

상태 머신 디자인 패턴을 익힌다.

CHAPTER 2. 디자인 패턴 II

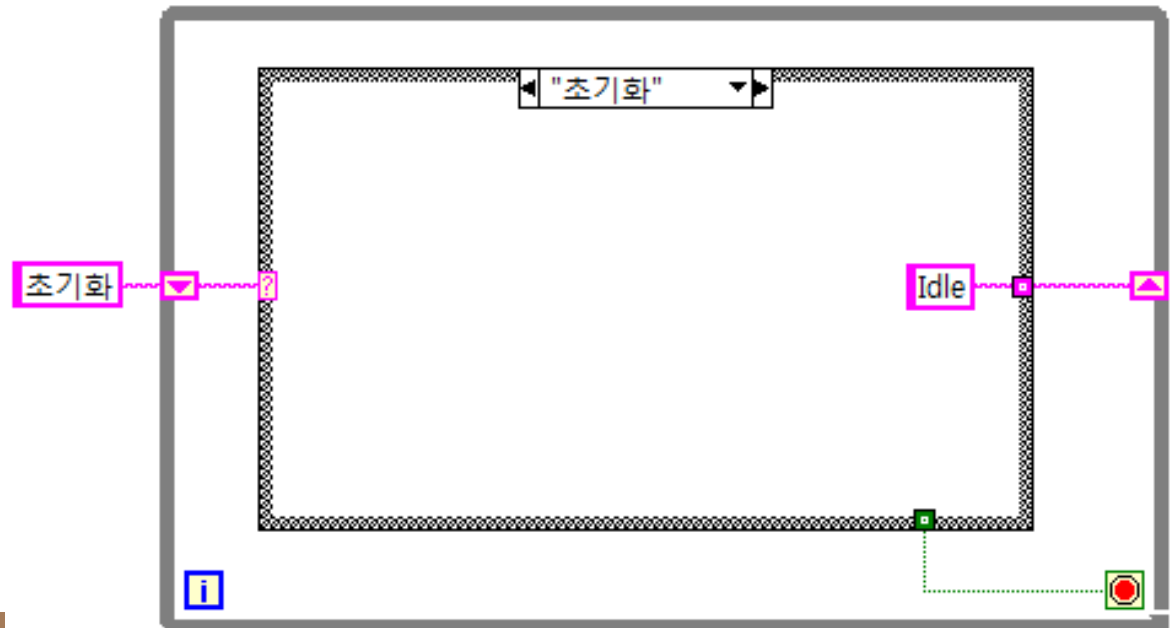
1. 상태 머신 디자인 패턴

일반 상태 머신

표준 상태 머신

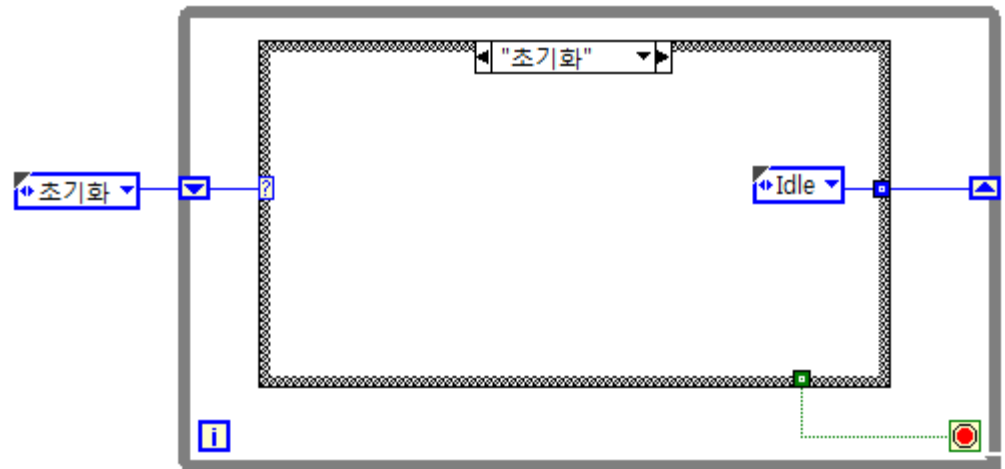
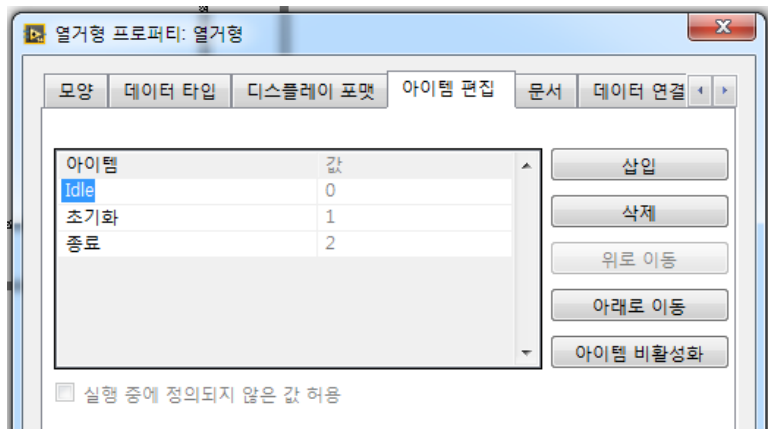
1. 일반 상태 머신

- State Machine의 구현
 - While 루프
 - 케이스 구조
 - 시프트 레지스터
 - 전환 코드 : 문자열



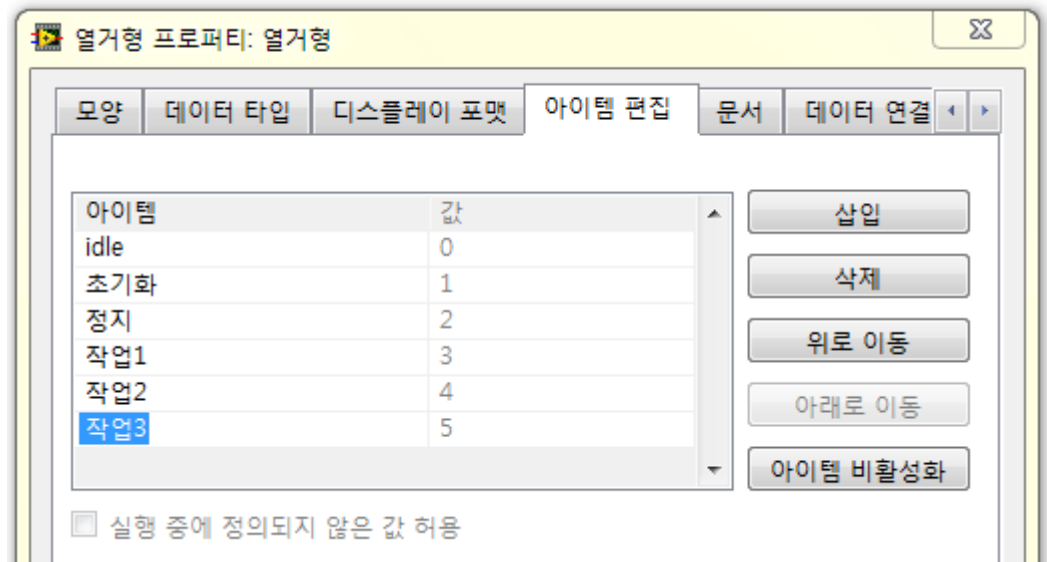
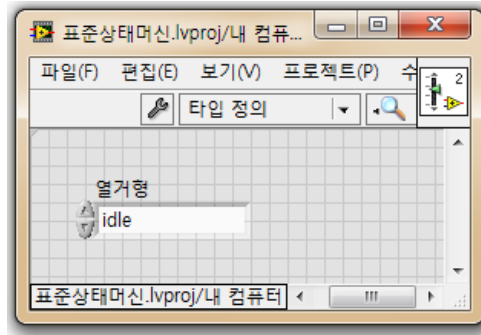
2. 표준 상태 머신

- Standard State Machine의 구현
 - While 루프
 - 케이스 구조
 - 시프트 레지스터
 - 전환 코드 : 타입 정의된 열거형 상수



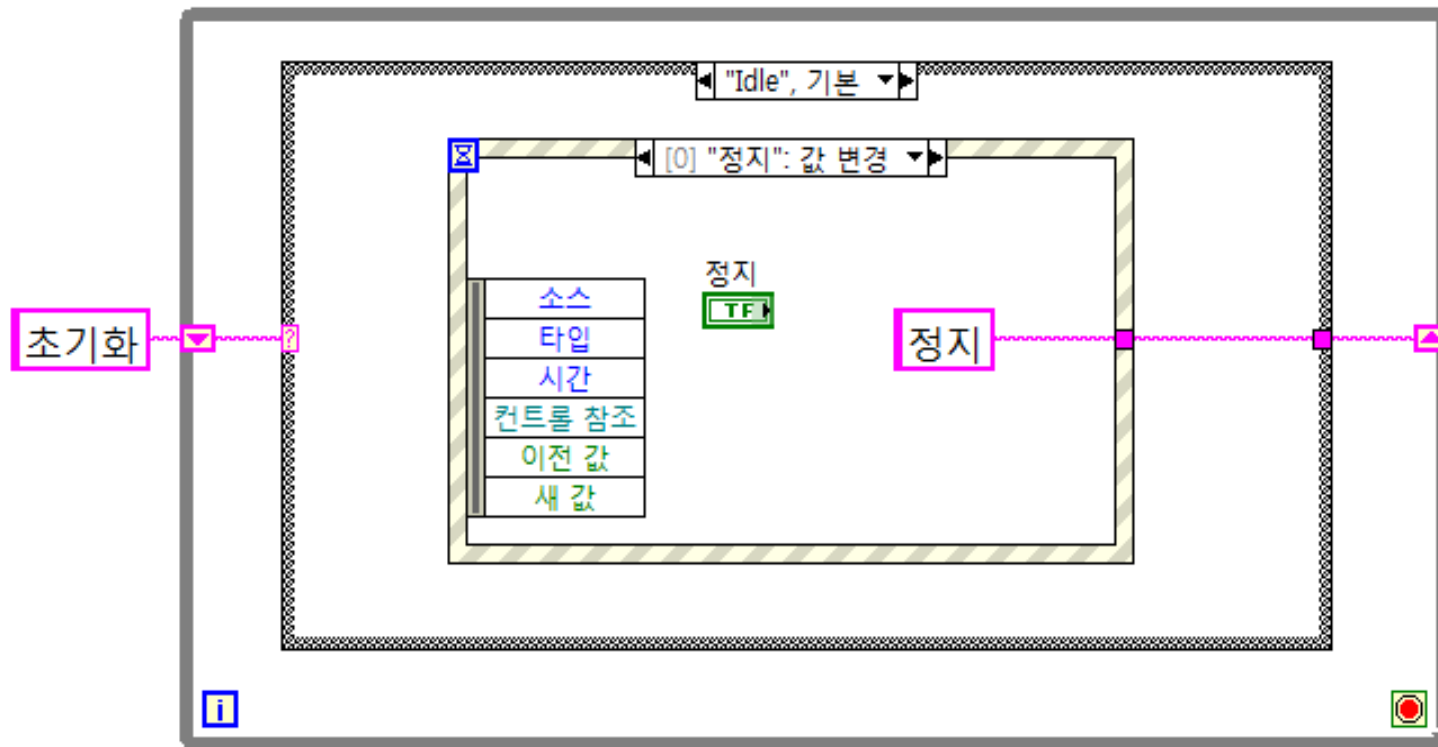
타입 정의 열거형 상수

- 숫자형 팔레트
- 왼쪽 상단에 검은색 삼각형 표시가 있음
- 타입 정의 만들기
- 아이템 편집



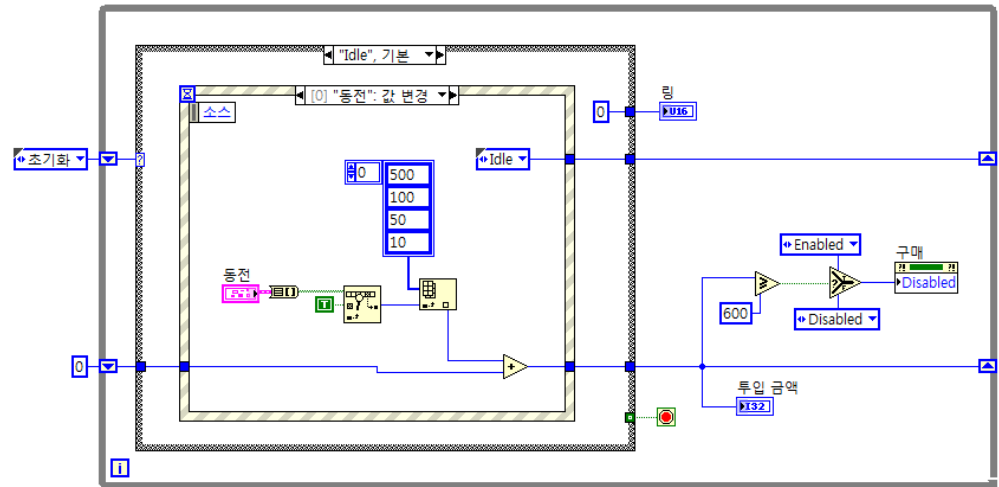
실습 2-1: 상태 머신

일반 상태 머신 - 문자열



실습 2-2: 콜라 자판기

표준 상태 머신 – 타입 정의된 열거형 상수

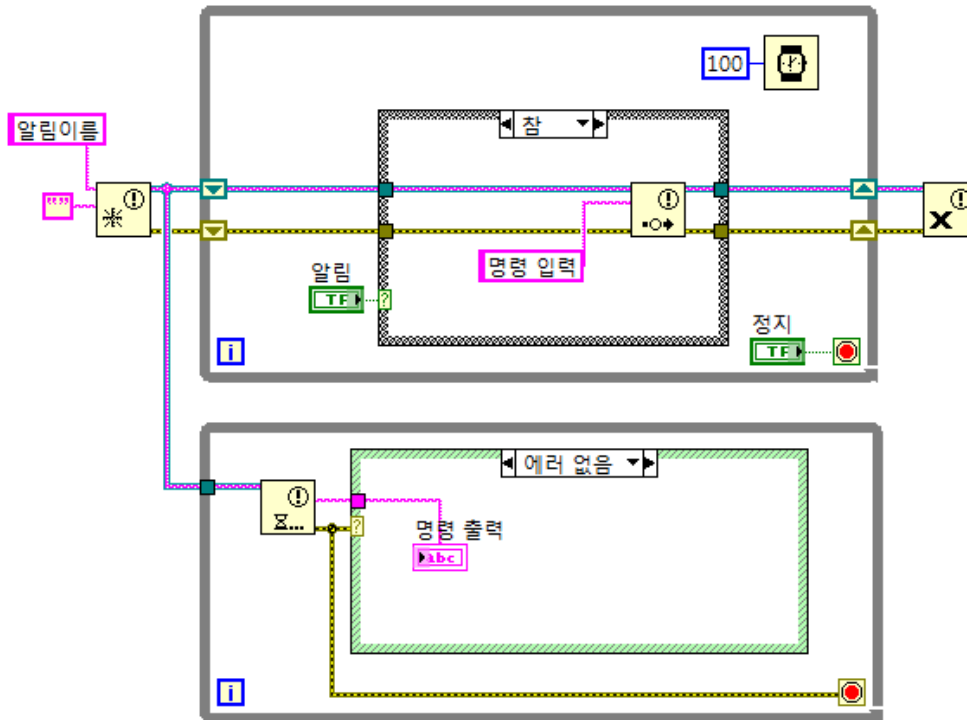


2. 마스터/슬레이브 디자인 패턴

알림자 이용

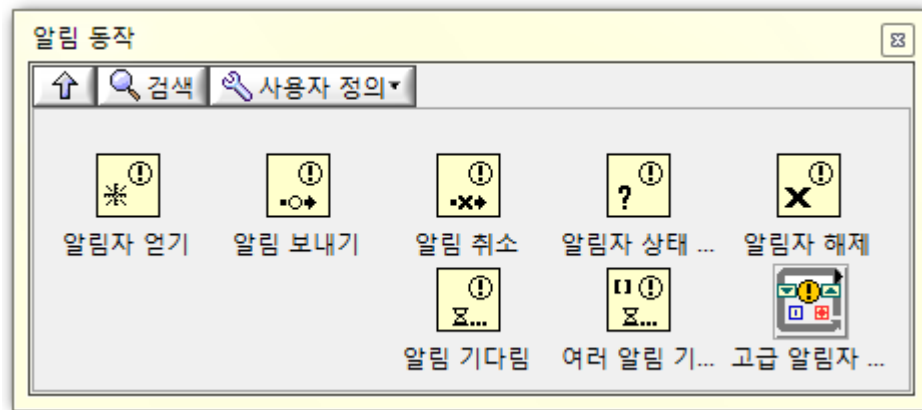
마스터/슬레이브 디자인 패턴

- 여러 개의 병렬 루프 간의 동기화
 - 한 개의 마스터 루프와 여러 개의 슬레이브 루프



알림 (Notify)

- 알림 동작 팔레트
 - 마스터: 알림 보내기
 - 슬레이브: 알림 기다림



알림의 특징

- 한 개의 데이터만 전달이 가능함.
- 동시에 같은 공유 데이터에 쓰면 안됨.
- 슬레이브는 마지막 메시지만 받음 → 데이터 손실이 있을 수 있음.
- 마스터 루프와 슬레이브 루프를 동기화함.
- 알림 확인을 위하여 **Polling**하지 않음.
- 알림자는 데이터를 버퍼링하지 않는다.

알림자 – 장점

병렬 루프 사이에 데이터를 전달하는데 알림자를 사용하여 얻을 수 있는 장점은 다음과 같습니다:

- 두 루프 모두 마스터 루프에 동기화되며, 슬레이브 루프는 마스터 루프가 알림을 보낼 때에만 실행
- 알림자를 사용하여 글로벌로 사용할 수 있는 데이터를 생성하고, 알림과 함께 데이터를 보낼 수 있음
- 알림자를 사용하여 효과적인 코드를 생성 – 마스터 루프에서 언제 데이터를 사용할 수 있는지 확인하기 위해 폴링을 할 필요가 없음

알림자 – 단점

- 알림자는 데이터를 버퍼링하지 않습니다.
- 슬레이브 루프가 데이터의 읽기 전에 마스터 루프가 추가적인 데이터를 보낼 경우, 데이터는 덮어쓰기 되므로 이전 데이터가 손실됩니다.

실습 1-2: 마스터 슬레이브 디자인 패턴

- 알림을 이용하여 두 개의 루프를 동기화한다.
- 마스터 루프에서 보낸 메시지를 슬레이브 루프에서 팝업 창으로 알려준다.