

로컬 변수, 글로벌 변수, 공유 변수에 대하여 배웁니다.

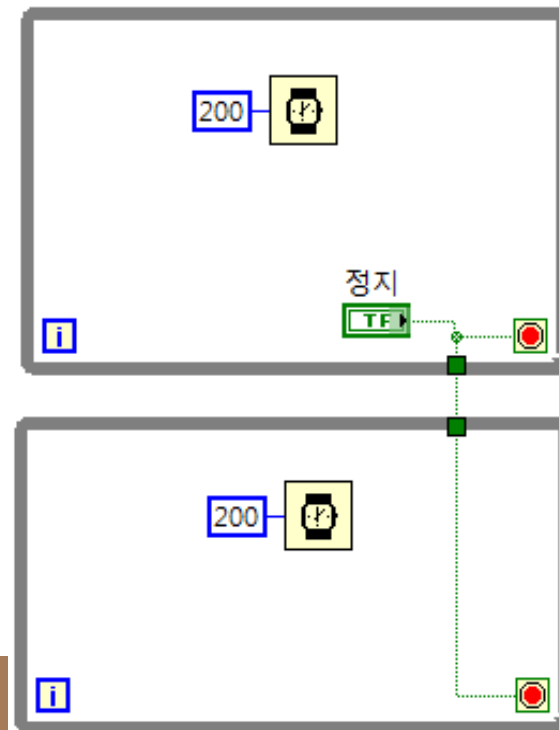
CHAPTER 8. 변수 사용하기

1. 로컬 변수

로컬 변수 만들기
사용하기

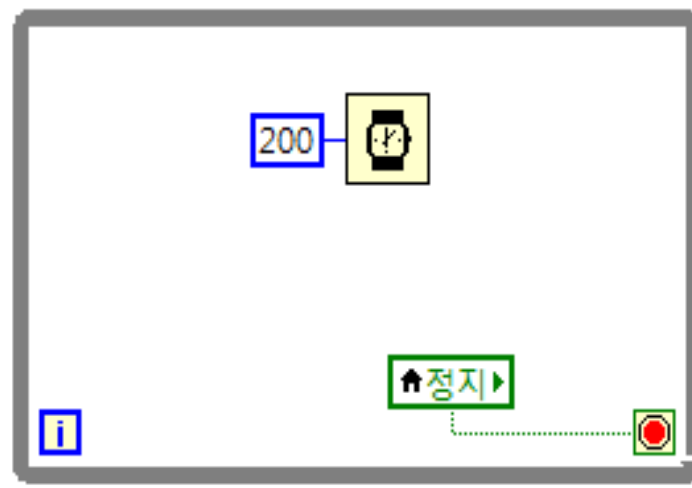
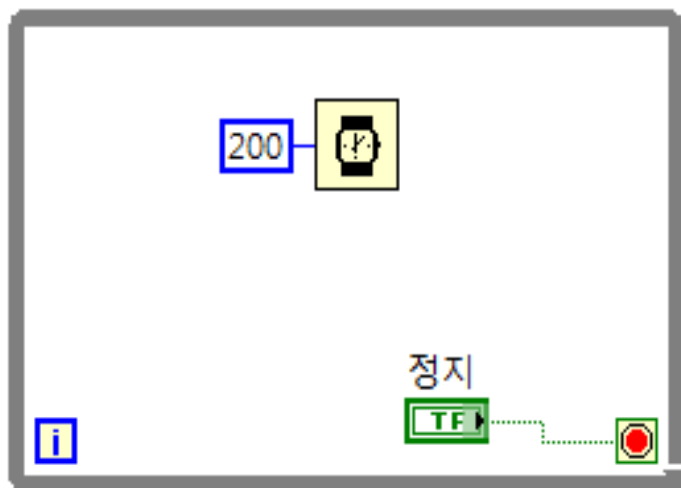
변수의 필요성

- LabVIEW는 와이어를 통하여 데이터가 전달되는 데이터 흐름 방식을 사용한다.
 - 함수나 노드들은 모든 필수 입력에 데이터가 입력된 뒤에 실행되고, 와이어 연결 순서에 따라서 차례로 함수들이 실행된다.
 - 그러나 와이어의 연결이 불가능한 경우가 있고, 이 경우에 사용할 수 있는 것이 변수이다.



변수의 필요성

- 로컬 변수
- 글로벌 변수
- 단일 프로세스 공유 변수
- 네트워크 공유 변수



로컬 변수

- 로컬 변수는 단일 VI 내에서 사용할 수 있는 변수이다.
- 로컬 변수를 만들기 위해서는 프런트패널에 이름이 있는 객체가 있어야 된다.



로컬 변수 만들기

- 단축메뉴로 생성 > 로컬 변수
- 함수 > 일반 > 구조 팔레트 → 빈 로컬 변수
- 프런트패널의 컨트롤과 인디케이터에 고유 라벨을 할당해야 한다.

로컬 변수의 주의점

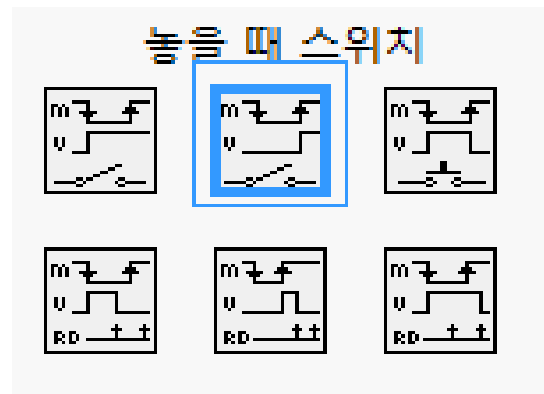
- 프런트패널에 동일한 이름의 객체가 있을 경우, 로컬 변수는 두 개의 객체에 대하여 대응하는 링크를 각각 만든다.
- 불리언의 래치 기능은 로컬 변수로 만들 수 없다.
- 불리언 로컬 변수는 초기화가 필요하다.

프로퍼티 노드를 이용한 값 전달

- 생성 > 프로퍼티 노드 > 값



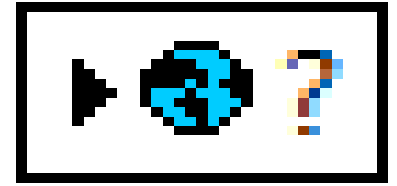
실습 8-1: 로컬 변수



2. 글로벌 변수

글로벌 변수 만들기
사용하기

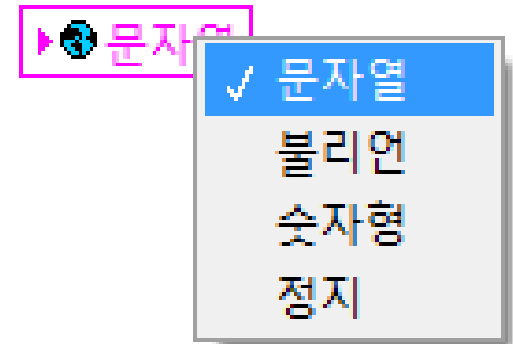
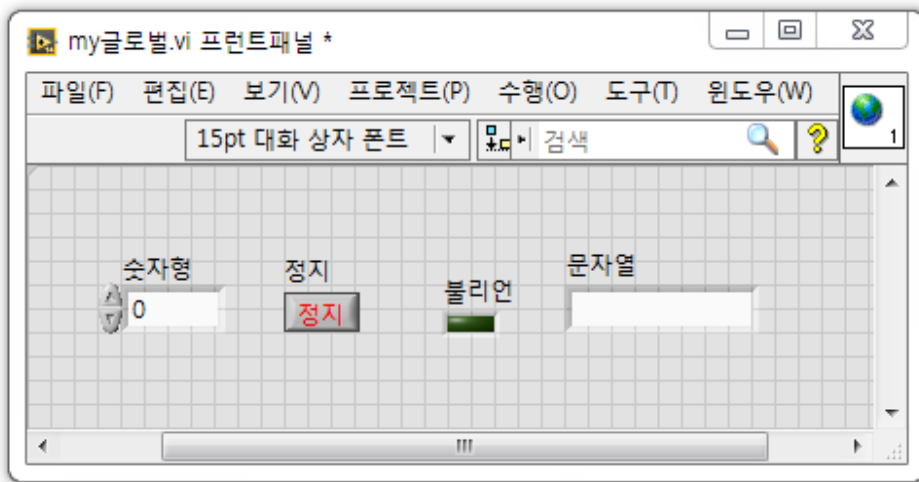
글로벌 변수



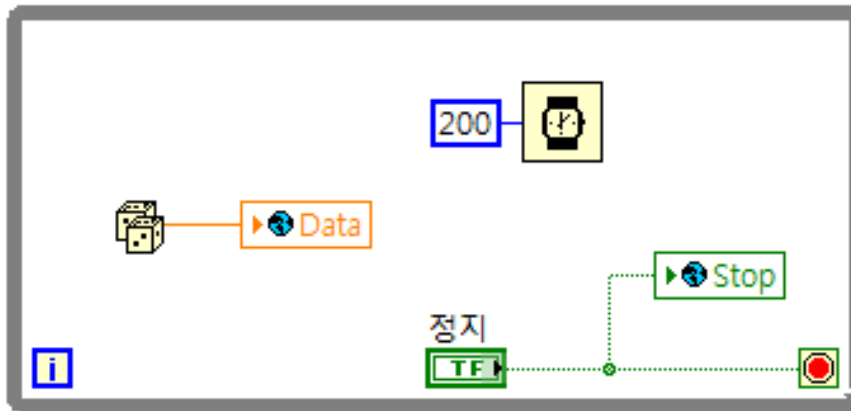
- 글로벌 변수는 단일 프로세스 안에 있는 여러 VI들 사이에서 데이터를 전달하는 목적으로 사용되는 변수이다.
- 로컬 변수를 대체하여 사용할 수도 있다.
- 글로벌 변수를 생성하면, LabVIEW는 블록다이어그램은 없고 프런트패널만 있는 특수한 글로벌 VI를 생성한다.

글로벌 변수 만들기

- 함수 > 일반 > 구조 팔레트의 글로벌 변수
 - 글로벌 변수를 일반 SubVI처럼 사용한다.
 - VI로 저장한다.
 - 함수 팔레트의 **VI** 선택을 이용한다.



실습 8-2: 글로벌 변수



3. 공유 변수

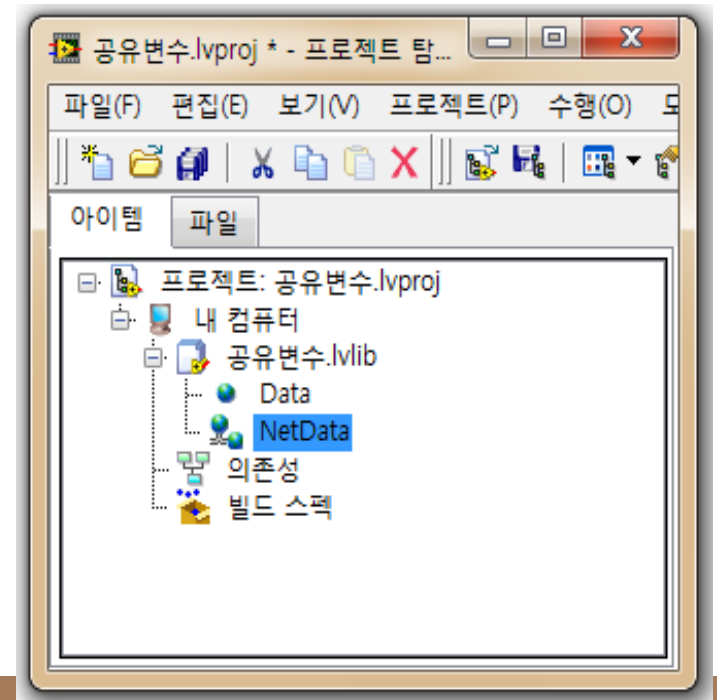
공유 변수 만들기
사용하기

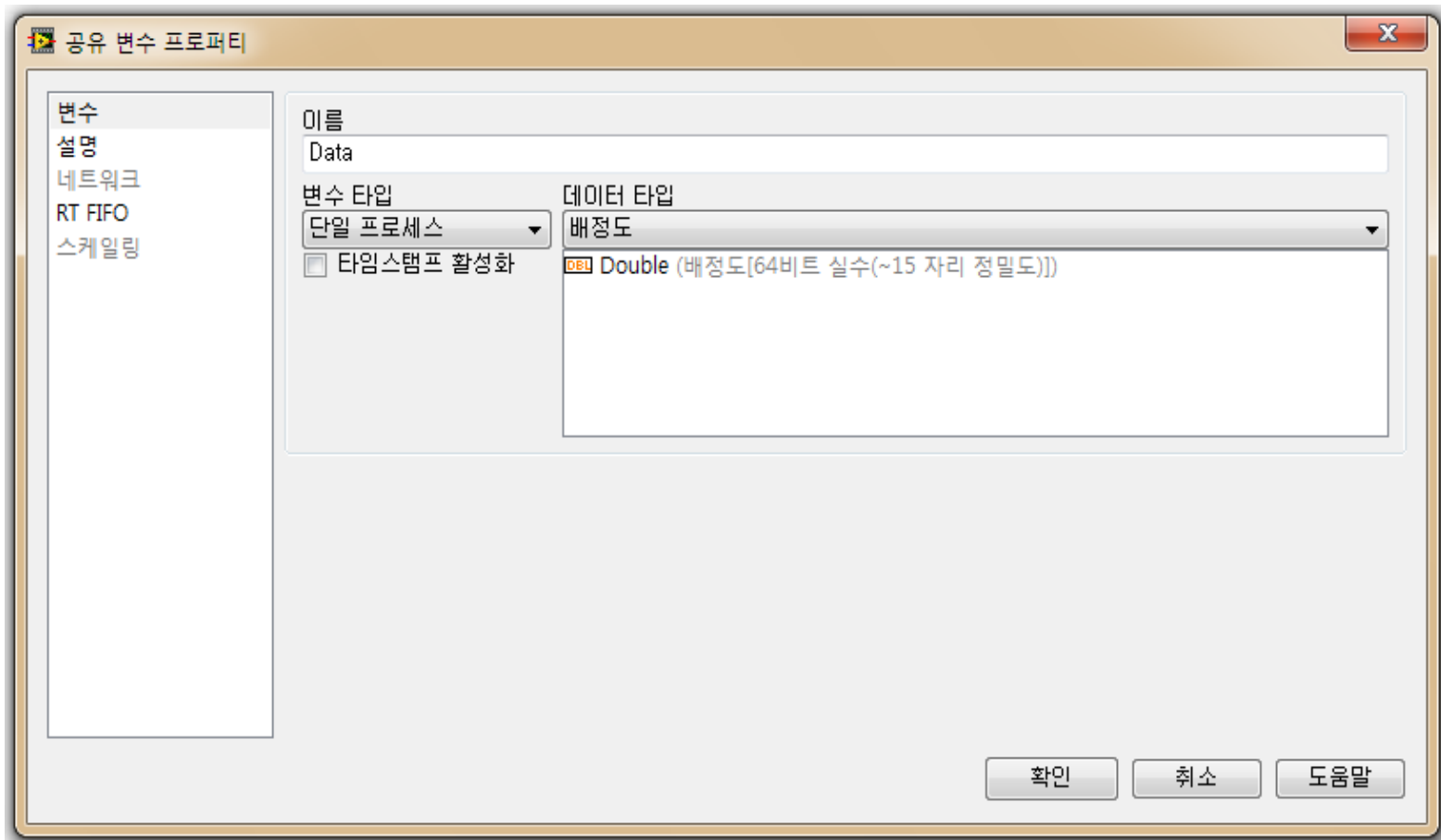
공유 변수

- 공유 변수는 공유 변수 엔진(Shared Variable Engine)을 통하여 데이터를 공유한다.
- 네트워크 상에서도 데이터 공유가 가능하다.
- 공유 변수 엔진은 **LabVIEW**를 설치할 때 사용자 컴퓨터에 기본 옵션으로 설치되며, **NI Publish-Subscribe Protocol**라는 기술을 사용하여 공유 변수 업데이트를 관리한다.

공유 변수 만들기

- LabVIEW 프로젝트에서만 만들 수 있다.
- 공유 변수는 프로젝트 상에서 라이브러리 아래에 만들어줘야 된다.
 - 새로 만들기 > 라이브러리
 - 새로 만들기 > 변수

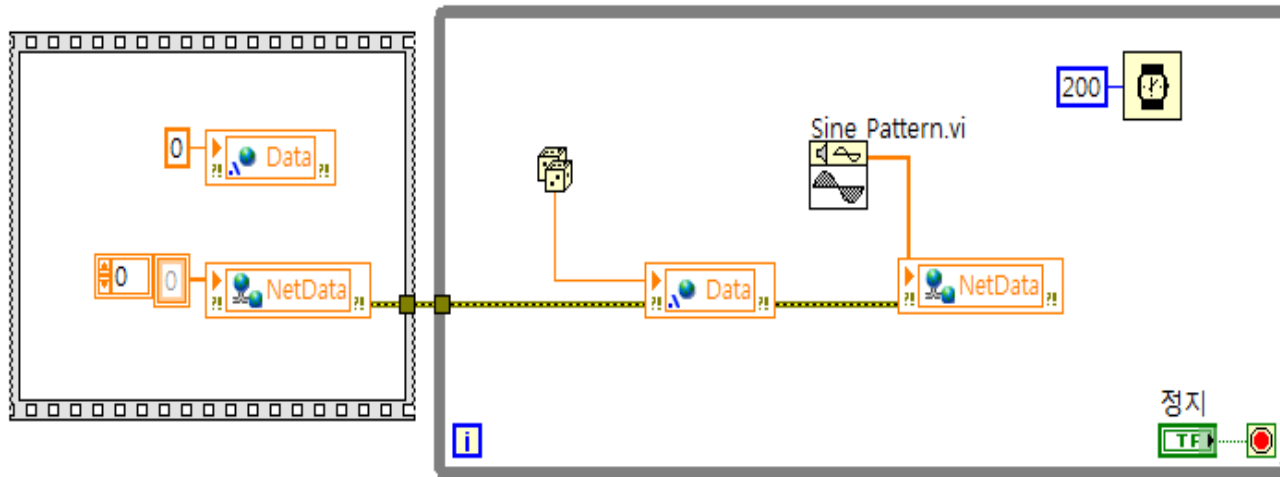




- 변수: 데이터 타입과 변수 타입을 포함한 공유 변수의 기본 설정을 결정
- 설명: 공유 변수에 대한 설명을 지정
- 네트워크: 버퍼링을 포함과 같은 공유 변수에 대한 네트워크 옵션을 설정
- **Real-Time FIFO**: 공유 변수의 리얼타임 **FIFO**를 활성화하고 설정
- 스케일링: 공유 변수에 대한 스케일링 옵션을 설정
- 왼쪽 리스트에서 [변수]가 선택되면, 다음과 같이 변수의 이름과 변수 타입, 데이터 타입을 설정하는 탭이 나타난다.
- 변수 타입으로는 네트워크와 단일 프로세스를 선택할 수 있다.
 - 네트워크는 동일 네트워크 상에서 원격 컴퓨터나 타겟에서 공유되는 변수를 생성한다.
 - 단일 프로세스는 한 컴퓨터 내에서 VI들 사이에 공유되는 변수를 생성한다.

공유 변수 사용

- 공유 변수를 공유 변수 엔진(SVE)에 배포해야만 해당 공유 변수를 네트워크를 통해 읽고 쓸 수 있다.
 - 공유 변수를 배포하면 공유 변수 엔진에서 공유 변수에 대한 메모리 공간이 생성된다.
 - 공유 변수의 배포를 해제하면 해당 공유 변수에 할당되었던 메모리 리소스가 해제된다.
- 배포: 공유 변수를 포함한 VI를 실행할 때, 자동으로 배포된다.
- 배포 해제: 배포 해제는 자동으로 되지 않는다.
- 배포 완료된 공유 변수는 프로젝트 상에서 선택하여, 블록다이어그램의 원하는 위치에 끌어서 붙여놓을 수 있다.



공유 변수 사용의 장점과 단점

- 공유 변수는 로컬 변수와 글로벌 변수를 대체하여 사용할 수 있다.
- 공유 변수는 프로젝트 상에서 생성하고 설정하도록 되어있기 때문에, 관리 가능한 변수이다.
- 공유 변수는 에러 입출력을 가지고 있다. → 에러 라인 연결을 통하여 실행 순서를 정하기 쉽다.
- 사용이 불편하다. → 배포 및 배포 해제 작업이 필요하다.
- 공유 변수는 공유 변수 엔진을 사용한다. → 추가적인 플러그 인을 사용하는 것이고, 서비스 충돌로 시스템이 다운될 수 있다.

실습 8-3: 공유 변수

