**REPORT**

**| Embedded System**

**| Programming on an embedded systems 3**

**| 컴퓨터정보공학과**

**| 12141540 박영창**

# 내용

11주차 실습에서는 임베디드 보드의 광센서에 대해 배웠습니다.

* CdS 센서
  + 황하 카디늄을 말하며, 빛의 밝기에 따라 전기저항 값이 변화하는 성질을 이용
  + 빛이 밝아질수록 저항 값은 커지고, 전압은 감소한다. 그 반대로 빛이 어두워질수록 저항 값은 작아지고 전압은 증가한다.
* A/D 컨버터 ( Analog to Digital Converter )
  + 아날로그 신호를 컴퓨터가 읽을 수 있는 병렬/직렬의 디지털 데이터로 변환시켜주는 장치
  + 측정하려는 아날로그 물리량의 범위 및 시스템 응용 목적에 따라 분해능이나 정밀도가 적합한 것을 사용한다
  + 분해능 ( resolution )
    - 디지털 출력 값을 한 등급만큼 변화시키기 위한 아날로그 입력의 최소 변화
    - n 비트 A/D 컨버터 경우 출력의 최소 데이터 범위는 1 / 2 ^ n
  + 변환 시간 ( conversion time )
    - A/D 컨버터가 변환을 수행하는데 필요한 시간
    - 초당 샘플링 속도로 나타낸다
* ATmega128의 A/D 컨버터 레지스터
  + ADMUX ( ADC Multiplexer Selection Register ) : 멀티플렉서 선택 레지스터
    - 비트 7 / 6 : REFS 1 / 0 ( Reference Selection Bit ) – 기준 전압 선택
    - 비트 5 : ADLAR ( ADC Left Adjust Result ) – ADCH / ADCL 의 정렬 방향 설정 ( 1일 때 왼쪽으로 정렬 )
    - 비트 4 ~ 0 : MUX 4 ~ 0 ( Analog Channel and Gain Selection ) – 아날로그 입력채널 선택
  + ADCSRA ( ADC Control and Status Register A ) : 제어 및 상태 레지스터 A
    - 비트 7 : ADEN ( ADC Enable ) – 1로 설정 시 ADC 모듈 enable
    - 비트 6 : ADSC ( ADC Start Conversion ) – 1로 설정 시 ADC 변환 시작
    - 비트 5 : ADFR ( ADC Free Running Select ) – 프리런닝 모드 설정 ( 0으로 설정 시 한 번만 ADC 변환 )
    - 비트 4 : ADIF ( ADC Interrupt Flag )
      * ADC 인터럽트 플래그 및 ADC 변환의 완료를 알리는 플래그
      * ADC 변환 완료 후 ADC Data Register가 업데이트되면 이 비트가 1로 set 되면서 인터럽트를 요청한다.
      * 이 때 ADIE 는 1로 설정되고 SREG 레지스터의 I 비트가 1로 설정되어 있다면 인터럽트가 발생되어 처리된다.
    - 비트 3 : ADIE ( ADC Interrupt Enable )
    - 비트 2 ~ 0 : ADPS 2 ~ 0 ( ADC Prescaler Select Bit ) – 프리스케일러 ( 클록 분주비 ) 선택
  + ADCH / ADCL ( ADC Data Register High / Low ) : 데이터 레지스터
    - 변환 결과 데이터가 10비트 양의 정수로 표시된다.
    - 이 값은 기준 전압에 대한 비율 상수를 의미한다.
    - ADMUX 레지스터의 ADLAR이 0인 경우 오른쪽으로 정렬하며 1인 경우는 잘 사용 X ?
  + ADC 프로그램 과정

1. ADMUX / ADCSRA 초기 설정
   * + - ADMUX : 오른쪽 정렬 / AREF 설정 / MUX 설정
       - ADCSRA : ADC enable / single conversion / 분주 설정
2. ADC 변환을 진행하며 인터럽트 대기 및 데이터 read

# 과제 목표

< 광센서 실습 과제 >

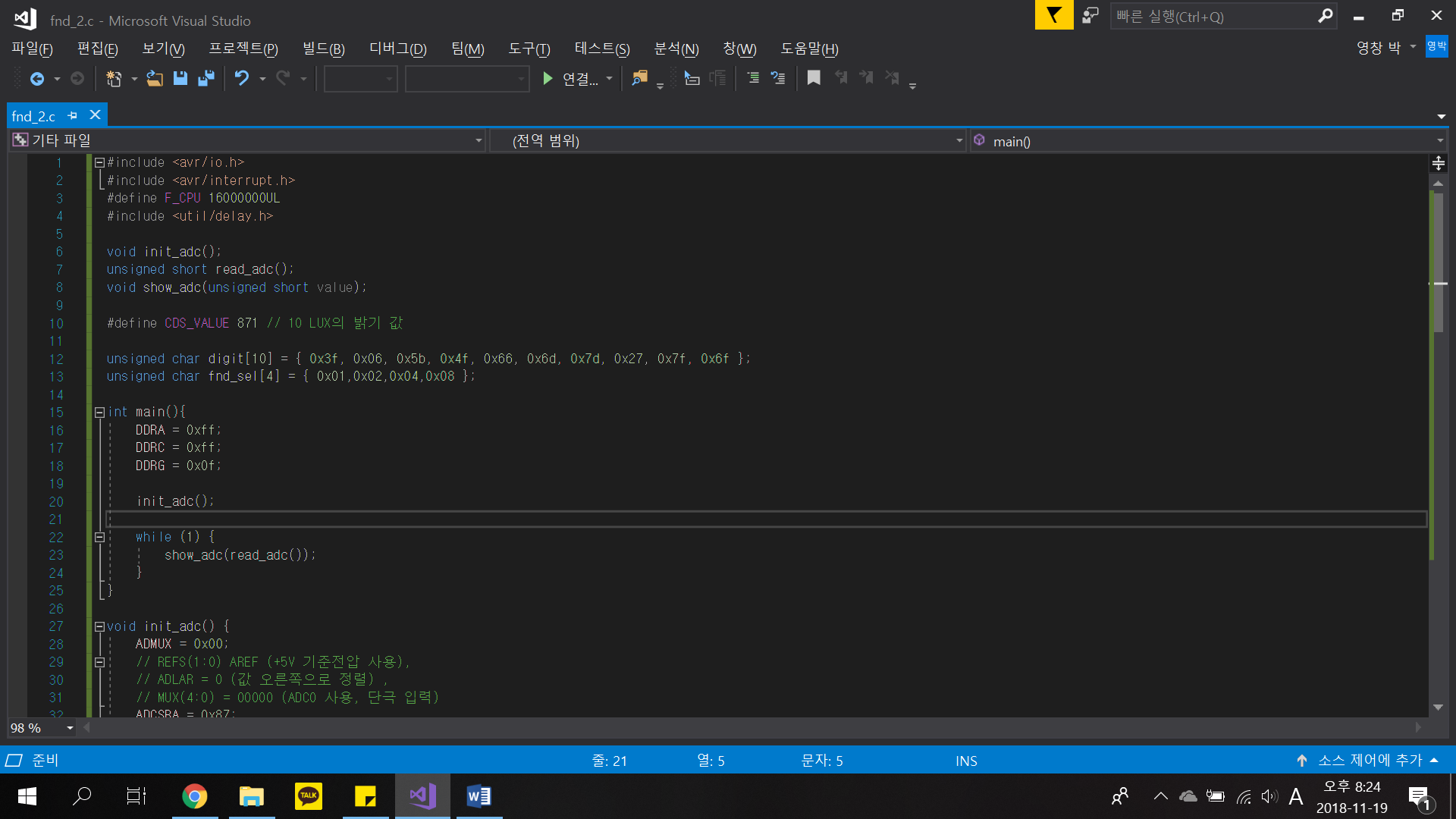
지금까지 배운 센서 2개 이상과 광센서를 이용하여 빛의 밝기 정도를 활용한 응용 프로그램 작성 ( LED , FND , BUZZER 중 2 + 광센서 )

Ex ) 광센서 일정 값 미만 -> LED ON / FND ON / BUZZER 울림

광센서 일정 값 이상 -> LED OFF / FND OFF / BUZZER 꺼짐

# 문제 해결

**< 전역 변수 및 함수 정의 >**



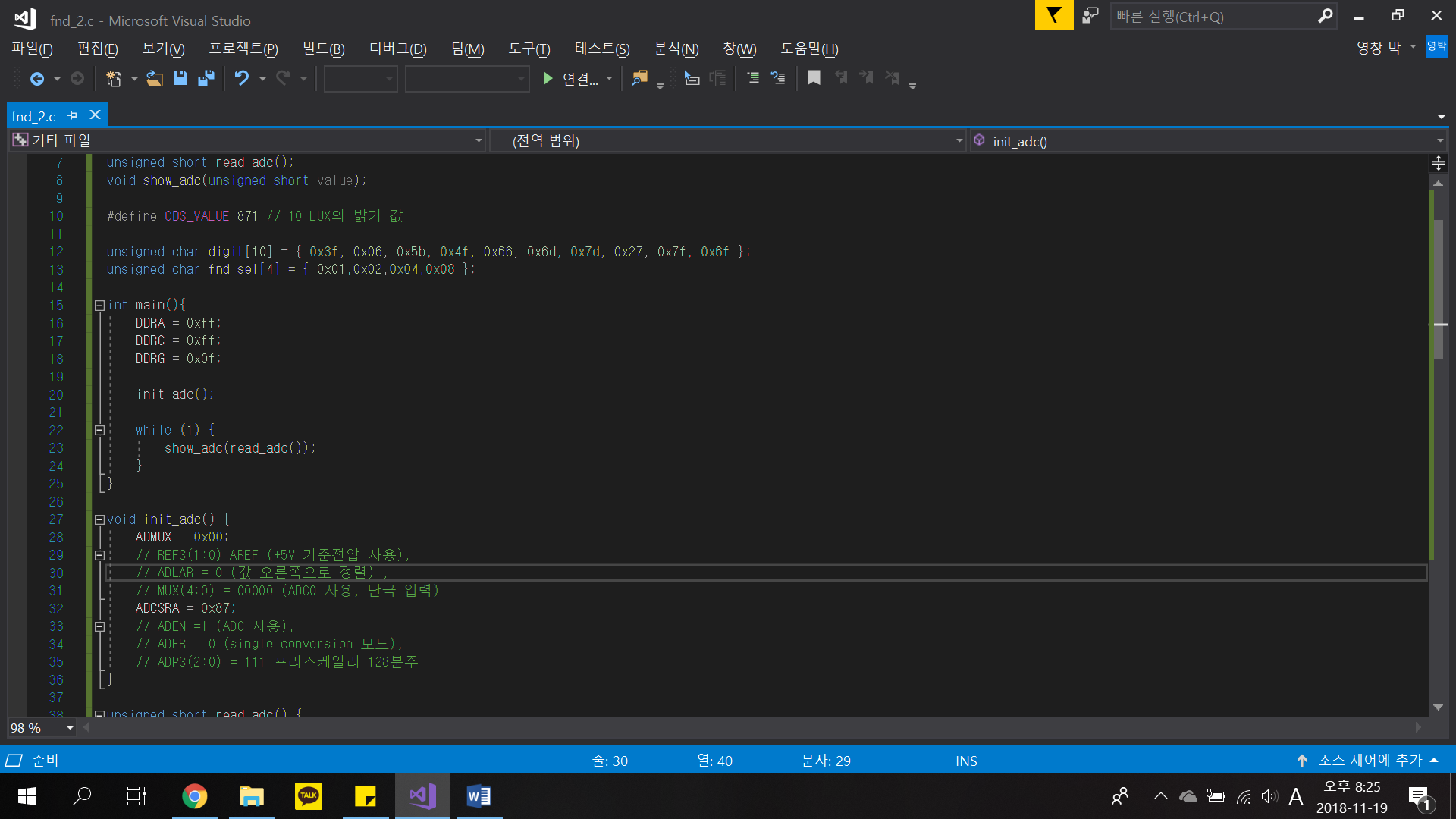
**init\_adc() : ADC 관련 레지스터 초기화 설정 함수**

**read\_adc() : 현재 광센서의 상태를 read**

**show\_adc() : 현재 광센서의 상태에 따라 다른 센서에 출력 ( LED / FND )**

**digit : FND로 출력할 각 숫자**

**fnd\_sel : FND의 출력 위치**

**< main >**

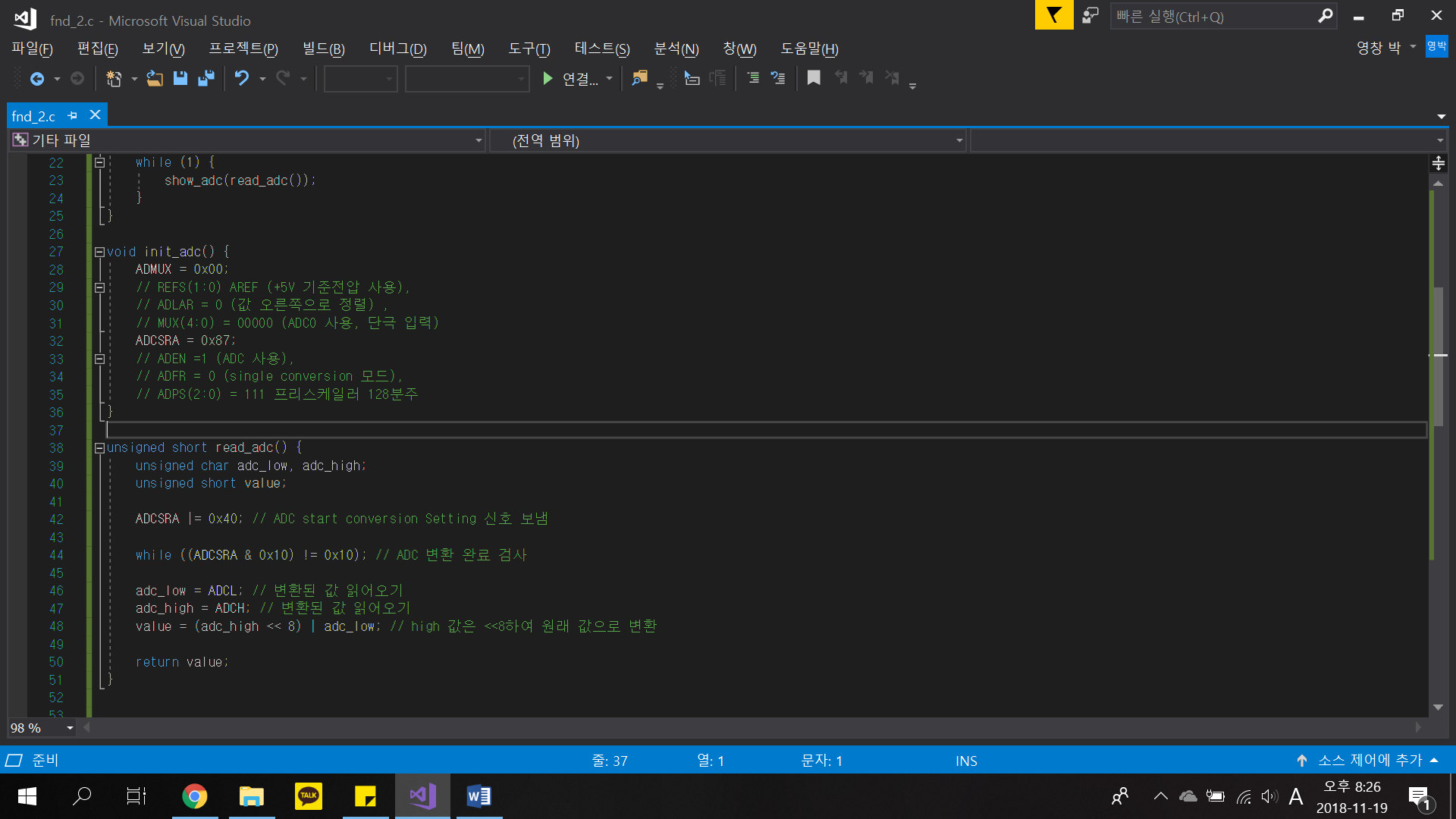
**DDRA 를 0xff로 초기화함으로써 LED 출력 설정**

**DDRC 를 0xff로 초기화함으로써 FND 출력 설정**

**DDRG 를 0x0f로 초기화함으로써 FND 출력 위치 설정**

**각 레지스터 초기화 후 ADC 관련 레지스터를 초기화 설정한 뒤 ADC 인터럽트를 기다리며 그를 통해 광센서의 상태를 읽음과 동시에 그에 따른 센서의 변화를 출력한다.**

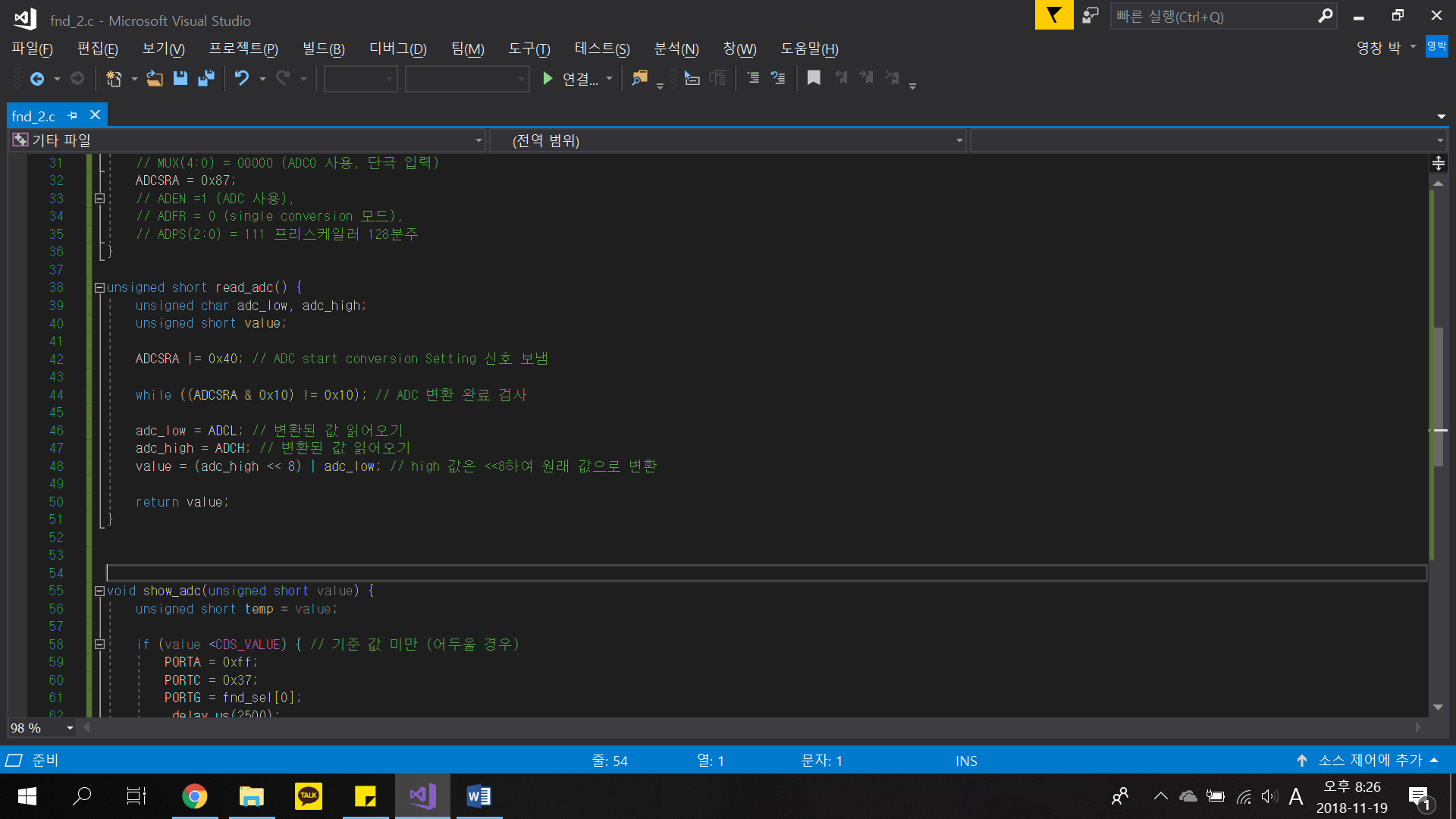
**< init\_adc >**



**ADMUX 를 0x00으로 초기화함으로써 오른쪽 정렬 / AREF / MUX를 설정**

**ADCSRA 를 0x87로 초기화함으로써 ADC enable / single conversion / 분주 설정**

**< read\_adc >**



**adc\_low / adc\_high : ADCL / ADCH의 값을 받아 오기 위한 char형 변수 ( 1 byte )**

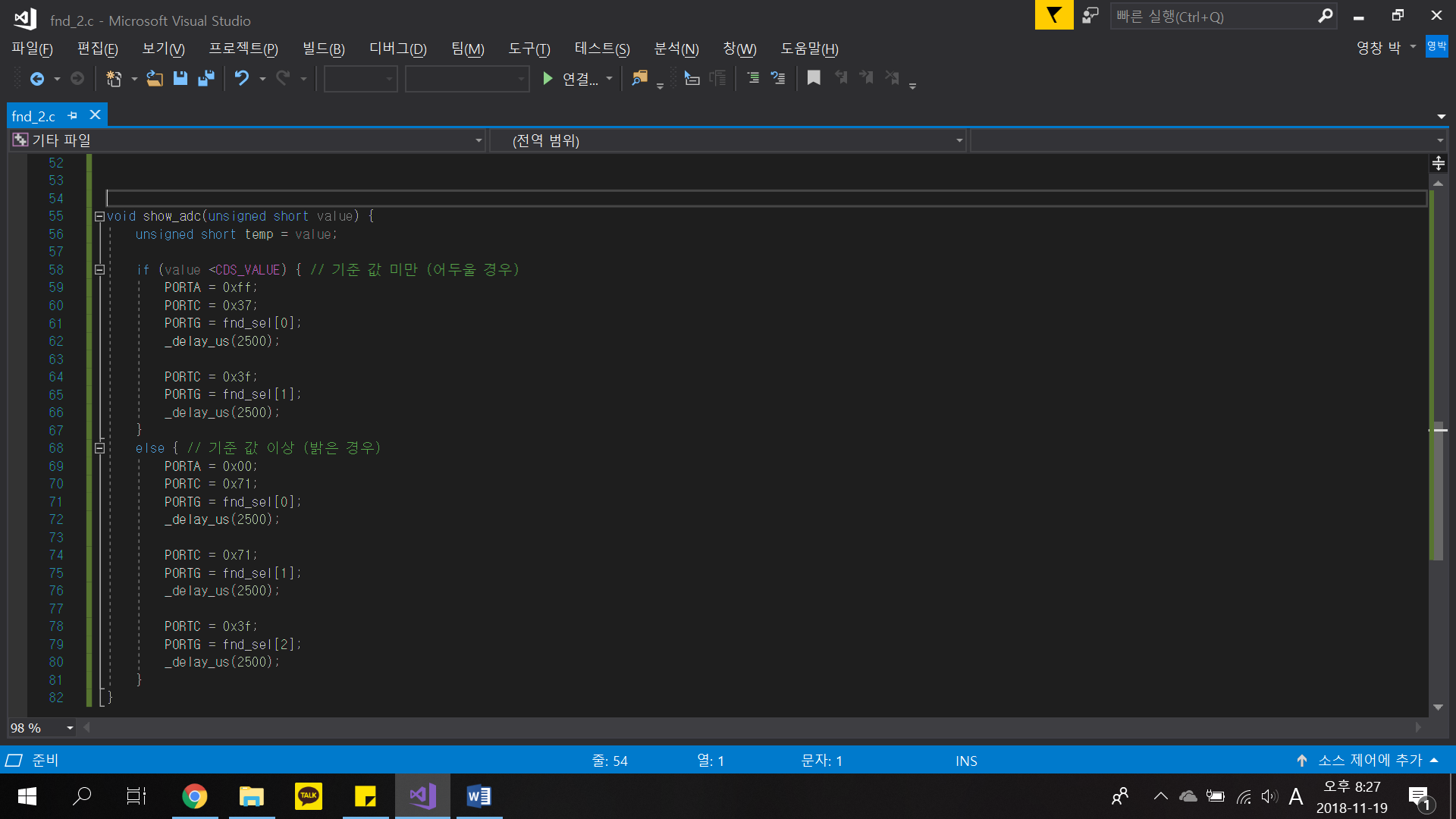
**value : ADCL / ADCH 값을 받아와 합쳐서 반환할 short형 변수 ( 2 byte )**

**ADCSRA의 6번 bit ( ADSC )를 1로 설정해 ADC변환의 시작 신호를 보낸다.**

**그 다음 ADCSRA의 4번 bit ( ADIF )가 완료될 때까지 대기한다. ( 변환이 완료되면 해당 bit가 1로 설정 )**

**변환된 값을 ADCL / ADCH로부터 각 변수로 읽어와 value 변수에 합친 뒤 반환한다.**

**< read\_adc >**

**인자로 들어온 2 byte 크기의 value값을 기준 값과 비교한다**

**비교 결과 기준 값보다 인자로 받은 value가 작은 경우 더 어두운 쪽이므로 LED를 on 시켜주고 FND에 “ON”을 출력해준다.**

**반대의 경우에는 더 밝은 쪽이므로 LED를 off시켜주고 FND에 “OFF”를 출력해준다.**

**FND에 출력하는 각 알파벳은 digit 배열 내에서 받아 오지 않고 FND 내의 각 LED에 해당하는 bit를 켜 알파벳 형태로 출력한다.**