|  |  |  |  |
| --- | --- | --- | --- |
| **4주차 결과보고서** | | | |
| **과목명** | 전자회로실험2 | **분반** | 005 |
| **학과** | 전자공학과 | **이름** | **이진호** |
| **제목** | ATmega128 USART LCD | | |

**1. 코드 작성**

(1)

과제 1

|  |
| --- |
| #define F\_CPU 16000000UL //16MHz 설정  #define BAUD 9600 //9600bfs 설정  #define MYUBRR F\_CPU/16/BAUD-1 //UBRR 계산  #define COMPARE\_MATCH 1 //타이머 모드 설정 (Overflow, CompareMatch)  #include <avr/io.h>  #include <avr/interrupt.h>  #include <util/delay.h>  unsigned int msec, sec; //타이머 글로벌 변수  //Over flow  #ifdef OVER\_FLOW  void Timer\_Init(){      TCCR0 = (4<<CS0);   //프리스케일러의 타이머 0에서 64로 설정      TIMSK = (1<<TOIE0); //오버플로우 인터럽트 활성화      TCNT0 = 6;  //포인트 카운터의 스타팅 포인터를 6으로 지정  }  //타이머 오버플로우의 인터럽트  ISR(TIMER0\_OVF\_vect){      msec++; //ms 증가      //ms가 1000이 될때      if(msec == 1000){          msec = 0; //msec를 0으로 초기화          sec++; //sec 증가      }      //sec가 100이 될때      if(sec == 100){          sec = 0; //sec 0으로 초기화      }      TCNT0 = 6;  //TCNT0 초기화  }  //Compare Match  #elif COMPARE\_MATCH  void Timer\_Init(){      TCCR0 = (4<<CS0);      TIMSK = (1 << OCIE0);      OCR0 = 249; //컴페어 매치를 250으로 만들때 카운트 증가시킨다.  }  //컴페어 매치의 인터럽트  ISR(TIMER0\_COMP\_vect){      msec++; //ms 증가      //ms가 1000이 될때      if(msec == 1000){          msec = 0; //msec를 0으로 초기화          sec++; //sec 증가      }      //sec가 100이 될때      if(sec == 100){          sec = 0; //sec 0으로 초기화      }  }  #endif  // 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      }  }  int main(void){      USART\_Init(MYUBRR); //USART 초기화      Timer\_Init(); //타이머 활성화        DDRF = 0xFF;    //출력 포트 F로 설정      SREG = 0x80;    //전역 인터럽트 활성화      int position = 0;        while(1){          USART\_Transmit\_String("Timer : ");          USART\_Transmit(sec/10 + 48);    //10의 자리수 아스키코드로 변환해서 USART 출력          USART\_Transmit(sec%10 + 48);    //1의 자리수 아스키코드로 변환해서 USART 출력          USART\_Transmit('\r');   //명령어 종료          PORTF = position++; //PORTF의 LED핀에 출력 후 숫자 1씩 증가            \_delay\_ms(300); //딜레이 300ms      }  } |
| 과제 2  #define F\_CPU 16000000UL //16MHz 설정  #define BAUD 9600 //9600bfs 설정  #define MYUBRR F\_CPU/16/BAUD-1 //UBRR 계산  #define COMPARE\_MATCH 1 //타이머 모드 설정 (Overflow, CompareMatch)  #include <avr/io.h>  #include <avr/interrupt.h>  #include <util/delay.h>  unsigned int msec, sec; //타이머 글로벌 변수  //Over flow  #ifdef OVER\_FLOW  void Timer\_Init(){      TCCR0 = (4<<CS0);   //프리스케일러의 타이머 0에서 64로 설정      TIMSK = (1<<TOIE0); //오버플로우 인터럽트 활성화      TCNT0 = 6;  //포인트 카운터의 스타팅 포인터를 6으로 지정  }  //타이머 오버플로우의 인터럽트  ISR(TIMER0\_OVF\_vect){      msec++; //ms 증가      //ms가 1000이 될때      if(msec == 1000){          msec = 0; //msec를 0으로 초기화          sec++; //sec 증가      }      //sec가 100이 될때      if(sec == 100){          sec = 0; //sec 0으로 초기화      }      TCNT0 = 6;  //TCNT0 초기화  }  //Compare Match  #elif COMPARE\_MATCH  void Timer\_Init(){      TCCR0 = (4<<CS0);      TIMSK = (1 << OCIE0);      OCR0 = 249; //컴페어 매치를 250으로 만들때 카운트 증가시킨다.  }  //컴페어 매치의 인터럽트  ISR(TIMER0\_COMP\_vect){      msec++; //ms 증가      //ms가 1000이 될때      if(msec == 1000){          msec = 0; //msec를 0으로 초기화          sec++; //sec 증가      }      //sec가 100이 될때      if(sec == 100){          sec = 0; //sec 0으로 초기화      }  }  #endif  // 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      }  }  int main(void)  {      USART\_Init(MYUBRR); //USART 초기화      Timer\_Init(); //타이머 활성화      int save\_sec = 0;      int flag1 = 0;  //버튼 1 flag 변수      int flag2 = 0; //버튼 2 flag 변수      DDRC = 0x00;   //입력 포트 C로 설정      SREG = 0x80; //전역 인터럽트 활성화      while (1)      {          //버튼 1을 누를때          if ((PINC & 0x01) == 0x01)          {              //flag1의 상태가 0일때              if (flag1 == 0)              {                  flag1 = 1; //flag1을 1로 전환              }              //flag1의 상태가 1일때              else if (flag1 == 1)              {                  flag1 = 2;  //flag1을 2로 전환              }              //flag1dml 상태가 2일때              else              {                  flag1 = 1; //flag1을 1로 전환              }              \_delay\_ms(500); //500ms 딜레이          }          //버튼 2를 누르면 초기화          else if ((PINC & 0x02) == 0x02)          {              sec = 0; //시간 초기화            }          //버튼 3          else if ((PINC & 0X04) == 0X04)          {              //flag2의 상태가 0일때              if (flag2 == 0)              {                  flag2 = 1; //flag2를 1로 전환              }              //flag2의 상태가 1일때              else              {                  flag2 = 0; //flag2를 0으로 전환              }              \_delay\_ms(500); //500ms 딜레이          }          //스위치1 첫번째 상태          //타이머 초기화          if (flag1 == 0)          {              sec = 0;          }          //스위치1 두번째 상태          //타이머 시작          else if (flag1 == 1)          {              //스위치3 첫번째 상태              if (flag2 == 0)              {                  USART\_Transmit\_String("Timer : ");                  USART\_Transmit(sec/10 + 48);    //10의 자리수 아스키코드로 변환해서 USART 출력                  USART\_Transmit(sec%10 + 48);    //1의 자리수 아스키코드로 변환해서 USART 출력                  USART\_Transmit('\r');   //명령어 종료                  save\_sec = sec; //save\_sec의 현재 시간 저장              }              //스위치3 두번째 상태 (첫번째 상태 출력 유지)              else if (flag2 == 1)              {              }          }          //스위치1 세번째 상태          //타이머 정지          else if (flag1 == 2)          {              sec = save\_sec; //save\_sec에 저장되어있던 시간을 sec에 저장          }          \_delay\_ms(300); //300ms 딜레이      }  } |

**2. 고찰**

이번 실험은 아트메가128을 이용하여 타이머를 USART에 출력하는 것과 이를 응용하는 실험을 진행하였다. 먼저 첫번째 실험은 타이머를 이용하여 시간을 측정하고, 1초마다 LED가 하나씩 새로 켜지는 프로그램을 작성하는 것이였다. 이는 간단하게 타이머 초기화 함수를 이용하여 초기화 설정을 하고, int형 값을 LED핀에 출력을 하고 이를 1씩 증가시키게 하면 1초마다 LED가 하나씩 새로 켜지게 할수 있기 때문에 간단하게 해결하였다. 문제는 두번째 실험이었는데, 스톱워치 프로그램을 작성하는데, 스위치 1을 누르면 시간이 멈추거나 흐르고, 스위치 2를 누르면 시간이 초기화 스위치 3을 누르면 랩 타임 기능을 설정하는 프로그램을 작성하는 것이었는데, 스위치 2는 스위치를 누를때 sec를 0으로 초기화 하면 되는 것으로 해결하였고, 스위치 1은 flag를 3단계로 나누어 1번은 시작 버튼을 누르기 전, 2번은 시작버튼을.눌렀을 때, 3번은 정지버튼을 눌렀을 때로 기능을 분리해서 구현하였다. 2번의 기능에서는 시간이 1초 지날 때 마다 현재의 시간을 save\_sec에 저장하였고, 3번의 정지 기능에서 sec에 save\_sec의 값을 넣어주어서 정지의 기능을 동작하도록 하였다.마지막으로 스위치3을 누를 때는 스탑워치가 시작일 때 동작하도록 하였고, 이 역시 flag를 2단계로 나누어 1단계는 출력, 2단계는 출력하지 않도록 코드를 구현하였다. 다만 구현하는 도중 동작을 하지않아서 확인해보니 스위치의 입력을 0X03으로 하였고 이를 0X04로 바꾸어 동작을 하니 해결하였다. 다음부터는 스위치 입력을 구현할 때 제대로 확인하고 짜야겠다.