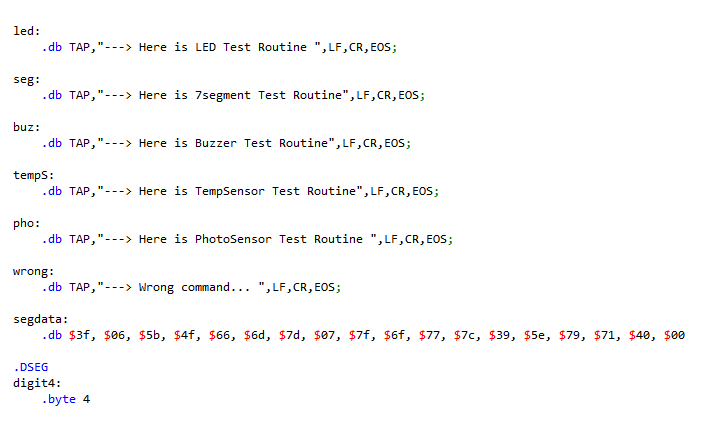
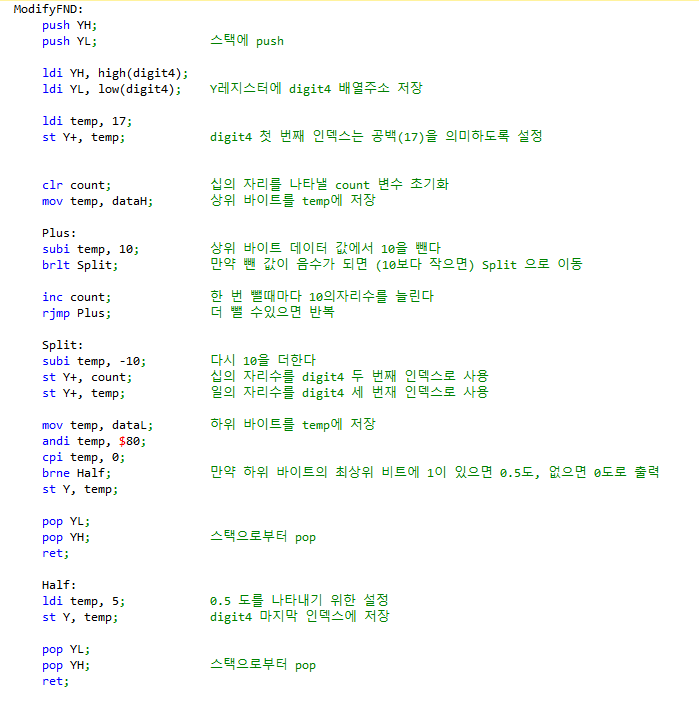
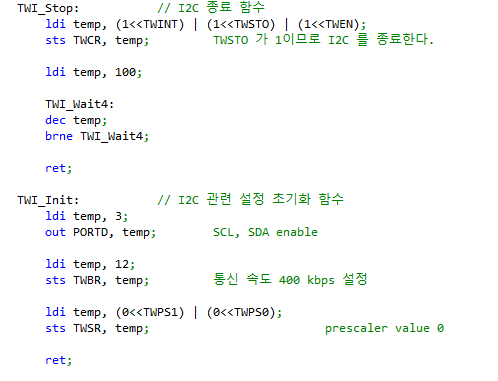
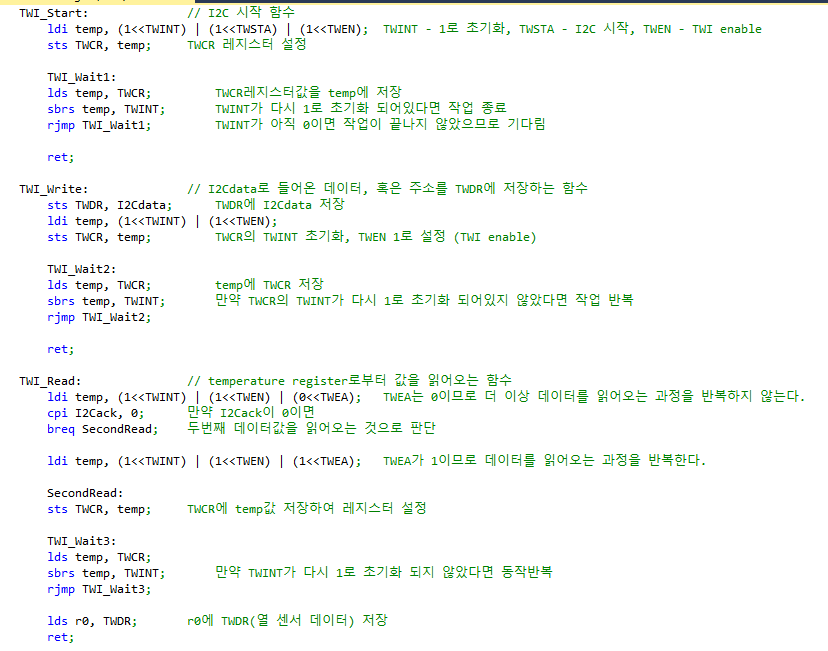
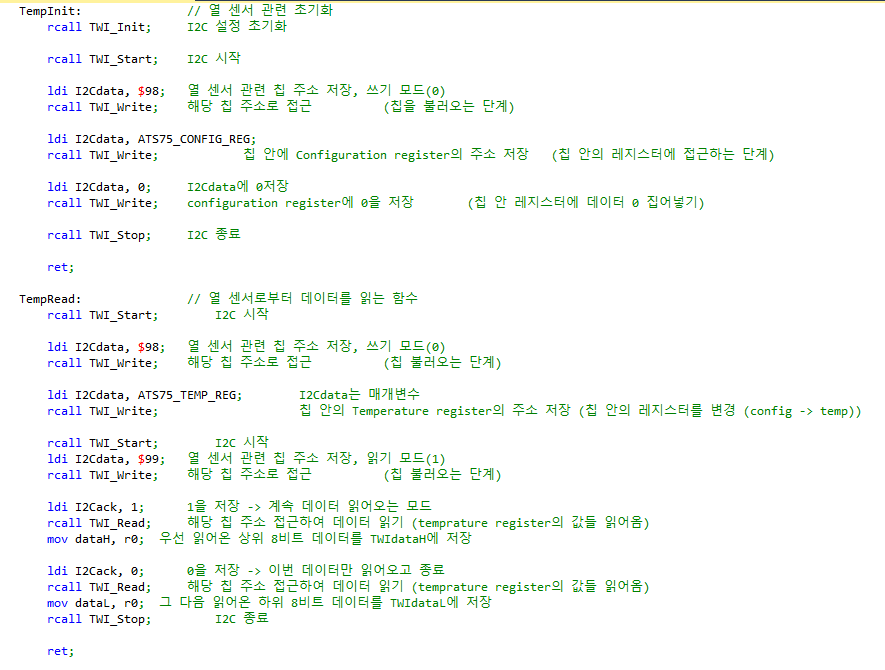
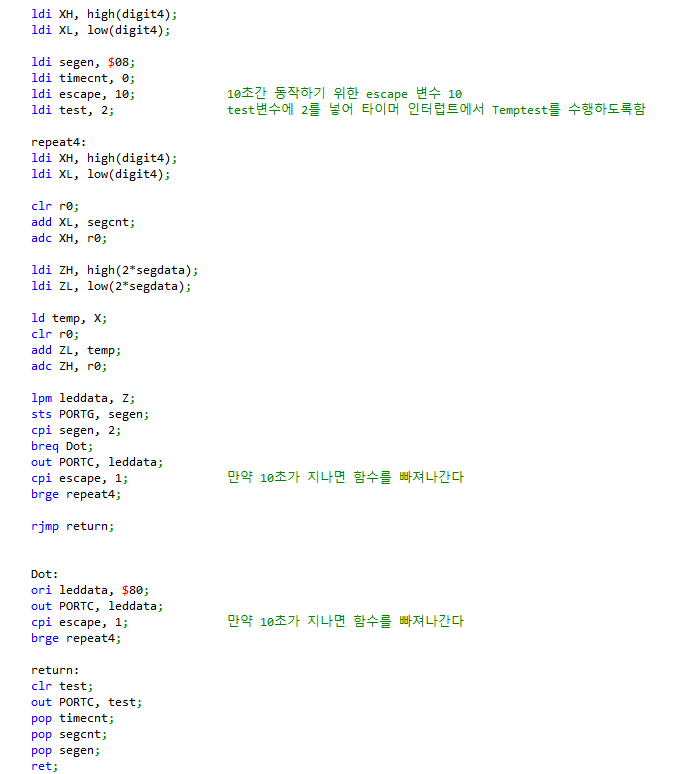
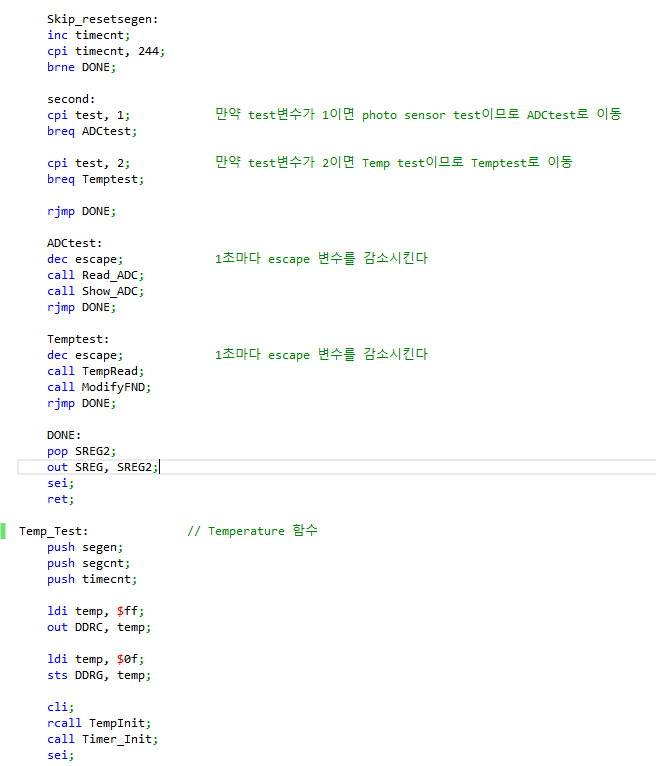
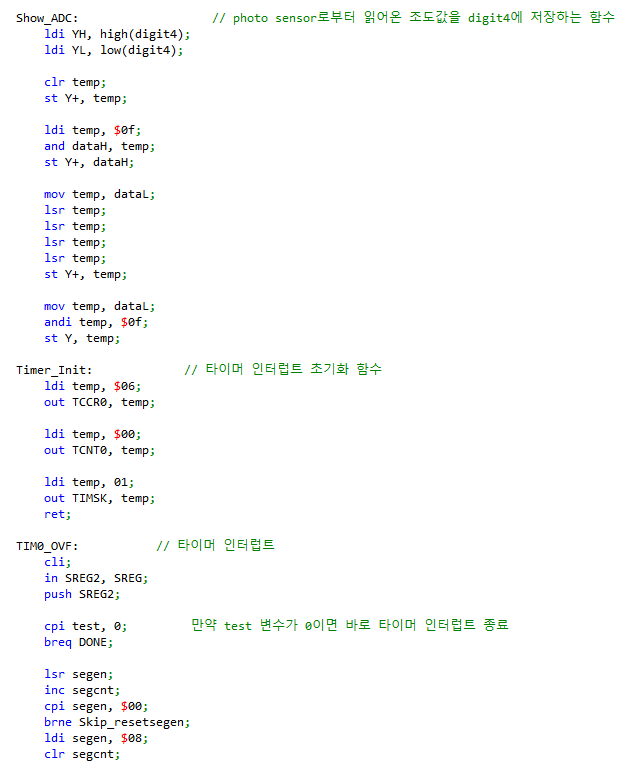
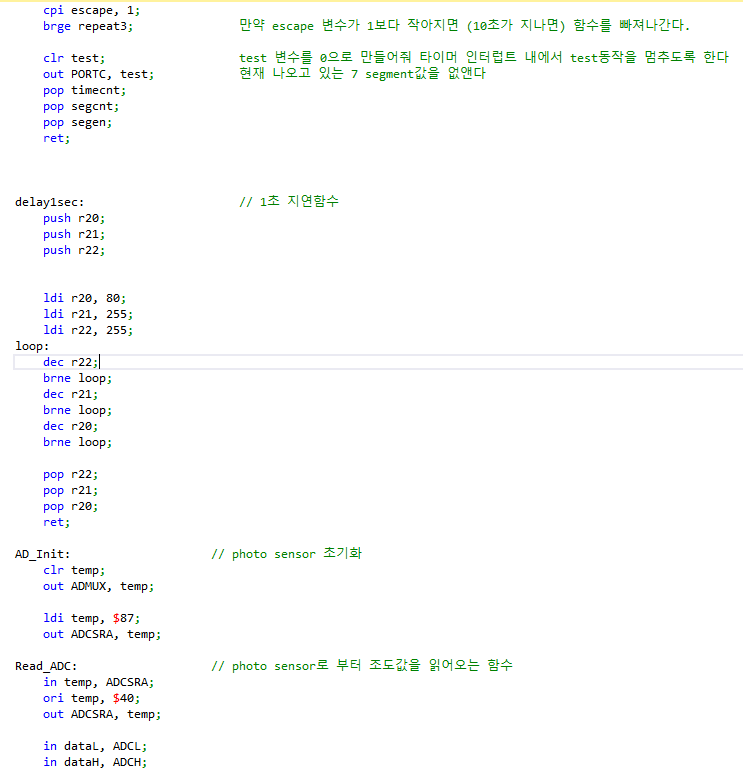
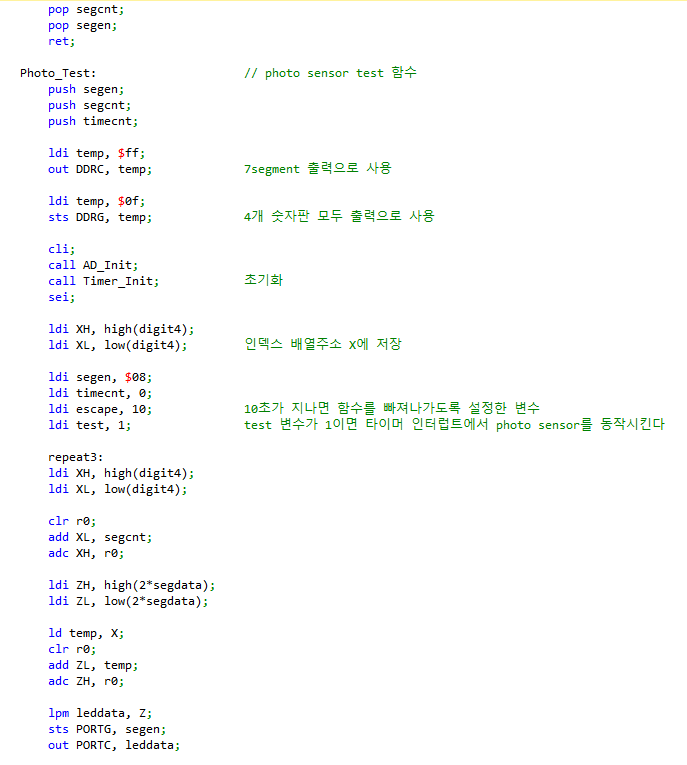
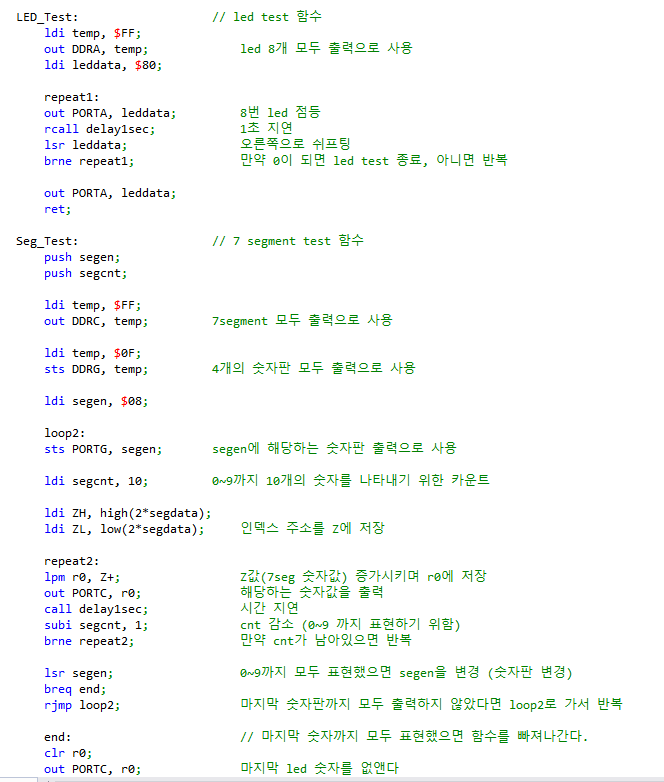
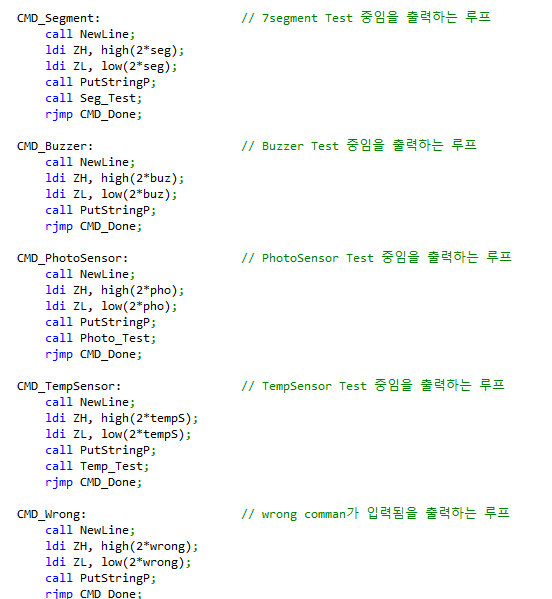
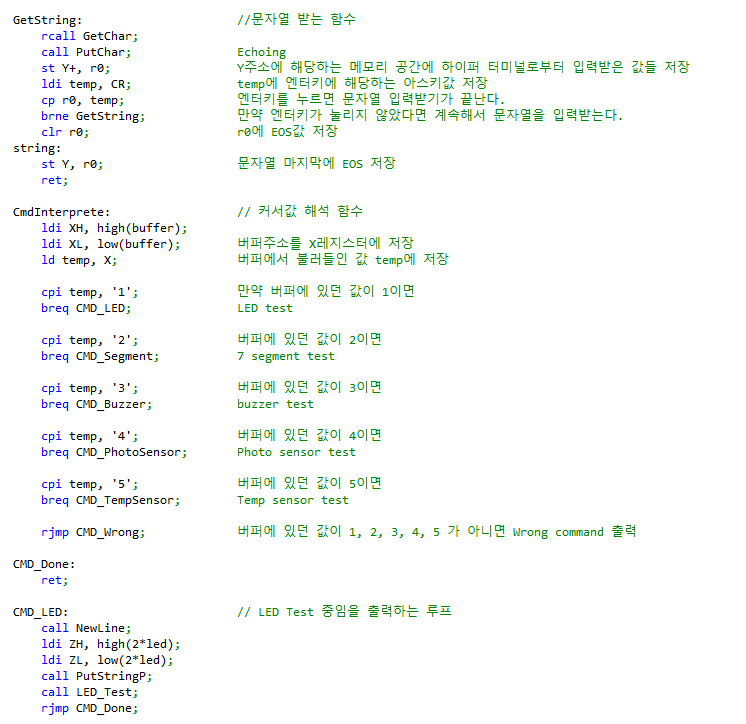
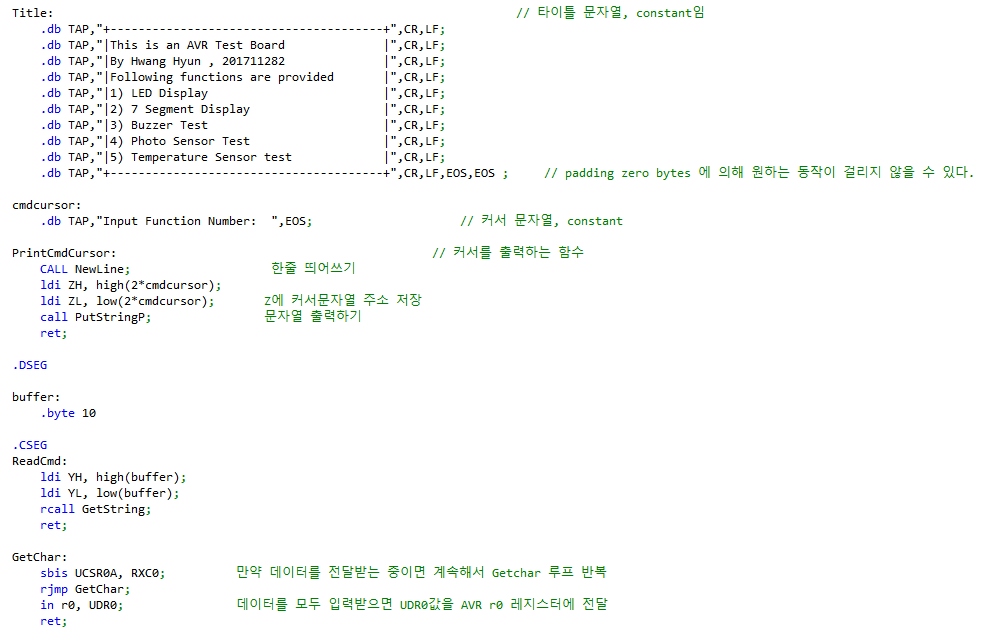
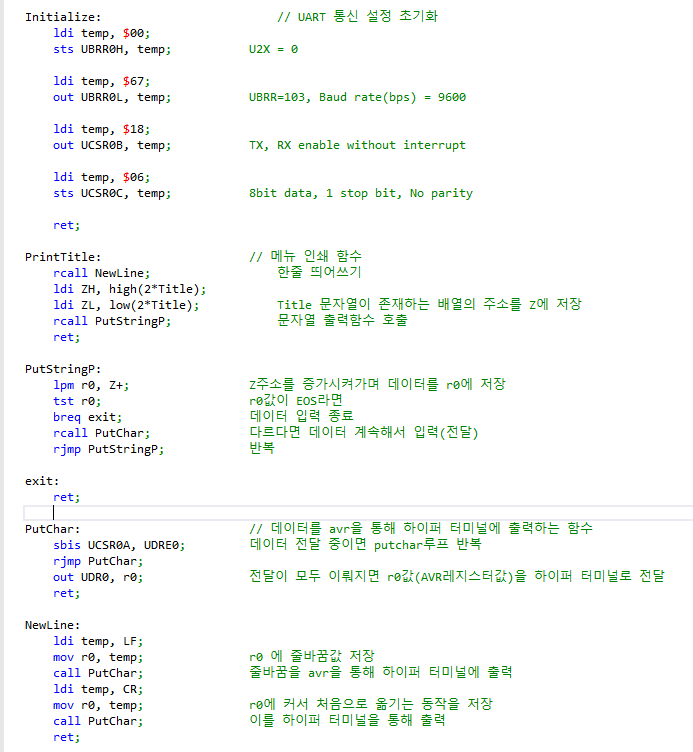
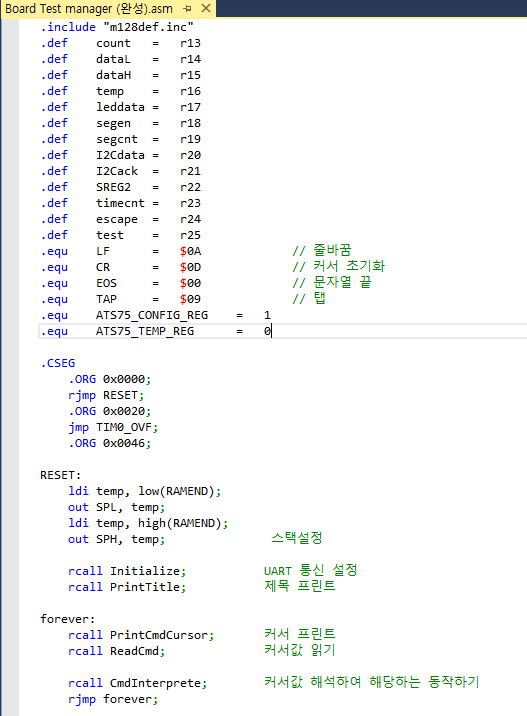
Board Test Manager



1번 test는 Led\_Test 함수의 호출로 수행하며 1초마다 8번 led부터 차례로 0번 led까지 점등시키는 test이다. 이는 delay1sec 함수를 사용하여 1초마다 점등시킬 led를 leddata레지스터의 lsr으로 구현하였으며 만약 leddata가 0이 되면 동작을 멈추고 led를 소등시켰다.

2번 test는 Seg\_Test 함수의 호출로 수행하며 0~9까지 모든 숫자판에 차례로 7segment값을 출력하는 test이다. 1초마다 7segment값이 바뀌는 것 역시 delay1sec 함수를 사용하여 수행하였으며 만약 0~9까지 숫자판에서 모두 번갈아 출력되면 숫자판을 lsr을 통해 바꾸는 방식이며 , 만약 마지막 숫자판까지 0~9까지의 7segment값 출력이 끝나면 모두 소등시키며 함수를 끝내도록 프로그래밍 하였다.

3번 test는 버저를 울리는 test로 따로 구현하지 않았다.

4번 5번 test는 test 변수에 따라 수행이 바뀐다. test 변수가 0이면 타이머 인터럽트내 아무런 동작도 하지않고, test 변수가 1이면 타이머 인터럽트내에서 초마다 photo sensor test를 수행하고, test 변수가 2이면 타이머 인터럽트내에서 초마다 temp sensor test를 수행한다.

4번 test는 photo sensor test로 Photo\_Test 함수를 호출하는 것으로 수행할 수 있다. 타이머 인터럽트를 이용하여 10초 동안 조도값을 읽어들이도록 설정하였다. 방법을 설명하자면 먼저 test 변수에 1, escape 변수에 10을 할당한다. 이를 이용해 타이머 인터럽트에서 1초마다 photo sensor test에 해당하는 루프로 이동, 필요한 조도값을 읽어들여 digit4에 저장, escape 변수값을 1씩 감소시킨다. Photo\_Test 함수내에서 escape 변수값이 1보다 작아지면 (10초가 지나면) test 변수를 0으로 만들고 함수를 빠져나가 test를 마친다.

5번 test는 temp sensor test로 Temp\_Test 함수를 호출하여 실행한다. 4번 test와 유사하게 타이머 인터럽트를 사용해 10초동안 주변환경의 온도값을 읽어들인다. 방법 역시 test 변수에 2, escape 변수에 10을 할당, 초마다 temp sensor test에 해당하는 루프로 이동해 주변 온도값을 읽어들여 digit4에 저장, escape 변수값을 1씩 감소시킨다. 만약 10초가 지나 escape 변수값이 1보다 작아지면 test 변수를 0으로 만들고 함수를 빠져나간다.

고찰

2학년 1학기에 논리회로와 2학년 2학기에 배웠던 컴퓨터구조 수업을 들었을 때 메모리구조, 레지스터, 어셈블리명령어등의 이론적인 수업을 할 땐 크게 이해가 되지 않고 뜬구름을 잡는듯했다. 3학년에 와서 마이크로 프로세스 응용을 배우면서 직접 어셈블리어 코딩을 수행하고, 메모리공간의 데이터가 저장되는 모습을 보면서 레지스터가 어떤 방식으로 데이터를 저장하고 가져오는지 , 데이터 메모리와 프로그램 메모리의 차이가 무엇인지를 명료하게 깨달았다. C언어보다 더욱 기계어에 가까운 어셈블리어는 처음엔 낯설고 불편하지만 익숙해지면 어떤 점에서는 C언어보다 더욱 직관적으로 메모리를 주고 받는다는 느낌을 느꼈다.

AVR 역시 처음엔 낯설고 어떻게 이런 기계를 컨트롤할 수 있을까 의문점이 많았는데 교수님께서 수업에서 알려주신대로 차례차례 실습을 따라하고 또 나름대로의 응용으로 과제들을 수행하면서 AVR을 다루는데 자신감이 생겼다. 특히 타이머 인터럽트의 개념은 처음엔 굉장히 이해하기 어려웠는데 타이머 인터럽트를 다양한 실습에서 여러 번 반복해서 사용하다 보니 정확한 시간제어동작이 필요할 때 큰 메리트가 있음을 느꼈다.

이외에도 UART 통신, Photo sensor, Temp sensor를 7segment를 이용해 표시하는 것도 동작을 눈으로 확인해볼 수 있어 흥미로웠고, hyperterminal을 이용해 Board Test Manager로 지금까지의 동작들을 복습, 정리할 수 있어 좋았다.

한학기 좋은 수업 감사드립니다.