Lab 2. 스테핑 모터 제어

실습 1. GPIO를 이용한 스테핑 모터 구동

1. 홈페이지에 제공된 DSP code를 다운로드하여 압축을 풀어 사용한다. 배포된 코드는 skeleton에 해당하며, 1상여자 구동방식을 적용하여 1초에 1 step씩 무한히 정회전 또는 역회전하는 코드로 배포되었다. 각 상에서 일정한 시간의 delay를 구현하였다. 아울러 타이머 인터럽트는 100 kHz로 구현되었다. WaitTFlagCnt() 함수의 설정값 1은 10usec delay에 해당한다.
2. Delay time을 변경하여 모터의 회전 속도를 변경해본다. Delay time이 과도하게 짧아서 모터가 회전하지 않는 상태까지 가변해본다.
3. 2에서 delay가 짧아서 회전하지 않는 상태에서 손으로 모터를 회전시켜보자. 일단 회전하고 나면 회전이 잘 이루어지는 것을 확인할 수 있다. 원인을 생각해보시오.
4. 스테핑 모터 구동회로는 내장 FPGA에 구현하였다. 관련 FPGA 레지스터는 다음과 같다.

Register	Bit 31 ~ 8	Bit 7 ~ 4				Bit 3 ~ 0			
STEPPER (0x2000024)	Reserved to 0	Left Stepping Motor Phase				Right Stepping Motor Phase			
		A	/A	B	/B	B	/B	A	/A

5. 배포된 코드는 Full Step 구동으로 작성되었다. Half Step 구동 방식으로 변경하여 적용해보시오.
6. Full Step과 Half Step 각각에서 모터 1 회전에 몇 스텝이 필요한지 실험적으로 확인해보시오. (실험에 사용된 스텝 모터의 스텝각은 1.8 deg 임을 고려하여 결과와 비교할 것)

실습 2. 스테핑 모터 구동 API 작성

1. 실습2 부터는 Full step 구동으로 구현한다.
2. 배포된 코드에는 시간 지연 기능으로 delay_ms() 함수도 있다. 하지만, delay_ms() 함수에서는 단순한 for 문을 사용하므로 delay time이 정확하지 않다. 본 과제에서는 시간 지연 함수로 WaitTFlagCnt() 함수를 사용한다. WaitTFlagCnt() 함수가 어떻게 동작하는지 코드를 분석해본다. 참고로 배포된 코드에서 타이머 인터럽트는 100kHz로 설정되었으므로 WaitTFlagCnt(1e5); 을 실행하면 1초 delay 된다.
3. 스테핑 모터를 정방향 또는 역방향으로 1 스텝 회전하는 함수로 void OneStepMove(unsigned int dir, unsigned int tDelayCnt) 를 작성한다. dir이 0이면 정방향, 1이면 역방향으로 회전한다. (정방향과 역방향은 각자 정의한다) 현재 인가된 상의 상태를 알아야 다음 상태를 생성할 수 있음에 유의한다. 모터에 인가하는 상을 변환한 후 tDelayCnt에 지정한 횟수 만큼 타이머 인터럽트가 발생할 때까지 대기한 후 함수에서 return한다.
4. void StepMoveCV(float angle, float spd) 함수를 작성한다. angle에 지정한 스텝 만큼 회전하되, 0보다 크면 정방향, 0보다 작으면 역방향으로 회전한다. 넹는 회전 스피드에 해당하며, deg/s (degree per sec) 단위로 입력한다. 함수 내부에서 angle과 spd로부터 회전 방향과 스텝 수, 그리고 스텝간 시간지연을 타이머 인터럽트 주기 수로 환산한 후 1에서 작성한 OneStepMove를 호출하여 구현한다.

실습 3. 속도 프로파일의 적용

1. 앞의 실습 2에서는 일정 속도로 스테핑 모터를 구동하는 실습을 진행하였다. 하지만, 처음부터 너무 빠른 속도로 구동을 시도할 경우에 탈조가 발생하여 의도한 대로 회전하지 않는 증상을 확인할 수 있다. 즉, spd 값을 너무 크게 설정하면 탈조가 발생한다.
2. 일정 각가속도로 속도를 증가시키는 사다리꼴 속도 프로파일을 적용시키는 방법에 대하여 생각하고, 이를 적용하여 함수 void StepMoveVP(float angle, float maxVel, float accel) 를 다음의 절차에 따라 작성한다. angle은 양/음수에 따라 회전 방향이 결정되지만, maxVel과 accel은 절대값으로만 사용한다.
3. 사다리꼴 속도 프로파일의 결정을 위해서는 가속도, 최대속도, 그리고 회전할 각도 등 3가지로부터 룩업테이블을 작성하여야 한다. 룩업테이블에는 각 스텝간 시간 지연을 타이머 인터럽트 발생 횟수로 저장한다. 전체 이동 스텝에 대하여 작성할 필요는 없으며, 가속 구간에 대해서만 작성한다. 이를 담당하는 함수로 unsigned int MakeVelProfile(float maxVel, float accel) 를 작성한다. 룩업테이블은 unsigned int array type으로 200개 정도의 array를 사용한다. 함수에서 200개 이상의 시간지연 값을 기록하지 않도록 주의하여야 하며, 가속 구간에서 이동할 스텝 수를 return 한다. 가속 구간의 스텝 수는 절대로 200을 넘어서는 안되며, 생성된 룩업테이블을 MACRO_PRINT로 출력하여 이상 없음을 **반드시** 확인한다. 실제로 잘 튜닝된 스텝 모터 구동에서는 가속 구간의 스텝 수가 일반적인 경우에 최대 50 스텝을 넘어가지 않는다.
4. StepMoveVP에서는 MakeVelProfile 함수를 호출하여 룩업테이블을 생성하고, 생성된 룩업테이블에 저장된 스텝 지연 시간과 가속 구간의 스텝 수를 활용하여 스텝 모터를 회전시킨다. 가속 구간의 스텝 수를 이용하면 전체 이동 스텝 중 현재가 가속, 등속, 감속 구간 중 어디에 속하는지 확인할 수 있다. 가속 구간에서는 매 스텝 이동 마다 룩업테이블을 읽어오는 인덱스를 증가시키고, 등속에서는 계속 같은 위치의 tDelayCnt 값을 읽어오며, 감속에서는 인덱스를 감소시키며 읽어온다.
5. 스텝 모터가 1회전하도록 StepMoveVP 함수를 호출하되, maxVel과 accel 값은 너무 크지 않은 값으로 적용하여 구동한다. 일단 1회전이 정상적으로 회전하면 maxVel과 accel을 점차 증가시켜서 탈조가 발생하는 maxVel과 accel 값을 확인한다. 그 값의 70%에 해당하는 값을 적용하면 안전하게 스텝 모터를 구동할 수 있다. 경우에 따라 기구의 공진 주파수와 상전환 주파수가 일치하면 진동으로 인해 탈조가 발생하기도 하므로 이 경우에 해당하는지 확인이 필요한다.