\_SET\_COUNTER()

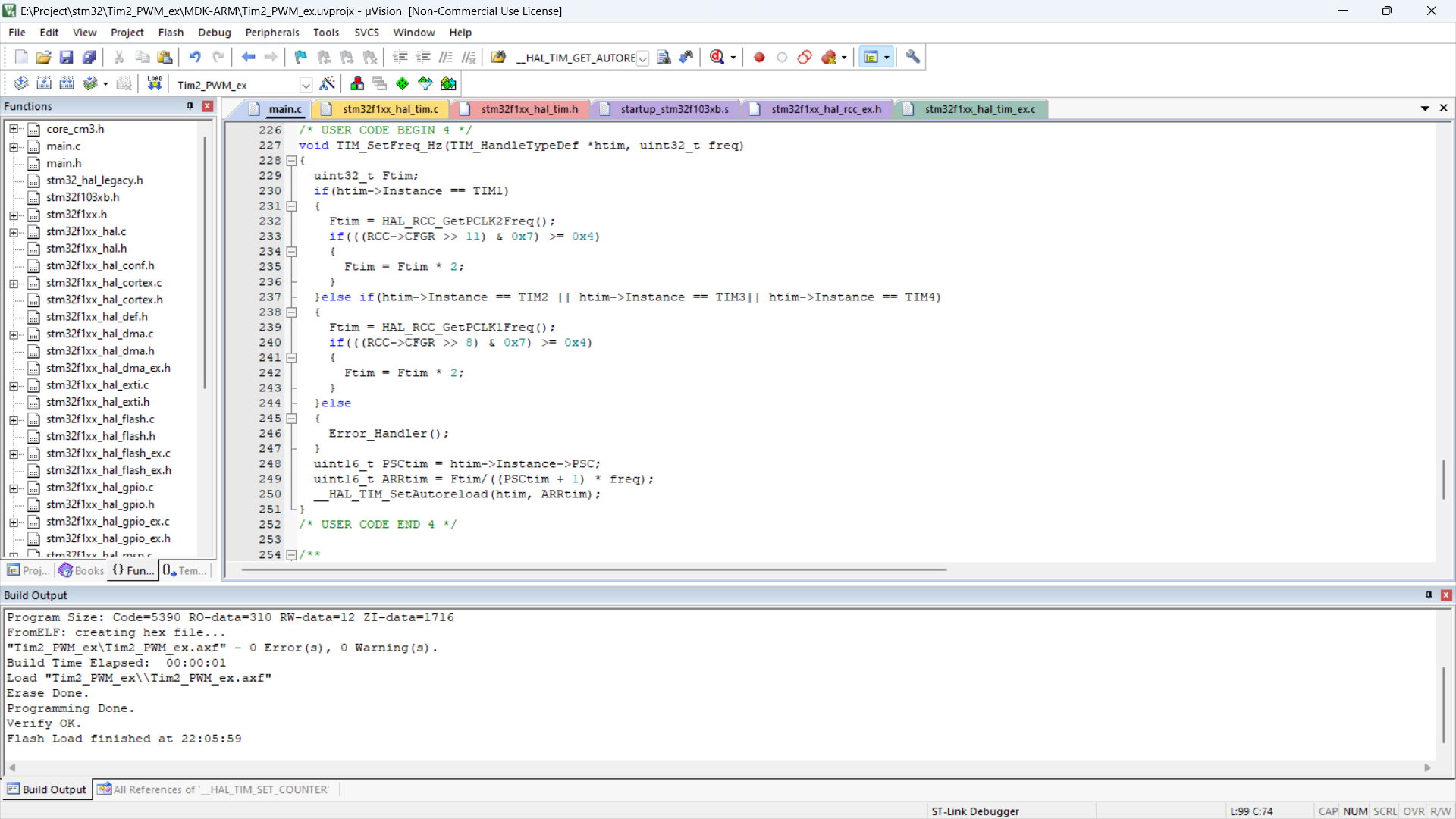
- có thể thay đổi Duty cycles băng hàm \_\_HAL\_TIM\_SET\_COMPARE()

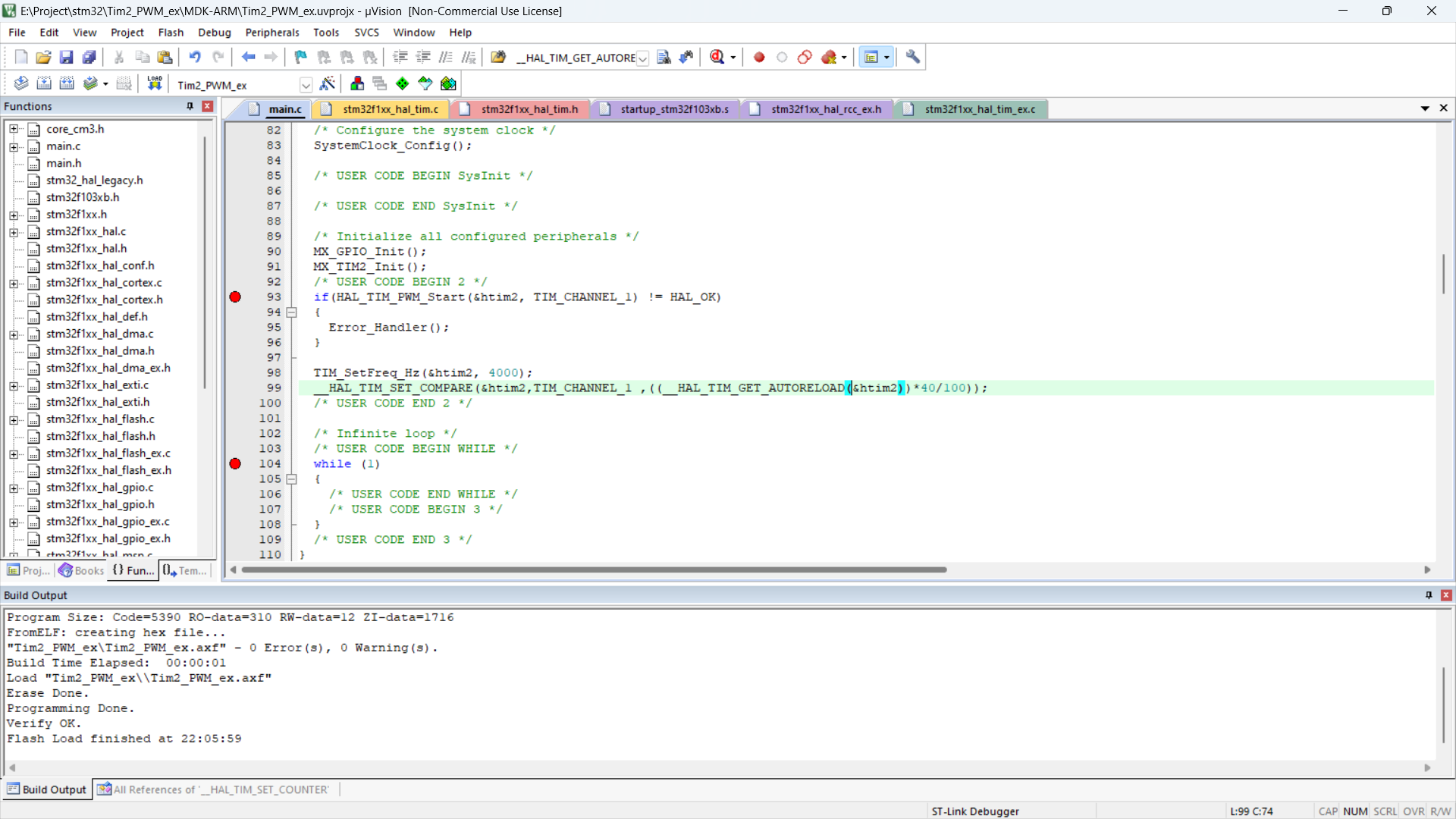


Kết quả 1 thu được tại chân TIM2\_CH1(PA0) với duty cycles = 40% (cấu hình mặc định)



Tạo hàm tạo xung clock PWM





Kết quả thu được ở chân PWM với tần số 4000hz 40% duty cycles

