

## Chương 4: PWM

```
using System;
using System.Data;

namespace MyProject
{
    /// <summary>
    /// Customer Master
    /// </summary>
    /// <remarks></remarks>
    public class Customer
    {
        private string pCustomerCode;
        private string pCustomerName;
        private string pAddress;
        private string pMobileNo;
        private string pEmail;

        public string CustomerCode
        {
            get { return pCustomerCode; }
            set { pCustomerCode = value; }
        }

        public string CustomerName
        {
            get { return pCustomerName; }
            set { pCustomerName = value; }
        }

        public string Add
        {
            get { }
        }
    }
}
```

# PWM là gì?

PWM hay Pulse Width Modulation là phương pháp điều chỉnh độ rộng của xung có chu kỳ cố định, nhằm tạo ra sự thay đổi điện áp tại đầu ra.

Điện áp logic chỉ có 2 trạng thái 0 và 1 tương ứng với 0V và Vcc (5V hoặc 3.3V).

Trường hợp muốn xuất ra mức điện áp khác 2 mức này có 2 cách:

- Dùng bộ DAC (Digital Analog Convert)
- Dùng PWM

Tuy nhiên các dòng vi điều khiển giá rẻ thường không có bộ DAC, nên thông dụng nhất là sử dụng PWM.

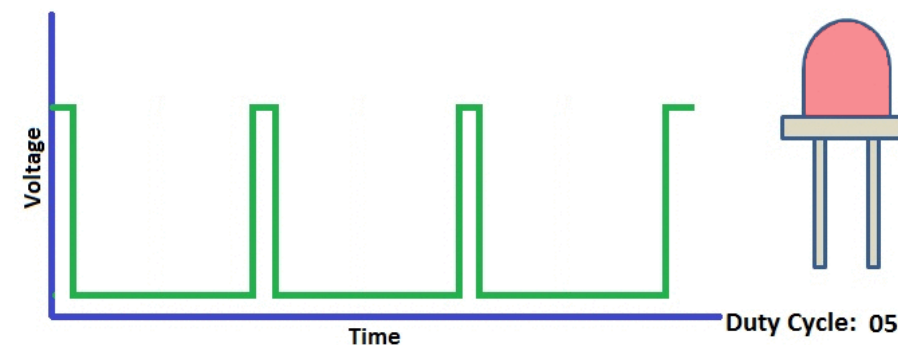
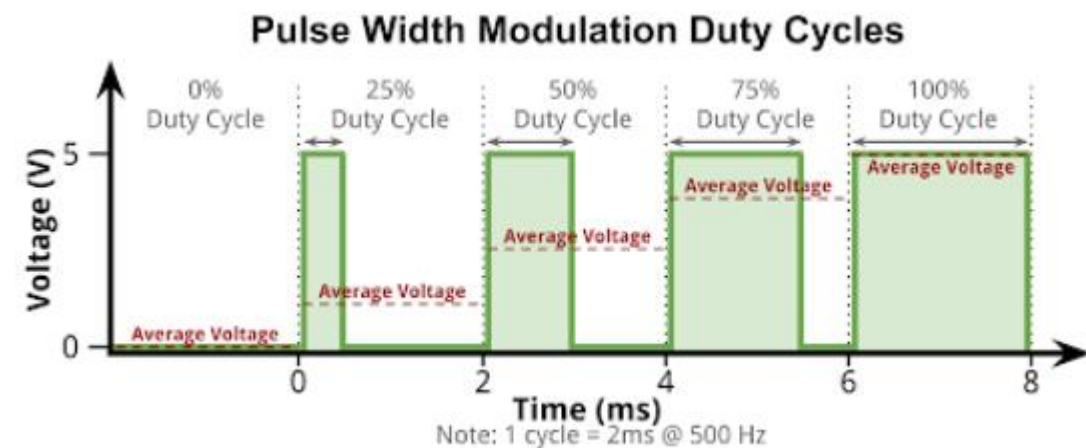
PWM sẽ tăng giảm điện áp hiệu dụng tùy vào Duty Cycle (% thời gian xung dương/thời gian toàn bộ chu kỳ)

Công thức:

$$V \text{ hiệu dụng} = \text{Duty} * V_{cc}$$

Ví dụ: băm xung có điện áp Vcc là 3.3V với Duty là 50% thì

$$V \text{ hiệu dụng} = 50\% * 3.3 = 1.65 \text{ V}$$



# Ứng dụng của PWM



PWM thường được sử dụng trong việc điều khiển tốc độ động cơ, điều khiển độ sáng led, điều khiển các loại động cơ đặc biệt như servo.



# Các tham số quan trọng của PWM

## ➤ Chu kỳ của xung PWM

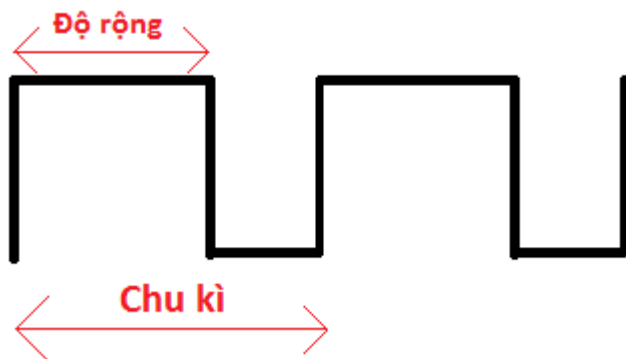
Là thời gian hoàn thiện 1 chu kỳ (bao gồm cả mức 1 và 0)

VD:  $T = 1\text{ms}$

## ➤ Độ rộng xung PWM

Là thời gian xung ở mức cao (mức 1) trên tổng thời gian.

VD: Duty = 50%



Khi điều khiển tăng giảm độ rộng xung, cần 1 tham số nữa là độ mịn PWM

## ➤ Độ mịn của PWM:

Là số lần Duty từ 0% tăng lên 100%

VD: Bấm xung với độ mịn 100 thì từ 0% lên 100% mất 100 lần.

Nhưng độ mịn 1000 thì từ 0% lên 100% mất 1000 lần

## ➤ Tốc độ tăng giảm

Là thời gian mỗi lần tăng hoặc giảm 1 nhịp PWM

Trên STM32 có thể PWM bằng 2 cách

- PWM mềm (dùng timer và chân GPIO)
- PWM cứng (dùng chân xuất PWM của STM32)

PWM mềm

- Ưu điểm: Không hạn chế chân sử dụng và timer sử dụng.
- Nhược điểm: MCU phải xử lý, có thể dẫn tới điều khiển không đúng do MCU có thể có các tác vụ khác đè lên

PWM cứng:

- Ưu điểm: Do bộ Timer xử lý, không liên quan tới MCU
- Nhược điểm: Bị hạn chế các chân và bộ Timer

STM32F1 có 3 Timer có thể dùng để xuất xung PWM gồm TIM2, TIM3, TIM4, trong đó mỗi kênh còn gồm 4 Channel nhỏ

Timer	Channel	Pin
TIM2	Channel 1	PA0
	Channel 2	PA1
	Channel 3	PA2
	Channel 4	PA3
TIM3	Channel 1	PA6
	Channel 2	PA7
	Channel 3	PA8
	Channel 4	PB1
TIM4	Channel 1	PB6
	Channel 2	PB7
	Channel 3	PB8
	Channel 4	PB9



# Thiết lập PWM trên STM32

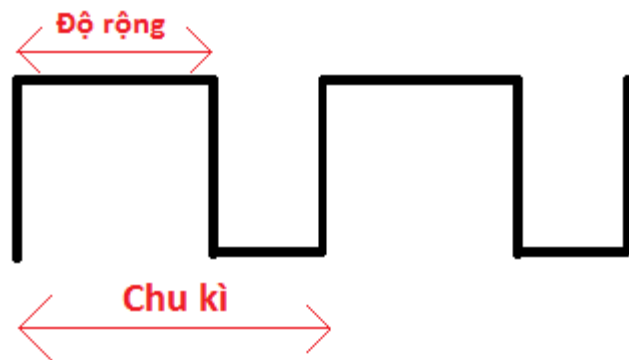


KHUÊ NGUYỄN CREATOR  
Chia sẻ đam mê

Chu kỳ PWM:

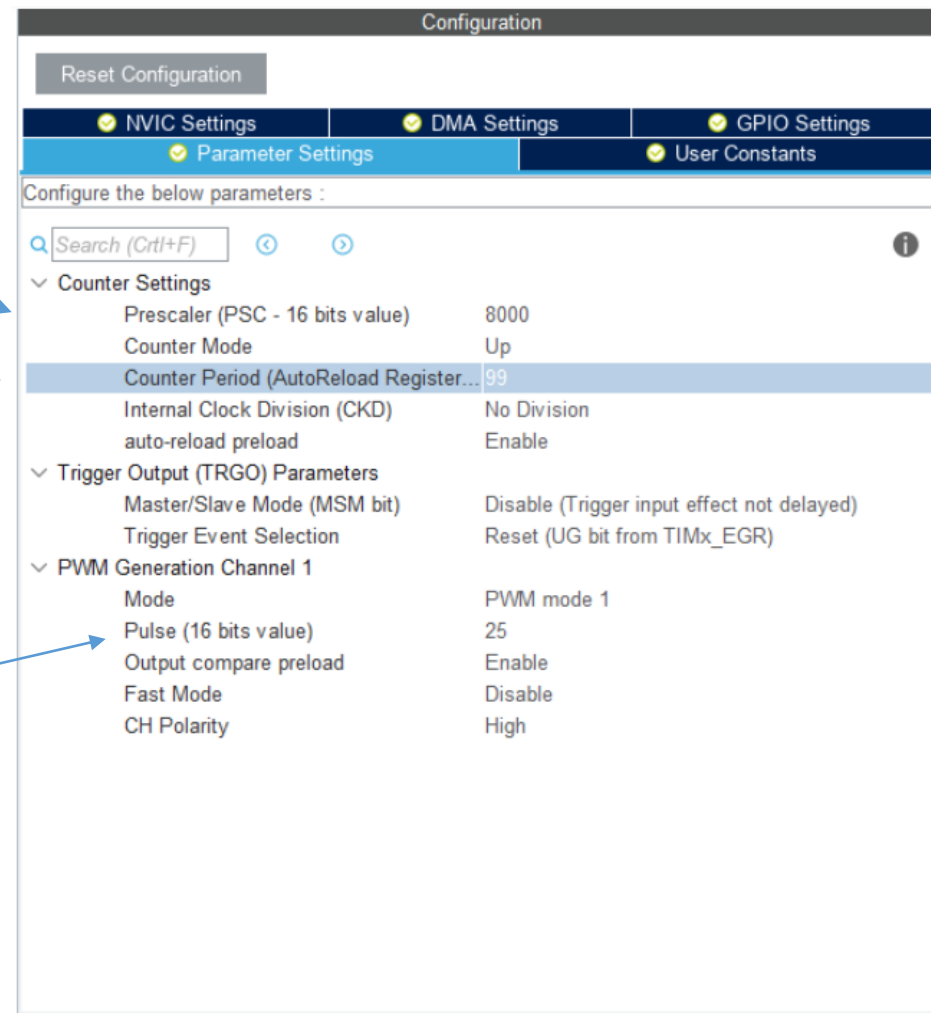
Thiết lập bằng 2 thanh ghi:

Prescaler và Period



Độ rộng xung PWM:

Thiết lập bằng thanh ghi Pulse



Độ mịn PWM của STM32 là 16 bit tương ứng với  $2^{16}$  lần

# Thực hành điều khiển điện áp



**KHUÊ NGUYỄN CREATOR**  
Chia sẻ đam mê

Bài 1: Thiết lập PWM có chu kỳ  $T = 1\text{ms}$   
Độ rộng xung là 50, độ mịn là 100.

Sau đó dùng đồng hồ đo mức điện áp trên đó

Bài 2: Thiết lập PWM có chu kỳ  $T = 1\text{ms}$ ,  
độ mịn là 100, độ rộng xung tăng dần từ 0 -  
100%, sau đó giảm dần từ 100 – 0% với tốc  
độ 5ms mỗi lần tang/giảm.

Mắc led vào chân PWM và xem kết quả.



```
88
89  /* Initialize all configured peripherals */
90  MX_GPIO_Init();
91  MX_TIM2_Init();
92  /* USER CODE BEGIN 2 */
93  HAL_TIM_PWM_Start(&htim2,TIM_CHANNEL_1);
94  /* USER CODE END 2 */
95
96  /* Infinite loop */
97  /* USER CODE BEGIN WHILE */
98  while (1)
99  {
100     /* USER CODE END WHILE */
101
102     /* USER CODE BEGIN 3 */
103     __HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_1, 50);
104     }
105     /* USER CODE END 3 */
106 }
107
```

HAL\_TIM\_PWM\_Start

Khởi động Timer chế độ PWM

\_\_HAL\_TIM\_SetCompare

Thay đổi độ rộng của xung



# Thực hành điều khiển Servor SG90

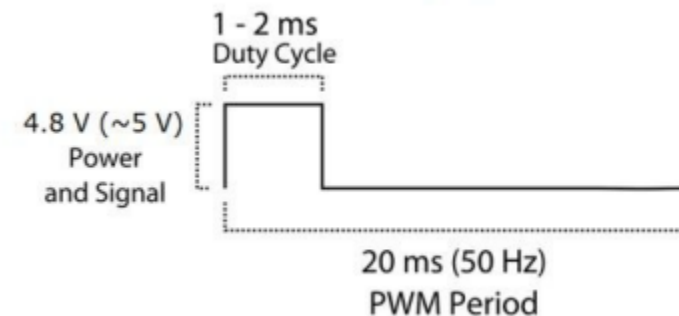
Servor SG90 Datasheet:

[http://www.ee.ic.ac.uk/pcheung/teaching/D  
E1\\_EE/stores/sg90\\_datasheet.pdf](http://www.ee.ic.ac.uk/pcheung/teaching/D<br/>E1_EE/stores/sg90_datasheet.pdf)

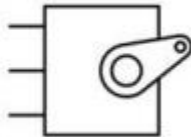
Cách điều khiển:

$T_{pwm} = 20ms$

Duty = 1ms (0%)  $\Rightarrow$  2m (100%)



PWM=Orange (PWM)  
Vcc = Red (+)  
Ground=Brown (-)



# Bài tập về nhà



**KHUÊ NGUYỄN CREATOR**  
Chia sẻ đam mê

Dùng nút nhấn để điều khiển động cơ servo. Nhấn nút để đảo chiều quay của servor.

# Tạo thư viện SG90



**KHUÊ NGUYỄN CREATOR**  
Chia sẻ đam mê

Tạo thư viện cho SG90