

## การใช้งาน PWM และ การควบคุมเซอร์โวมอเตอร์

---



รศ.ณรงค์ บวบทอง  
ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยธรรมศาสตร์

### หัวข้อ

---

- วัตถุประสงค์
- PWM คืออะไร
- การใช้งาน โมดูล PWM ของ PSoC
- Application Programming Interface (API) เกี่ยวกับ PWM
- เซอร์โวมอเตอร์ (Servo motor ) และการใช้งาน

## วัตถุประสงค์

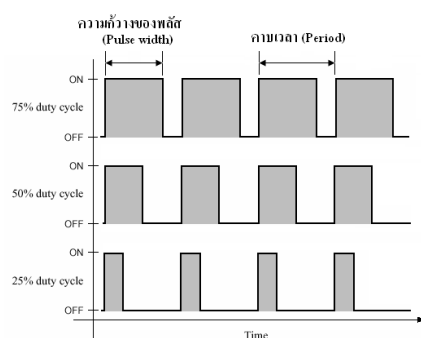
- เพื่อให้เข้าใจการทำงานของ PWM และสามารถเขียนโปรแกรมควบคุมการทำงานของ PWM ได้
- เพื่อให้เข้าใจการทำงานของเซอร์โวมอเตอร์ และสามารถเขียนโปรแกรมควบคุมการทำงานของ เซอร์โวมอเตอร์ได้ ด้วยการใช้โมดูล PWM

## PWM คืออะไร

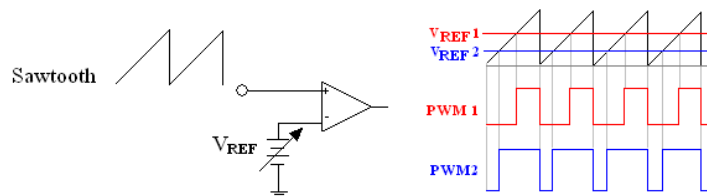
PWM หรือ Pulse Width

Modulation คือสัญญาณพัลส์ที่มีค่าความถี่คงที่ แต่ความกว้างของพัลส์เปลี่ยนแปลงได้

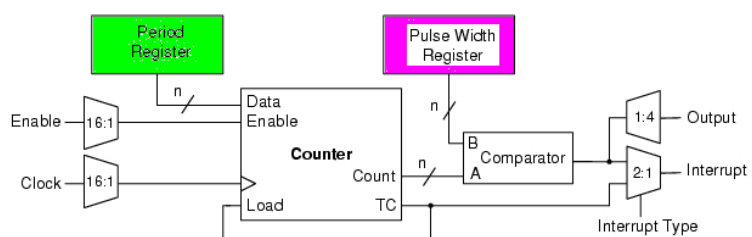
PWM เป็นวิธีหนึ่งที่นิยมใช้กันมากในงานควบคุม เช่นการควบคุมความเร็วมอเตอร์



## การสร้างสัญญาณ PWM ด้วยวิธีทาง Analog

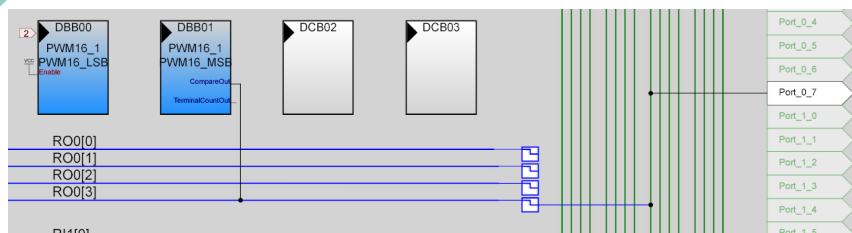
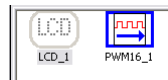


## การสร้างสัญญาณ PWM ด้วยวิธีทาง Digital



PWM Block Diagram, Data Path width  $n = 8$  or  $16$

## การใช้งานโมดูล PWM ของ PSoC



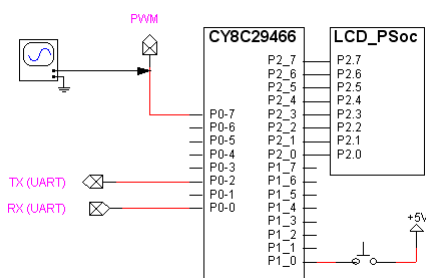
## การกำหนดพารามิเตอร์ Global Resource

$$\begin{aligned} VC1 &= \text{SysClk}/8 \\ VC2 &= VC1/3 \\ VC2 &= \text{SysClk}/(8 \times 3) \\ &= 24\text{MHz}/24 = 1 \text{ MHz} \\ VC3 &= 8 \times \text{Baudrate} \\ &= 8 \times 9600 = 76.8 \text{ K} \\ VC3 &= VC1/n \\ n &= 3000/76.8 \\ &= 39.0625 \\ \text{ดังนั้น } VC3 \text{ Divider} &= 39 \end{aligned}$$

Global Resources	Value
Power Setting [ Vcc / SysClk freq	5.0V / 24MHz
CPU_Clock	SysClk/8
32K_Select	Internal
FLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	8
VC2= VC1/N	3
VC3 Source	VC1
VC3 Divider	39
SysClk Source	Internal
SysClk*2 Disable	No
Analog Power	SC 0n/Ref Low
Ref Mux	[Vdd/2]+/-BandGap
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
SwitchModePump	OFF
Trip Voltage [LVD (SMP)]	4.81V (5.00V)
LVDThrottleBack	Disable
Watchdog Enable	Disable

## การกำหนดพารามิเตอร์ PWM และพอร์ต

PWM16\_1\_WritePeriod(2000);  
 ดังนั้นได้ค่าคาบเวลาเท่ากับ ?



PWM16_1	
User Module Parameters	Value
Clock	VC2
Enable	High
CompareOut	Row_0_Output_3
TerminalCountOut	None
Period	0
PulseWidth	0
CompareType	Less Than Or Equal
InterruptType	Terminal Count
ClockSync	Sync to SysClk
InvertEnable	Normal

LCD_1	
User Module Parameters	Value
LCDPort	Port_2
BarGraph	Enable

Name	Port	Select	Drive	Interrupt
Port_0_7	P0[7]	StdCPU	High Z Analog	DisableInt
Port_1_0	P1[0]	StdCPU	Pull Down	DisableInt
Port_1_1	P1[1]	StdCPU	High Z Analog	DisableInt

## API ของ PWM16

- PWM16\_Start
- PWM16\_Stop
- PWM16\_WritePeriod
- PWM16\_WritePulseWidth

## PWM16\_Start

---

ฟังก์ชัน `PWM16_Start`

รายละเอียด

Starts the PWM16 User Module. If the enable input is high, the counter will begin to down count.

การใช้งานด้วยภาษา C

```
void PWM16_Start(void);
```

พารามิเตอร์

ไม่มี

ค่าส่งกลับ

ไม่มี

## PWM16\_Stop

---

ฟังก์ชัน `PWM16_Stop`

รายละเอียด

Stops the counter operation.

การใช้งานด้วยภาษา C

```
void PWM16_Stop(void);
```

พารามิเตอร์

ไม่มี

ค่าส่งกลับ

ไม่มี

## PWM16\_WritePeriod

---

ฟังก์ชัน `PWM16_WritePeriod`

รายละเอียด

Writes the Period register with the period value. The period value will be transferred from the Period register to the Counter register immediately, if the PWM16 is stopped or when the counter reaches the zero count.

การใช้งานด้วยภาษา C

```
void PWM16_WritePeriod(WORD wPeriod);
```

พารามิเตอร์

wPeriod: wPeriod value is a value from 0 to 216-1. MSB is passed in the X register and LSB is passed in the Accumulator.

ค่าส่งกลับ

ไม่มี

## PWM16\_WritePulseWidth

---

ฟังก์ชัน `PWM16_WritePulseWidth`

รายละเอียด

Writes the PulseWidth register with the pulse width value.

การใช้งานด้วยภาษา C

```
void PWM16_WritePulseWidth(WORD wPeriod);
```

พารามิเตอร์

wPulseWidth: wPulseWidth value is the value from 0 to the period value. MSB is passed in the X register and LSB is passed in the Accumulator.

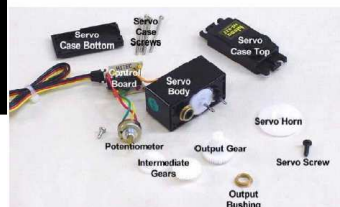
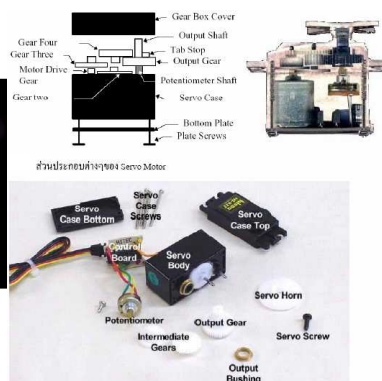
ค่าส่งกลับ

ไม่มี

## เซอร์โวมอเตอร์ (Servo motor) และการใช้งาน

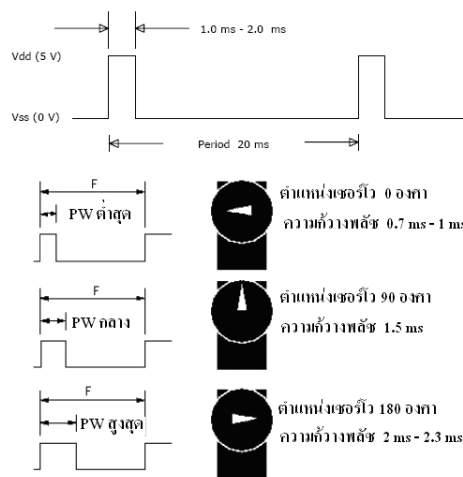
- Servo motor คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับ ชุดเกียร์ และ ส่วนควบคุม ต่างๆ ไว้ ใน โมดูลเดียวกัน หรือ ภายในกล่องพลาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC,GND และ สายสัญญาณควบคุม(Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้าย หรือ ขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณพัลส์วีดมอด (PWM) แบบ TTL Level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ใน ช่วงประมาณ 4 ถึง 6 โวลท์

## ลักษณะของเซอร์โวมอเตอร์





## การควบคุมเซอร์โวมอเตอร์



หมายเหตุ ตำแหน่งมุมที่เรียกแบบนี้ เพื่อให้มุมเป็นเลขจำนวนเต็มบวก

## การกำหนดค่าของ PWM สำหรับควบคุมเซอร์โวมอเตอร์

จากความถี่ 1 MHz :  $T = 1/F = 1/1\text{MHz} = 1 \mu\text{s}$   
 หมายความว่าถ้านับสัญญาณนี้ 1 ไซเคิลจะได้เวลา 1  $\mu\text{s}$   
 ดังนั้นถ้าต้องการเวลา 20 ms ต้องใช้ =  $20 \text{ ms} / 1 \mu\text{s}$   
 = 20000 ไซเคิล

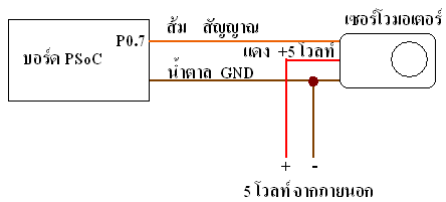
ดังนั้นการสร้างความกว้างของพัลส์ของโมดูล PWM เท่ากับ 20000  
`PWM16_1_WritePeriod(20000);`

ที่มุม 0 องศา ต้องการเวลา 0.7 ms ต้องใช้ =  $0.7 \text{ ms} / 1 \mu\text{s}$   
 = 700

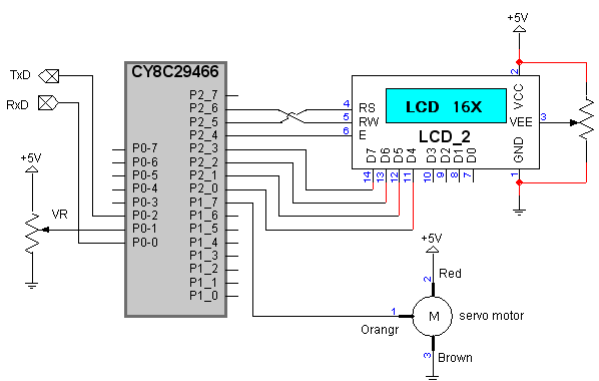
ที่มุม 90 องศา ต้องการเวลา 1.5 ms ต้องใช้ =  $1.5 \text{ ms} / 1 \mu\text{s}$   
 = 1500

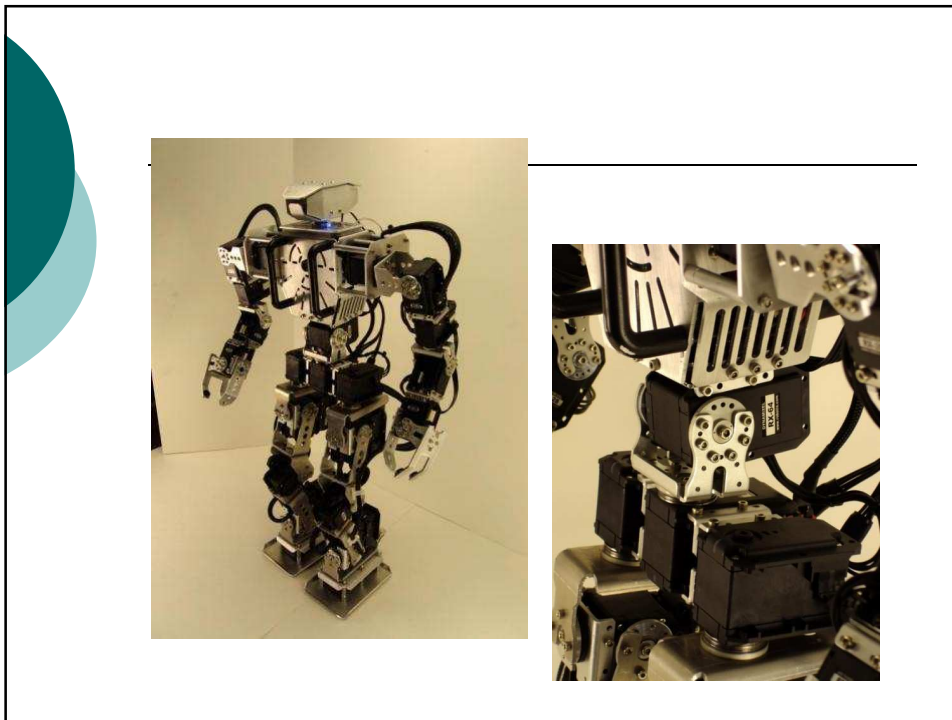
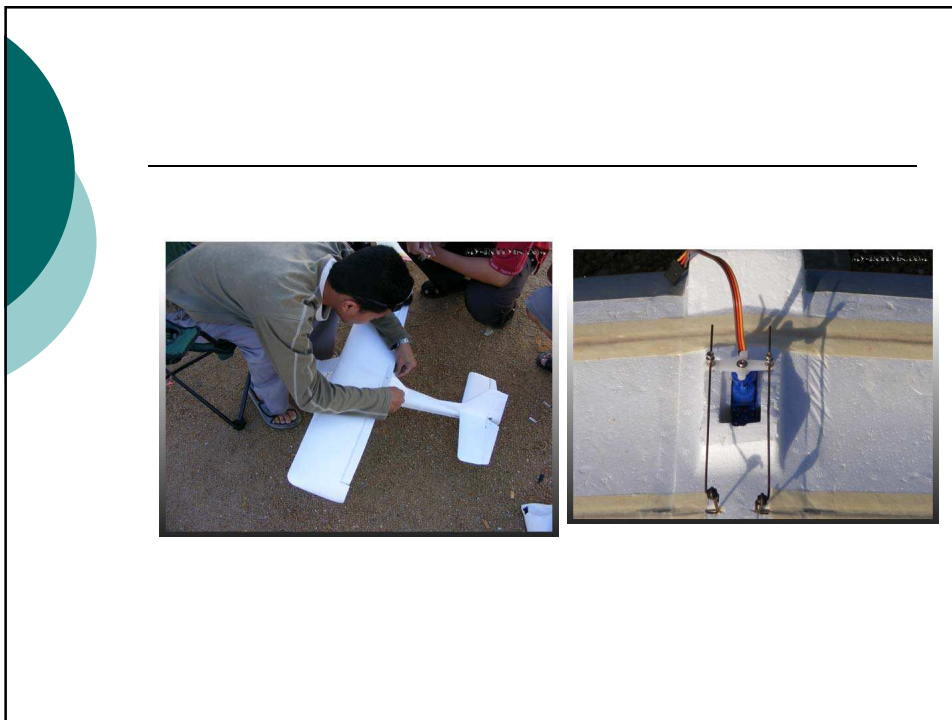
ที่มุม 180 องศา ต้องการเวลา 2.3 ms ต้องใช้ =  $2.3 \text{ ms} / 1 \mu\text{s}$   
 = 2300

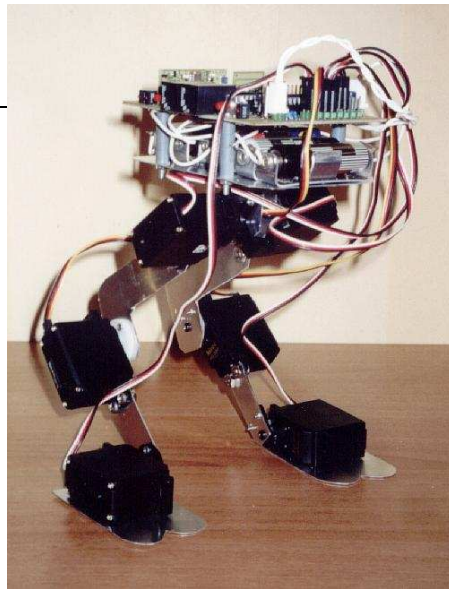
วงจรทดลอง



วงจรทดลอง







## จบแล้วครับ

