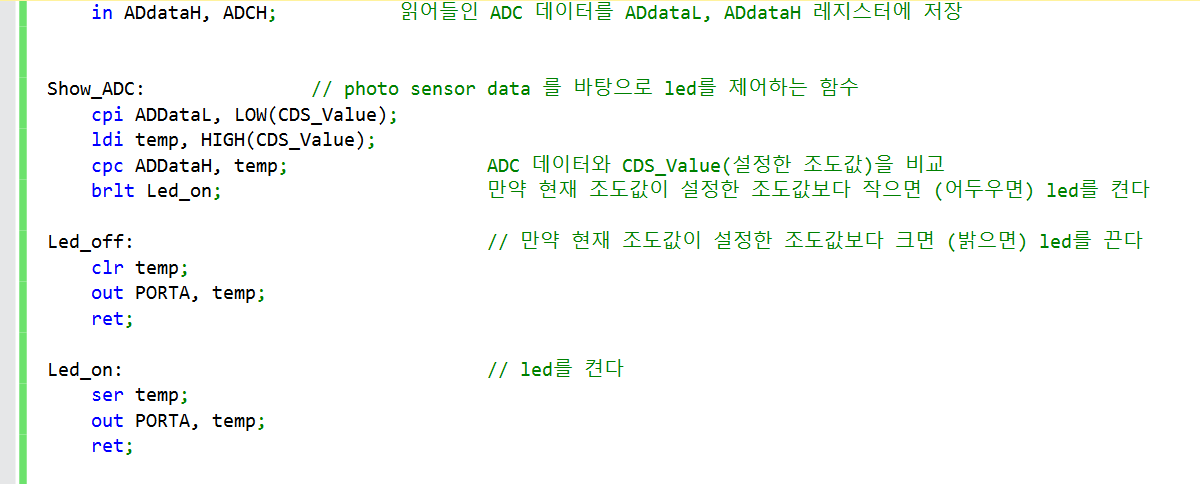
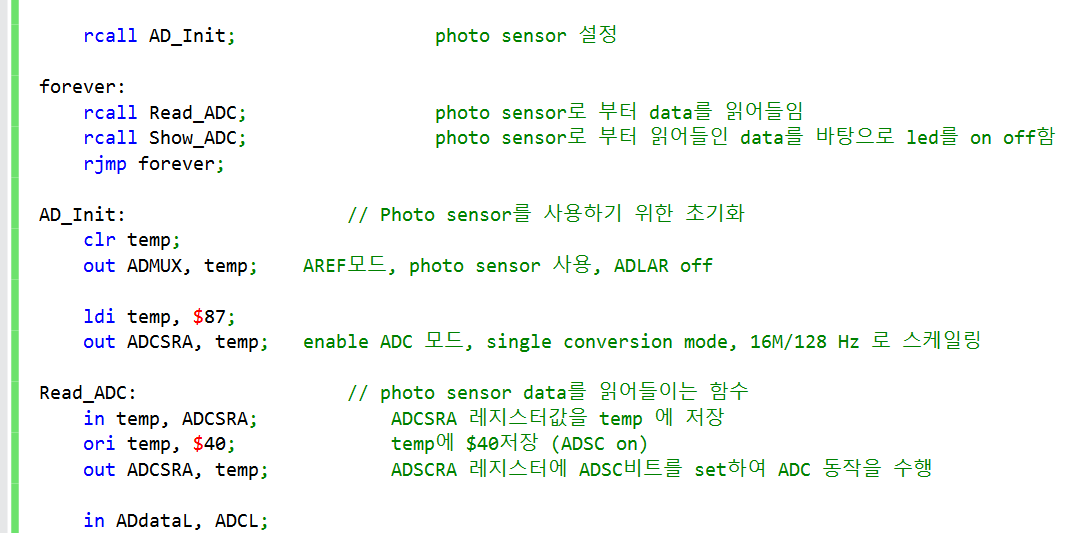
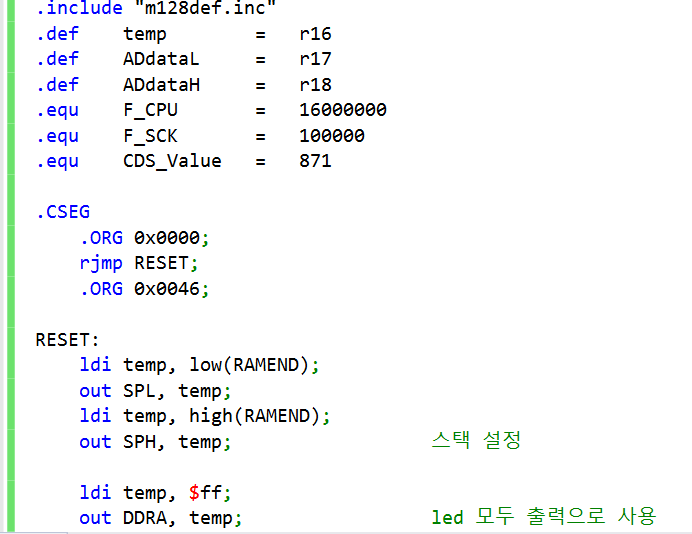
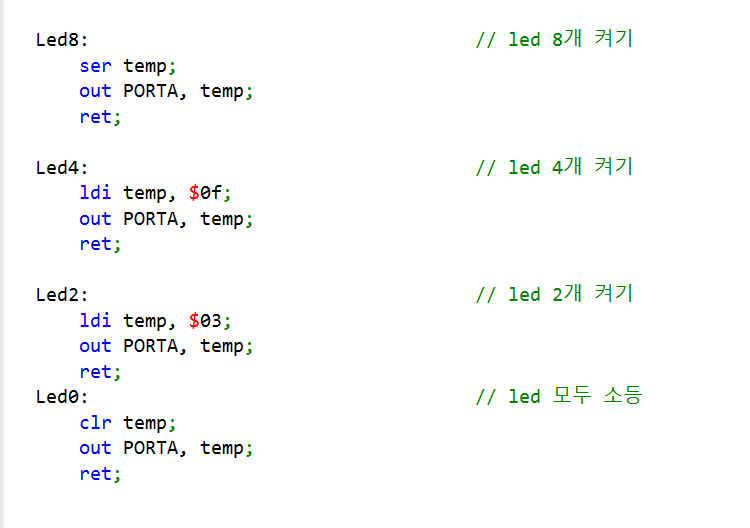
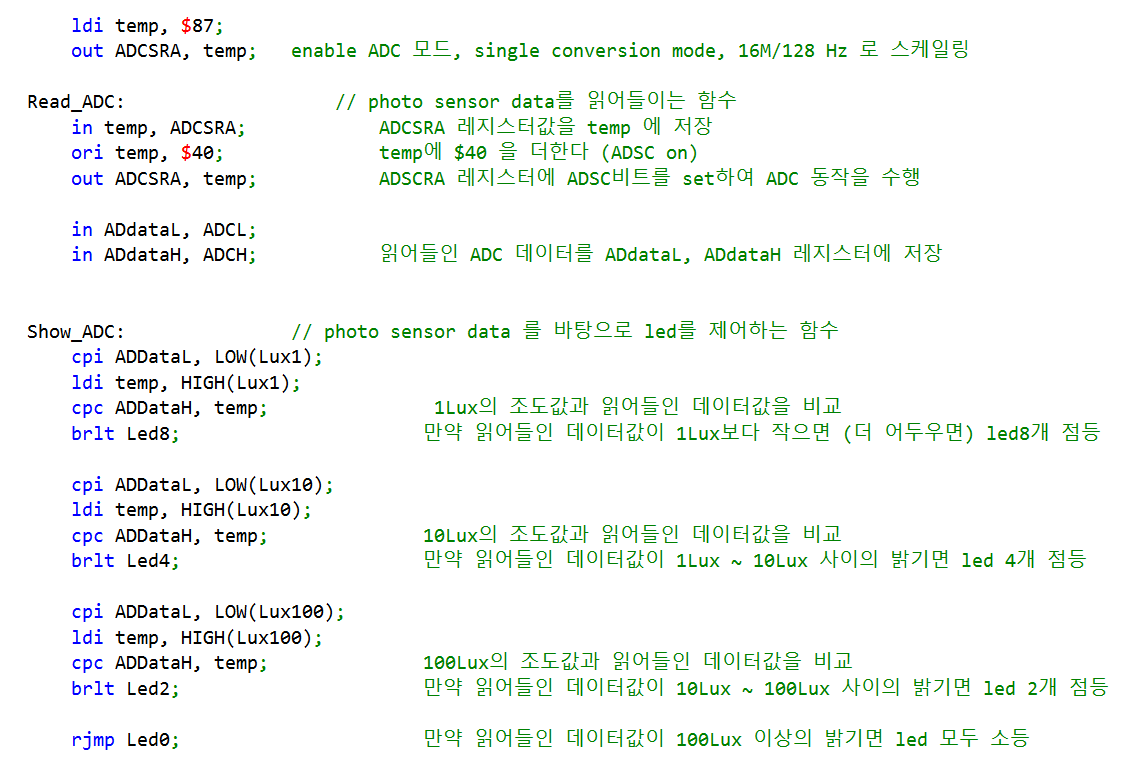
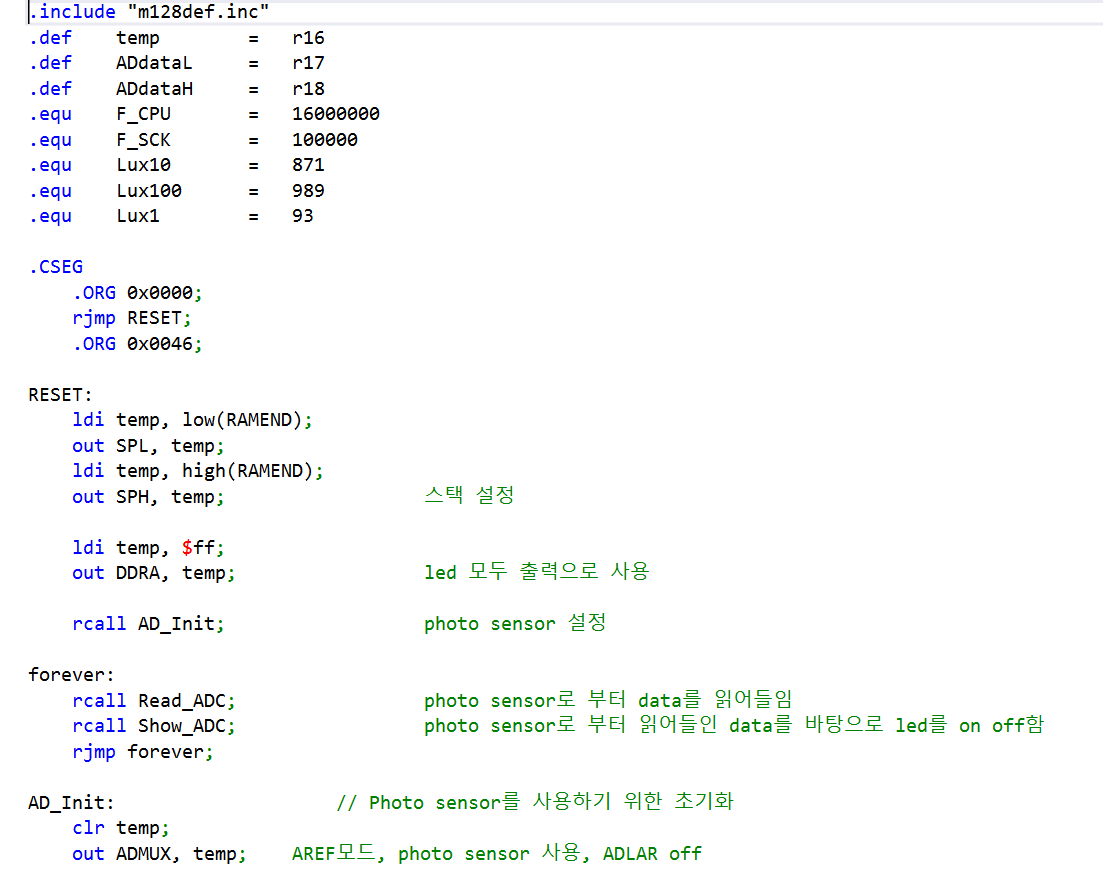
Lab 10-1



CDS\_Value (설정한 조도값 , 10Lux) 와 현재 광센서로부터 읽어들인 조도값을 비교하여 현재 환경이 어두우면 led on 루프로 이동하여 led를 켜고, 밝으면 led off 루프로 이동하여 led를 끄는 프로그램이다. 읽어들이는 조도값은 16비트 데이터이므로 cpi, cpc를 통해 16비트를 모두 비교해야 한다. ADSC가 1로 set 되면 conversion 을 수행하는 점을 이용하여 Read\_ADC 에서 조도값을 읽어들인다. Conversion을 모두 수행하면 ADSC는 다시 0이 된다.

Lab 10-2

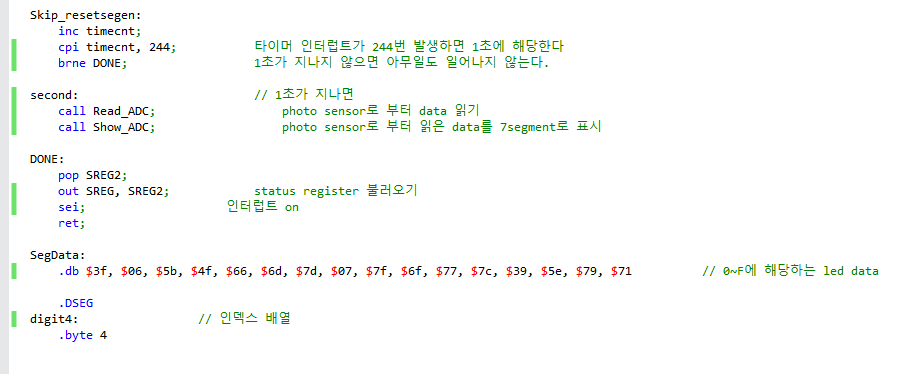
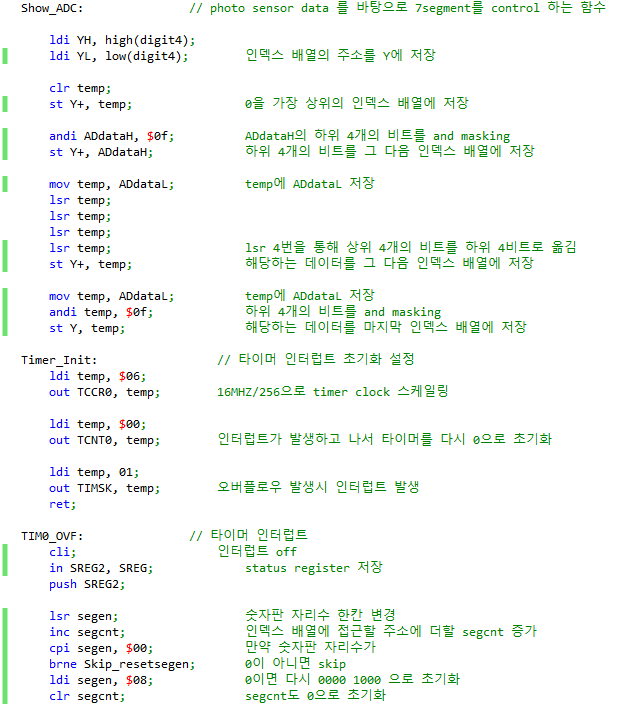
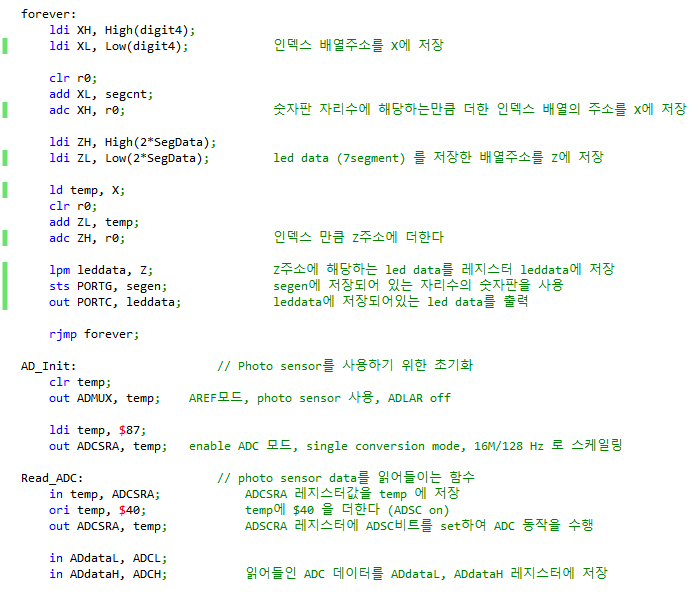
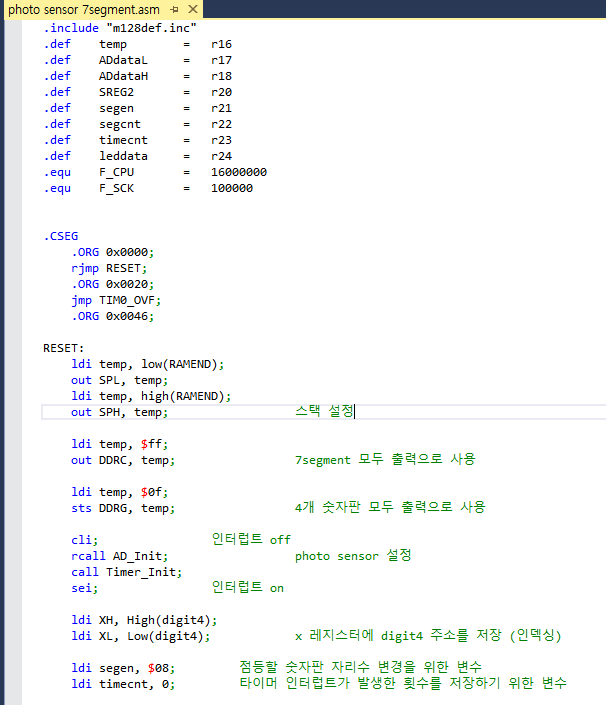


Lab 10-1에서의 실습 내용을 응용하여 1Lux 이하, 1Lux~10Lux , 10Lux~100Lux, 100Lux 이상의 4가지 단계를 만들어 각 단계에 해당하는 환경(광센서 데이터값)마다 다른 led 점등 개수를 설정한다.

어두운 환경일수록 led 점등 개수를 늘리고, 밝은 환경일수록 led 점등 개수를 줄인다.

Show\_ADC 함수를 변형하였으며 해당하는 단계의 광센서 데이터값이 읽히면 비교를 통해 해당하는 단계로 점프하도록 루프컨트롤을 사용하여 구현하였다.

Lab 10-3



데이터를 photo sensor로부터 읽어들이는 과정은 똑같다. 즉 Read\_ADC의 부분은 수정하지 않고 Show\_ADC 부분과 타이머 인터럽트를 추가하여 1초마다 photo sensor로부터 data를 받아들이고 그 값을 상위 비트와 하위 비트를 4bit씩 나누어 4개의 숫자판에 차례로 출력하는 실습이다.

ADLAR = 0 이기 때문에 AD data는 ADCH 에 0000 00xx 로 들어오고 , ADCL 에 xxxx xxxx 의 형태로 들어온다. 따라서 이를 인덱스로 사용하기 위해선 4비트씩 나누는 작업을 수행해야 한다.

1초마다 동작하도록 설정하기 위해서 타이머 인터럽트가 발생할 때 마다 Read\_ADC, Show\_ADC를 수행하도록 하였다.

고찰

광센서를 이용하여 현재 환경의 조도값을 읽어들이고 이를 통하여 LED를 제어하거나 7segment를 통해 현재 조도값을 출력하는 실습을 수행하였다. 프로그래밍 자체는 어렵지 않았으나 광센서의 가변저항을 통해 얻은 전압값을 0~ 1023 의 스케일링을 통해 조도값으로 변환하는 과정을 이해하는데 오랜시간이 걸렸다. 이러한 변환방법과 ADMUX, ADCSRA 레지스터의 설정, AD data가 상위 ADCH, 하위 ADCL 로 나누어져 들어온다는 점을 이해하면 크게 어렵지 않게 원하는 동작들을 설정할 수 있었다.