



## PWM (Pulse Width Modulation) 功能使用方法

### 1 適用產品：

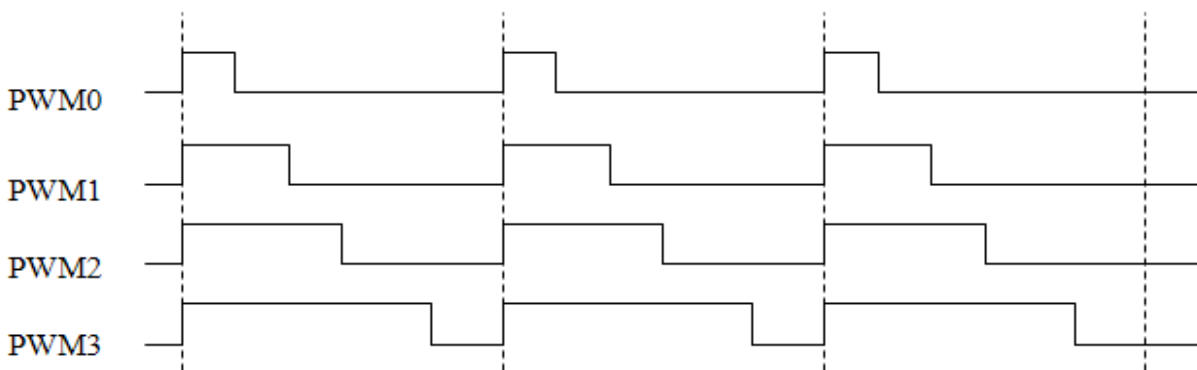
- 1.1 SM59R16A2/ SM59R08A2
- 1.2 SM59R16A5/ SM59R09A5/ SM59R05A5/ SM59R16A3/ SM59R09A3/ SM59R05A3
- 1.3 SM59R16G6/ SM59R09G6/ SM59R05G6
- 1.4 SM59R04A2/ SM59R04A1/ SM59R03A1/ SM59R02A1
- 1.5 文件說明：SM59R 系列注意 PWM 因架構差異，各型號說明請參考以下相對章節。

### 2 SM59R16A5/ SM59R09A5/ SM59R05A5/ SM59R16A3/ SM59R09A3/ SM59R05A3/ SM59R16G6/

SM59R09G6/ SM59R05G6/ SM59R04A2/ SM59R04A1/ SM59R03A1/ SM59R02A1 PWM 使用說明:

(SM59R16A2/ SM59R08A2 請參考章節 3 說明)

- 2.1 提供 4 組獨立的 PWM 信號
- 2.2 輸入時脈可由暫存器 PWMCS[2:0]選擇
- 2.3 PWM 為 10-bit 模式，但最大資料計數可由 PWMMDH 及 PWMDL 設定為 1~1024。



### 2.4 PWM 相關的特殊暫存器 PWM Special Function Register (SFR)

Mnemonic	Description	Direct	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RESET
PWM											
PWMC	PWM Control register	B5h	PWMCS[2:0]			-	PWM3EN	PWM2EN	PWM1EN	PWM0EN	00H
PWMD0H	PWM 0 Data register high byte	BCh	PWMP0	-	-	-	-	-	PWMD0[9:8]		00H
PWMD0L	PWM 0 Data register low byte	BDh	PWMD0[7:0]								00H
PWMD1H	PWM 1 Data register high byte	BEh	PWMP1	-	-	-	-	-	PWMD1[9:8]		00H
PWMD1L	PWM 1 Data register low byte	BFh	PWMD1[7:0]								00H
PWMD2H	PWM 2 Data register high byte	B1h	PWMP2	-	-	-	-	-	PWMD2[9:8]		00H

Specifications subject to change without notice, contact your sales representatives for the most recent information.



PWMD2L	PWM 2 Data register low byte	B2h	PWMD2[7:0]							00H
PWMD3H	PWM 3 Data register high byte	B3h	PWMP3	-	-	-	-	-	PWMD3[9:8]	00H
PWMD3L	PWM 3 Data register low byte	B4h	PWMD3[7:0]							00H
PWMMDH	PWM Max Data register high byte	CEh	-	-	-	-	-	-	PWMMD[9:8]	00H
PWMMDL	PWM Max Data register low byte	CFh	PWMMD[7:0]							FFH

**2.5 SM59R16A5/SM59R09A5/SM59R05A5/SM59R16A3/SM59R09A3/SM59R05A3 之 PWM 輸出**  
可選擇由 P2[4:7]或 P4[0:7]，預設值為 P4[0:7]，其設定之特殊暫存器為 AUX(0x91h)

Mnemonic: AUX Address: 91h

7	6	5	4	3	2	1	0	Reset
BRS	-	P4SPI	P4UR1	P4IIC	P0KBI	P2PWM	DPS	00H

P2PWM : P2PWM = 0 – PWM function on P4.  
P2PWM = 1 – PWM function on P2.

**2.6 SM59R16G6/ SM59R09G6/ SM59R05G6 之 PWM 輸出**可選擇由 P2[4:7]或 P4[0:7]，預設值為 P4[0:7]，其設定之特殊暫存器為 AUX(0x91h)

Mnemonic: AUX Address: 91H

7	6	5	4	3	2	1	0	Reset
BRGS	-	P4SPI	P1UR	P4IIC	-	P2PWM	DPS	00H

P2PWM : P2PWM = 0 – PWM function on P4.  
P2PWM = 1 – PWM function on P2.

Mnemonic: PWMC

Address: B5h

7	6	5	4	3	2	1	0	Reset
PWMCS[2:0]		-	PWM3EN	PWM2EN	PWM1EN	PWM0EN	00H	

PWMCS[2:0]: 時脈選擇位元(PWM clock select)

PWMCS [2:0]	Mode
000	Fosc
001	Fosc/2
010	Fosc/4
011	Fosc/6
100	Fosc/8
101	Fosc/12
110	Timer 0 overflow
111	Timer 0 external input (P3.4/T0)



- PWM3EN: 通道 3 致能位元(PWM channel 3 enable control bit)  
PWM3EN = 1 - PWM channel 3 致能.  
PWM3EN = 0 - PWM channel 3 禁能.
- PWM2EN: 通道 2 致能位元(PWM channel 2 enable control bit)  
PWM2EN = 1 - PWM channel 2 致能.  
PWM2EN = 0 - PWM channel 2 禁能.
- PWM1EN: 通道 1 致能位元(PWM channel 1 enable control bit)  
PWM1EN = 1 - PWM channel 1 致能.  
PWM1EN = 0 - PWM channel 1 禁能.
- PWM0EN: 通道 0 致能位元(PWM channel 0 enable control bit)  
PWM0EN = 1 - PWM channel 0 致能.  
PWM0EN = 0 - PWM channel 0 禁能.

**Mnemonic: PWMD0H** **Address: BCh**

7	6	5	4	3	2	1	0	Reset
PWMP0	-	-	-	-	-	PWMD0[9:8]		00H

**Mnemonic: PWMD0L** **Address: BDh**

7	6	5	4	3	2	1	0	Reset
PWMD0[7:0]								00h

- PWMP0: 通道 0 閒置準位選擇位元(PWM channel 0 idle polarity select)  
“0” –通道 0 閒置時為低準位(PWM channel 0 will idle low)  
“1” –通道 0 閒置時為高準位(PWM channel 0 will idle high)
- PWMD0[9:0]: 通道 0 資料暫存器(PWM channel 0 data register)  
可調整 PWM 通道 0 Duty Cycle

**Mnemonic: PWMD1H** **Address: BEh**

7	6	5	4	3	2	1	0	Reset
PWMP1	-	-	-	-	-	PWMD1[9:8]		00H

**Mnemonic: PWMD1L** **Address: BFh**

7	6	5	4	3	2	1	0	Reset
PWMD1[7:0]								00H

- PWMP1: 通道 1 閒置準位選擇位元(PWM channel 1 idle polarity select)  
“0” –通道 1 閒置時為低準位(PWM channel 1 will idle low)  
“1” –通道 1 閒置時為高準位(PWM channel 1 will idle high)
- PWMD1[9:0]: 通道 1 資料暫存器(PWM channel 1 data register)  
可調整 PWM 通道 1 Duty Cycle

**Mnemonic: PWMD2H** **Address: B1h**

7	6	5	4	3	2	1	0	Reset
PWMP2	-	-	-	-	-	PWMD2[9:8]		00H

Specifications subject to change without notice, contact your sales representatives for the most recent information.



**Mnemonic: PWMD2L**

**Address: B2h**

7	6	5	4	3	2	1	0	Reset
PWMD2[7:0]								00H

PWMP2: 通道2閒置準位選擇位元(PWM channel 2 idle polarity select)

“0” –通道 2 閒置時為低準位(PWM channel 2 will idle low)

“1” –通道 2 閒置時為高準位(PWM channel 2 will idle high)

PWMD2[9:0]: 通道 2 資料暫存器(PWM channel 2 data register)

可調整 PWM 通道 2 Duty Cycle

**Mnemonic: PWMD3H**

**Address: B3h**

7	6	5	4	3	2	1	0	Reset
PWMP3	-	-	-	-	-	PWMD3[9:8]		00H

**Mnemonic: PWMD3L**

**Address: B4h**

7	6	5	4	3	2	1	0	Reset
PWMD3[7:0]								00H

PWMP3: 通道 3 閒置準位選擇位元(PWM channel 3 idle polarity select)

“0” –通道 3 閒置時為低準位(PWM channel 3 will idle low)

“1” –通道 3 閒置時為高準位(PWM channel 3 will idle high)

PWMD3[9:0]: 通道 3 資料暫存器(PWM channel 3 data register)

可調整 PWM 通道 3 Duty Cycle

**Mnemonic: PWMMDH**

**Address: CEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	PWMMD[9:8]		00H

**Mnemonic: PWMDL**

**Address: CFh**

7	6	5	4	3	2	1	0	Reset
PWMMD[7:0]								FFH

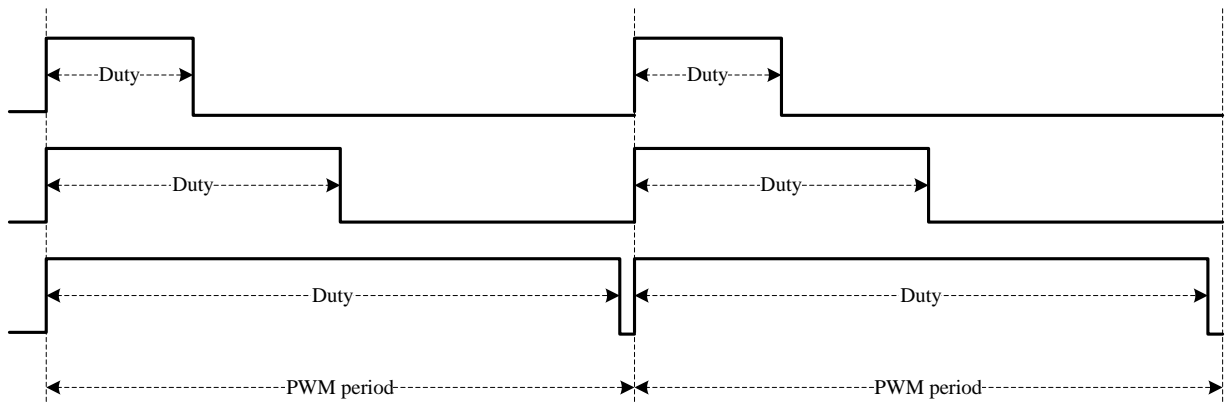
PWMMD[9:0]: 最大資料設定暫存器(PWM Max Data register)

PWM 由 0000 計數至 PWMMD[9:0]，當計數等於 PWMMD[9:0]時會產生溢位。

PWMPx = 0

Duty = PWMDx[9:0]

Period = PWMMD[9:0]



$$\text{PWM period} = \frac{\text{PWMMD} + 1}{\text{PWM clock}}$$

$$\text{Leader pulse} = \frac{\text{PWMDx}}{\text{PWM clock}}$$

## 2.7 PWM 中斷

### 2.7.1 向量表(Interrupt vectors table)

Interrupt Request Flags	Interrupt Vector Address	Interrupt Number *(use Keil C Tool)
IE0 – External interrupt 0	0003h	0
TF0 – Timer 0 interrupt	000Bh	1
IE1 – External interrupt 1	0013h	2
TF1 – Timer 1 interrupt	001Bh	3
RI0/TI0 – Serial channel 0 interrupt	0023h	4
TF2/EXF2 – Timer 2 interrupt	002Bh	5
PWMIF – PWM interrupt	0043h	8
SPIIF – SPI interrupt	004Bh	9
ADCIF – A/D converter interrupt	0053h	10
KBIIF – keyboard Interface interrupt	005Bh	11
LVIIF – Low Voltage Interrupt	0063h	12
IICIF – IIC interrupt	006Bh	13
RI1/TI1 – Serial channel 1 interrupt	0083h	16

\*See Keil C about C51 User's Guide about Interrupt Function description



### 2.7.2 中斷相關暫存器(Interrupt SFR)

Mnemonic	Description	Direct	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RESET
<b>Interrupt</b>											
IEN0	Interrupt Enable 0 register	A8h	EA	-	ET2	ES0	ET1	EX1	ET0	EX0	00h
IEN1	Interrupt Enable 1 register	B8h	EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h
IEN2	Interrupt Enable 2 register	9Ah	-	-	-	-	-	-	-	ES1	00h
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIF	KBIIF	ADCIF	SPIIF	PWMIF	00H
IP0	Interrupt priority level 0	A9h	-	-	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	00h
IP1	Interrupt priority level 1	B9h	-	-	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	00h

PWM 中斷可參考以下設定：

**(1) PWM 中斷致能設定：**

```
IEN0 |= 0x80;           //Enable interrupt All
IEN1 |= 0x01;           //Enable interrupt PWM
```

**(2) PWM 中斷程序表示：**

```
void IIC_interrupt(void) interrupt 8{
    if(IRCON != 0x01)
    {
        IRCON &= (~0x01);           //Clear interrupt flag
    }
}
```

## 2.8 PWM 範例程式

### 2.8.1 PWM Assembly 程式範例

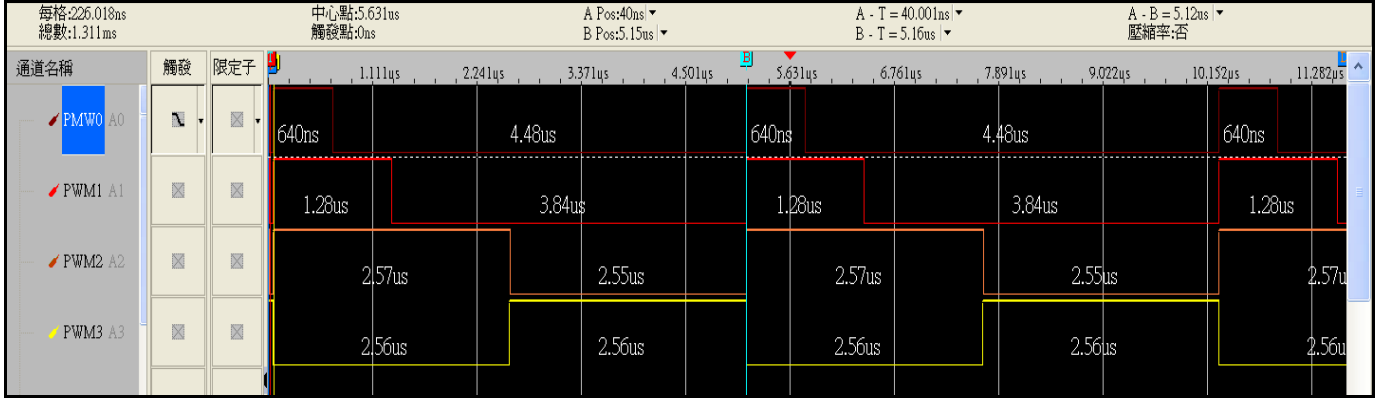
Description



Main program	<pre>//===== ===== // //          S Y N C M O S   T E C H N O L O G Y // //===== ===== #include "SM59R04A2.h"  #define d_Fosc          0x00 #define d_Fosc_div2    0x20 #define d_Fosc_div4    0x40 #define d_Fosc_div6    0x60 #define d_Fosc_div8    0x80 #define d_Fosc_div12   0xA0 #define d_T0_overflow  0xC0 #define d_T0_EX        0xE0 #define d_PWMCS        d_Fosc_div12 //PWMCS[2:0] (PWM clock select 0~7) #define d_PWMXEN        0x0F //PWM0~3 Channel Enable (0~F) #define d_IDLE_HI      0x80 #define d_IDLE_LO      0x00 #define d_PWMP0        d_IDLE_HI //PWM0 Channel idle polarity select #define d_PWMP1        d_IDLE_HI //PWM1 Channel idle polarity select #define d_PWMP2        d_IDLE_HI //PWM2 Channel idle polarity select #define d_PWMP3        d_IDLE_LO //PWM3 Channel idle polarity select #define d_PWMIE        0x00 //PWM Interrupt Enable bit  //P2PWM only for SM59R16A5/ SM59R09A5/ SM59R05A5/ SM59R16A3/ SM59R09A3/ SM59R05A #define d_P2PWM        0x02 //PWM channel on P2/P4 select      org    0000h //Start     call  PWM_initialize Start:     jmp    Start PWM_initialize:     mov    PWMMDH, #000 //set pwm max period     mov    PWMMDL, #100     mov    PWMD0H, #d_PWMP0 //set idle polarity     orl    PWMD0H, #0 //set pwm0 period     mov    PWMD0L, #10     mov    PWMD1H, #d_PWMP1 //set idle polarity     orl    PWMD1H, #0 //set pwm1 period     mov    PWMD1L, #20     mov    PWMD2H, #d_PWMP2 //set idle polarity     orl    PWMD2H, #0 //set pwm2 period     mov    PWMD2L, #50     mov    PWMD3H, #d_PWMP3 //set idle polarity     orl    PWMD3H, #0 //set pwm3 period     mov    PWMD3L, #50 //    orl    AUX, #d_P2PWM //Set PWM channel on P2/P4     mov    PWMC, #d_PWMCS //PWM CLK select     orl    PWMC, #d_PWMXEN //PWM channel enable     ret PWM_Disable:     //mov    IEPWM, #0 //Disable PWM Interrupt     mov    PWMC, #0 //Disable PWM Function     ret end</pre>
--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



2.8.2 PWM C語言程式範例，各通道產生的時序圖，Fosc=25MHz：



2.8.3 PWM C語言程式範例

Description	





```
Main program //=====
//
//          S Y N C M O S   T E C H N O L O G Y
//
//=====
// Device:SM59R04A2/SM59R03A2/SM59R02A2
//=====
#include "SM59R04A2.h"

void PWM_init(void)
{
    PWMD0H =0x00;           //idle =low
    PWMD0L =0x10;           //clock=16
    PWMD1H =0x00;           //idle =low
    PWMD1L =0x20;           //clock=32
    PWMD2H =0x00;           //idle =low
    PWMD2L =0x40;           //clock=64
    PWMD3H =0x80;           //idle =high
    PWMD3L =0x40;           //clock=64

// PWMMDH =0x03;           //PWM period =PWMMD +1 =1024
// PWMMDL =0xFF;           //PWM period =PWMMD +1 =128
    PWMMDH =0x00;           //PWM period =PWMMD +1 =128
    PWMMDL =0x7F;

    IE  |=0x80;             //Enable interrupt All
    IEN1 |=0x01;            //Enable interrupt PWM
    PWMC |=0x0F;            //PWM clock source=Fosc, enable all channel
}

void PWM_Int_disable(void)
{
    IEN1 &=(~0x01);        //Disable interrupt PWM
}

void PWM_interrupt(void) interrupt 8
{
    if(IRCON |=0x01)
    {
        IRCON &=(~0x01);    //Clear interrupt flag
    }
}

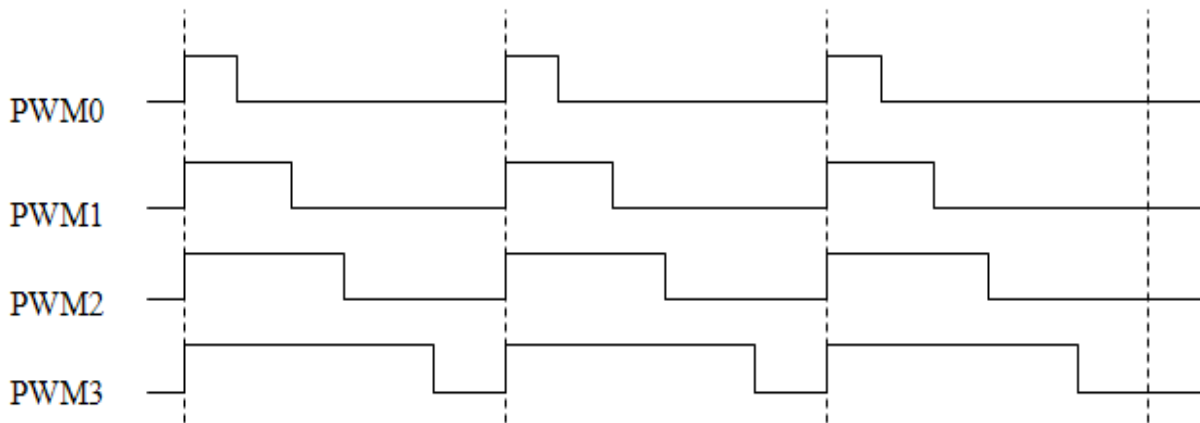
void main(void)
{
    PWM_init();

    while(1)
    {}
}
}
```



### 3 SM59R16A2/ SM59R08A2 PWM 使用說明：

- 3.1 提供 4 組獨立的 PWM 信號
- 3.2 有三組預除器；輸出頻率  $FPWM = FOSC / Div$
- 3.3 三種資料模式：8, 10, 12 bit mode
- 3.4 4 組 PWM 信號可同時動作(條件為參考頻率  $FPWM$  相同，如下圖所示)



### 3.5 Pin Description

SM59R16A2 / SM59R08A2 Package: TQFP & LQFP			
Pin	Name	I/O	Description
9	P4.0/PWM2	i/o	Bit 0 of port 4 & PWM Channel 2
10	P4.1/PWM3	i/o	Bit 1 of port 4 & PWM Channel 3
22	P1.5/PWM1/CC1	i/o	Bit 5 of port 1 & Timer 2 compare/capture Channel 1 & PWM Channel 1
23	P1.4/PWM0/CC0	i/o	Bit 4 of port 1 & Timer 2 compare/capture Channel 0 & PWM Channel 0



### 3.6 SFR 特殊控制暫存器及特殊狀態暫存器介紹：

#### 3.6.1 Special Function Register (SFR)

A map of the Special Function Registers is show as below:

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
F8	IICS	IICCTL	IICA1	IICA2	IICRWD				FF
F0	B	SPIC1	SPIC2	SPITXD	SPIRXD	SPIS			F7
E8	P4	MD0	MD1	MD2	MD3	MD4	MD5	ARCON	EF
E0	ACC								E7
D8	BRGS								DF
D0	PSW								D7
C8	T2CON		CRCL	CRCH	TL2	TH2			CF
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	C7
B8	IEN1	IP1	S0RELH	S1RELH	PWMD0H	PWMD0L	PWMD1H	PWMD1L	BF
B0	P3	PWMD2H	PWMD2L	PWMD3H	PWMD3L	PWMC	WDTC	WDTK	B7
A8	IEN0	IP0	S0RELL	ADCC1	ADCC2	ADCDH	ADCCL	CLKR	AF
A0	P2								A7
98	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL			9F
90	P1	P5	DPS	KBLS	KBE	KBF			97
88	TCON	TMOD	TL0	TL1	TH0	TH1		IFCON	8F
80	P0	SP	DPL	DPH	DPL1	DPH1		PCON	87

Note: Special Function Registers reset values and description for SM59R16A2/SM59R08A2

#### 3.6.2 Pulse Width Modulation (PWM)

Mnemonic	Description	Direct	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RESET
PWM											
PWMC	PWM Control register	B5h	-	-	PWMM[1:0]		PWM3 EN	PWM2 EN	PWM1 EN	PWM0 EN	00h
PWMD0H	PWM 0 Data register high byte	BCh	-	-	-	-	PWMD0[11:8]				00h
PWMD0L	PWM 0 Data register low byte	BDh	PWMD0[7:0]								00h
PWMD1H	PWM 1 Data register high byte	BEh	-	-	-	-	PWMD1[11:8]				00h
PWMD1L	PWM 1 Data register low byte	BFh	PWMD1[7:0]								00h
PWMD2H	PWM 2 Data register high byte	B1h	-	-	-	-	PWMD2[11:8]				00h
PWMD2L	PWM 2 Data register low byte	B2h	PWMD2[7:0]								00h
PWMD3H	PWM 3 Data register high byte	B3h	-	-	-	-	PWMD3[11:8]				00h
PWMD3L	PWM 3 Data register low byte	B4h	PWMD3[7:0]								00h

Specifications subject to change without notice, contact your sales representatives for the most recent information.



<b>Mnemonic: PWMC</b>							<b>Address: B5h</b>	
7	6	5	4	3	2	1	0	Reset
-	-	PWMM[1:0]		PWM3EN	PWM2EN	PWM1EN	PWM0EN	00h

**PWMM[1:0]: PWM 模式旗標:**

**PWM output freq. = Fosc / Div.**

PWMM[1:0]	Mode	Div.
00	8-bit mode	256
01	10-bit mode	1024
10	12-bit mode	4096
11	8-bit mode	256

here Fosc is the external crystal or oscillator frequency

**PWM3EN: PWM Channel 3 致能旗標:**

PWM3EN = 1 – PWM Channel 3 致能.

PWM3EN = 0 – PWM Channel 3 禁能.

**PWM2EN: PWM Channel 2 致能旗標:**

PWM2EN = 1 – PWM Channel 2 致能.

PWM2EN = 0 – PWM Channel 2 禁能.

**PWM1EN: PWM Channel 1 致能旗標:**

PWM1EN = 1 – PWM Channel 1 致能.

PWM1EN = 0 – PWM Channel 1 禁能.

**PWM0EN: PWM 0 Channel 0 致能旗標:**

PWM0EN = 1 – PWM Channel 0 致能.

PWM0EN = 0 – PWM Channel 0 禁能.

<b>Mnemonic: PWMD0H</b>							<b>Address: BCh</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	PWMD0[11:8]			00h	

<b>Mnemonic: PWMD0L</b>							<b>Address: BDh</b>	
7	6	5	4	3	2	1	0	Reset
PWMD0[7:0]							00h	

**PWMD0[11:0]: PWM channel 0 資料暫存器:**

可調整 PWM Channel 0 Duty Cycle

<b>Mnemonic: PWMD1H</b>							<b>Address: BEh</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	PWMD1[11:8]			00h	

<b>Mnemonic: PWMD1L</b>							<b>Address: BFh</b>	
7	6	5	4	3	2	1	0	Reset
PWMD1[7:0]							00h	

**PWMD1[11:0]: PWM channel 1 資料暫存器:**

可調整 PWM Channel 1 Duty Cycle



Mnemonic: PWMD2H							Address: B1h	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	PWMD2[11:8]				00h

Mnemonic: PWMD2L							Address: B2h	
7	6	5	4	3	2	1	0	Reset
PWMD2[7:0]								00h

**PWMD2[11:0]: PWM channel 2 資料暫存器：**  
可調整 PWM Channel 2 Duty Cycle

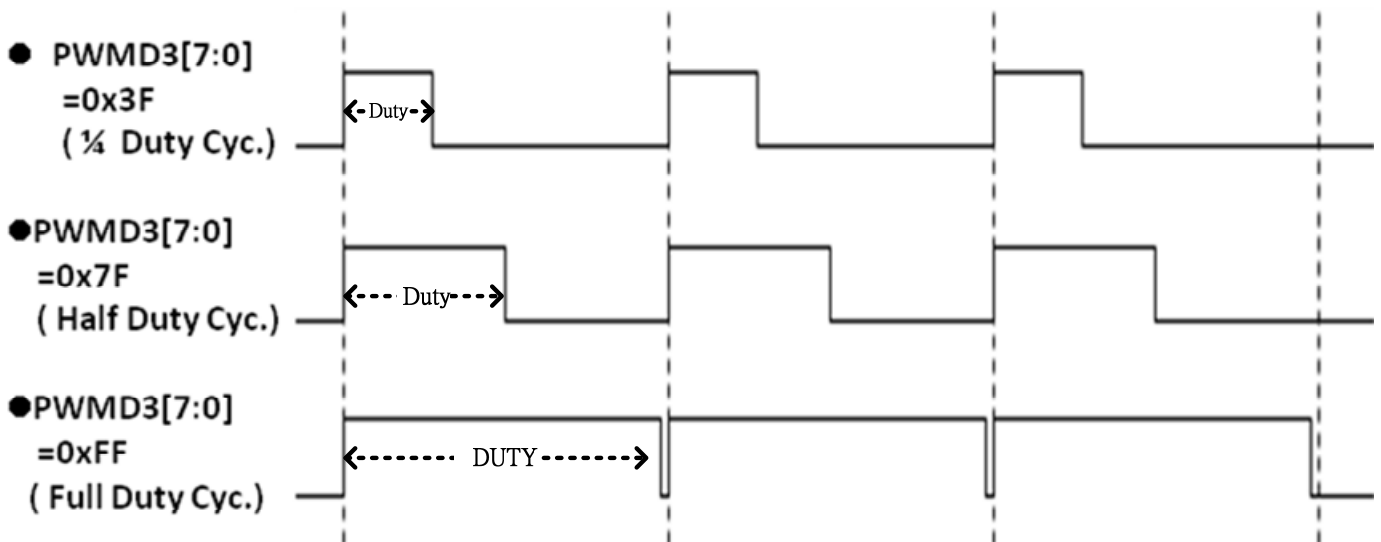
Mnemonic: PWMD3H							Address: B3h	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	PWMD3[11:8]				00h

Mnemonic: PWMD3L							Address: B4h	
7	6	5	4	3	2	1	0	Reset
PWMD3[7:0]								00h

**PWMD3[11:0]: PWM channel 3 資料暫存器：**  
可調整 PWM Channel 3 Duty Cycle

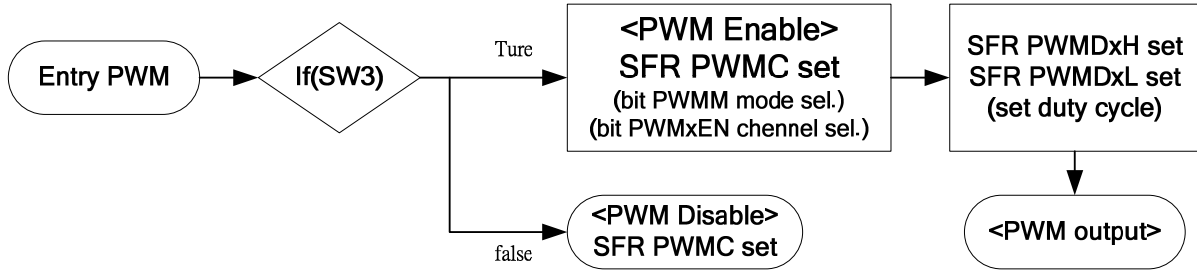
**說明：**

以PWM 8-bit模式，PWMD3資料暫存器為例，PWMD3值愈小duty cycle愈小，值愈大則反之。圖示如下：





3.7 PWM 應用流程圖：



3.8 C 語言程式範例：

Describe:	Program:
main	<pre> //===== //INCLUDE FILES //===== #include "..\h\SM59R16A2.h" #include "..\Demo\Demo.h" #include "..\PWM\pwm.h"  //===== //MAIN //===== void main() {     unsigned char Press_SW;     while(1)     {         Press_SW = ScanKeypad();         if(!Press_SW) //SW3 Press         {             // sel 8-bit mode=Fosc/256, enable channel 3,2,1,0             PWMC = 0x0F ; //Enable PWM              PWM0Duty(0x00, 0xff); //MPU Pin 23 full Cyc.             PWM1Duty(0x00, 0x7f); //MPU Pin 22             PWM2Duty(0x00, 0x3f); //MPU Pin 09             PWM3Duty(0x00, 0x1f); //MPU Pin 10         }         else         {             PWMC = 0x00; //Disable PWM         }     } // End of while(1) }  //===== void PWM0Duty(unsigned char PWM0_H, unsigned char PWM0_L) {     PWMD0H = PWM0_H;     PWMD0L = PWM0_L; } //===== void PWM1Duty(unsigned char PWM1_H, unsigned char PWM1_L) </pre>



```
{
    PWMD1H = PWM1_H;
    PWMD1L = PWM1_L;
}

//=====
void PWM2Duty(unsigned char PWM2_H, unsigned char PWM2_L)
{
    PWMD2H = PWM2_H;
    PWMD2L = PWM2_L;
}

//=====
void PWM3Duty(unsigned char PWM3_H, unsigned char PWM3_L)
{
    PWMD3H = PWM3_H;
    PWMD3L = PWM3_L;
}
```