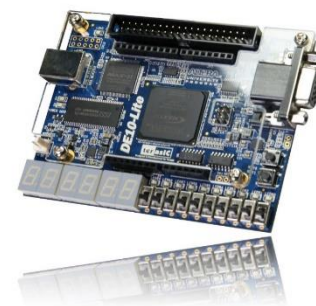


# GRAFCET電路設計-PWM

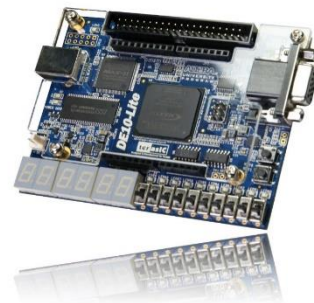
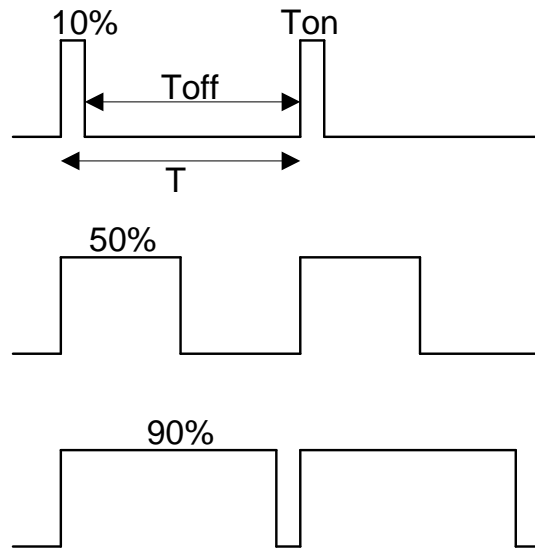
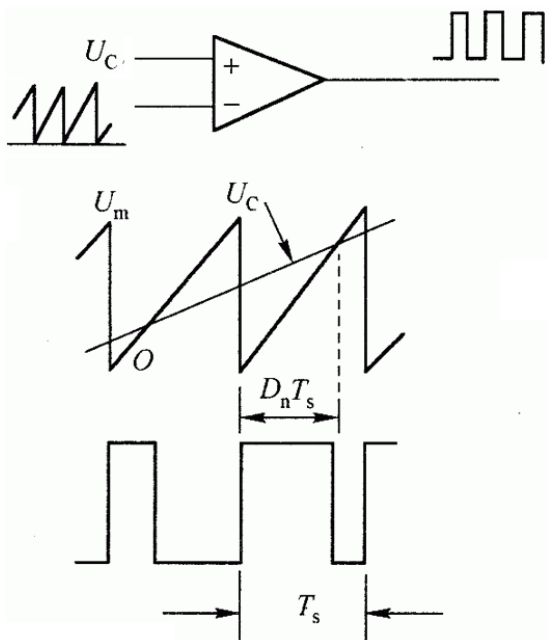
# Outline

- PWM介紹
- 傳統驅動電路 vs. PWM驅動電路
- PWM原理
- PWM控制應用實例一
- PWM控制應用實例二
- PWM控制應用實例三



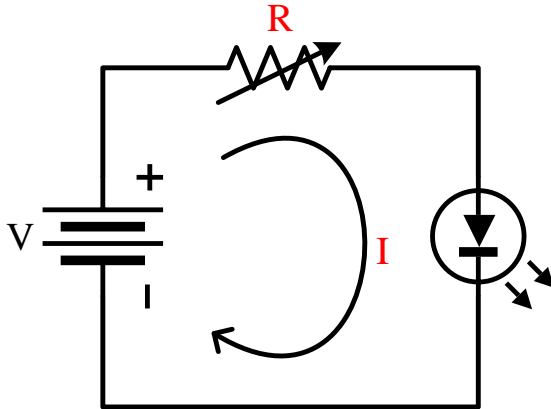
# PWM介紹

- 脈波寬度調變(Pulse Width Modulation, PWM)，簡稱**脈寬調變**，是將類比訊號轉換為脈波的一種技術，一般轉換後脈波的週期固定，但脈波的工作週期會依類比訊號的大小而改變。



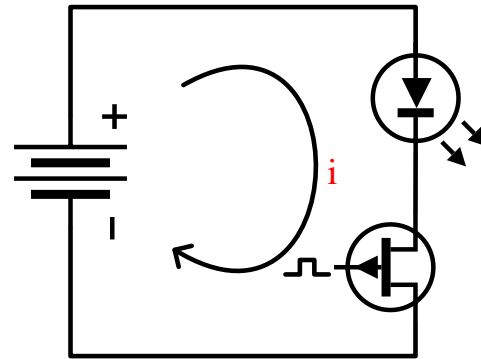
# PWM介紹

- 傳統驅動電路 vs. PWM驅動電路
  - 傳統驅動電路損失大
  - PWM驅動電路可讓損失減至最小
  - PWM驅動容易控制



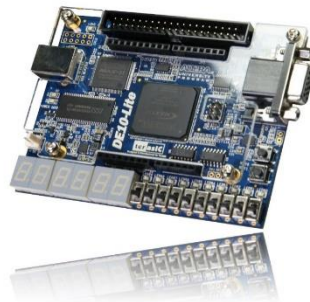
$$I = V \div R$$

$$P_R = I^2 \cdot R$$

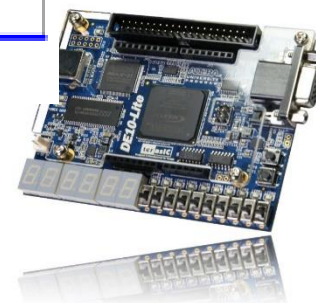
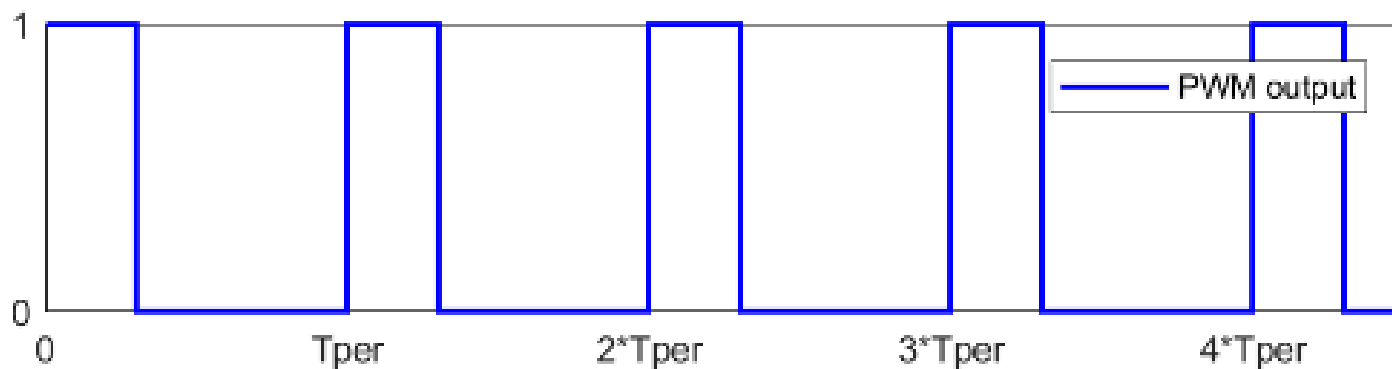
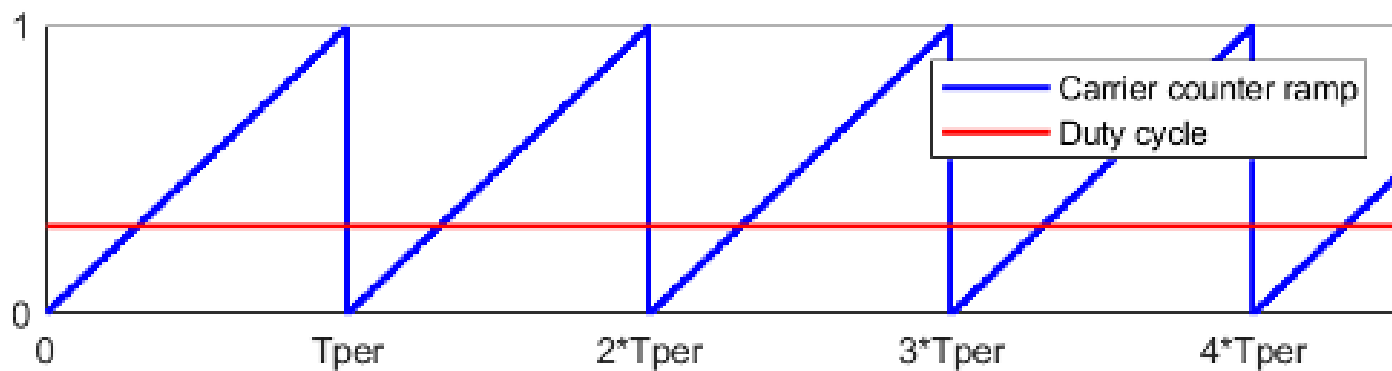


$$\bar{i} = \frac{1}{T} \int_0^T f(t) dt$$

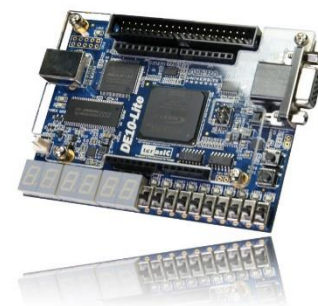
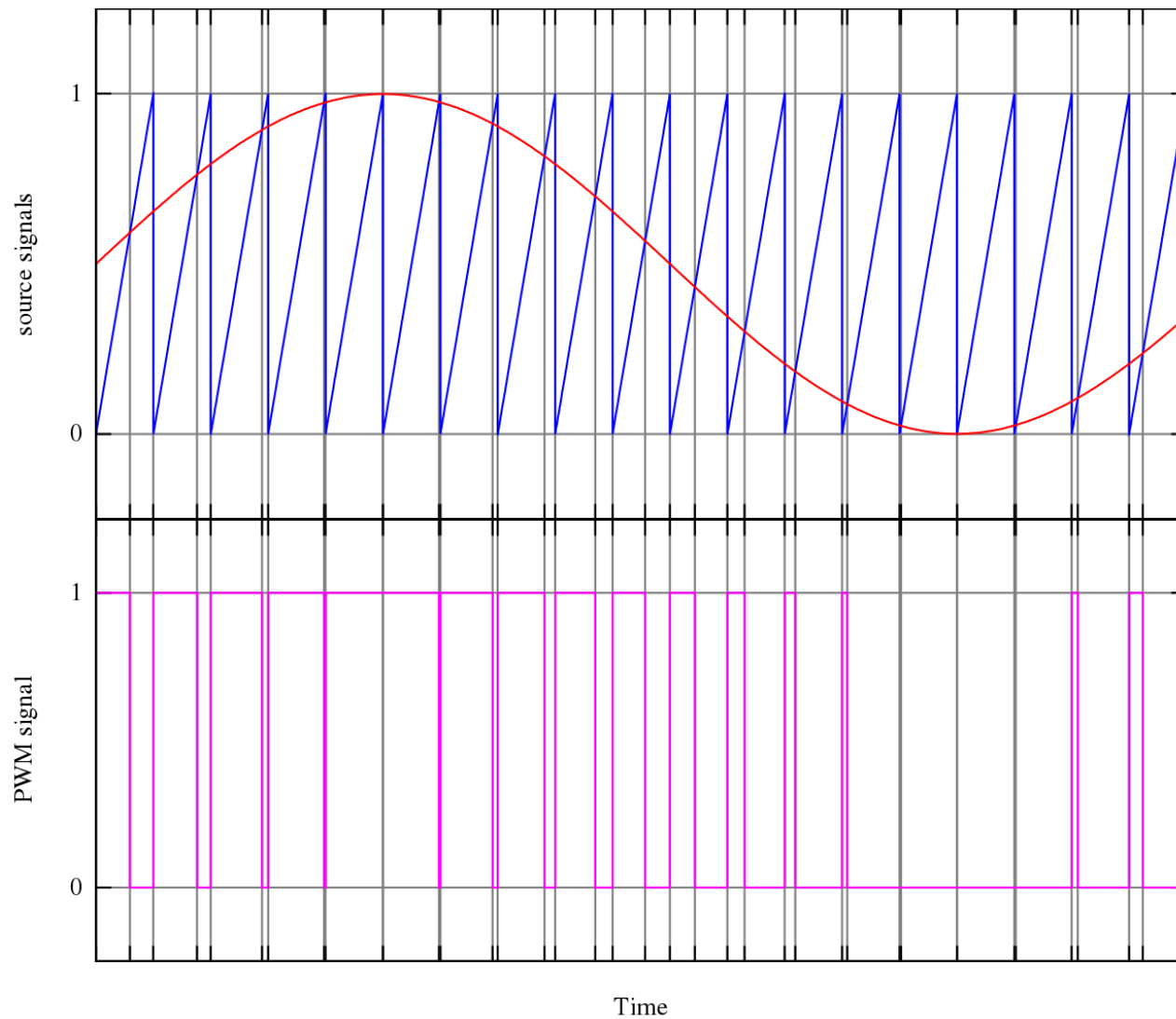
$$R_{DS\_ON} \downarrow \downarrow$$



# PWM原理(1/2)

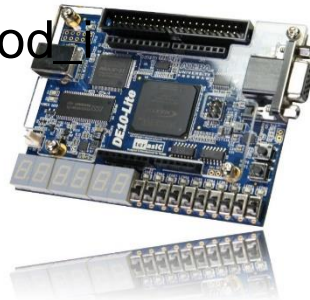
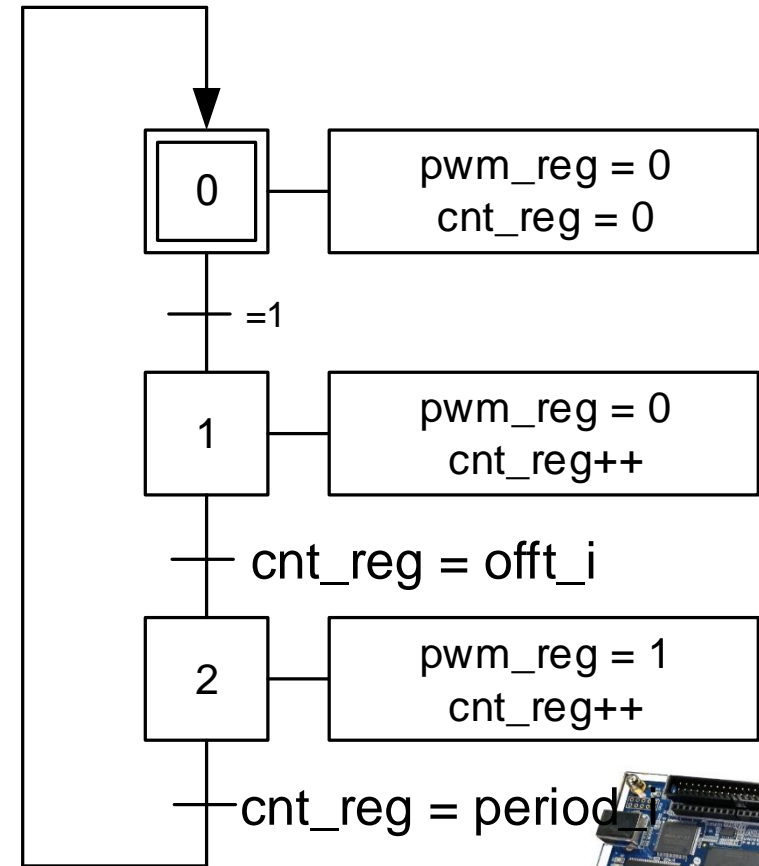
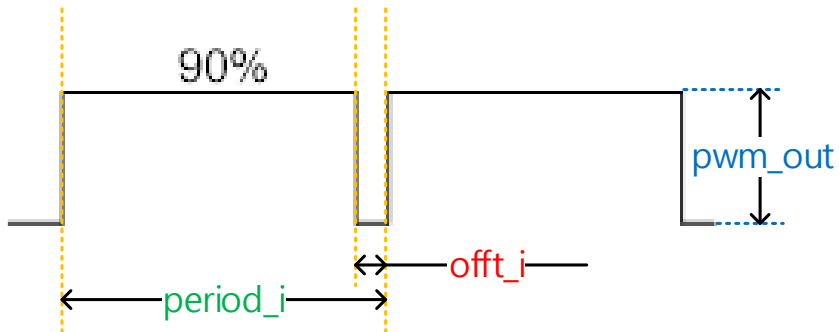


# PWM原理(2/2)



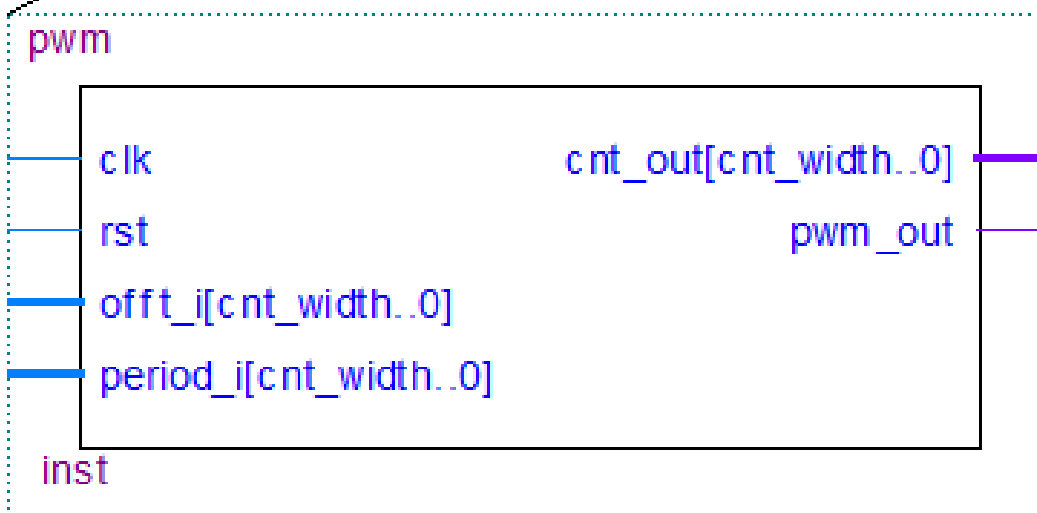
# PWM GRAFCET 程序控制器

- rst
- clk
- period\_i
- offt\_i
- **cnt\_out**
- pwm\_out



# PWM GRAFCET 程序控制器

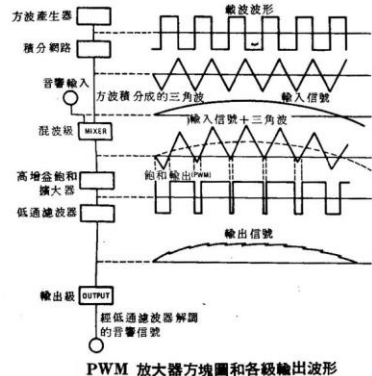
Parameter	Value	Type
CNT_WIDTH	7	Signed Integer



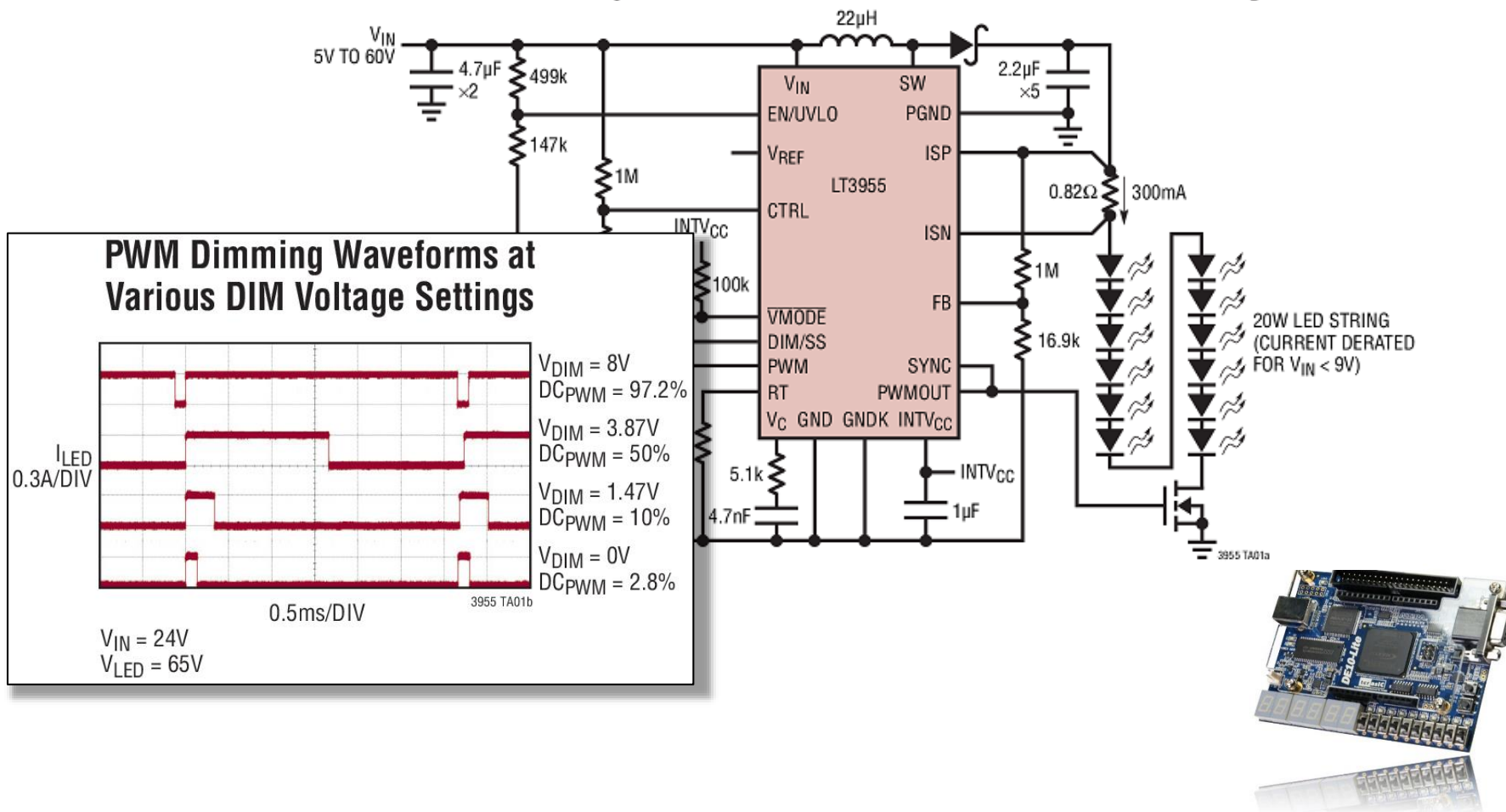


# PWM控制應用實例一

- 調光電路

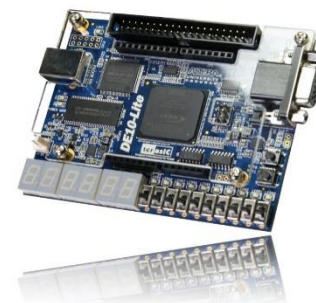
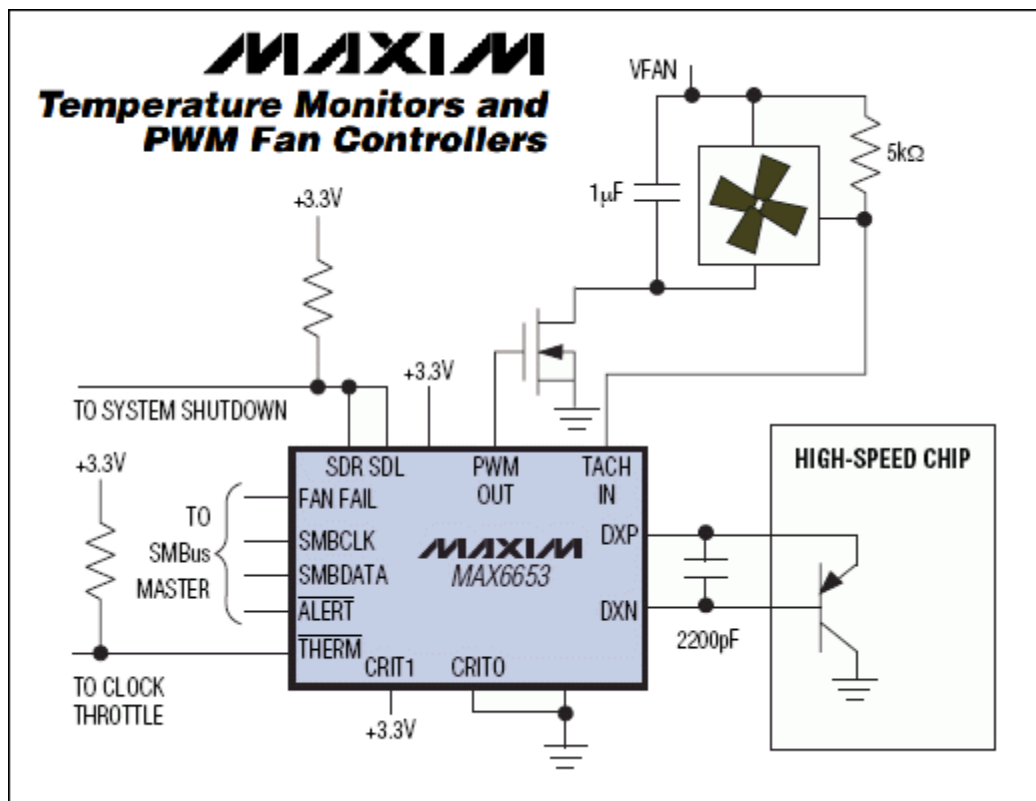


## 94% Efficiency 20W Boost LED Driver with Internal PWM Dimming



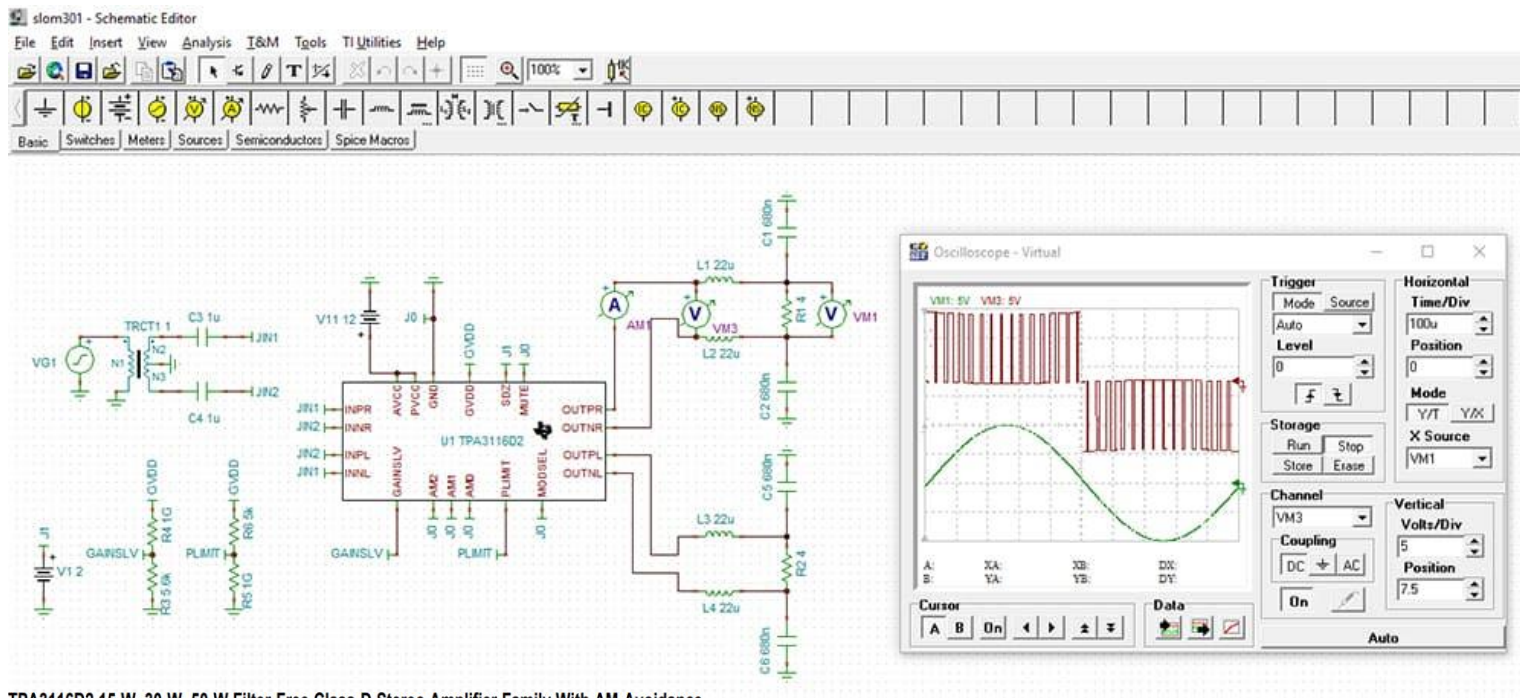
# PWM控制應用實例二

- DC風扇控制

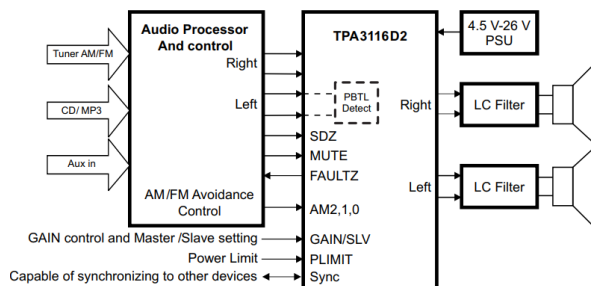


# PWM控制應用實例三

## • D類放大器



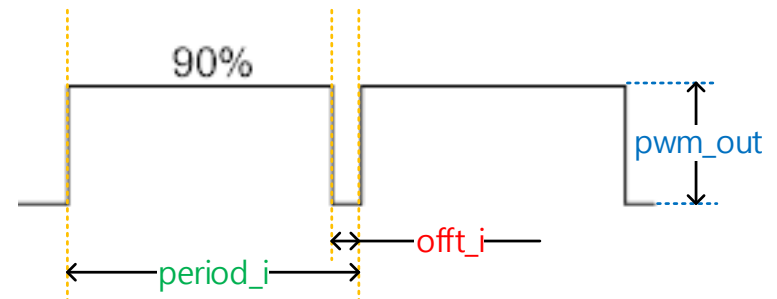
TPA3116D2 15-W, 30-W, 50-W Filter-Free Class-D Stereo Amplifier Family With AM Avoidance



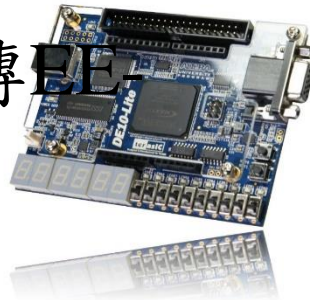
# 隨堂練習

- 請使用 VHDL 完成下列 PWM 控制電路，並完成紀錄，包括 **GRAFCET 離散事件建模**、**VHDL Source Code**、**模擬波形圖**。

Sig.	Dir.	Bit	Desc.
clk	in	1	時脈
period_i	in	8 (Default)	如圖
offt_i	in	8 (Default)	如圖
cnt_out	out	8 (Default)	
pwm_out	out	1	如圖

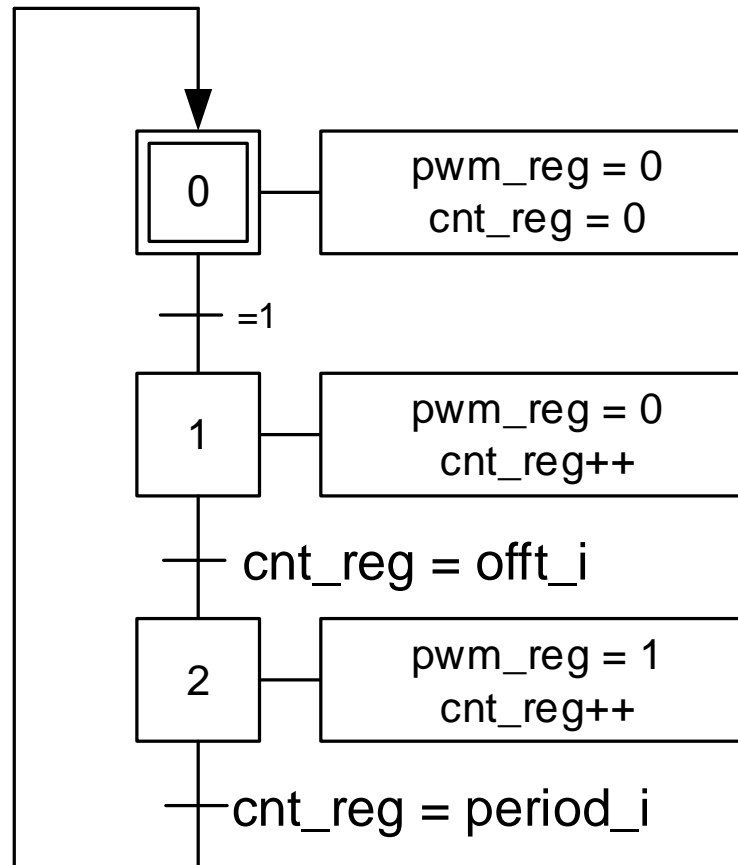


- 本次實驗完成後請，將專案與報告壓縮上傳 PE Class。
- Lecture10\_組別XX.ZIP



# 隨堂練習

- GRAFCET離散事件建模



# 隨堂練習

- VHDL Source Code

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity pwm is
generic(CNT_WIDTH:integer:=7);
port(
  clk,rst:in std_logic;
  offst_i: in std_logic_vector(CNT_WIDTH downto 0);
  period_i: in std_logic_vector(CNT_WIDTH downto 0);
  cnt_out: out std_logic_vector(CNT_WIDTH downto 0);
  pwm_out: out std_logic
);
end pwm;

--
ARCHITECTURE action OF pwm IS
  signal x0,x1,x2,pwm_reg: std_logic;
  signal cnt_reg:std_logic_vector(CNT_WIDTH downto 0);
begin
  process(clk,rst)
  begin
    if rst='0' then
      x0<='1';
      x1<='0';
      x2<='0';
    elsif clk'event and clk='1' then
      if x0='1' then x0<='0'; x1<='1';
      elsif x1='1' and cnt_reg=offst_i then x1<='0'; x2<='1';
      elsif x2='1' and cnt_reg=period_i then x2<='0'; x0<='1';
      end if;

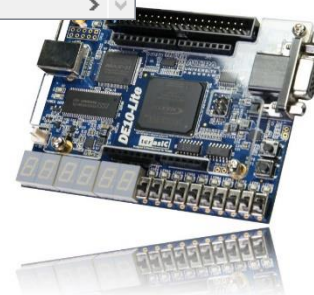
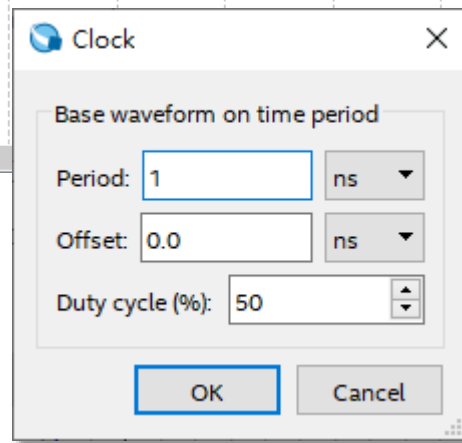
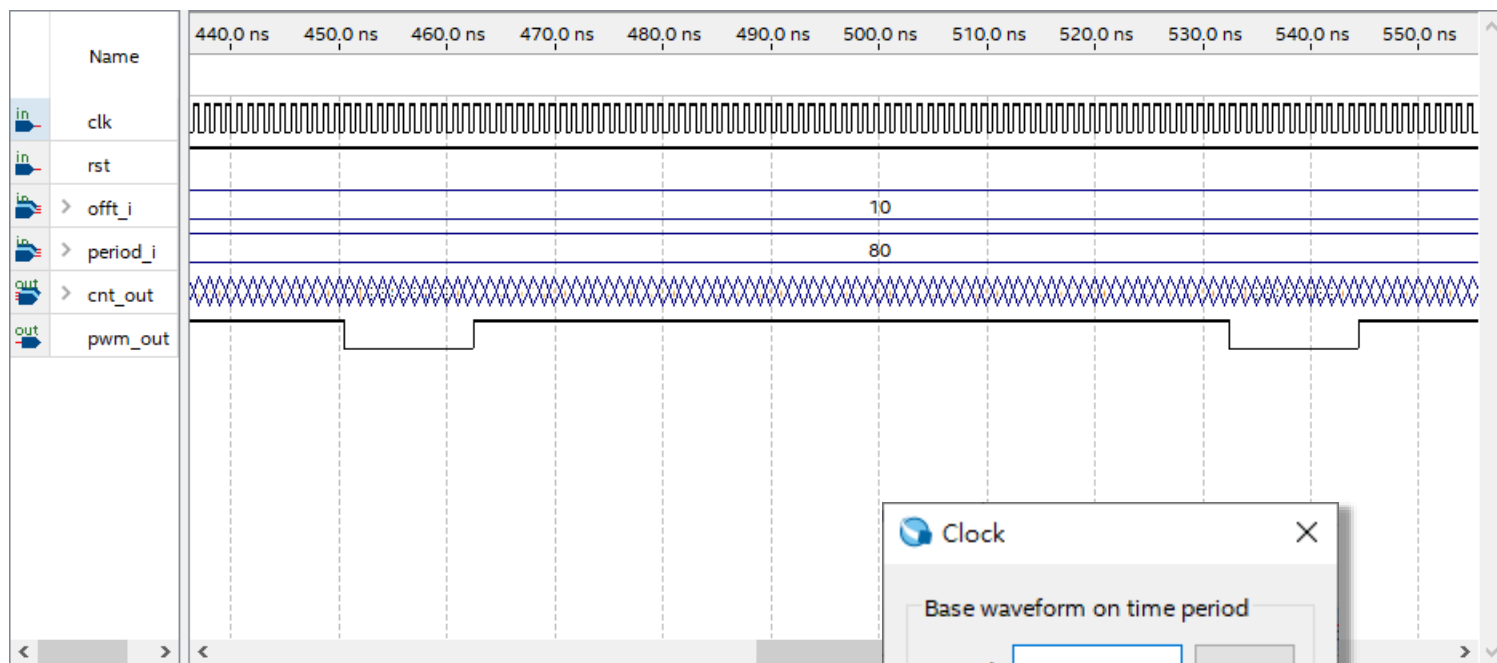
      if x0='1' then pwm_reg<='0'; cnt_reg<=(others=>'0');
      elsif x1='1' then pwm_reg<='0'; cnt_reg<=cnt_reg+1;
      elsif x2='1' then pwm_reg<='1'; cnt_reg<=cnt_reg+1;
      end if;
    end if;
  end process;

  cnt_out<=cnt_reg;
  pwm_out<=pwm_reg;
end action;
```



# 隨堂練習

- 模擬波形圖





- RTL Schematic

