**전자회로실험 보고서**



학과 전자공학과

학번 1218165

이름 김재욱

제출일 2020-11-11

**실습 이론**

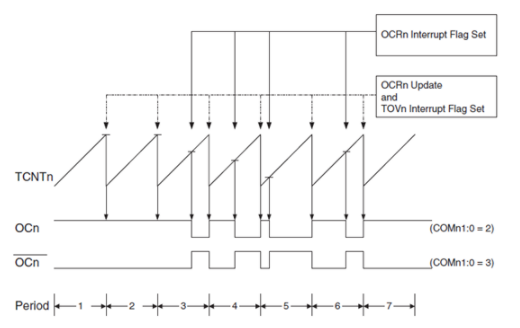
timer/counter 정리

timer/counter는 총 4개로 이뤄져 있으며 0/2는 8bit 1/3은 16비트 타이머이다. 8bit의 경우 overflow가 총 256 번 발생한다. 모드 또한 4가지인데 Normal 모드, CTC 모드, Fast PWM 모드, Phase Correct PWM 모드가 존재한다.

timer/counter에서 pwm모드 동작법

pwm 에서는 OCR이라는 비교 매치가 사용된다. (timer/counter2기준) 이때 pwm 동작 모드에 따라 달라지며 비반전,반전에따라 달라지므로 각 경우를 나눠서 설명한다.

Fast pwm

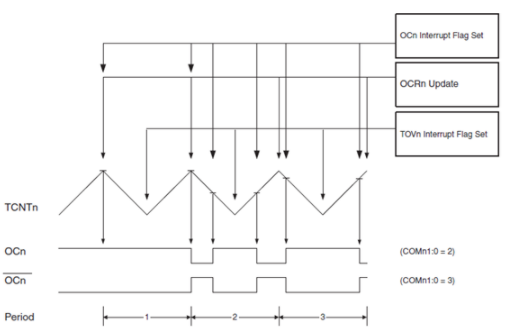


TCNT2는 0부터 255까지 세면서 오버플로우 되는 것을 반복한다. 0xff에서 0x00 넘어가서 다시 0xff까지 센다.OCR2의 값을 0~255R까지 설정하면 TCNT2값을 OCR2의 값과 비교하여 OC2의 출력을 결정한다.

비반전 비교 출력 모드에서는 OC2핀이 HIGH로 출력되다가 비교해 일치하면 LOW가 출력되고 TCNT2가 다시 0x00이 될 때 HIGH가 출력된다.

반전 비교 출력모드에서는 OC2핀이 LOW 로 출력되다가 비교해 일치하면 HIGH가 출력되고 TCNT2가 다시 0x00이 될 때 LOW가 출력된다.

Phase correct pwm



TCNT2는 0부터 255까지 세면서 오버플로우 되는 것을 반복한다 fast pwm과 다르게0xff에서 0x00으로 줄어들면서 0x00에 도달한다.

비반전 출력모드에서는 업카운트 중에  TCNT2=OCR2 이면 OC2에 LOW 가 출력되고 다운카운트 중에 비교 일치하면 HIGH가 출력된다.

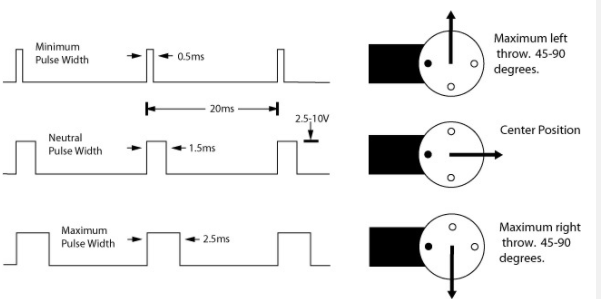
반전 출력모드에서는 업카운트 중에  TCNT2=OCR2 이면 OC2에 HIGH가 출력되고 다운카운트 중에 비교 일치하면 LOW 가 출력된다.

**Clear Timer on Compare Match (CTC) 모드**

TCNT2는 0부터 255까지 세면서 OCR2와 비교 일치하면 0x00으로 set되고TIFR레지스터의 OCF2비트는 1로 설정되면서 인터럽트가 발생한다. 그리고 해당되는 인터럽트 핸들러 벡터(루틴)가 실행될 때 하드웨어에 의해서 OC20는 0으로 클리어된다.

서보모터

서보모터는 PWM 신호를 기반으로 동작한다. 50Hz의 PWM 주파수를 가지므로 20ms의 신호를 필요로 한다.



위의 그림을 통해 0도는 20ms중 0.5ms만 활성화되어야하며 90도는 1.5ms, 180도는 2.5ms를 필요로 한다.

**실습내용 및 결과**

실습1

|  |
| --- |
|  |
| #define *F\_CPU* 16000000  #include <avr/io.h>  #include <avr/interrupt.h>  // CTC Mode  volatile unsigned char led = 0x0f; // 초기 led  volatile unsigned int count = 0;  ISR(TIMER2\_COMP\_vect){  count++; // 인터럽트 횟수 카운트  if(count == 125){ // 125번째 인터럽트시 (1초)  PORTF = (~PORTF);// 반전 출력  count = 0; // 카운트 초기화  }  }  int main(){  DDRF = 0xff; // LED 포트를 출력으로 설정  TCCR2 = 0b00001101; // CTC모드, 프리스케일러 분주비 1024  TIMSK = 0b10000000; // 타이머2의 비교일치 인터럽트 설정  OCR2 = 124; // 비교일치 값 지정  TCNT2 = 0; // 초기값  SREG = 0x80; // 전역 인터럽트 허용  PORTF = led;  while(1){  }  return 0;  } |
| 결과를 확인 하였고 |

실습2

|  |
| --- |
|  |
| #define *F\_CPU* 16000000UL  #include <avr/io.h>  #include <avr/interrupt.h>  #include <math.h>  #include <util/delay.h>  #include <stdio.h>  char buffer[5];  int bufindex = 0;  int angle =0;  int on\_time = 5; //0도에 해당  int off\_time = 195; //0도에 해당  volatile unsigned int count = 0;  volatile unsigned char rx\_data;  volatile unsigned int mode =0;  void reset(){  for(int i= 0; i<5;i++){  buffer[i] =*NULL*;  }  bufindex=0;  }  ISR(TIMER2\_COMP\_vect){// OCR에 의한 비교일치가 한번 할때마다 0.1ms 걸림(8)  if (count == on\_time && mode==1){ //활성화  mode = 0;  count=0;  }  if(count == off\_time && mode ==0){ //비활성화  mode = 1;  count=0;  }  count++;  }  ISR(USART0\_RX\_vect){  rx\_data = UDR0;  if(rx\_data =='e'){ //enter(를 문자 'e'로 간주)가 입력되면  angle = *atoi*(buffer); // 배열을 숫자로 변환  reset(); // null로 초기화  on\_time = map(angle); // 활성화 시간으로 바꾸기  off\_time=200-on\_time;  }  else{  buffer[bufindex] = rx\_data;  bufindex++;  }  UDR0 =rx\_data;  }  // 0~180 도 이면 0도에 0.5ms ,180도에 2.5ms 이므로 5~25로 변환 해야함  int map(int x,int in\_min,int in\_max,int out\_min,int out\_max){  return(x-in\_min)\*(out\_max-out\_min)/(in\_max-in\_min)+out\_min;  }  int main(void)  {  DDRB |= (1<<DDB7); //LED용  DDRD =0xff;  //프리스케일러로 8를 선택  TCCR2 |= (1<<CS21);  //TCCR2 |= (1<<CS20);    //FAST PWM 모드, OCR 범용 I/O  TCCR2 |= (1<<WGM01);  TCCR2 |= (1<<WGM00);    TCCR2 |= (1<<COM01);    //카운터 초기화  TCNT2 = 0;  OCR2 =150;      UCSR0A = 0x00;  UCSR0B = 0b10011000; // RXCIE0=1, TXEN0=1, RXEN0=1 ==> 수신완료 인터럽트 허가, 수신허가.송신허가  UCSR0C = 0b10000110; // 비동기 통신, 패리티 없음, ==>고정  // 데이터 비트 : 8비트, 정지비트 : 1비트  UBRR0H = 0; //보오 레이트 레지스터  UBRR0L = 103; //보오 레이트 레지스터, fosc=14.7456MHz, BAUD=9600bps,    SREG = 0x80; // 전역 인터럽트 허용    /\* Replace with your application code \*/  while (1)  {  if(mode ==1){  PORTB = 0b00000001;  }  else if(mode ==0){  PORTB = 0b00000000;  }  }  } |
| 시리얼 통신을 통해 angle이 바뀌는 것을 확인 했으나 서보모터가 동작하지 않았다. |

**고찰**

코드 설명을 간략히 하자면 먼저 20ms의 주기를 만들기 위해 on\_time과 off\_time을 만들어 on\_time+off\_time =200이 되도록 하였다. 프리스케일러 분주비가 8이므로 8bit에 해당하는 timer2 는 256번 마다 오버프로우가 발생하여 인터럽트 된다. 즉 한번 인터럽트 될때마다 1/(16000000/8)\*256초 즉 0.1ms가 걸린다. 그래서 20ms 주기를 위한 on\_time+off\_time =200이다.

USART 인터럽트 서비스 함수에서 서보모터의 각도를 받아야하기 때문에 최대 3자리까지 숫자를 입력받아야한다. 일반적인 시리얼통신에서는 한번에 하나의 숫자만 받을수 있으므로 숫자가 모두 입력될때까지 이전 숫자를 버퍼링 해야한다. 그리고 atoi 함수(char 배열을 숫자로 변환하는 함수)를 통해 angle에 각도 값을 저장한다.

그리고 0~180 범위의 각도를 출력하기 위해서는 on\_time은 5~25사이로 변해야하며 map함수를 작성하여 angle값을 on\_time으로 바꾸도록 하고 off \_time은 200 -on\_time을 통해 저장한다.

Timer 인터럽트 서비스 함수에서 count 가 on\_time ,off\_time이 될때마다 mode를 바꿔서 main 문에서 동작 할 수 있게 한다.

main문의 동작은 mode =0 과 mode =1에 의해 구분된다. Mode=0은 off\_time동안 서보모터에 0V를 주기 위함이고 mode =1은 on\_time동안 서보모터에 5V를 주기 위함이다.