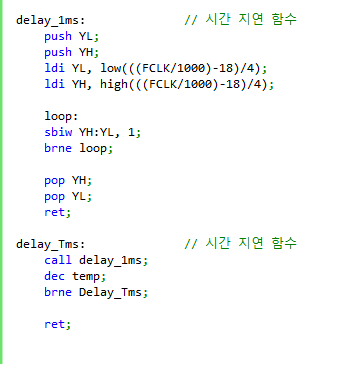
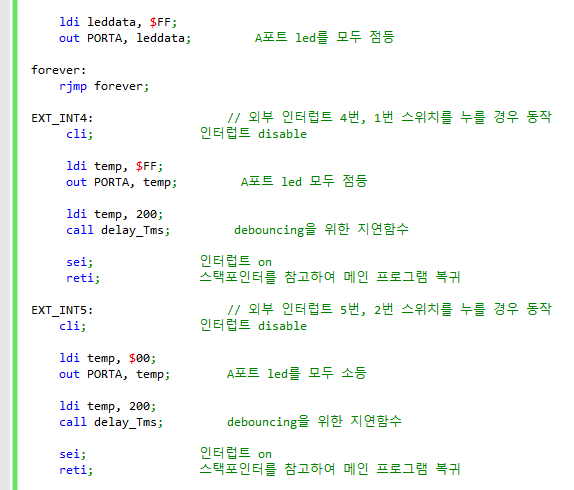
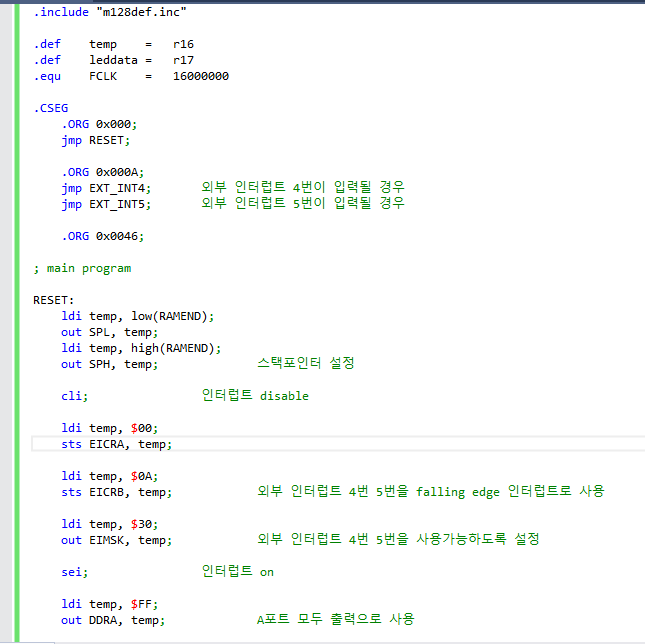
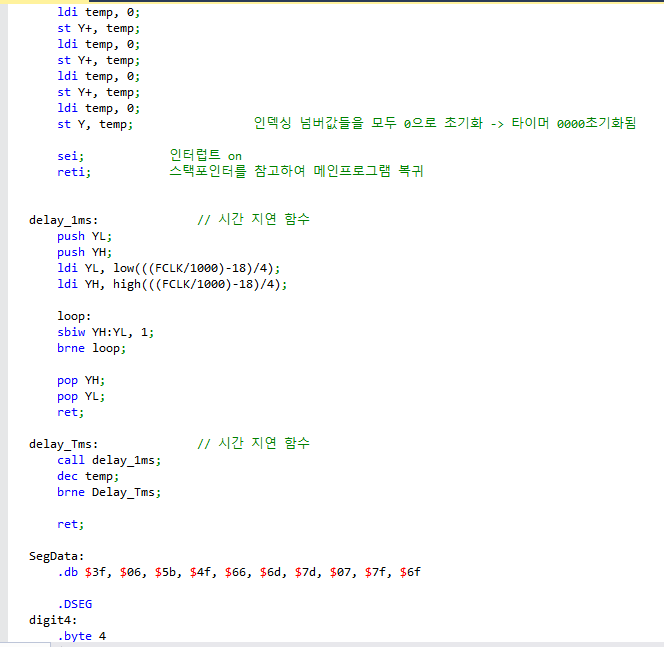
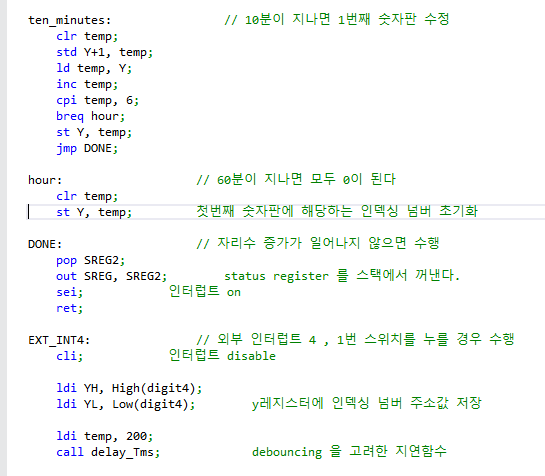
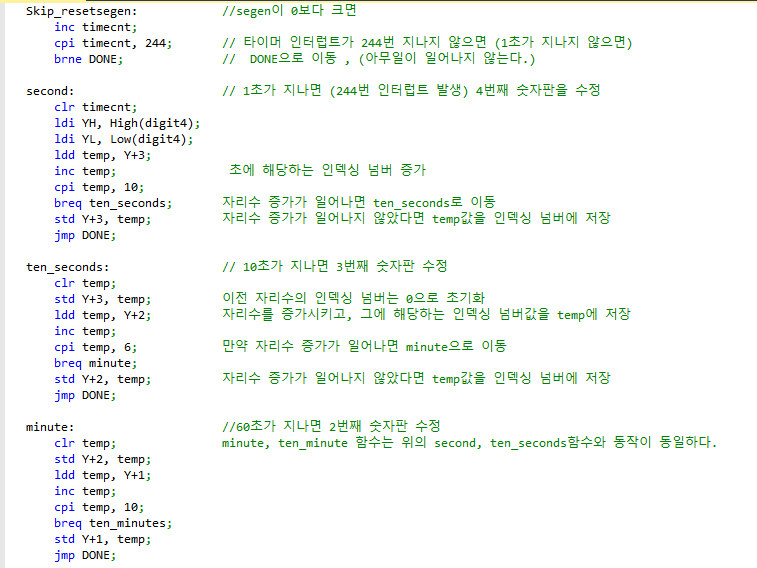
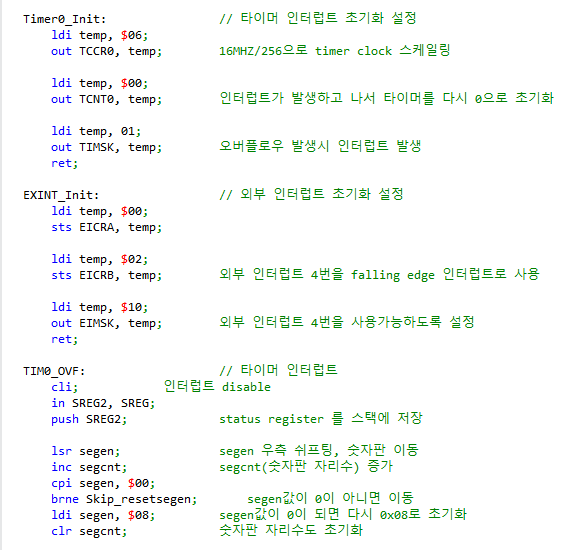
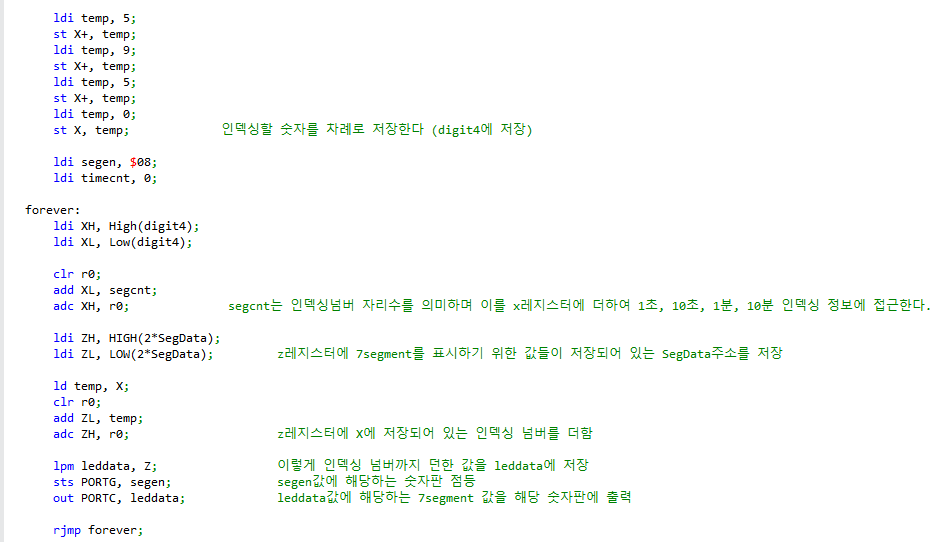
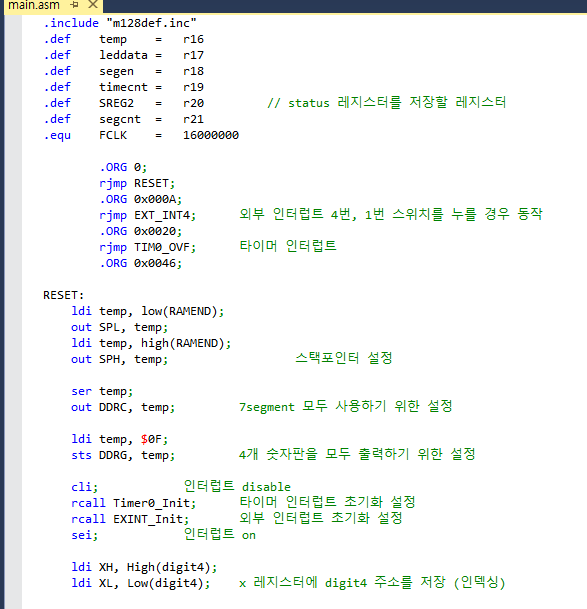
1)

PreLab7



falling edge에서 동작하는 시스템을 구성하였다. 외부 인터럽트4, 5를 이용해 led를 점등, 소등한다는 것은 결국 쉽게말해 AVR의 1번 스위치를 누르면 외부 인터럽트4로 이동하여 led점등, 2번 스위치를 누르면 외부 인터럽트5로 이동하여 led 소등의 동작을 수행하는 것을 말한다. 이때 사람이 실제로 스위치를 누를 때 bouncing 이 일어나므로 오류가 생기지 않도록 스위치를 누를 때 바로 led 점등, 소등의 동작을 수행하는 것이 아닌 스위치를 누른 후 잠깐의 delay를 주어 입력신호가 중복되지 않도록 하였다.

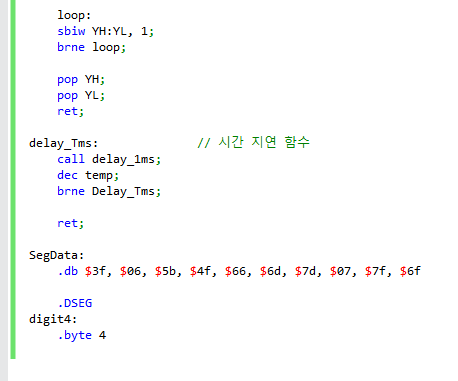
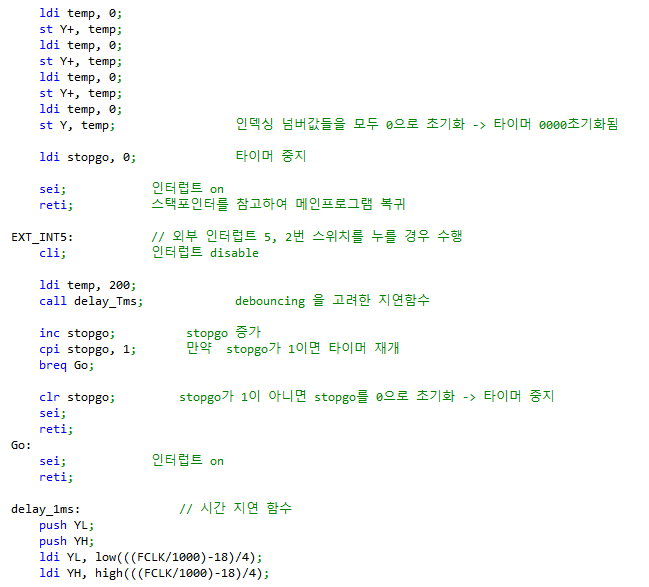
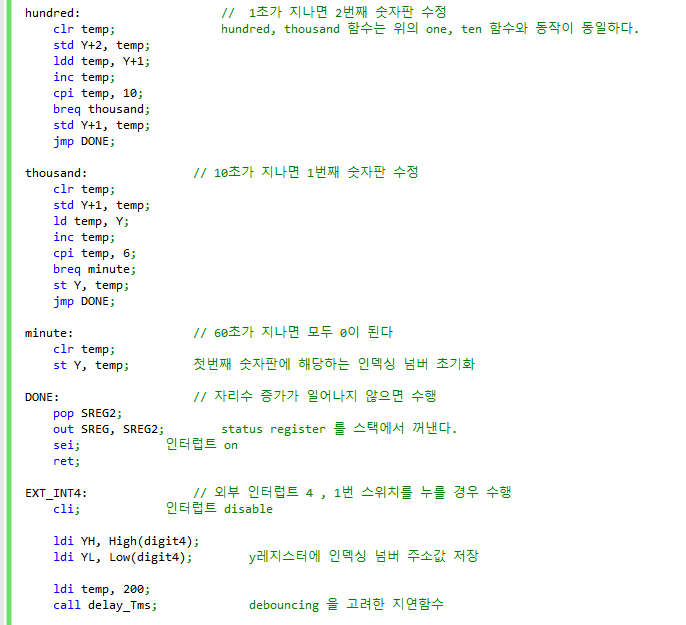
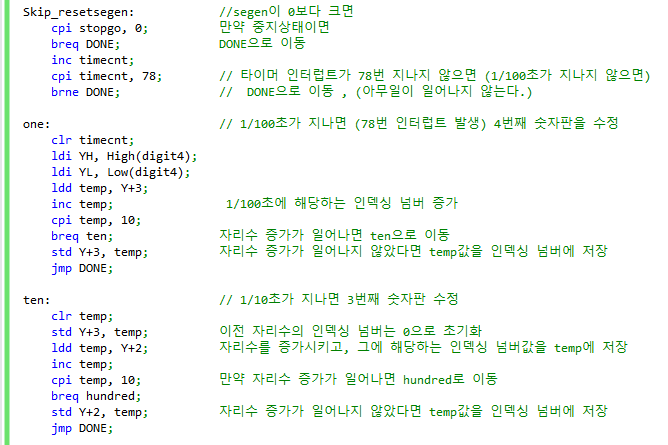
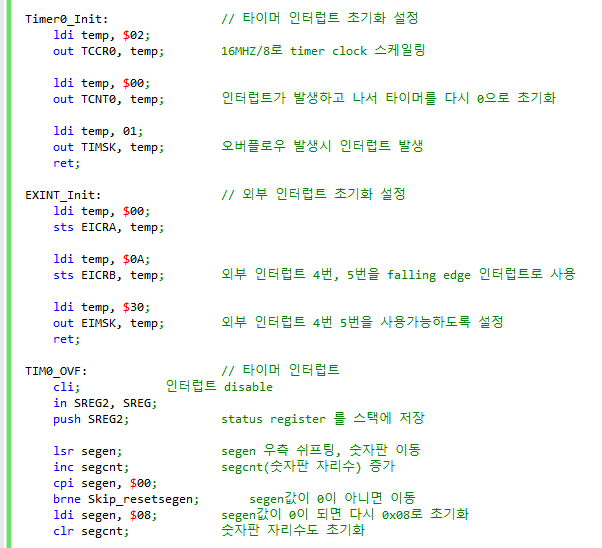
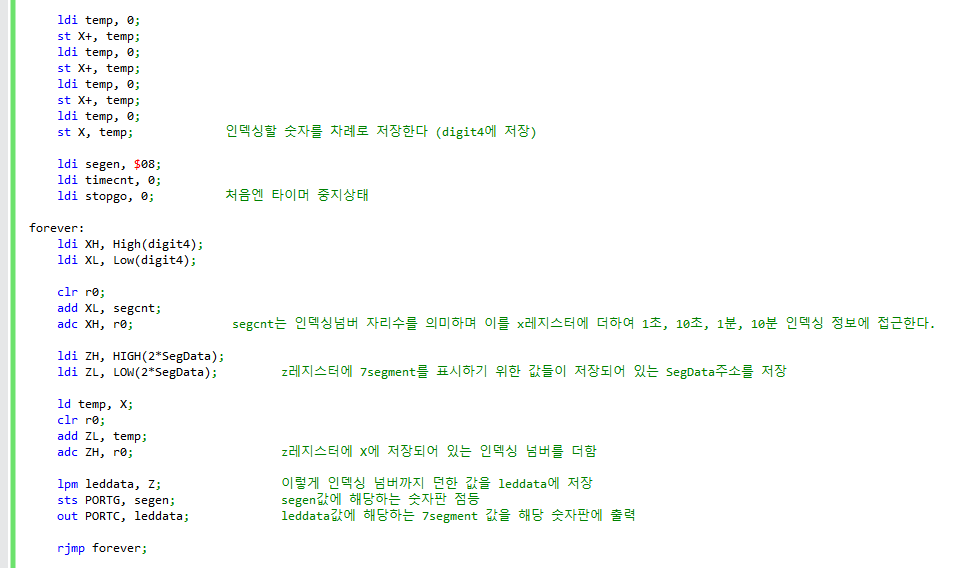
2)



lab 6에서 수행하였던 초 단위 타이머를 구성하고 스위치1을 눌러 (외부 인터럽트 4) 인덱싱 넘버값들을 0으로 초기화하여 결국은 타이머를 0000으로 초기화하는 프로그램이다. 이때도 역시 prelab에서와 같이 debouncing을 고려하여 스위치를 누르고 딜레이를 주어 타이머 초기화가 중복되지 않도록 설정하였다. 복잡해 보이지만 스위치1번을 눌렀을 때의 동작만을 추가해주면 된다.

3)

lab 7-2)



1/100초부터 시작되는 타이머이기 때문에 이전에 1초부터 시작되는 타이머와 다른 clock 스케일링을 가진다. 이전 실습에서의 타이머는 TCCR0레지스터에 0x06(clk/256)값을 넣어주었는데 이렇게 하면 인터럽트 244번마다 1초이므로 1/100초는 2.44번이 되는데 인터럽트 카운트는 자연수만 가능하므로 더욱 정확성을 올리기 위하여 TCCR0레지스터에 0x02(clk/8)값을 넣어주었다. 이를 통해 인터럽트 78번 발생시 1/100초와 유사하게 만들 수 있다.

외부 인터럽트의 경우 1번 스위치를 누르면 0000으로 타이머 초기화 및 타이머 중지를 수행할 수 있고 2번 스위치를 누르면 타이머의 동작/ 일시정지 를 수행할 수 있다. 이는 stopgo 레지스터를 통해 수행하였다. stopgo레지스터는 0의 상태일 경우 타이머 인터럽트에서 시간초가 증가하지 않도록 설정하였고 , 2번 스위치를 누르면 stopgo 레지스터를 바꿔 타이머를 재개, 중지시킬 수 있도록 하였다.

어셈블리어를 통해 AVR을 컨트롤 하는 LAB6, LAB7은 처음에 실습할 땐 굉장히 어렵고 난해하게 느껴졌지만 AVR의 복잡해보이는 여러가지 포트와 포트를 컨트롤하는 레지스터들의 기능을 이해하고 나면 코딩 자체는 크게 어렵지 않음을 느꼈다. 따라서 AVR 실습에서의 최우선 목표는 이러한 AVR의 구조를 이해하는 것임을 느꼈다.