|  |  |  |  |
| --- | --- | --- | --- |
| **10주차 결과보고서** | | | |
| **과목명** | 전자회로실험2 | **분반** | 005 |
| **학과** | 전자공학과 | **이름** | **이진호** |
| **제목** | ATmega128 PWM | | |

**1. 코드 작성**

(1)

과제 1

|  |
| --- |
| #define F\_CPU 16000000UL  #define BAUD 9600  #define MYUBRR F\_CPU/16/BAUD-1  #include <avr/io.h>  #include <util/delay.h>  #include <string.h>  //USART 초기화 함수  void USART\_init(unsigned int ubrr)  {  UBRR0H = (unsigned char)(ubrr >> 8); //0으로 초기화  UBRR0L = (unsigned char) ubrr; //그대로 출력  UCSR0B = (1 << RXEN0) | (1 << TXEN0); //송수신 가능  //0000 0000 0000 0011 -> 0000 0000 0000 1100 char size 12로 설정  UCSR0C = (3 << UCSZ0);  }  //USART 데이터 전송 함수  void USART\_Transmit(char data)  {  while (!(UCSR0A & (1 << UDRE0))); //입력이 안들어오면 반복  UDR0 = data; //UDR0에 데이터 저장  }  //USART 문자열 전송 함수  void USART\_Transmit\_String(char \*str)  {  //문자열이 NULL이 될때 까지  while( \*str !='\0')  {  USART\_Transmit(\*str++); //문자 출력  }  }  //USART 데이터 수신 함수  char USART\_Receive()  {  while (!(UCSR0A & (1 << RXC0))); //입력이 안들어오면 반복  return UDR0; //UDR0 반환  }  void USART\_Receive\_String(char \*str)  {  int i = 0;  for(i = 0; str[i-1] != '\r' ;i++)  {  str[i] = USART\_Receive();  }  str[i] = '\0';  }  void Timer\_Init(){  TCCR0 =( 1 << WGM00 ) | (1 << WGM01 ) | ( 1 <<COM01) | ( 4 << CS0);  //Fast PWM 모드 설정  //5번 1번 => 비반전 비교 출력 모드 , CS0 에 4번 초기화  TIMSK = (1 << OCIE0);  }  int main(void)  {  char str[3]; //문자열 저장  int num[3]; //숫자 저장  int result = 0;  int n = 1;  USART\_init(MYUBRR); //USART 초기화  Timer\_Init(); //Timer 초기화  DDRB = 0xFF; //B를 출력 포트로 설정  int i;    while (1)  {  result = 0; //result 입력값 초기화  str[3] = NULL; //문자열 초기화  n=1; //n값 초기화  USART\_Receive\_String(str); //문자열 입력  USART\_Transmit\_String(str); //USART에 문자열 출력  int length = strlen(str); //문자열 길이 저장  //숫자를 1의자리 수부터 문자열 길이 만큼 순서대로 저장  for(i=0;i<length;i++){  num[length-i-1] = str[i]-48;  }  //1의 자리 수부터 역으로 가져와 result에 입력값 저장  for(i=0;i<length;i++){  result += n\*num[i];  n\*=10;  }  \_delay\_ms(1000); //delay 1000ms  OCR0 = result; //result의 값에 따른 모터 이동  }  } |
| ﻿과제 2  CTC 모드란    Normal 모드에서는 TCNT0가 오버플로우가 되었을때 인터럽트가 발생하는 반면, CTC(Clear Timer on Compare Match) 모드는 OCRnA의 레지스터에 원하는 값을 넣게되면 TCNT0가 OCRnA의 값만큼 증가된 뒤 인터럽트(Compare Match Interrupt)가 발생하게 된다. 즉 사용자가 원하는 값으로 인터럽트를 일으킬 수 있는 장점이 있다.  CTC 에서는 TCNT0 == OCRnA 가 되었을 때 TCNT0 = 0으로 되므로, ISR초기화 필요 없이 compare match interrrupt를 이용하여 구현을 하게 된다.  ﻿ |

**2. 고찰**

이번 실험은 아트메가128을 이용하여 pwm에 관해서 진행하였다.

먼저 예제를 이용하여 간단하게 pwm의 동작 원리에 대해서 알아보았다. TCCR0에서 PWM모드를 설정하고, OCR0의 레지스터에 들어가는 값에 따라 모터가 동작하는 원리 였다. 이를 이용해 사용자에게 USART 통신을 통해서 PWM OCR 레지스터 값을 입력 받아 모터를 이동시키는 프로그램을 구현해 보았다. 이는 먼저 USART에서 입력 받는 값은 char형이기 때문에 이를 문자열로 저장하는 함수를 만들었고, 문자열로 저장한 값을 다시 정수형으로 변환하여 저장하는 정수형 변수를 만들었다. 그리고 여기에 문자열로 저장한 값을 역순으로 정수형 변수에 저장한 후, 전체값을 숫자로 변환하여 OCR0의 레지스터에 값을 입력하도록 구성하였다.