



ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh

Môn học

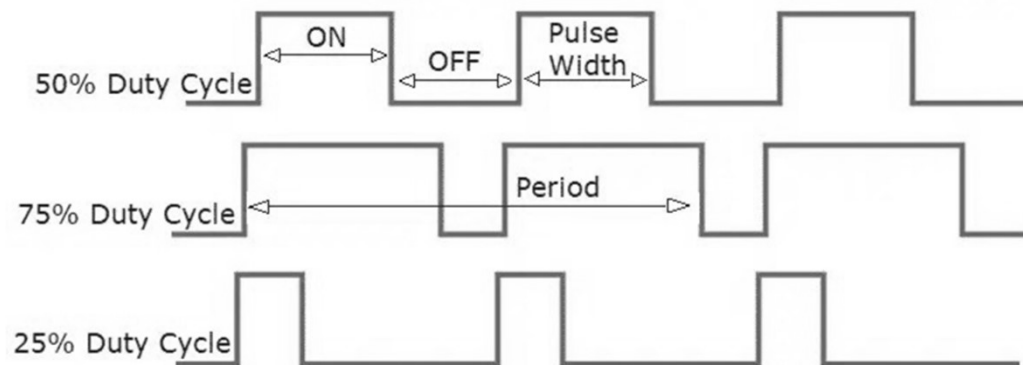
Thực tập Vi Điều Khiển²

Tuần 9

Pulse With Modulation (PWM)

PWM là một phương pháp điều chế độ rộng xung bằng cách thay đổi thời gian bật/tắt tín hiệu.

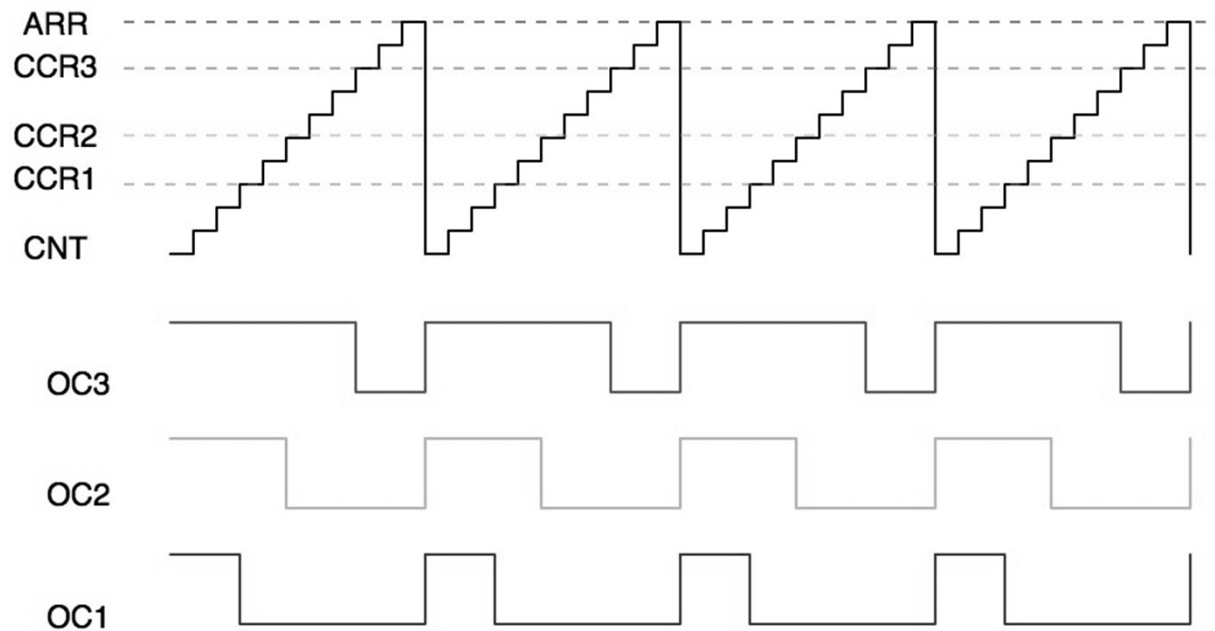
- ❖ Xung PWM thường được sử dụng trong các mạch analog được tạo bởi ngõ ra digital từ vi điều khiển khi mà việc chuyển đổi Digital-Analog không cần thiết. Nhiều từ hệ thống được tối giản nhất vì tín hiệu vẫn là tín hiệu số.
- ❖ Các tín hiệu như thế này được sử dụng rất nhiều trong các ứng dụng điều khiển công suất như điều chỉnh độ sáng và điều khiển tốc độ động cơ. Độ sáng của màn hình máy tính.



Pulse With Modulation (PWM)

Quá trình tạo xung PWM bằng bộ đếm của Timer trong STM32

Three PWM signals from the Output Compare Channels of a general purpose timer



Các bước thiết lập chế độ PWM

Step 1: Lựa chọn kênh PWM: CH1, CH2, CH3, CH4

Step 2. Thiết lập Output AF mode cho chân đã lựa chọn

Step 3. Cấp xung clock cho bộ Timer sử dụng

Step 4: Thiết lập bộ chia (PSC) và giá trị đặt trước (ARR) cho timer. (hai thông số này quyết định chu kỳ của xung PWM)

Step 5: Khai báo thông số độ rộng xung PWM

Step 6: Khai báo kênh PWM cần sử dụng trong các kênh CH1, CH2, CH3, CH4

❖ Kênh CH1, CH2 ở thanh ghi CCMR1 : cụm bit: OC1M hoặc OC2M

❖ Kênh CH3, CH4 ở thanh ghi CCMR2 : cụm bit: OC3M hoặc OC4M

Step 7: Kích hoạt kênh PWM

Step 8: Kích hoạt Timer tương ứng

Step 1. Lựa chọn channel và Timer

Lựa chọn timer và channel tương ứng để phát xung PWM.

❖ F1 Series

Timer	Timer 1	Timer 2	Timer 3	Timer 4	Timer 5
CH1	PA8 (PE9)	PA0 (PA15)	PA6 (PB4, PC6)	PB6 (PD12)	
CH2	PA9 (PE11)	PA1 (PB3)	PA7 (PB5, PC7)	PB7 (PD13)	
CH3	PA10 (PE13)	PA2 (PB10)	PB0 (PC8)	PB8 (PD14)	
CH4	PA11 (PE14)	PA3 (PB11)	PB1 (PC9)	PB9 (PD15)	

Xác định chân cần phát xung PWM tương ứng với channel đã lựa chọn. Các chân mặc định màu đỏ.

Step 2. Thiết lập Output AF mode cho chân đã lựa chọn

Sau khi đã lựa chọn được channel. Tiến hành cấu hình output cho chân tương ứng với chế độ Output Alternate Function mode

❖ F1 Series

❖ *Cấp xung clock cho GPIOx*

❖ *Thiết lập output AF mode – bằng các thanh ghi GPIOx_CLR hoặc GPIOx_CHR*

❖ F2,3,4 Series

❖ *Cấp xung clock cho GPIOx*

❖ *Thiết lập output AF mode – bằng các thanh ghi GPIOx_MODER, GPIOx_OTYER, GPIOx_OSPEEDR*

❖ *Thiết lập AF mode cho Timer tương ứng bằng thành ghi GPIOx_AFRL hoặc GPIOx_AFRH*

Vị trí AF của timer tra hình ở Figure 26, mục 8.3.2 thuộc STM32F4 Series Manual Reference

Step 3. Cấp xung clock cho bộ timer sử dụng

Được quản lý bởi module RCC (Reset and Clock Control). Để Timer hoạt động được thì cần cấp xung cho Timer tương ứng tại các thanh ghi quản lý bus APB1ENR hoặc APB2ENR

❖ F1 Series : RCC_APB1ENR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC EN	PWR EN	BKP EN	Res.	CAN EN	Res.	USB EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART3 EN	USART2 EN	Res.
		rw	rw	rw				rw		rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	876543210								
SPI3 EN	SPI2 EN	Reserved		WWD GEN	Reserved		TIM14 EN	TIM13 EN	TIM12 EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN
rw	rw						rw		rw	rw	rw	rw	rw	rw	rw

Ví dụ: **RCC** -> **APB1ENR** |= **(1<<0)** ; // Clock for Timer 2

Step 4. Thiết lập bộ tần số xung PWM

Tần số xung PWM được quyết định bởi 2 giá trị: PSC (Prescaler) và ARR

Ví dụ: Cần Tạo xung PWM có tần số 10KHz --- với Timer Frequency: 16Mhz

$$(\text{Prescaler} + 1)(\text{ARR}) = 16,000,000 * \frac{1}{10,000} = 1,600$$

1. Chọn: PSC = 0 \Rightarrow ARR = 1,600
2. Chọn: PSC = 15 \Rightarrow ARR = 160
3. ...

Ví dụ:

TIM2	->	PSC	=	0	;
TIM2	->	ARR	=	1600	;

Step 5. Thiết lập độ rộng xung PWM

Độ rộng xung PWM sẽ được quyết định bởi giá trị thanh ghi $TIMx_CCRy$ ($y: 1,2,3,4$)

$TIMx_CCRy$: Thanh ghi cho phép đảo trạng thái ngõ ra trong 1 chu kỳ đếm của timer ($0 \leq CCRy \leq ARR$)

Ví dụ:

TIM2

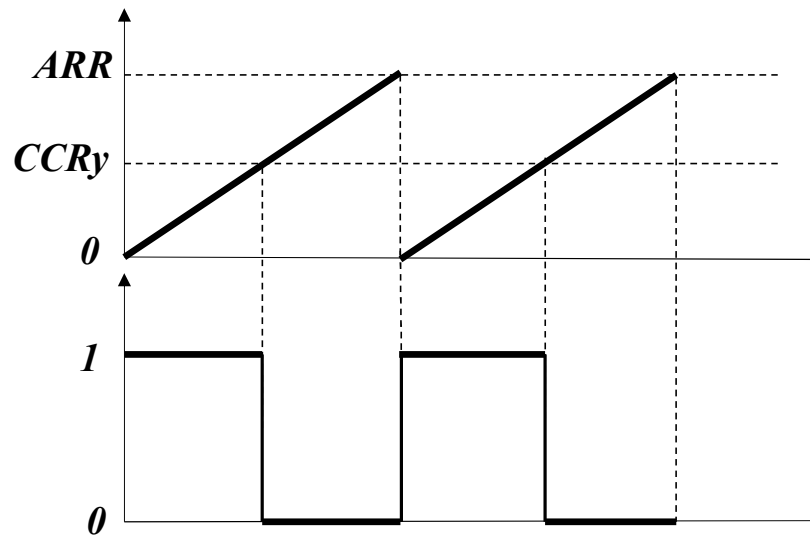
->

CCR1

=

100

;



Step 6. Khai báo kênh PWM của timer

Sau khi thiết lập các thông số, tiến hành khai báo kênh PWM đã lựa chọn cho vi điều khiển bằng 2 thanh ghi: *TIMx_CCMR1* (kênh CH1, CH2) hoặc *TIMx_CCMR2* (kênh CH3, CH4).

TIMx_CCMRx: Thanh ghi cho phép khai báo các chế độ làm việc khác của timer (PWM, Input Capture,...)

TIMx_CCMR1

<i>IMx_CCMR1</i>						00: output				00: output					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
	IC2F[3:0]			IC2PSC[1:0]					IC1F[3:0]			IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

00: output

CC1S/CC2S: Khai báo kênh CH1 hoặc CH2 là output/intput

OC1M/OC2M: Khai báo PWM mode 1 hoặc PWM mode 2

110: PWM mode 1: Channel kích hoạt (mức 1) khi $TIMx \rightarrow CNT < TIMx \rightarrow CCRx$ (in upcounting mode)

111: PWM mode 2: Channel không kích hoạt (mức 0) khi $TIMx \rightarrow CNT < TIMx \rightarrow CCRx$ (in upcounting mode)

Step 7. Kích hoạt kênh PWM của timer

*Sau khi thiết lập các thông số, tiến hành khai kích hoạt kênh PWM của timer bằng thanh ghi: **TIMx_CCER***

TIMx_CCER: Thanh ghi cho phép kích hoạt các kênh hoạt động của timer

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	Reserved		CC3P	CC3E	Reserved		CC2P	CC2E	Reserved		CC1P	CC1E
		rw	rw			rw	rw			rw	rw			rw	rw

CC1E/CC2E/CC3E/CC4E: Kích hoạt các kênh CH1, CH2, CH3, CH4 của Timer

CC1P/CC2P/ CC3P/CC4P: Phân cực trạng thái logic ban đầu ở các kênh CH1,CH2, CH3, CH4

- 0: active high (mức logic 1)
- 1: active low (mức logic 0)

Step 8. Kích hoạt timer

Sau khi thiết lập các thông số, tiến hành khai kích hoạt timer tương ứng bằng thanh ghi: TIMx_CR1

Ví dụ:

```
TIM2 -> EGR |= (1<<0) ;
```

```
TIM2 -> CR1 |= (1<<0) ;
```

Note: Cả 4 kênh PWM có thể cùng lúc hoạt động với cùng tần số bằng cách khai báo chân, kích hoạt, và khởi tạo các thông số quyết định độ rộng xung PWM

Bài tập 1. Viết chương trình tạo xung PWM có tần số 20kHz ở 2 kênh.

CH1 và CH4 với độ rộng xung CH1 là 50%, CH4 là 80%

Bài tập 2. Viết chương trình điều khiển tốc độ động cơ sử dụng IC chuyên dụng L298 theo yêu cầu sau:

1. Nhấn B0 tốc độ động cơ tăng dần (mỗi lần nhấn tăng 10% độ rộng xung).
2. Nhấn B1 tốc độ động cơ giảm dần (mỗi lần nhấn giảm 10% độ rộng xung).
3. Nhấn B2 động cơ đảo chiều quay .