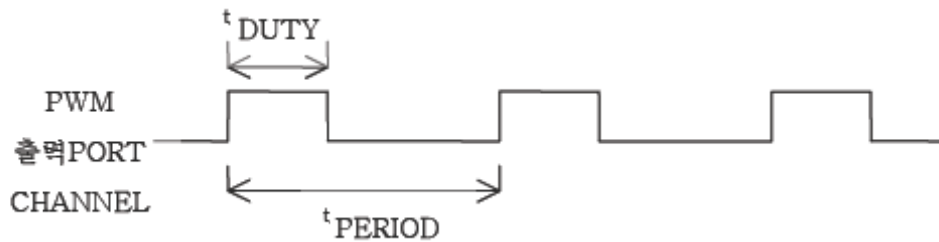


MC9S12DJ64 마이크로컨트롤러를 이용한 PWM 제어

1. PWM(Pulse Width Modulator)의 용도는 Stepper Motor의 회전속도/회전 수 제어, 혹은 LCD의 Back Light 제어 등이 있으며 전력제어 분야에 여러 용도로 응용되고 있습니다.
2. 그 일례로 이 기고에서는 LCD의 Back Light 제어를 위해 MC9S12DJ64 내의 PWM module을 어떻게 셋팅해 주고 어떻게 Control Program을 작성했는지를 제시해 봅니다.
3. PWM Module의 제어결과 나타나는 파형의 형상과 그 변화되는 모양은 다음과 같습니다.
 - A. 파형의 형상



PWM 파형에 사용되는 용어로 DUTY와 PERIOD가 있고 그 두 개의 시간축의 관계를 표현한 용어로 DUTY CYCLE이 있는데 이 DUTY CYCLE을 통상 PWM을 표현하는데 사용합니다.

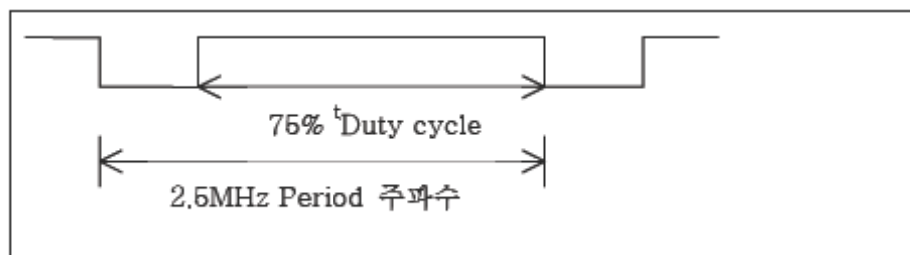
$$\text{즉 Duty Cycle} = \frac{t_{\text{Duty}} \times 100\%}{t_{\text{Period}}}$$

여기서 tDuty는 제어하고자 하는 대상에 따라 High Side의 폭이 될 수도 있고 Low Side의 폭이 될 수도 있습니다. 이 극성을 제어하기 위한 Control Register를 MC9S12DJ64 마이크로컨트롤러는 PWM Module 내에 갖고 있습니다.

tPeriod의 용도는 제어하고자 하는 대상의 제어속도와 관련이 있는데 얼마나 빠르고 세밀한 간격으로까지 제어해야 하는지에 따른 Pulse의 Width Resolution을 결정하는데 사용됩니다. 그리고 전체 제어하고자 하는 시간 폭 tPeriod의 몇 퍼센트를 tDuty가 차지하고 있는가에 의해 제어하고자 하는 제어 대상의 전력 소모량 또는 Pulse On time/Off time이 결정됩니다.

4. 예를 들어 Microcontroller의 Clock source가 10MHz이고 tDuty의 극성이 Low(Ø)인 경우 tPeriod를 4로 하고 tDuty를 1로 하면 $\frac{3}{4} \times 10\%$ 의 tDuty cycle을 갖게 되며 약 75%의 On time을 갖게 됩니다.

이때 PWM의 주파수는 Clock source: 10MHz/tPeriod:4 즉 10/4(MHz) 2.5MHz가 됩니다.



위와 같은 파형이 생성됩니다.

5. LCD Back light PWM 제어 예.

A. PWM Module 초기화

- ① LCDBKLHTPWMDISABLE; PWM Module ch2 Disable
- ② PWMPOLBIT.b.b2=1;
- ③ PWMCLKBIT.b.b2=1;
- ④ PWMPRCLKBIT.b.b6=1;
- ⑤ PWMPRCLKBIT.b.b5=0
- ⑥ PWMPRCLKBIT.b.b4=0
- ⑦ PWMCAEBIT.b.b2=0
- ⑧ PWMCTL=0X00;
- ⑨ PWMSCLB=250;
- ⑩ PWMCNT2=0X00;
- ⑪ PWMPER2=25;
- ⑫ PWMDTY2=PWMPER2-4
- ⑬ LCDBKLHTPWMENABLE;

이제 상기 초기화 내용의 한 줄 한 줄의 의미를 살펴봅니다.

① LCDBKLHTPWMDISABLE; PWM Module ch2 Disable

이 예에서는 LCD의 Back Light(조명)를 제어하기 위해 PWM Module의 Channel 2 출력을 사용하여 제어하는데 일단 PWM Module 초기화에 앞서 PWM Module CH2의 출력을 Off 시켜 놓는 부분입니다.

LCDBKLHTPWMDISABLE의 Macro 정의는 다음과 같습니다.

```
#define LCDBKLHTPWMDISABLE PWMEBIT.b.b2=0
```

여기서 PWMEBIT는 PWME라는 Special Function Register의 BIT Control을 위해 정의된 변수입니다.

PWME는 PWM Enable Register인데 그 구성은 다음과 같습니다.

내부 주소는 _PORT volatile char PWME_IO(0xa0);

PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
bit7	bit6	bit5	bit4	Bit3	Bit2	Bit1	Bit0

PWME7~0 : PWM Module CH8개의 Enable Bit를 나타냄.

각 bit가 1 : Enable 즉 PWM 출력 On(해당 CH)

각 bit가 0 : Disable 즉 PWM 출력 Off(해당 CH)

따라서 ①의 이 명령은 CH2의 PWM 출력을 Off 시킵니다.

② PWMPOLBIT.b.b2=1;

PWMPOL(PWM Polarity Register)

이 레지스터는 각 PWM Channel 파형의 시작하는 부분의 극성을 결정해주는



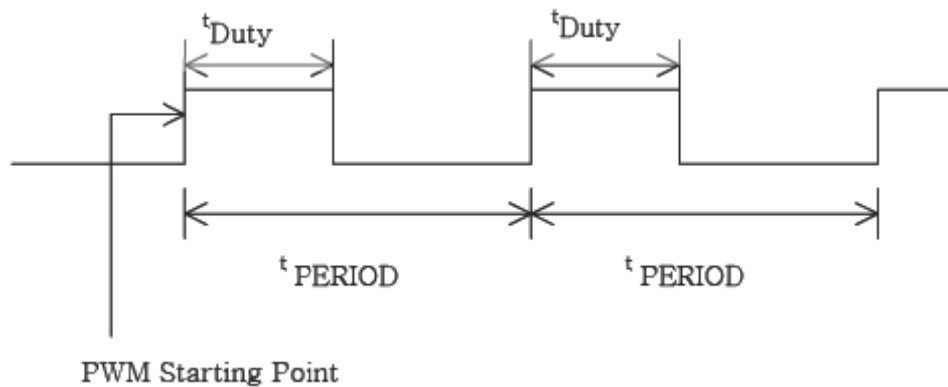
역할을 갖고 있습니다.

그 구성을 보면 다음과 같습니다.

`_PORT volatile char PWMPOL_IO(0xa1);`

PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
bit7	bit6	bit5	bit4	Bit3	Bit2	Bit1	Bit0

각 bit 마다 PWM 출력의 해당 Channel의 극성을 제어하는 역할을 갖고 있습니다. PPOLZ bit를 "1"로 하면 즉 "PWMPOLBIT.b.b2=1;"로 Program하면 PWM 출력 Channel Z의 시작 Pulse 극성은 High가 되고 tDuty count에 도달되고 난 후 Low로 됩니다.



`PWMCLKBIT.b.b2=1;`

PWMCLK(PWM Clock Select Register)

각각의 PWM Channel들을 2개의 Clock Source 중 하나를 선택하여 PWM 출력 파형을 생성하는데 사용합니다. 여기에서 선택을 할 수 있도록 해주는 Control Register가 PWMCLK이고 그 구성은 다음과 같습니다.

주소 define:

`_PORT volatile char PWMCLK_IO(0xa2);`

구성 bit:

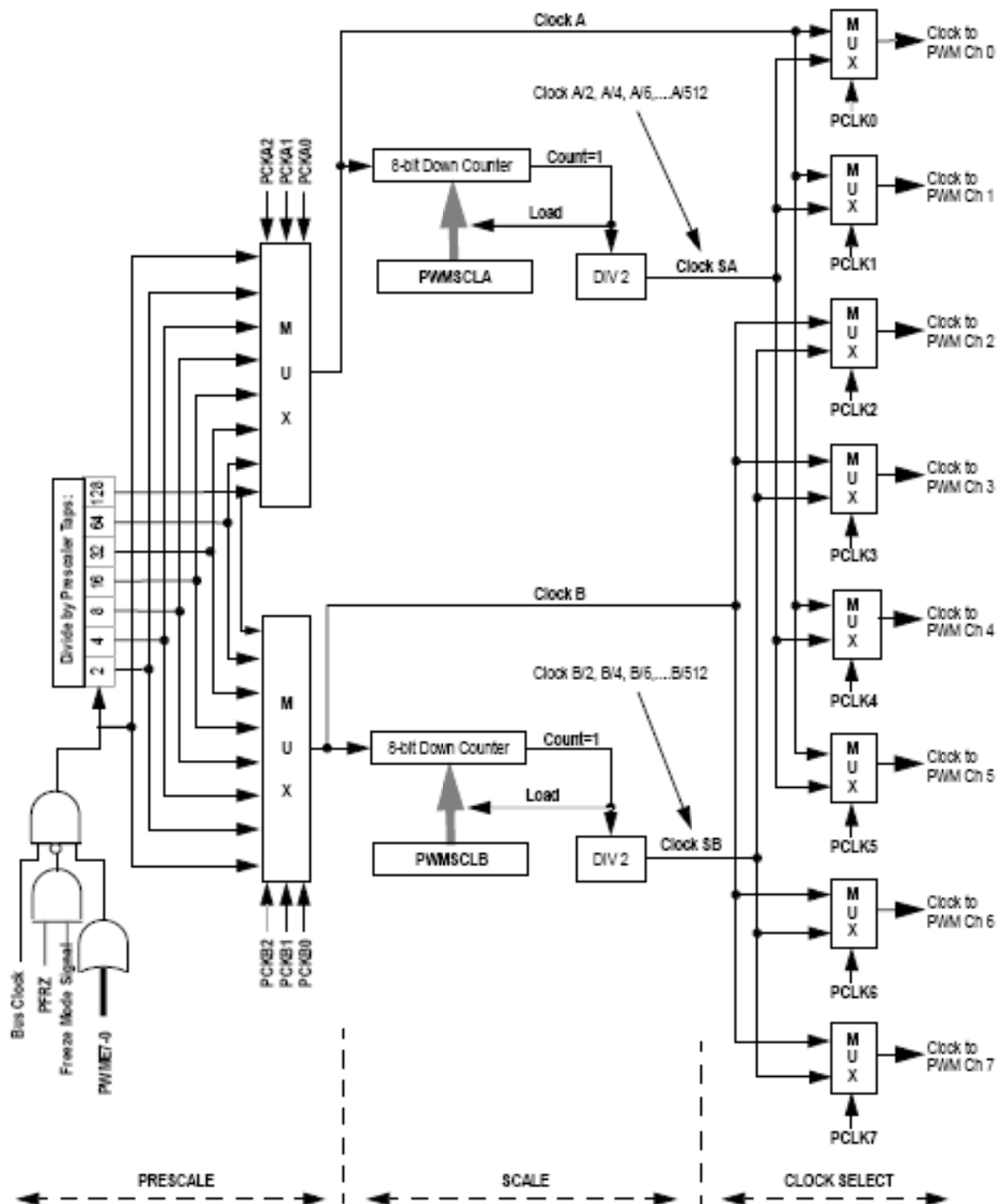
PCLK7	PCLK6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
bit7	bit6	bit5	bit4	Bit3	Bit2	Bit1	Bit0

이 각각의 bit는 PWM Output Channel 8개를 각각 제어하며 아무 때고 이 각각의 bit 값을 변경할 수 있으며 이 변경된 값에 따라 그 즉시 Clock Source가 바뀌고 이에 따른 PWM 출력 파형의 Period가 바뀝니다.

그러나 PWM Signal이 생성 되어지고 있는 가운데 이 Clock Source 선택 bit를 조작 변경하면 생성되던 PWM Pulse 중 일부분이 High에서 Low 또는 그 반대로

그러면 선택할 수 있는 Clock Source 는 어떤 것이 있고 그 선택 구조는 어떻게 되어 있는지 그 구조를 살펴보면 다음과 같습니다.

Bus Clock(E Clock이라고 명명함)을 직접 분주하여 생성된 Clock과 이 분주된 Clock을 다시 또 분주하여 생성되는 Clock이 바로 그 두 가지의 Clock Source입니다.



그림을 보면 Clock A, Clock B 그리고 Clock SA, Clock SB라는 Clock Source들이 있는데 이 4가지의 Clock들이 바로 PWM Module Channel들의 Clock Source가 됩니다.

그림에서 Clock A나 Clock B는 Bus Clock(E Clock)과 동일하거나 아니면 x2배~x128배의 분주기를 통해 출력되거나 합니다.

그림에서 Divide by prescaler taps라고 하는 것은 E Clock을 분주하기 위한 Prescaler를 의미 합니다.

이 분주를 위해 준비된 PWM Module의 Control Register가 PWMPRCLK라고 하는 Register입니다.

이 Clock Source의 초기화하는 것은 ④⑤⑥⑨번에서 다시 논의하기로 하고 우선 ③번의 PWMCLKBIT.b.b2=0; 으로 했을 시 PWM Module에 어떠한 변화가 오는지를 보면 다음과 같습니다.

PWMCLK는 PWM의 Source Clock을 Select하는 Register인데 b2 즉 PCLK2 bit를 1로 하면 Clock SB가 PWM Channel 2를 위한 Source Clock으로 선택 되도록 한다는 것입니다.

따라서 LCD BACK light의 PWM Control을 위해 PWM Module의 Channel 2에 Clock Source로는 E Clock의 Source를 분주해서 사용하는 Clock B를 PWMSCLB로 또 분주하여 생성되는 Clock SB가 선택 되었습니다.

이제 ④⑤⑥번 Line을 한꺼번에 살펴보면 모두다 PWMPRCLK Register의 bit들을 조작 변경하는 것을 알 수 있는데 이로 인한 변화를 보면 다음과 같습니다.

우선 PWMPRCLK의 bit6~bit4까지의 논리값이 100 이므로 Bus Clock/16의 Prescaler를 선택하게 되므로 8MHz의 Bus Clock을 사용하는 경우 500KHz의 Clock Source를 PWM 제어를 위해 사용하게 됩니다. 이와 같은 값은 아래 Table을 참조하여 계산합니다.

PWMPRCLK의 Bit 구성은

-PORT Volatile char PWMPRCLK_IO(0xa3) 내부주소.

0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0



Bit6	Bit5	Bit4	Clock B의 값
PCKB2	PCKB1	PCKB0	
0	0	0	Bus clock
0	0	1	Bus clock/2
0	1	0	Bus clock/4
0	1	1	Bus clock/8
1	0	0	Bus clock/16
1	0	1	Bus clock/32
1	1	0	Bus clock/64
1	1	1	Bus clock/128

Table 1 Clock B Prescaler Selects

Bit6	Bit5	Bit4	Clock B의 값
PCKA2	PCKA1	PCKA0	
0	0	0	Bus clock
0	0	1	Bus clock/2
0	1	0	Bus clock/4
0	1	1	Bus clock/8
1	0	0	Bus clock/16
1	0	1	Bus clock/32
1	1	0	Bus clock/64
1	1	1	Bus clock/128

Table 2 Clock A Prescaler Selects

이제 이 Clock B를 재 분주하여 Clock SB를 생성하면 다음과 같이 됩니다.

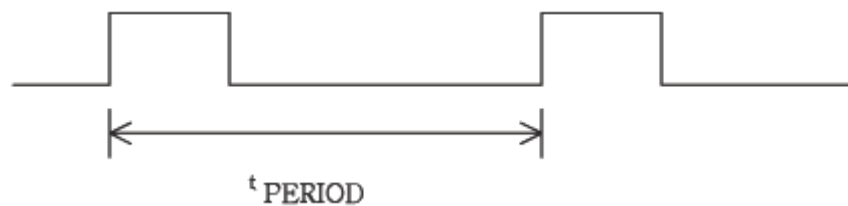
이것은 ⑨ PWMSCLB=250; Line의 실행에 의해 생성됩니다.

결과는 Clock SB= Clock B/(2 x PWMSCLB)

$$= 500\text{KHz} / (2 \times 250) =$$

$$= 500 / 500\text{KHz} = 1\text{KHz}$$

즉



tPERIOD의 기본 Frequency가 1KHz 인 것입니다.

⑪번의 PWMPER2 Register의 값에 의해 $t_{PERIOD} = 1(KHz)/PWMPER2$ 가 됩니다
즉 PWMPER2가 25이면 $t_{PERIOD} = 400(Hz)$ 가 됩니다.

결론적으로

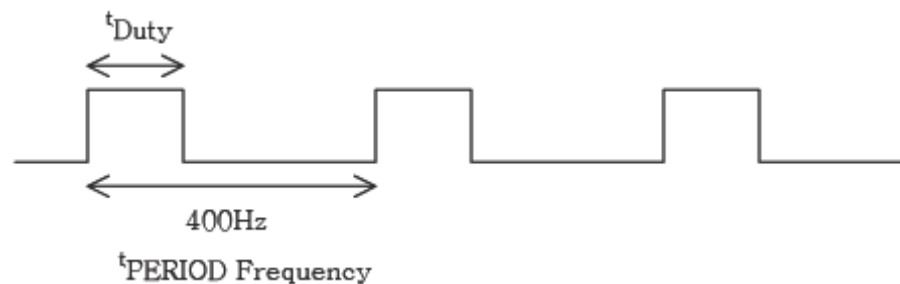
16(MHz)의 Oscillator를 사용하는 경우 E Clock 즉 Bus Clock은 Microcontroller의 기본 분주 2분주를 거쳐서 8(MHz)가 되고 이 8(MHz)의 주파수를 PWMPRCLK(PWM Prescaler Clock Select Register)로 분주하고 만일 PWMCLK의 해당 CH Bit가 0으로 되어 있으면 이 8(MHz)를 분주한 Clock B 혹은 A를 PWM Module의 clock source로 쓰게 되며 PWMCLK의 해당 CH Bit가 1로 되어있으면 이 8(MHz)를 분주한 Clock B 혹은 A를 다시 PWMSCLB 혹은 PWMSCLA PWM Scale Register에 의해 분주된 Clock SB 혹은 Clock SA를 PWM의 Clock Source로 사용하게 됩니다

그리고 이 Clock Source를 PWMPERX(PWM Channel Period Register)의 값으로 나누어서 결국 tPeriod의 Frequency를 결정지을 수 있게 됩니다.

이 일련의 과정을 예를 들어 그림으로 그려보면 다음과 같습니다.

예제)

일단 Micom Master Oscillator는 16MHz, 출력되는 PWM Channel은 PWM CH2로 하고 출력 파형은 다음과 같이합니다.



PWM CH2의 출력 Duty Cycle은 75%가 되게 합니다.

(해답의 흐름)

우선 Clock Source의 흐름

