

클러스터에 대하여 배웁니다.
클러스터 함수 사용법을 익힙니다.
케이스 구조에 대하여 배웁니다.

CH 5. 클러스터와 케이스 구조

1. 클러스터

클러스터의 정의와 사용법

클러스터

- 클러스터는 다양한 타입의 데이터를 묶어서 그룹화한다.
- 클러스터는 한 방향을 향한다.
 - 컨트롤 클러스터와 인디케이터 클러스터로 구현된다.
 - LabVIEW의 데이터 흐름 원칙을 준수한다.
- 클러스터는 데이터의 묶음이다.
 - 묶어 주는 순서

클러스터 만들기

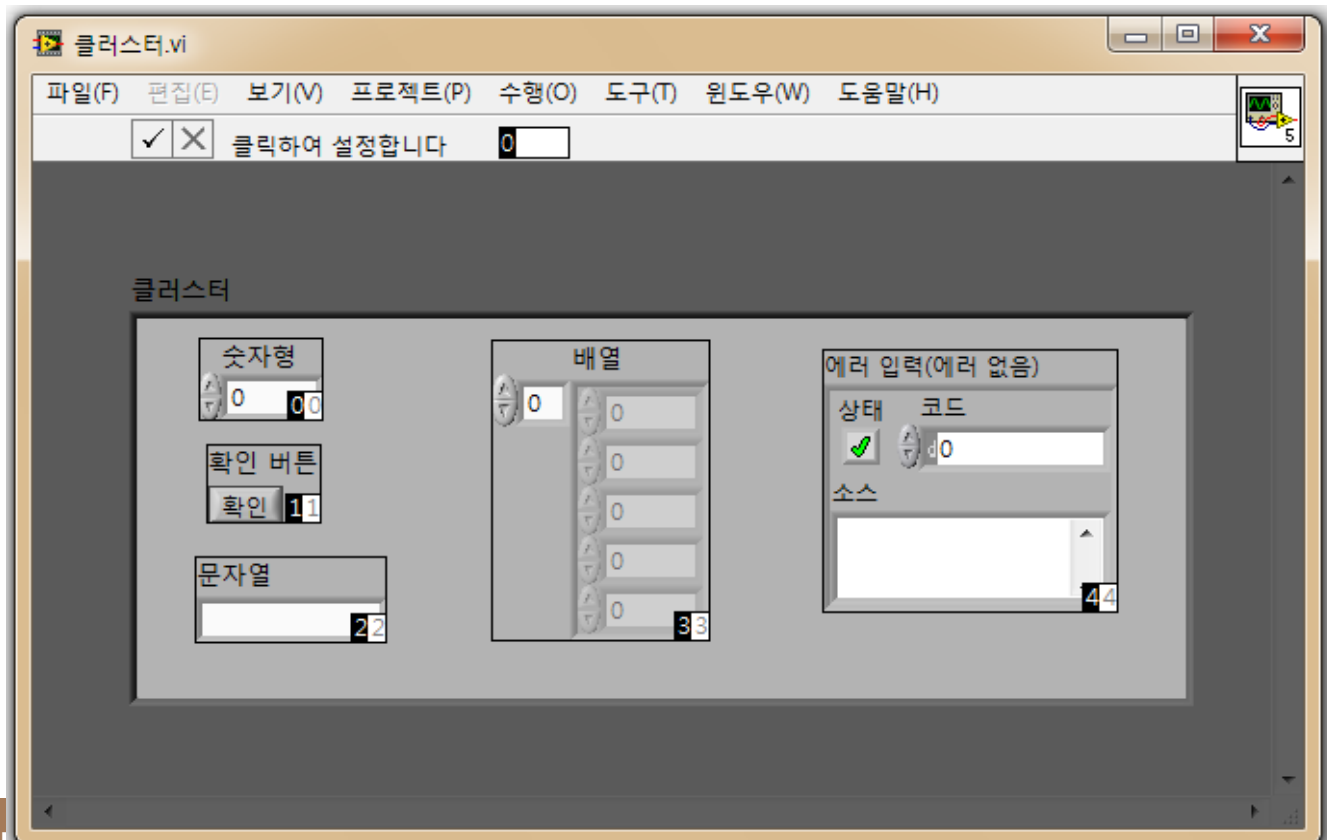
- 빈 클러스터를 찾아서 위치시키고, 묶어줄 원소를 클러스터 속에 차례로 넣어준다.

클러스터

<p>숫자형</p> <p>0</p> <p>확인 버튼</p> <p>확인</p> <p>문자열</p> <p></p>	<p>배열</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p> <p>0</p>	<p>에러 입력(에러 없음)</p> <table border="1"><tr><td>상태</td><td>코드</td></tr><tr><td></td><td>d0</td></tr><tr><td colspan="2">소스</td></tr><tr><td colspan="2"><p></p></td></tr></table>	상태	코드		d0	소스		<p></p>	
상태	코드									
	d0									
소스										
<p></p>										

클러스터 내의 컨트롤 순서 재설정

- 클러스터 원소는 그 위치에 관계 없이 논리적인 순서를 가진다.
- [클러스터 내의 컨트롤 순서 재설정]



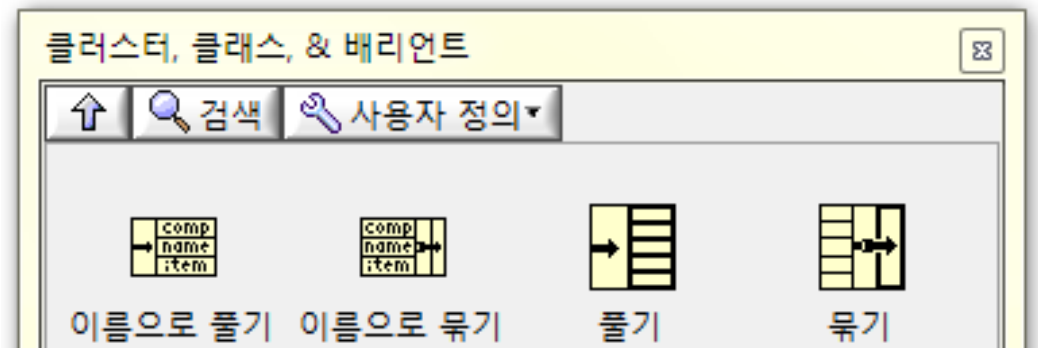
2. 클러스터 함수

묶기 함수

풀기 함수

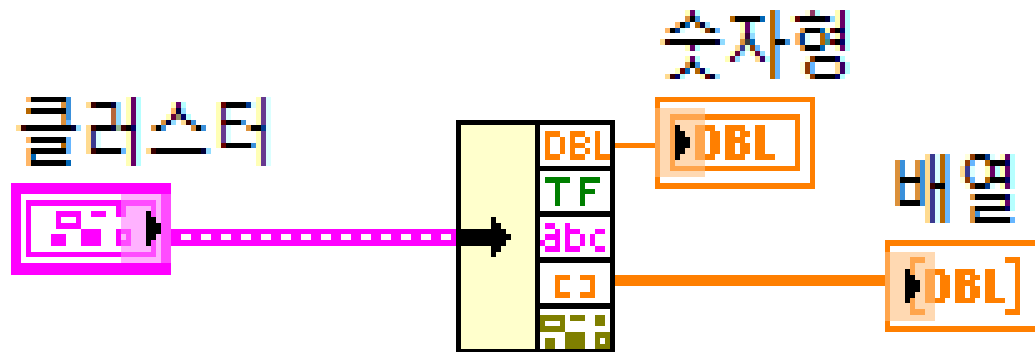
클러스터 함수

- 묶기 함수
 - 묶기
 - 이름으로 묶기
- 풀기 함수
 - 풀기
 - 이름으로 풀기



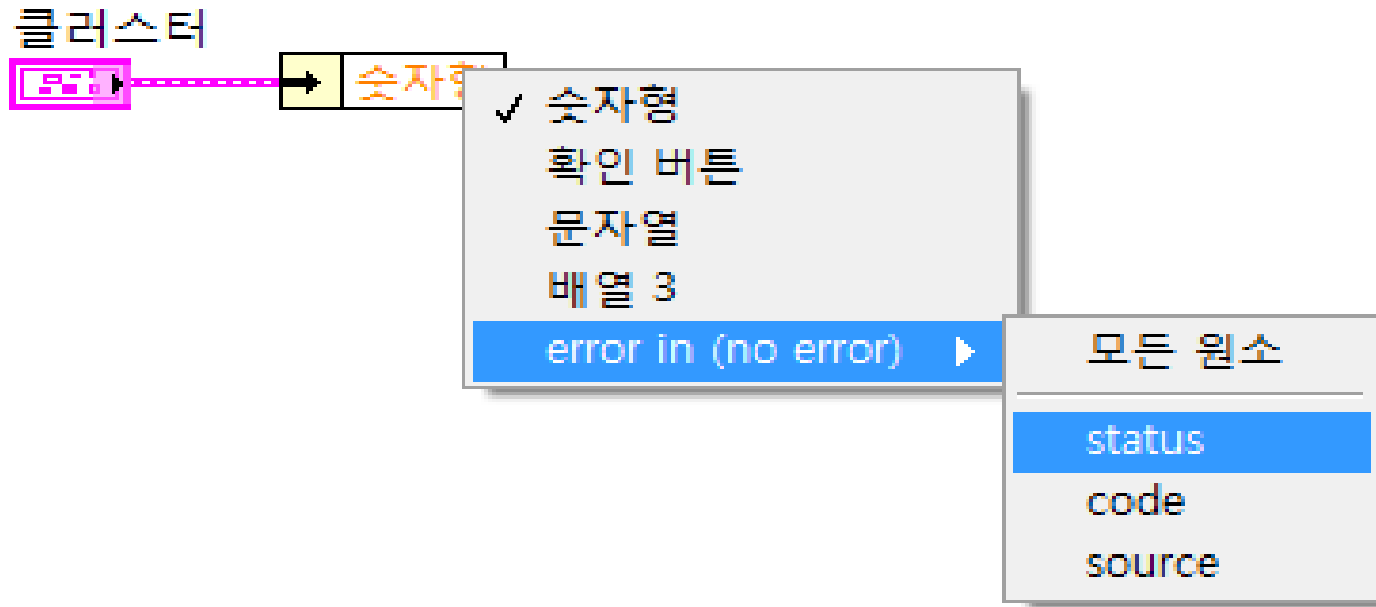
풀기

- 풀기는 클러스터의 모든 원소를 풀어주는 함수
- 모든 원소를 그 원소의 데이터 타입으로 풀어준다.



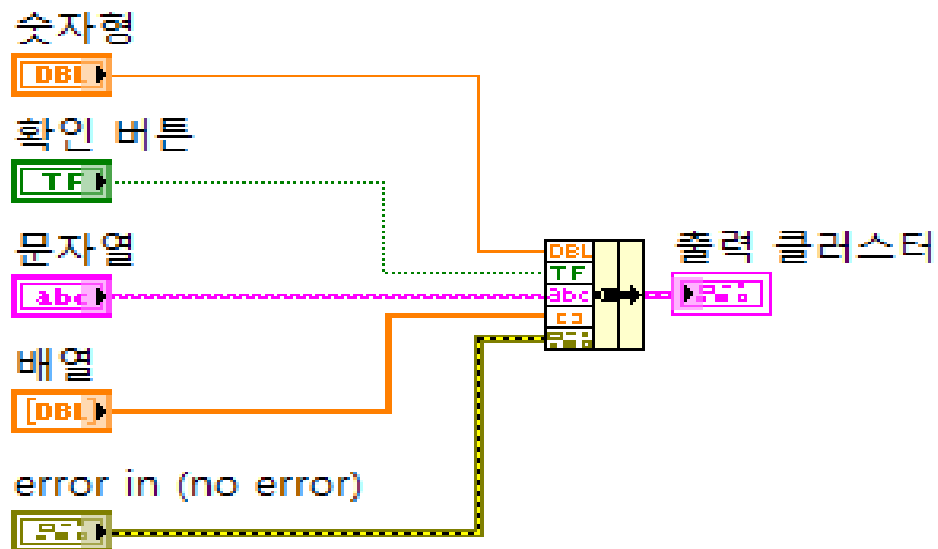
이름으로 풀기

- 클러스터에서 한 개 또는 여러 개의 원소를 이름으로 풀어내는 함수



묶기

- 개별 원소를 클러스터로 모은다.
- 입력을 늘려서 여러 개의 입력으로 바꿀 수 있다.
- 입력할 수 있는 원소에는 제한이 없다.



숫자형

DBL

확인 버튼

TF

문자열

abc

배열

DBL

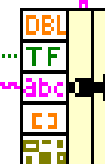
error in (no error)

TF



T

항상 참입니다.



출력 클러스터



이름으로 묶기

- 하나 또는 그 이상의 클러스터 원소를 대체하는 목적으로 사용된다.



실습 5-1: 클러스터 풀기와 묶기

클러스터

숫자형

0

확인 버튼

확인

문자열

배열 데이터

0

0

0

0

0

0

에러 입력(에러 있음)

상태

코드

0

소스

클러스터 2

숫자형

0

확인 버튼

확인

문자열

배열 데이터

0

0

0

0

0

0

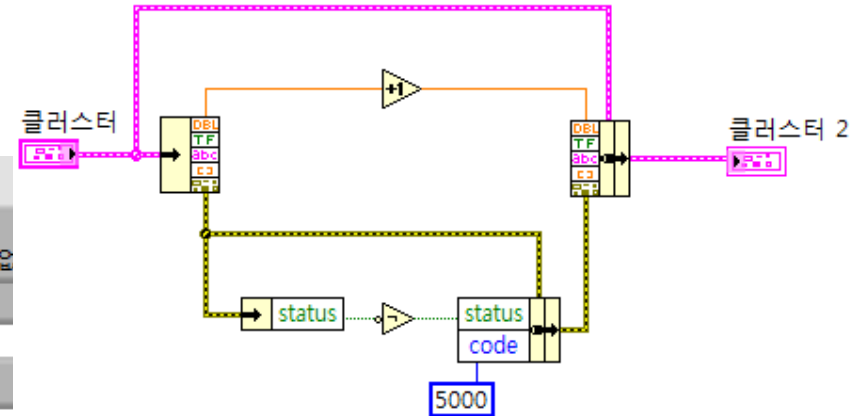
에러 입력(에러 없음)

상태

코드

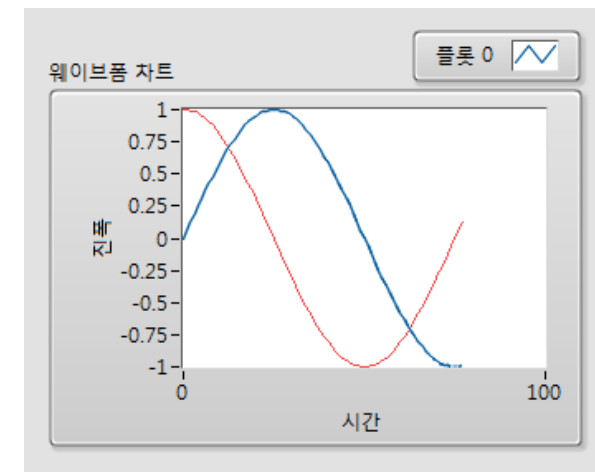
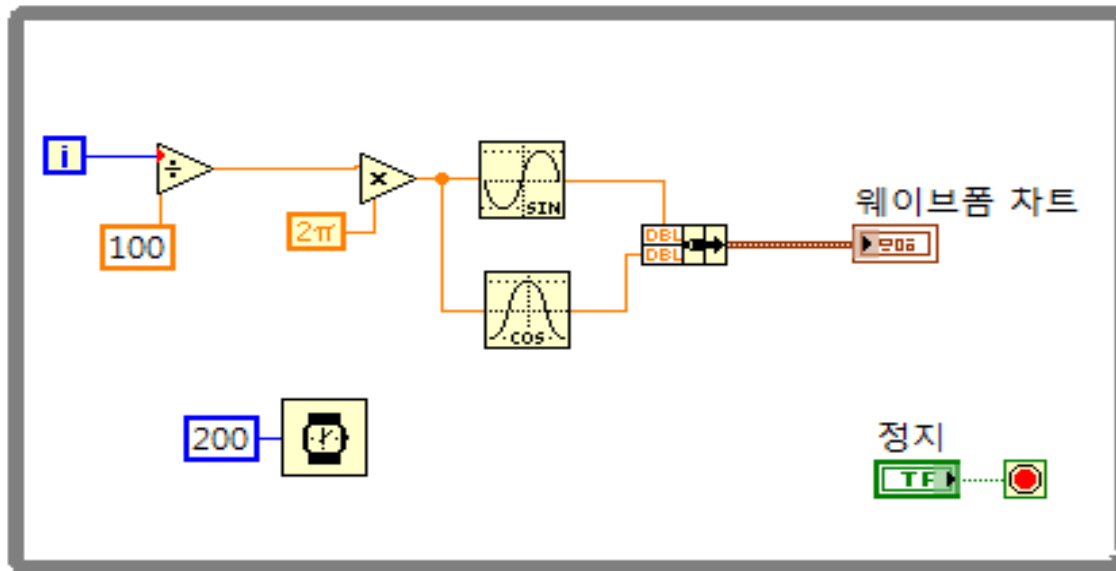
0

소스

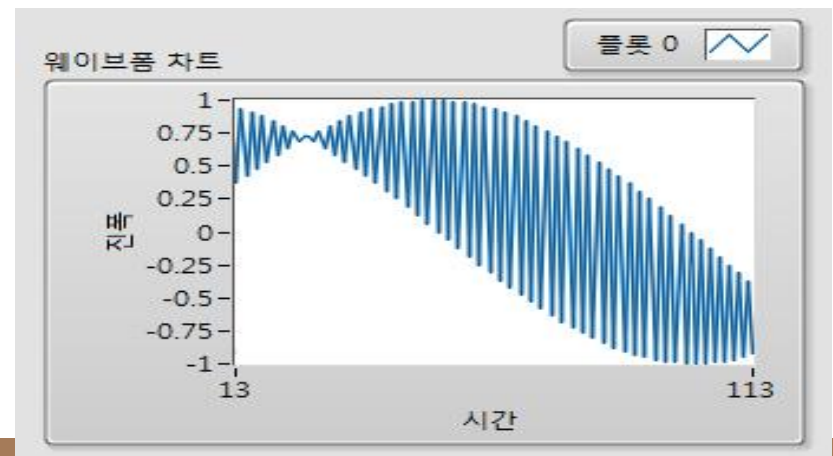
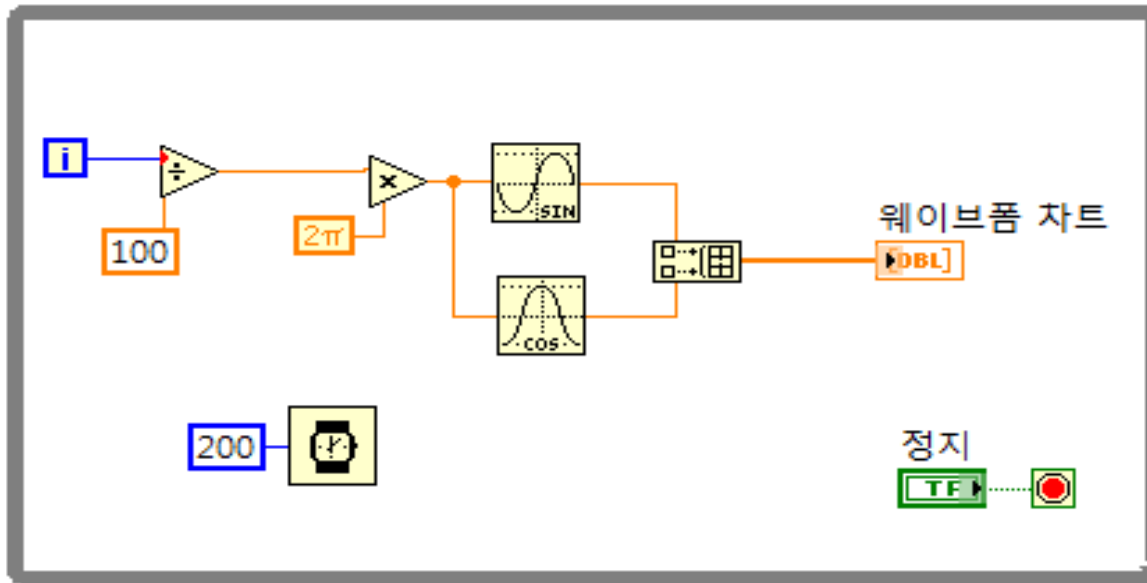


웨이브폼 차트와 클러스터

- 웨이브폼 차트를 이용하여 여러 채널을 플롯하고자 하는 경우에, 클러스터를 이용하여 묶어준다.

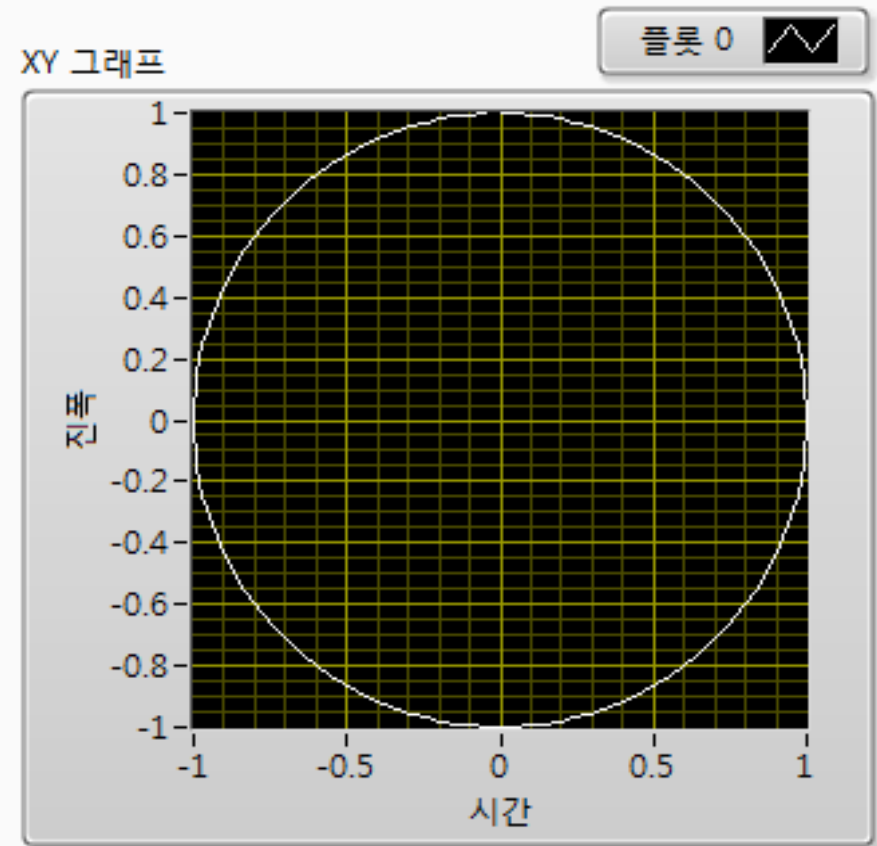
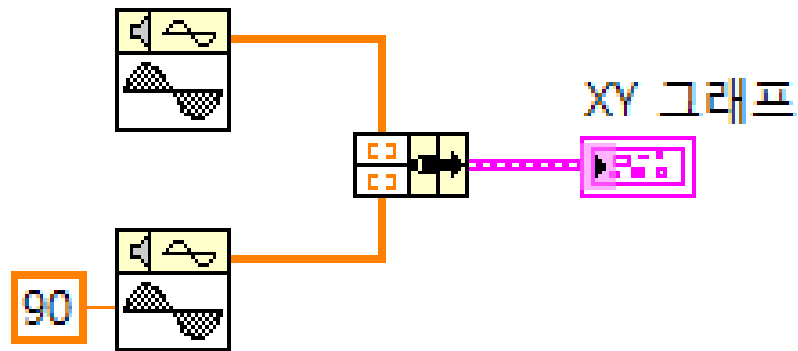


- 만약 뭉치 함수를 배열 만들기로 바꾸면, 한 개 채널로 플롯 된다.

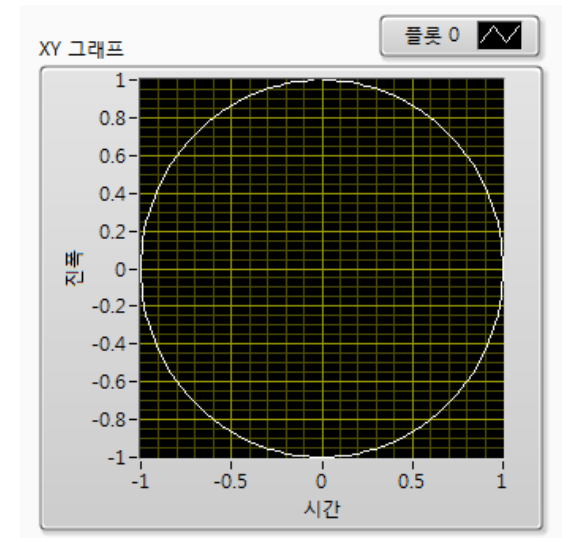
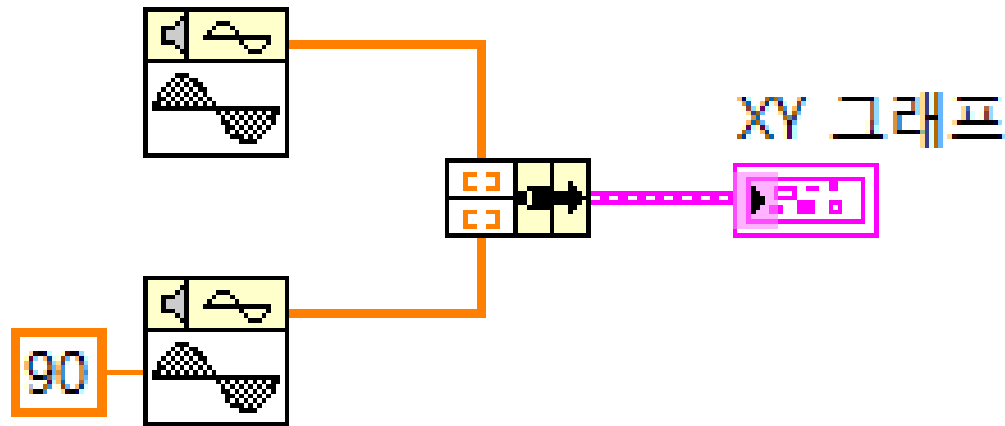


XY 그래프와 클러스터

- X 배열과 Y 배열을 클러스터로 묶어서 XY 그래프로 플롯

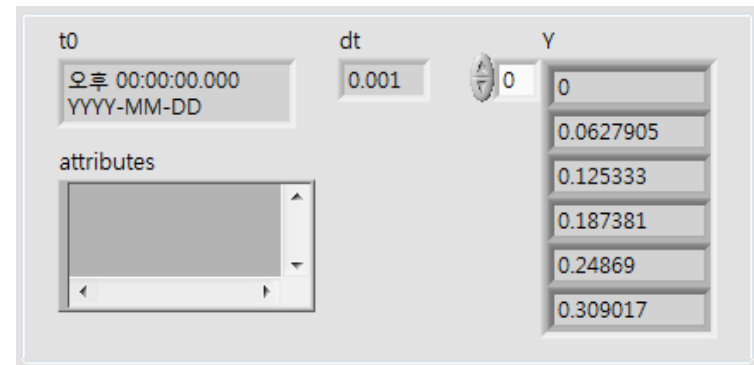
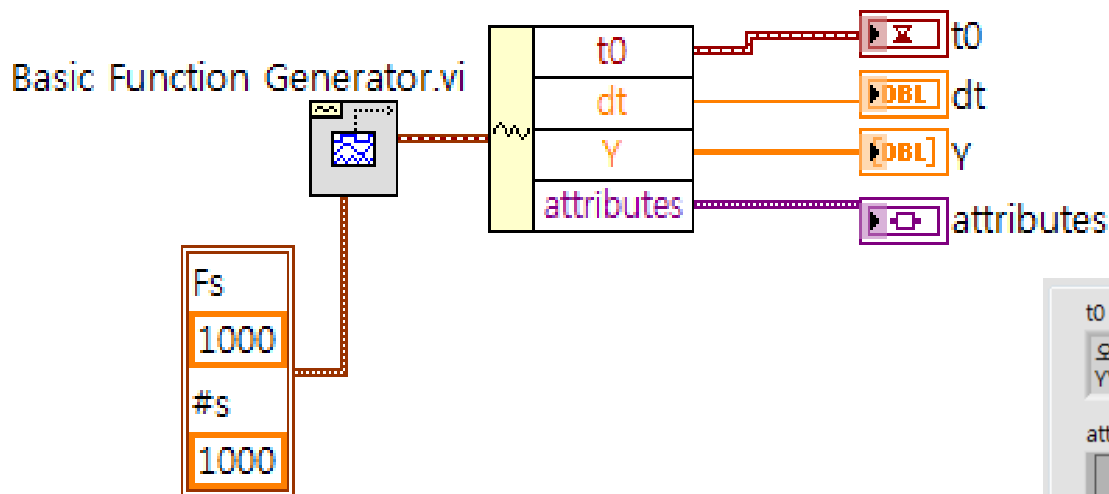


실습 5-2: 원 그리기

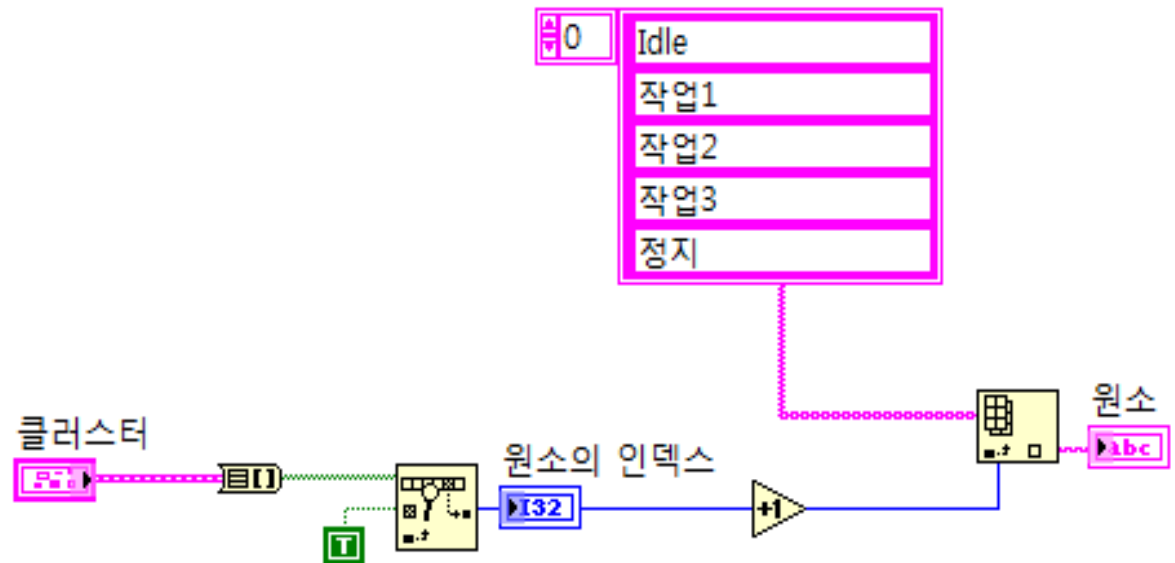


웨이브폼 데이터 타입

- 웨이브폼 데이터 타입은 t0, dt, Y 배열, Attributes로 구성된 구조체이다.
- 근본적으로 클러스터의 한 종류이지만, 클러스터 함수로 풀거나 묶을 수 없도록 보호되어 있다.



실습 5-3: 사용자 작업 선택 기법

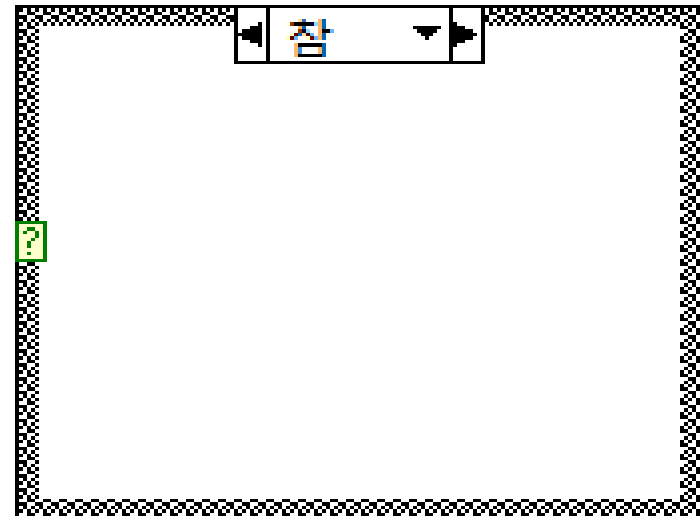


3. 케이스 구조

케이스 구조의 정의와 사용법

케이스 구조

- **Case** 또는 **IF** 문을 구현
- 케이스 구조는 하나 또는 그 이상의 서브다이어그램 또는 케이스를 가지며, 구조가 실행되면 그 중 하나만이 실행
 - 케이스 라벨
 - 케이스 선택자



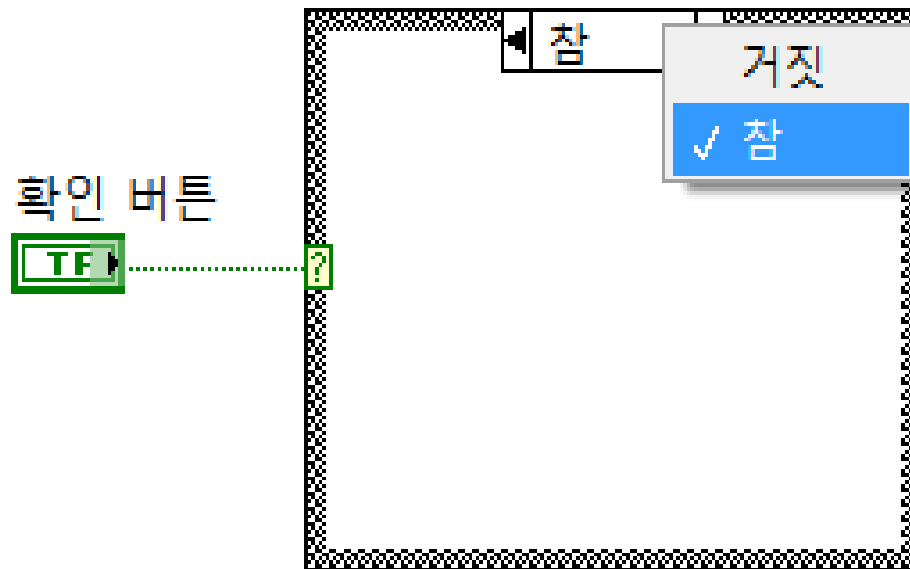
케이스 구조

1. 케이스 구조는 층층으로 쌓여있는 구조이다.
2. 사용할 케이스를 선택하기 위해서, 선택자 라벨의 증감 화살표를 클릭한다.
3. 케이스 구조는 한번에 한 개의 케이스만 실행할 수 있다.
4. 케이스 선택자 터미널에 연결된 값이 어떤 케이스를 실행할 지 결정한다.

- 기본(Default) 케이스를 지정할 수 있다.
- 케이스 선택자
 - 불리언
 - 문자열
 - 정수 숫자형
 - 열거형 타입
 - 에러 클러스터

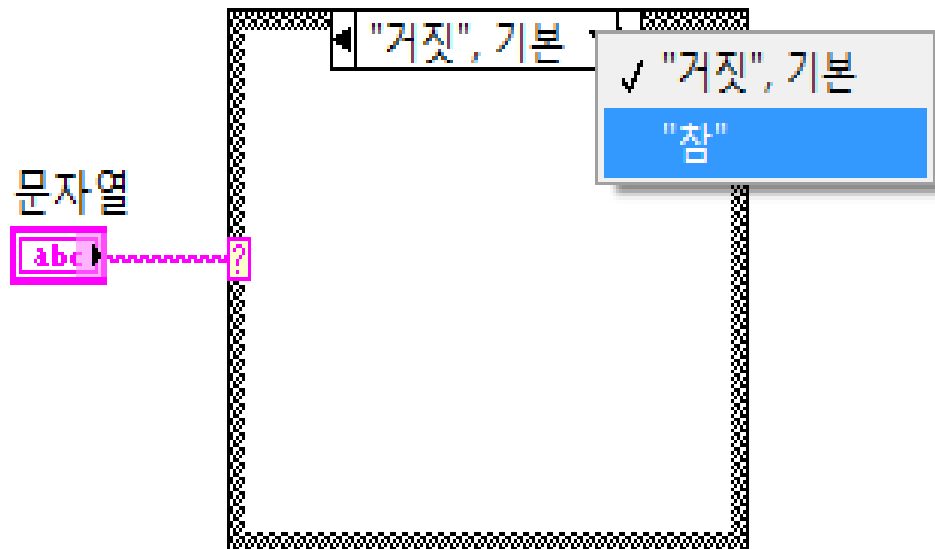
불리언 선택자

- <참> 케이스
- <거짓> 케이스



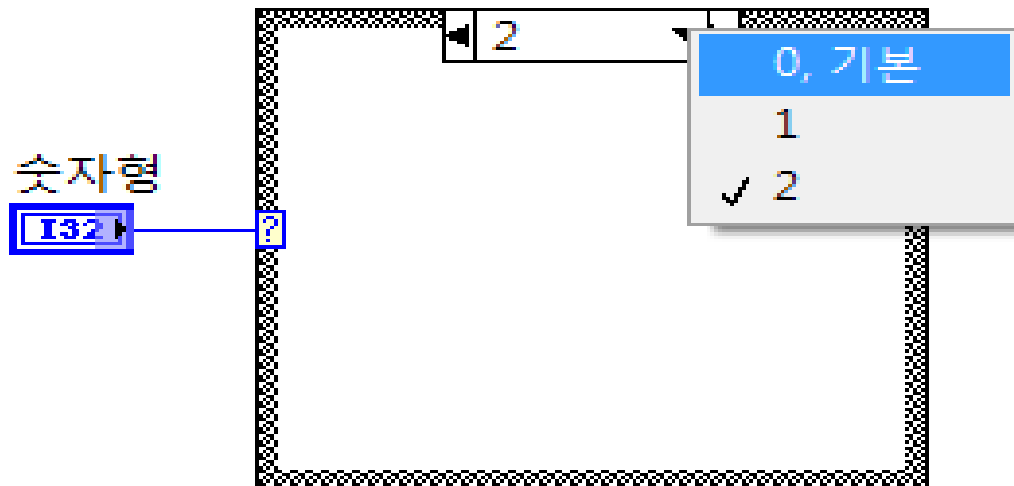
문자열 선택자

- <"거짓", 기본> 케이스와 <"참"> 케이스로 바뀐다.
- 문자열은 **ASCII** 문자표에서 제공되는 모든 문자를 사용할 수 있다.
- [다음 케이스 추가]

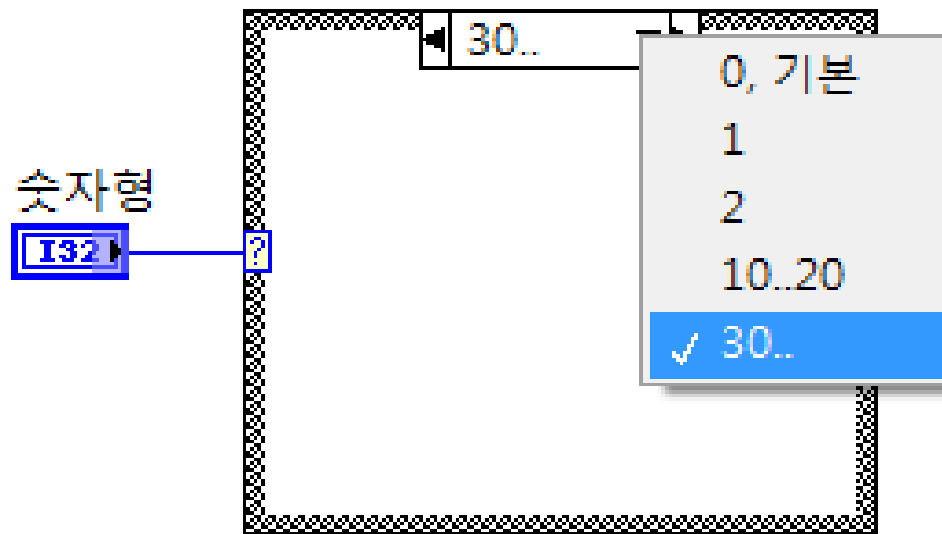


숫자형(Integer) 선택자

- <0, 기본> 케이스와 <1> 케이스
- [다음 케이스 추가]

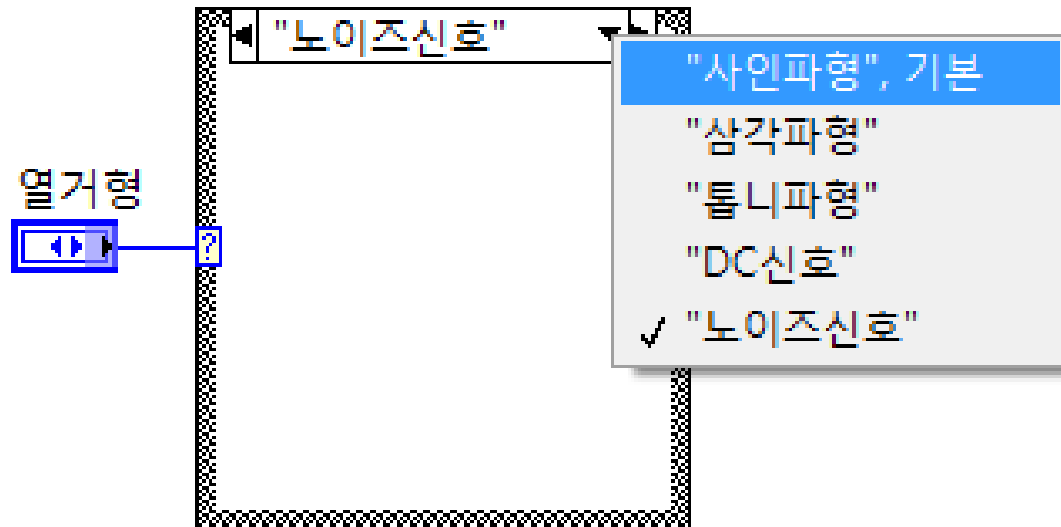


- $\langle 10..20 \rangle$ 10부터 20까지의 모든 정수
- $\langle 30.. \rangle$ 30보나 크거나 같은 모든 정수



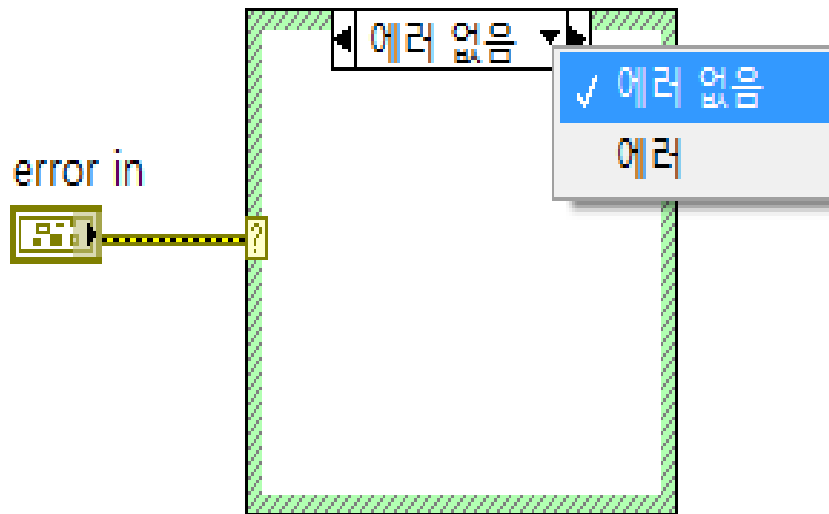
열거형 선택자

- 열거형 데이터 타입은 부호 없는 정수형이다.
- 열거형, 탭 컨트롤 등은 케이스 선택자 전용이다.



에러 클러스터 선택자

- <에러>는 빨간색 테두리
- <에러 없음>은 초록색 테두리로 표시

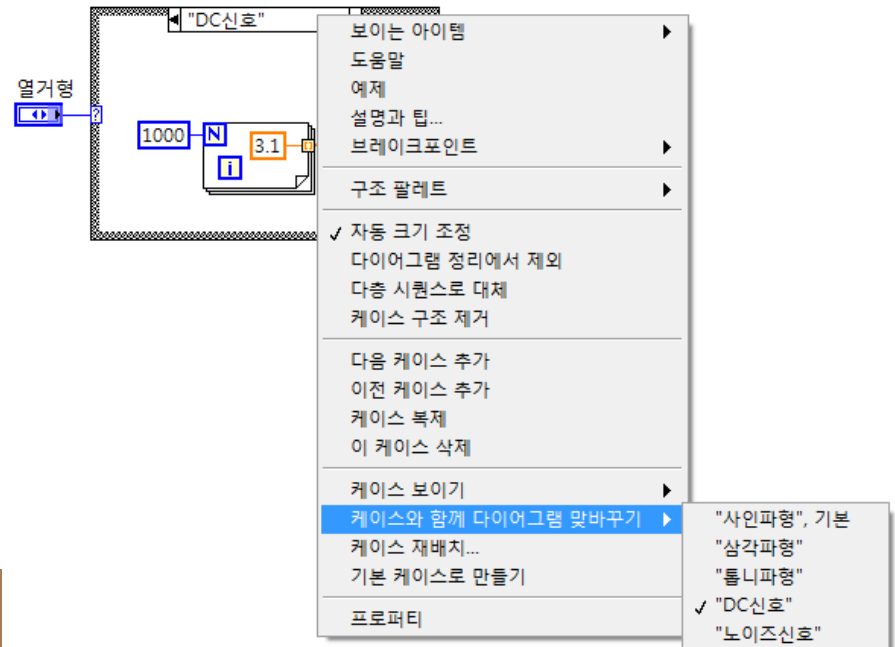


서브다이어그램 추가 및 복제하기

- [다음 케이스 추가]
- [이전 케이스 추가]
- [케이스 복제]

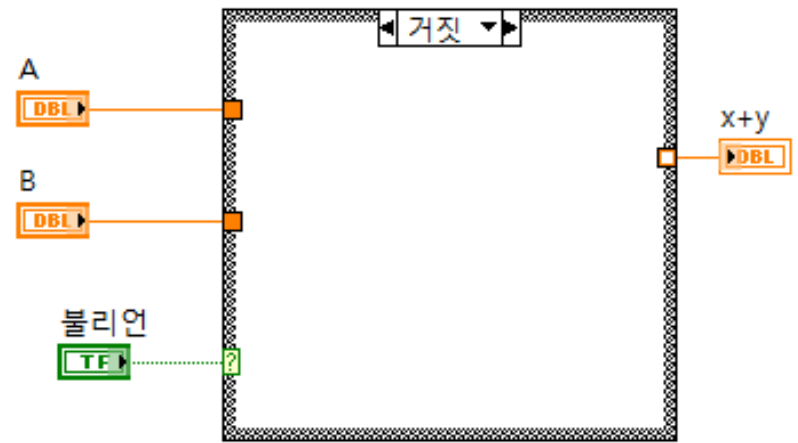
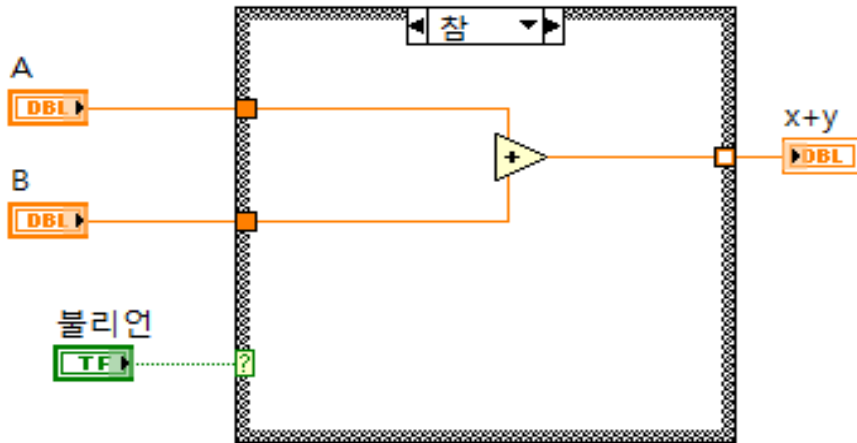
삭제하기, 재배치, 다이어그램 교환

- [이 케이스 삭제]
- [케이스 재배치]
- [케이스와 함께 다이어그램 맞바꾸기]



케이스 구조의 입력과 출력

- LabVIEW는 데이터 흐름으로 프로그램이 구현된다.
- 데이터 흐름이 끊어지면 에러가 발생한다.



- 출력 터널에 값이 모두 연결되지 않은 경우에는 에러가 발생
 - 출력 연결
 - [연결되지 않으면 기본값 사용]

실습 5-4: 열거형 케이스 선택자

