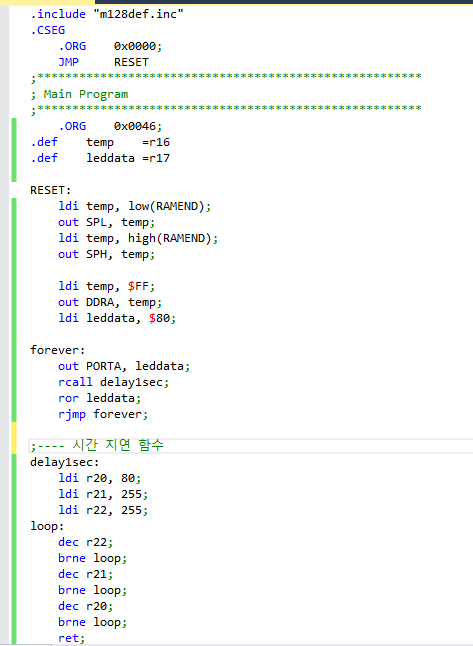
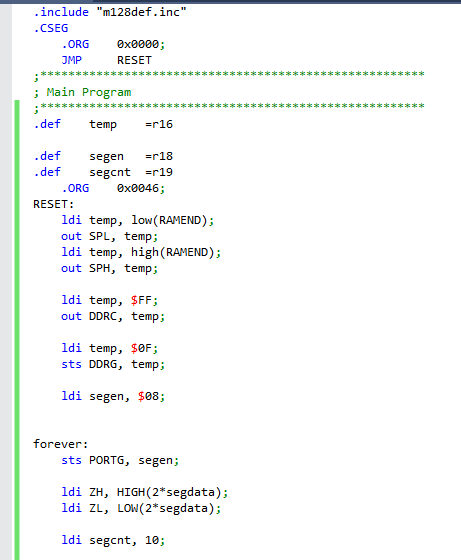
Lab 5-a

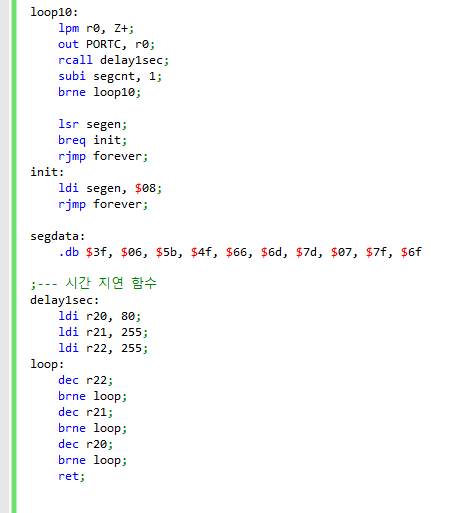


먼저 함수호출을 위한 스택포인터를 설정해준다. 그 뒤 temp 레지스터를 통해 1111 1111값을 DDRA에 넣어주어 8개의 LED를 모두 사용하도록 한다. leddata레지스터를 이용해 1000 0000 값을 PORTA 에 넣어주어 첫번째 LED에 불이들어오도록 설정한다. 그 뒤 1초의 시간지연을 위한 delay함수를 호출, 호출이 끝나면 ror 명령어를 통해 1000 0000 값을 우측으로 한 칸 쉬프팅한다. 이는 루프를 돌면서 두번째 LED에 불이 들어오게 하기 위함이며 rjmp forever를 통해 forever 루프를 무한 반복, ror명령어를 사용하였기 때문에 1은 사라지지 않고 계속해서 8비트내에서 우측으로 쉬프팅하게 된다. (만약 LSB에서 쉬프팅이 발생하면 MSB로 다시 1이 이동한다.) 이를 통해 과제에서 원하는 동작을 수행할 수 있다.

Lab 5-b



함수의 호출을 위한 스택포인터를 설정한다. 그 후 temp레지스터를 통해 DDRC에 1111 1111을 전달, 7segment 모든 LED를 사용할 수 있도록 한다. 또한 7segment 숫자 LED 4개를 제어하기 위한 G 포트, 즉 temp 레지스터를 이용하여 DDRG 에 0000 1111 값을 넣어준다. 그 뒤 segen 레지스터에 0000 1000 을 넣어주고 sts PORTG, segen 명령어를 통해 첫 번째 숫자판에만 불이 들어오도록 한다. Z레지스터에는 숫자판에 0~9가 들어오도록 하기 위한 배열주소를 할당한다.



루프를 이용하여 PORTC에 segdata 배열값들이 차례 차례 오도록 설정한다. 이를 통해 0~9값이 숫자판에 출력된다. lsr segen을 통해 0000 1000 을 우측으로 쉬프팅한다. 이를 통해

sts PORTG, segen 는 두번째 숫자판의 불만 들어오게 한다. lsr segen 을 반복하다 0000 0000 이 되면 breq init 을 통해 다시금 0000 1000 으로 segen 값을 변경, 첫 번째 숫자판에만 불이 들어오도록 설정한다.

고찰

어셈블리어로 작성한 코드를 이용하여 마이크로 프로세서, ATmega 128을 제어하는 실습과제를 수행하였다. 마이크로 프로세서는 CPU와 여러 제어장치, 출력장치를 섞어놓은 하나의 처리장치로서 여러가지 기능들을 수행하게 되는데 이를 손으로 직접제어하는 것은 굉장히 번거롭고 불편할 일일 것이다. 사람의 손으로 일일히 회로를 구성하여 7segment에 불이 들어오게 하는 실험은 2학년 2학기 전기전자기초실험에서도 진행하였었는데 굉장히 번거롭고, (사람손으로 직접구성하기 때문에) 오류도 많이 발생하며, 시간을 많이 잡아먹는 일이었다. 반면 이번 실습에선 USB 포트만 연결하여 컴퓨터로 쉽게 작성한 코드들을 디바이스에 전달해주었기 때문에 무엇보다도 쉽고 빠르게 원하는 동작들을 수행할 수 있었다. 물론 코드를 이용한 일련의 동작과정들을 정확히 이해하는 것 역시 필요하다.