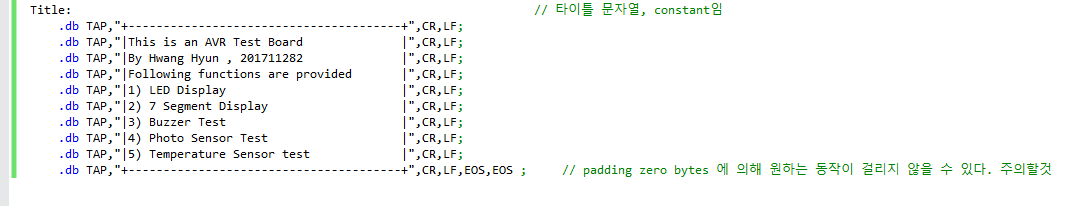
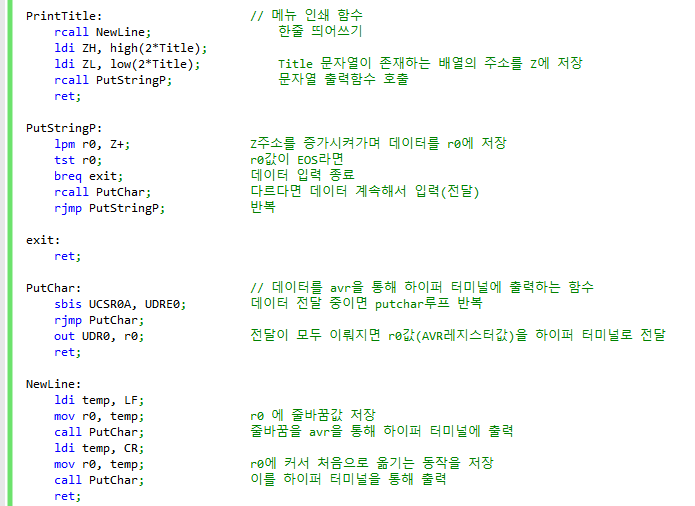
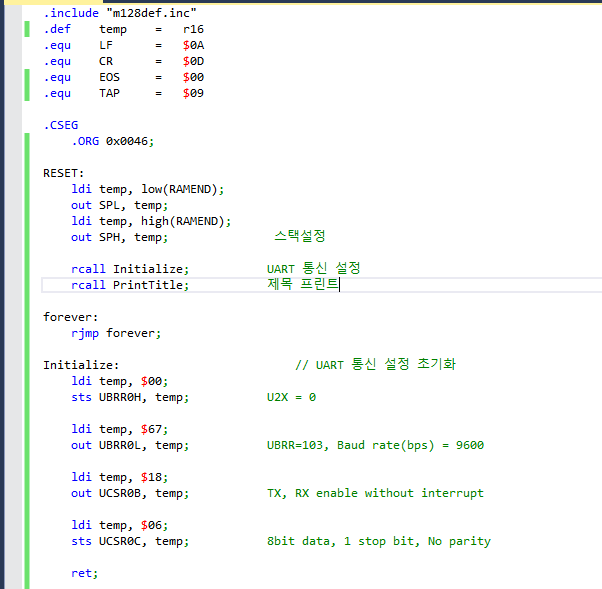
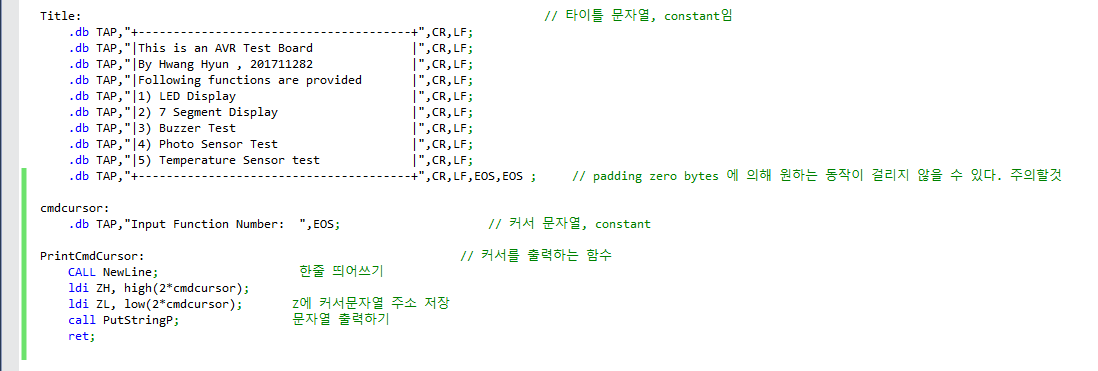
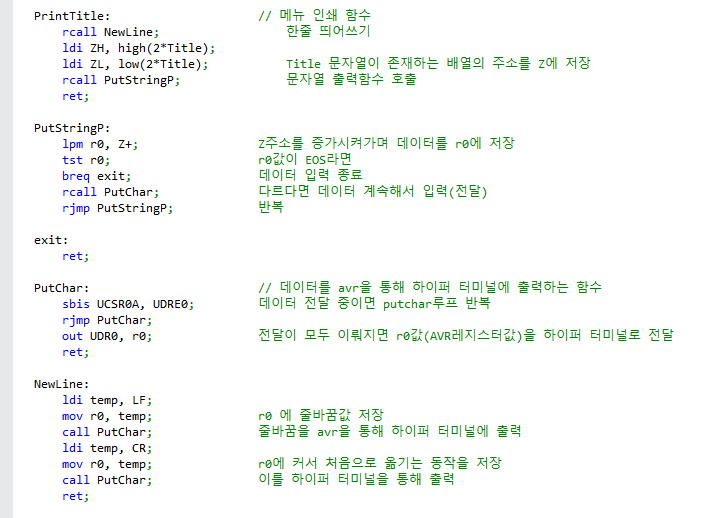
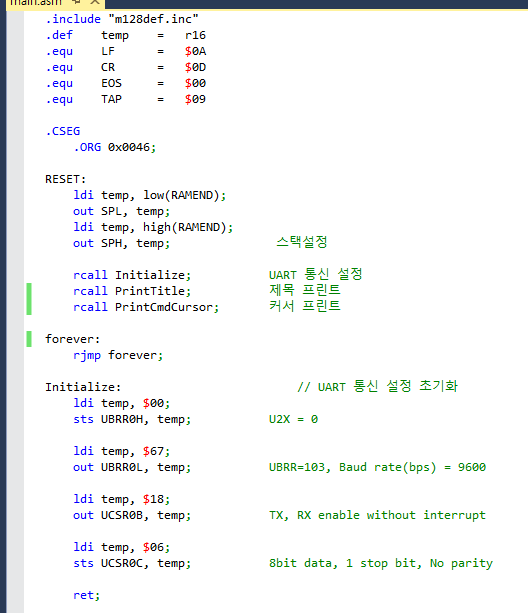
Lab 9-1



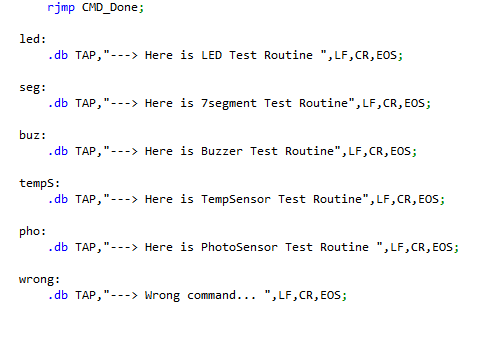
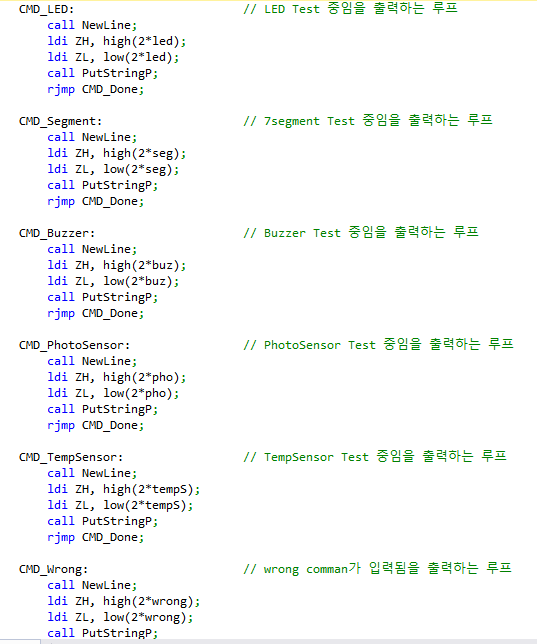
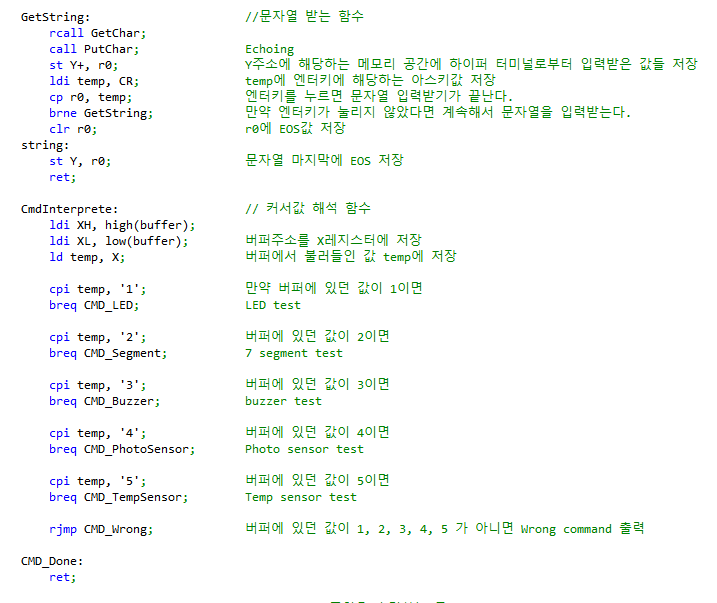
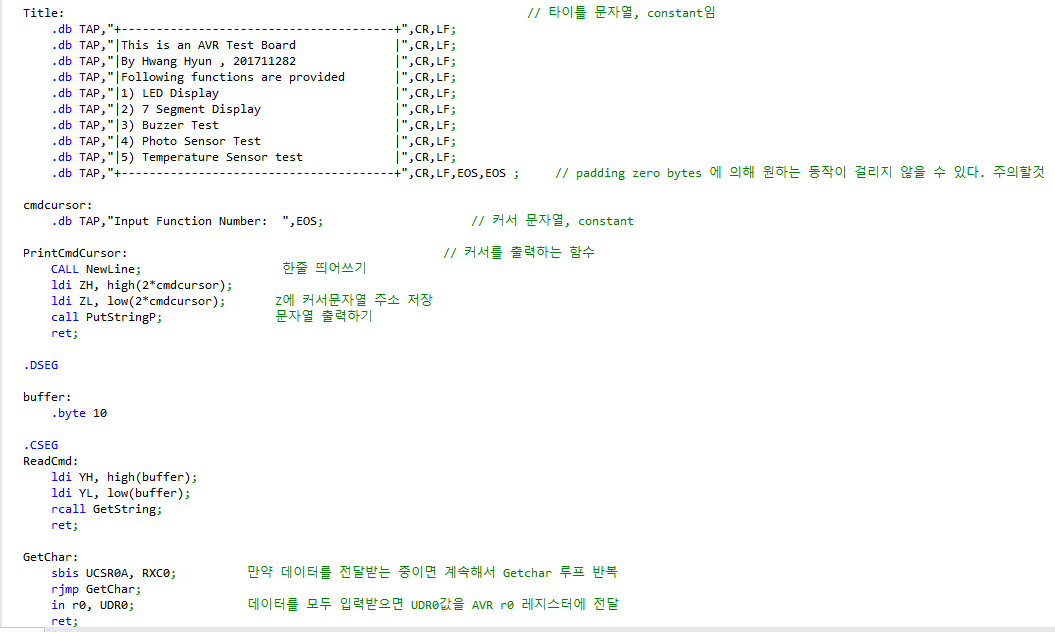
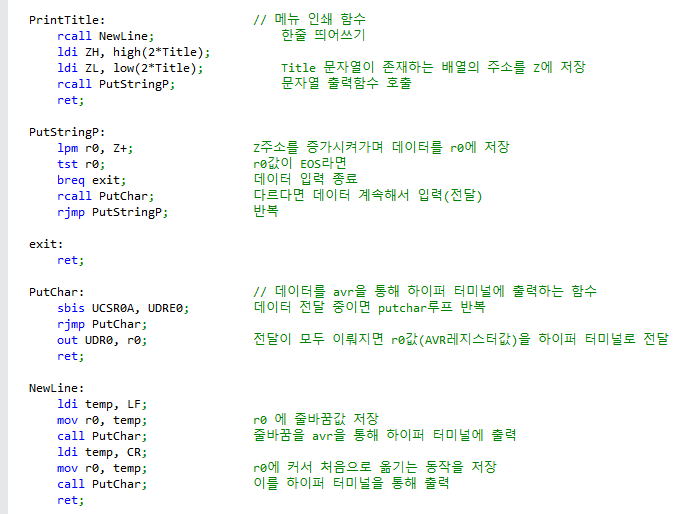
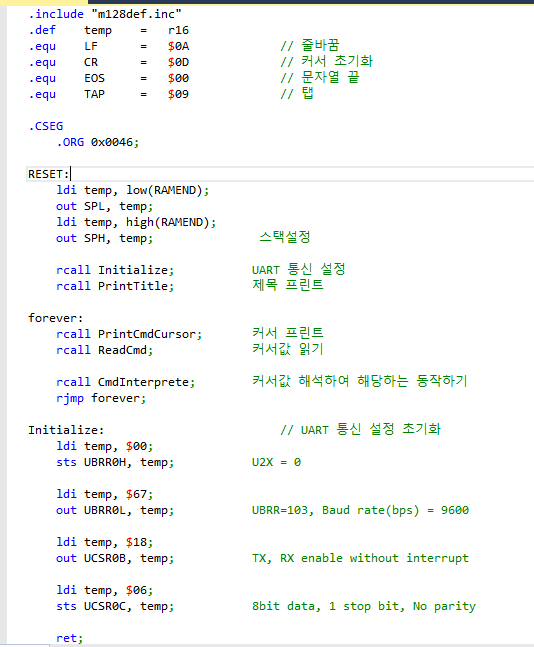
PutstringP 함수를 사용하여 메뉴부분을 출력하는 프로그램이다. 프로그래밍을 하면서 유의할 점은 PutstingP함수에서의 종료 조건을 r0 = EOS 일 때로 설정해놓았는데 만약 Title에서 문자열을 구성할 때 문자열의 길이가 홀수가 되어 padding zero bytes가 발생하면 중간에 EOS를 만난것처럼 문자열의 출력이 끊길 수 있다. 따라서 이를 유의하여 프로그래밍을 해주어야 한다.

lab 9-2



메뉴를 출력하는 lab 9-1 실습과 마찬가지로 lab 9-2 실습은 커서를 출력하는 프로그램을 만들면 된다. lab 9-1과 큰 차이는 없고 다만 유의할 점은 PutStingP에서 사용하는 주소는 Z레지스터를 통해 전달되기 때문에 커서문자열의 주소 역시 Z레지스터에 저장하여야 한다.

lab 9-3



AVR board TEST를 하는 실습으로 커서를 통해 원하는 메뉴값을 입력하면 입력값이 버퍼로 저장되고, 버퍼값을 CmdInterprete을 통해 해석하여 입력한 메뉴에 해당하는 TEST를 진행함을 나타내는 문자열을 출력한다. 이때 함수호출을 통해 입력한 메뉴에 해당하는 TEST를 진행하는 것이 아니고 점프와 같은 loop 컨트롤을 통해 해당 메뉴 TEST 루틴을 진행한 후 CMD\_Done으로 다시 점프하여 CmdInterprete 동작을 수행한다.

고찰

lab 8과 같이 이번 실습 역시 UART 통신을 다룬다는 점에서 크게 벗어나진 않는다. UART통신을 이용하여 하이퍼 터미널을 통해 AVR의 여러가지 기능들이 정상동작하는지 TEST를 할 수 있는 AVR TEST Board를 만드는 것이 이번 실습의 주요 목표였다. UART통신을 통해 문자를 입력받고, 문자를 출력하는 과정 자체는 저번 실습과 동일하나 이번 실습의 주요한 차이점은 입력받은 커서값을 해석하는 함수에서 커서값에 해당하는 메뉴, 즉 TEST를 수행할 때 해당 메뉴의 함수를 만들어서 함수를 호출하는 방식이 아니고, loop를 만들어 jump를 통해 해당 메뉴의 TEST 루틴을 수행하고 , 다시 jump를 통해 해석 함수를 마치는 방식으로 프로그래밍을 하였다는 점이다. 이렇게 한 이유는 함수 호출의 경우 함수 호출을 끝마치면 그 다음 명령어를 수행하는데, 이런 방식으로는 원하는 TEST만 진행할 수 없기 때문이다. 이번주 실습 역시 이러한 점만 주의하여 큰 어려움 없이 저번주 실습 내용을 활용하여 진행할 수 있었다.