**REPORT**

**| Embedded System**

**| Programming on an embedded systems 2**

**| 컴퓨터정보공학과**

**| 12141540 박영창**

# 내용

10주차 실습에서는 임베디드 보드의 스위치/버저의 사용 및 인터럽트에 대해 배웠습니다.

* 스위치
  + 두 개의 스위치가 각각 입출력포트 PE4, PE5에 할당되어 있다.
  + 눌려지지 않았을 때는 1, 눌려져 있을 때는 0이 되도록 설계되어 있다.
  + DDRE를 초기화함으로 스위치 입력을 설정할 수 있다.
  + PINE값을 통해 스위치의 현재 상태를 알 수 있다.
* 버저
  + 버저는 PB4 포트에 의해 제어된다.
  + DDRB를 출력으로 사용하기 위해 0x10으로 지정한다.
  + PORTB에 값을 설정함으로 소리가 나고, 0으로 설정함으로 소리가 멈춘다. 이 주기를 통해 파형이 결정되고 음이 결정된다.
  + 각 음계의 주파수를 통해 설정할 값을 구할 수 있다.
* 인터럽트
  + ATmega128 인터럽트

1. 종류
   * + - 차단 가능한 외부 인터럽트
       - 리셋 포함 총 35개의 인터럽트 벡터를 가짐 ( 외부 핀을 통한 인터럽트는 8개 )
2. 인터럽트 벡터
   * + - 리셋이 가장 우선순위가 높다
       - 최하위 주소에 있는 벡터는 최상위 주소에 있는 벡터에 비해 우선순위가 높다.
3. 인터럽트 트리거
   * + - Edge trigger : 입력 신호가 변경되는 순간을 트리거로 사용
         * 하강 에지 ( falling edge ) : ‘1’ => ‘0’ 변경 시점 트리거
         * 상승 에지 ( rising edge ) : ‘0’ => ‘1’ 변경 시점 트리거
       - Level trigger : 입력 신호가 일정 시간동안 원하는 레벨을 유지하면 트리거로 사용
4. 레지스터
   * + - SREG : ALU의 연산 후 상태와 결과를 표시하는 레지스터 ( 인터럽트 경우에는 제어 레지스터의 의미를 가짐 )
       - EIMSK : 외부 인터럽트의 개별적인 허용 제어 레지스터

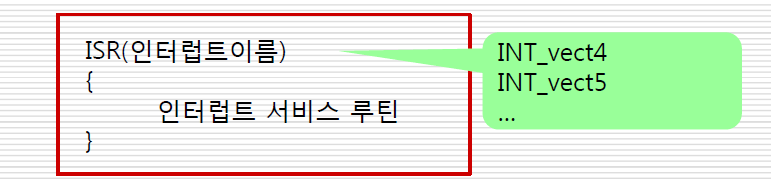
어떤 bit가 1로 set되면 해당 bit 번호의 인터럽트를 사용 가능하게 함

* + - * EICR : 외부 인터럽트의 트리거 설정에 사용된다.

EICRA : 외부 인터럽트 0~3의 트리거 설정

EICRB : 외부 인터럽트 4~7의 트리거 설정

1. 인터럽트 설정 과정
   * + - SREG의 전체 인터럽트 허용 비트를 1로 세팅
       - 특정 인터럽트 가능 비트를 모두 1로 세팅 ( INT4, INT5, TIMER 등 )
       - EICRA 세팅으로 level 및 edge 트리거 설정
2. ISR 등록



* 타이머
  + MCU 내부 clock을 세는 장치 -> 일정시간 경과 후에 인터럽트를 발생시킨다.
  + 타이머 인터럽트

1. 오버플로우 인터럽트 : 카운터 값이 오버플로우 시 발생(8bit 경우 : 0xff->0x00시 발생)
2. 출력비교 인터럽트 : 카운터 값이 출력비교 레지스터 값과 같게 되는 순간 발생
   * 8비트 타이머/카운터 특징
     + 0번과 2번 타이머/카운터를 사용
     + 비동기 동작 모드를 갖는 8비트 업/다운 카운터
     + 10비트의 프리스케일러 보유
     + 인터럽트 기능
   * 관련 레지스터
3. TCCRn : n번째 타이머/카운터 제어 레지스터 -> 타이머/카운터의 전반적인 동작 결정
4. TIMSK : 타이머 인터럽트 마스크 레지스터
   * + - TOIE : 오버플로우 인터럽트 enable
       - OCIE : 출력 비교 인터럽트 enable
   * 타이머 오버플로우 인터럽트 발생 과정
5. TCNT0값을 설정 ( 0~255 )
6. TCNT0값이 타이머 주기마다 증가하며, 255보다 커질 경우 오버플로우 인터럽트 발생

# 과제 목표

1. < 스위치 실습 과제 >

1/100 스톱워치를 FND를 사용해 만들되 스위치 인터럽트로 두 가지 기능 구현

SW1의 기능 : 한번 누르면 Start, 다시 누르면 Stop을 반복

SW2의 기능 : 스톱워치 초기화

1. < 버저 실습 과제 >

버저, 스위치 인터럽트와 타이머 인터럽트를 사용해 다음을 구현

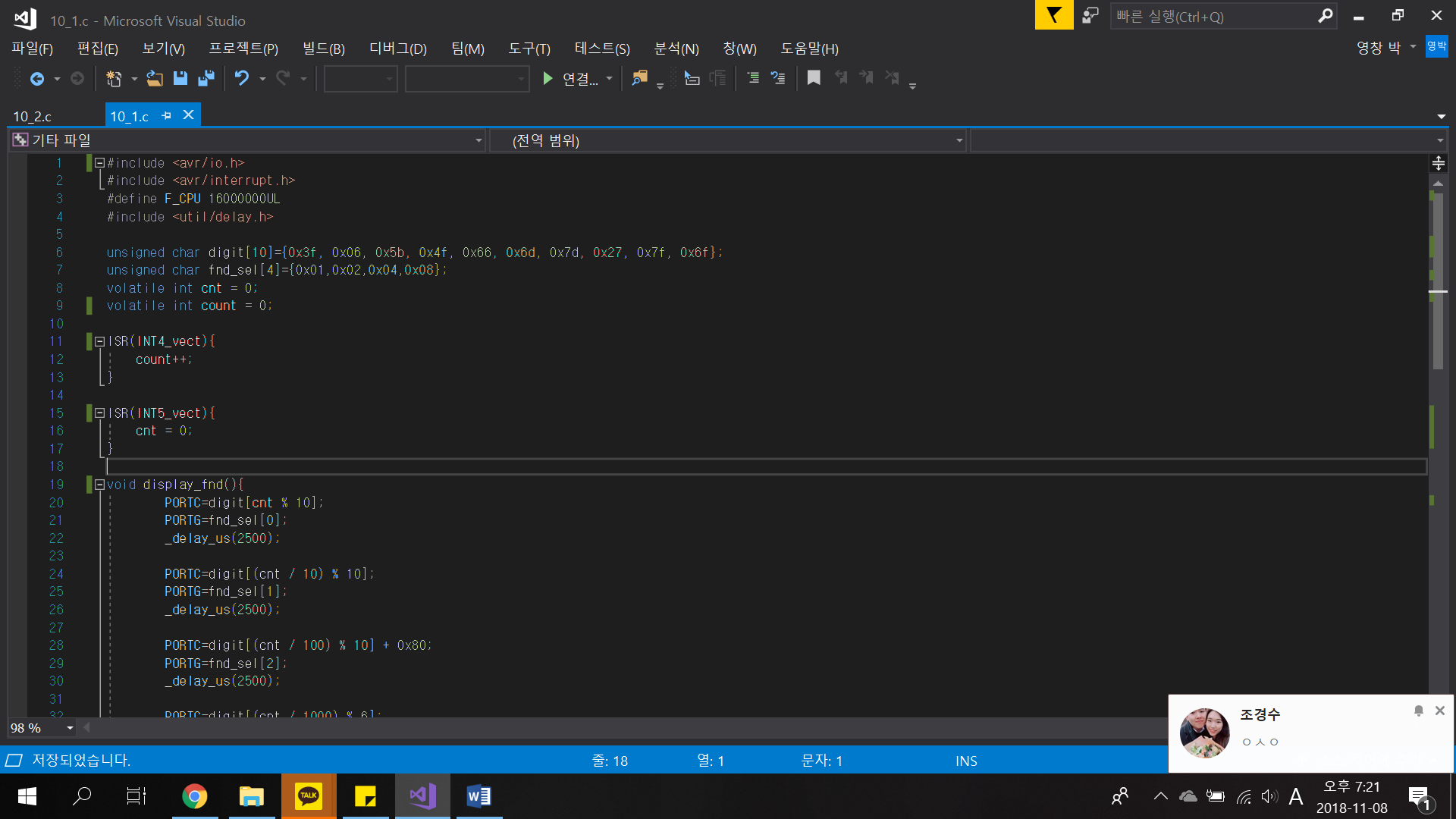
SW1의 기능 : 버저 ON/OFF

SW2의 기능 : 버저 음계 증가 ( 도 레 미 파 솔 라 시 도 반복 )

# 문제 해결

1. **< 스위치 실습 과제 >**

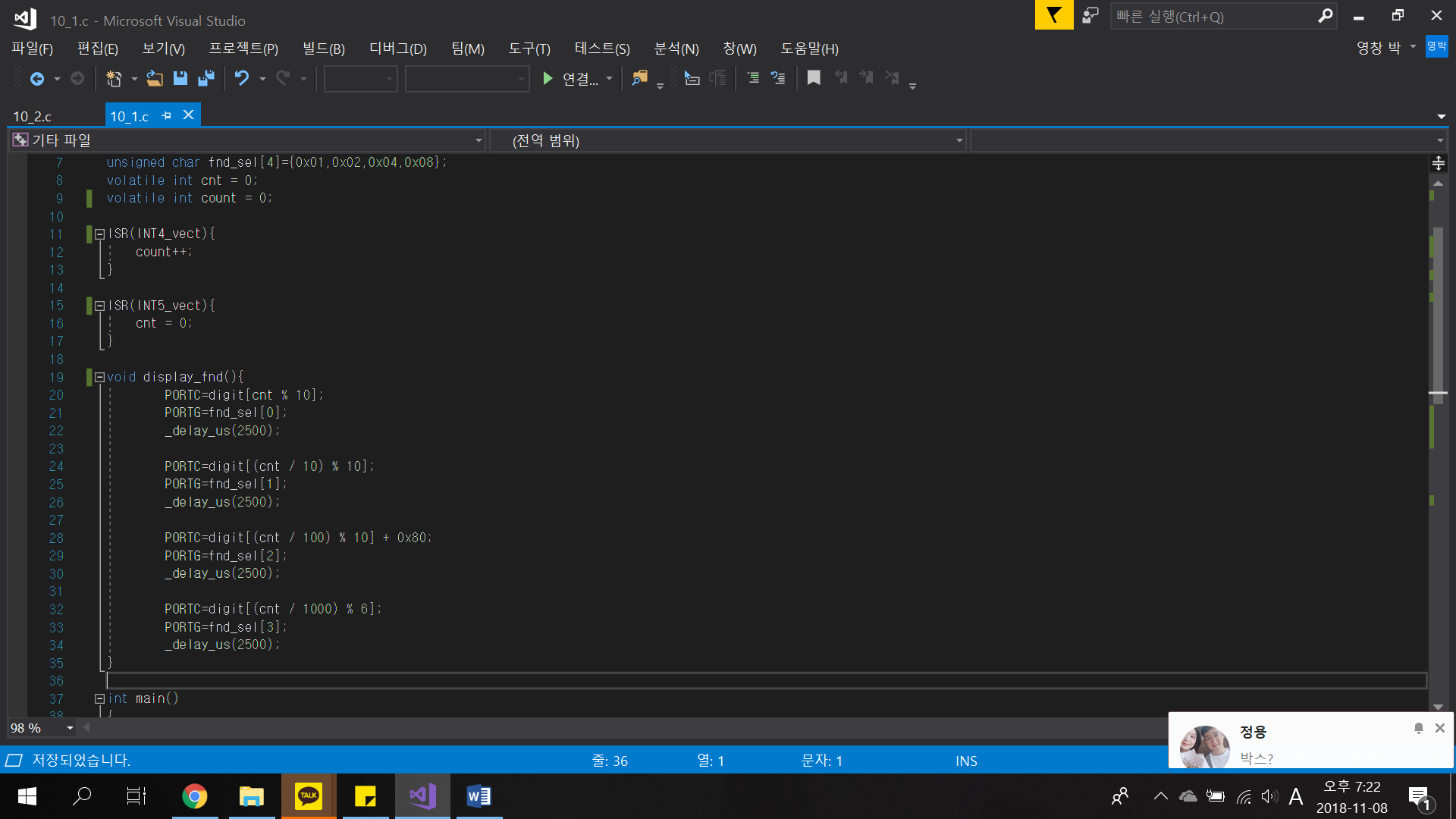
**변수 정의**



**digit : FND에 출력할 수의 값 fnd\_sel : FND 위치값**

**cnt : 1/100단위로 증가할 수 count : ON/OFF를 구분할 값**

**함수 정의**

**INT4\_vect ISR**

**SW1이 눌렸을 때 count 1 증가**

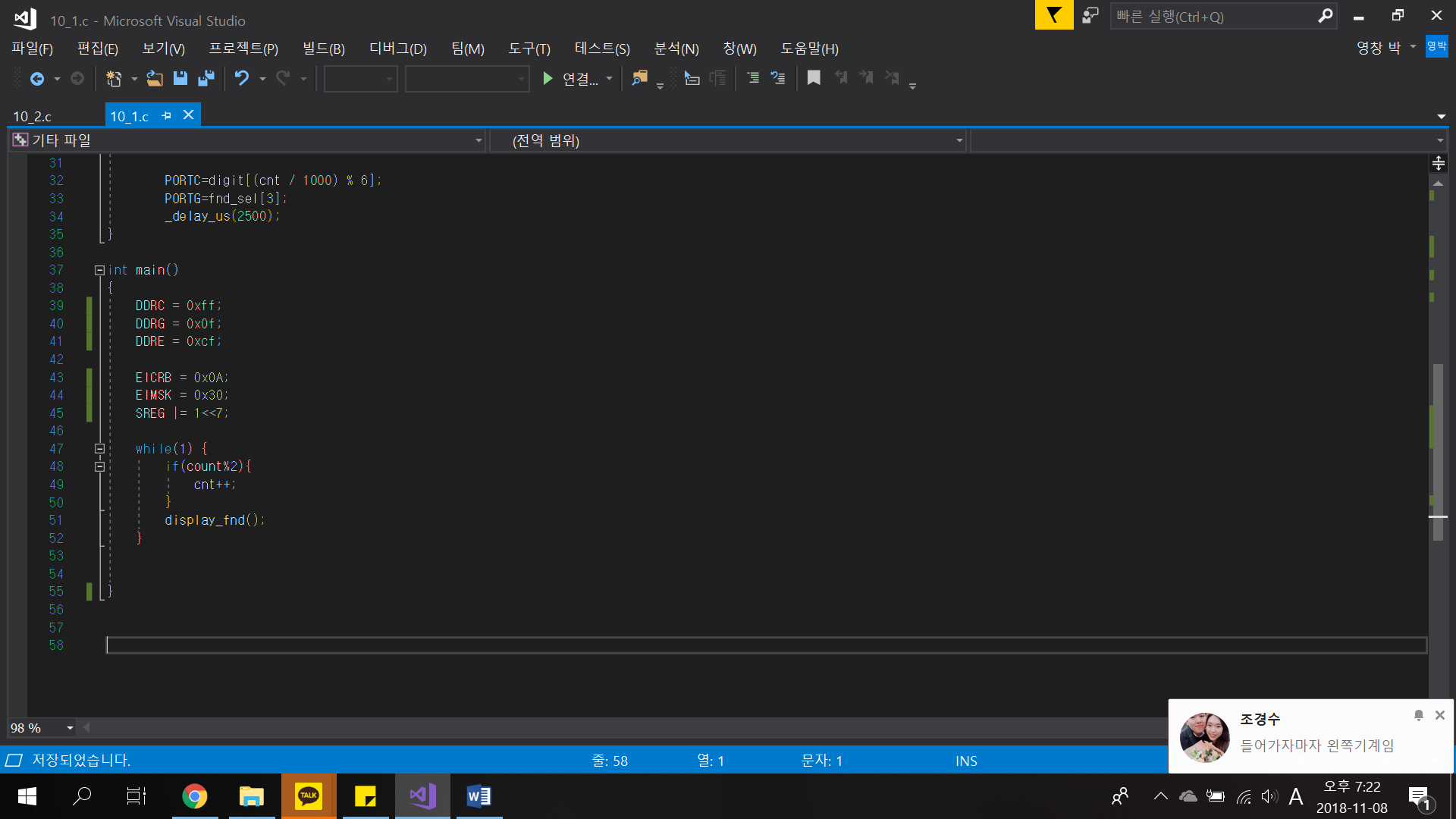
**INT5\_vect ISR**

**SW2이 눌렸을 때 cnt 0으로 초기화**

**display\_fnd**

**현재 cnt에 따른 값을 FND에 출력**

**레지스터 정의 및 메인 함수**

**DDRC : 모든 FND 출력 설정을 위해 0xff로 설정**

**DDRG : 모든 FND 선택 신호 출력 설정을 위해 0x0f로 설정**

**DDRE : SW1, SW2를 입력으로 설정하기 위해 11001111으로 설정**

**EICRB : INT4, INT5를 falling edge로 설정**

**EIMSK : INT4, INT5의 인터럽트 enable**

**SREG : 모든 bit를 1로 set함으로 인터럽트를 enable**

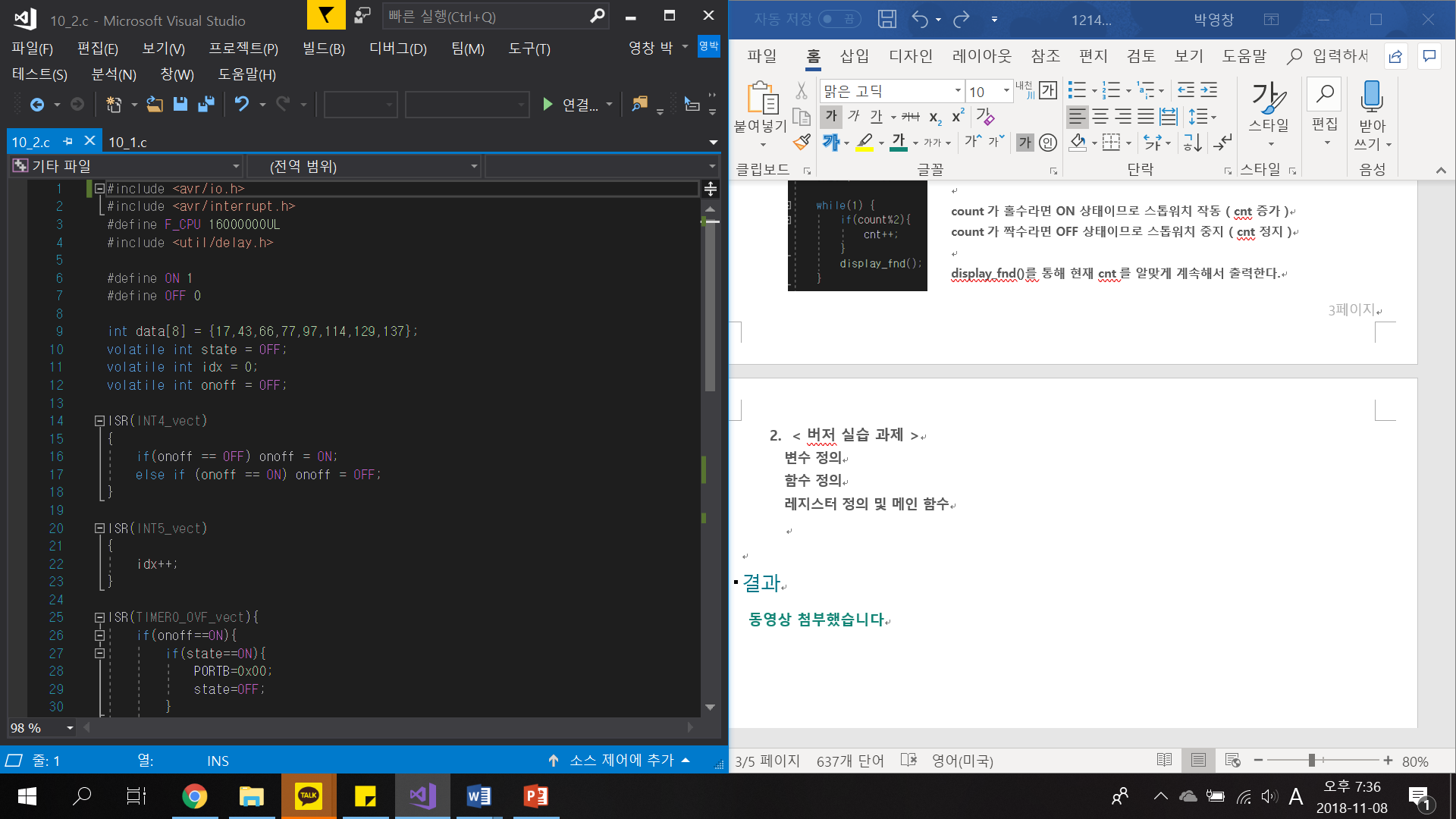
**count가 홀수라면 ON 상태이므로 스톱워치 작동 ( cnt 증가 )**

**count가 짝수라면 OFF 상태이므로 스톱워치 중지 ( cnt 정지 )**

**display\_fnd()를 통해 현재 cnt를 알맞게 계속해서 출력한다.**

1. **< 버저 실습 과제 >**

**변수 정의**



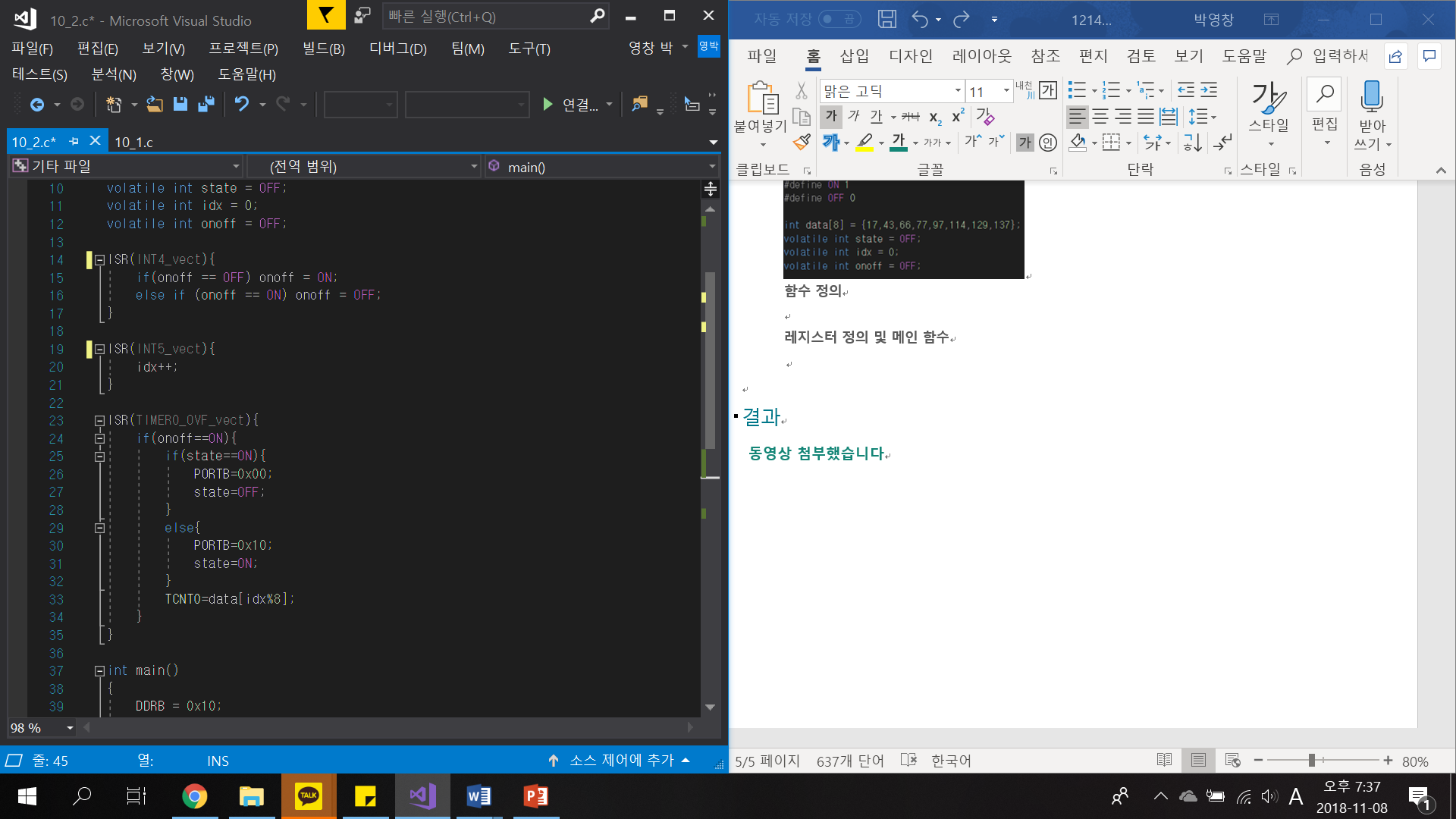
**data : 각 음계의 32분주 기준 주파수 값**

**state : 주파수를 파형으로 표현하기 위한 상태 변수**

**idx : 현재 음계를 나타내는 수**

**onoff : 버저 ON/OFF 상태**

**함수 정의**

**INT4\_vect ISR**

**SW1이 눌렸을 때 버저 ON/OFF 상태 변경**

**INT5\_vect ISR**

**SW2가 눌렸을 때 음계 인덱스 1 증가**

**TIMERO\_OVF\_vect ISR**

**타이머 오버플로우 인터럽트 발생 시**

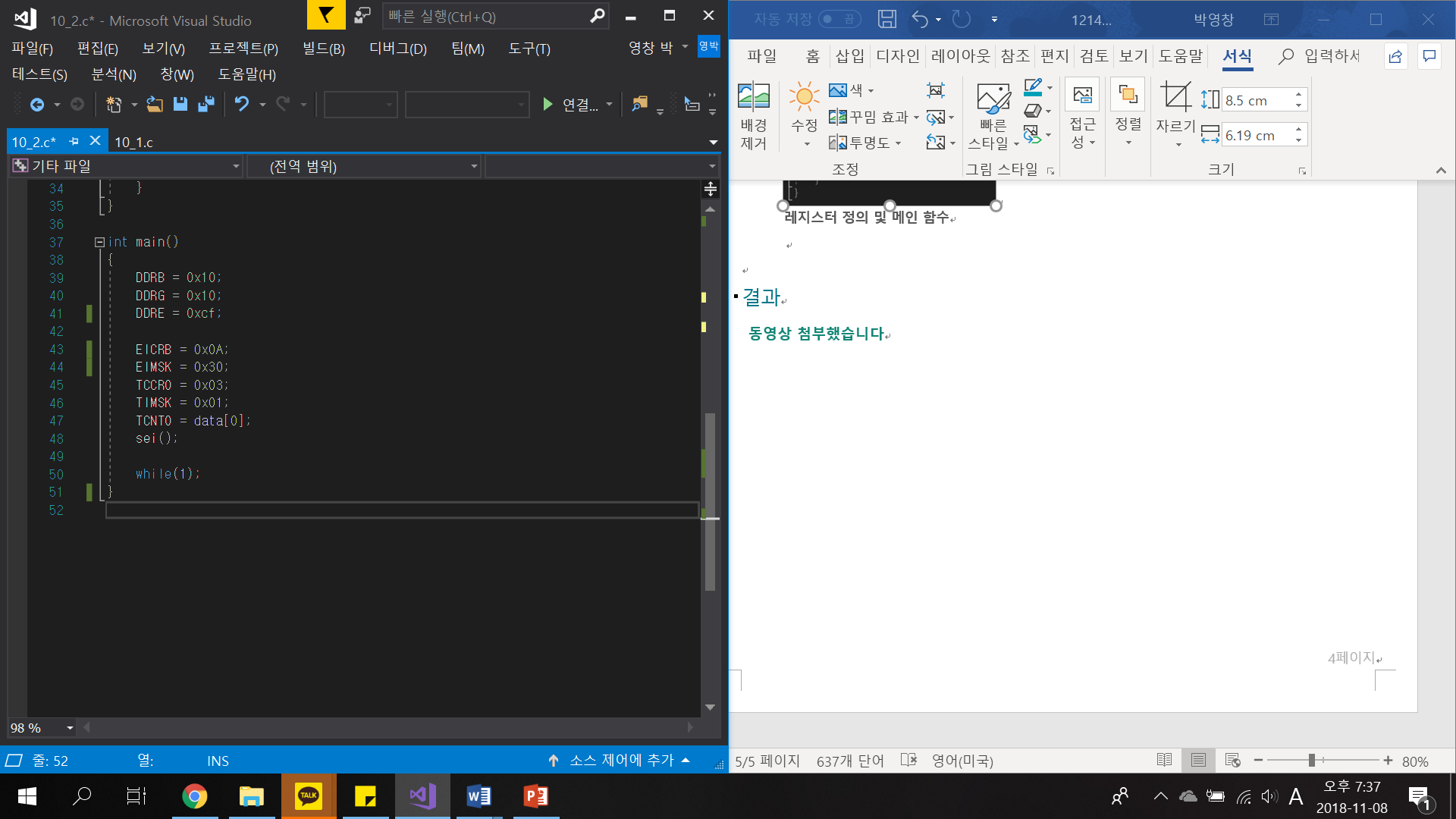
**버저 ON/OFF 상태를 확인 후 ON이라면**

**PORTB를 출력 상태와 출력하지 않는 상태로**

**번갈아 가며 실행해 주파수에 맞는 파형을 만든다.**

**주기는 TCNT0의 값을 통해 지정한다.**

**레지스터 정의 및 메인 함수**

**DDRB : 버저 출력 설정**

**DDRE : SW1, SW2를 입력으로 설정하기 위해 11001111으로 설정**

**EICRB : INT4, INT5를 falling edge로 설정**

**EIMSK : EIMSK : INT4, INT5의 인터럽트 enable**

**TCCR0 : 0번째 타이머/카운터 제어 및 32분주 설정**

**TIMSK : 타이머 오버플로우 인터럽트 enable**

**TCNT0 : 타이머 오버플로우 발생 전의 값을 지정**

**sei()를 통해 인터럽트를 enable시키며 내용이 없는 while문을 통해 인터럽트만을 기다린다.**

# 결과

**동영상 첨부했습니다**